Algorithm Agility for RPKI

Roque Gagliano

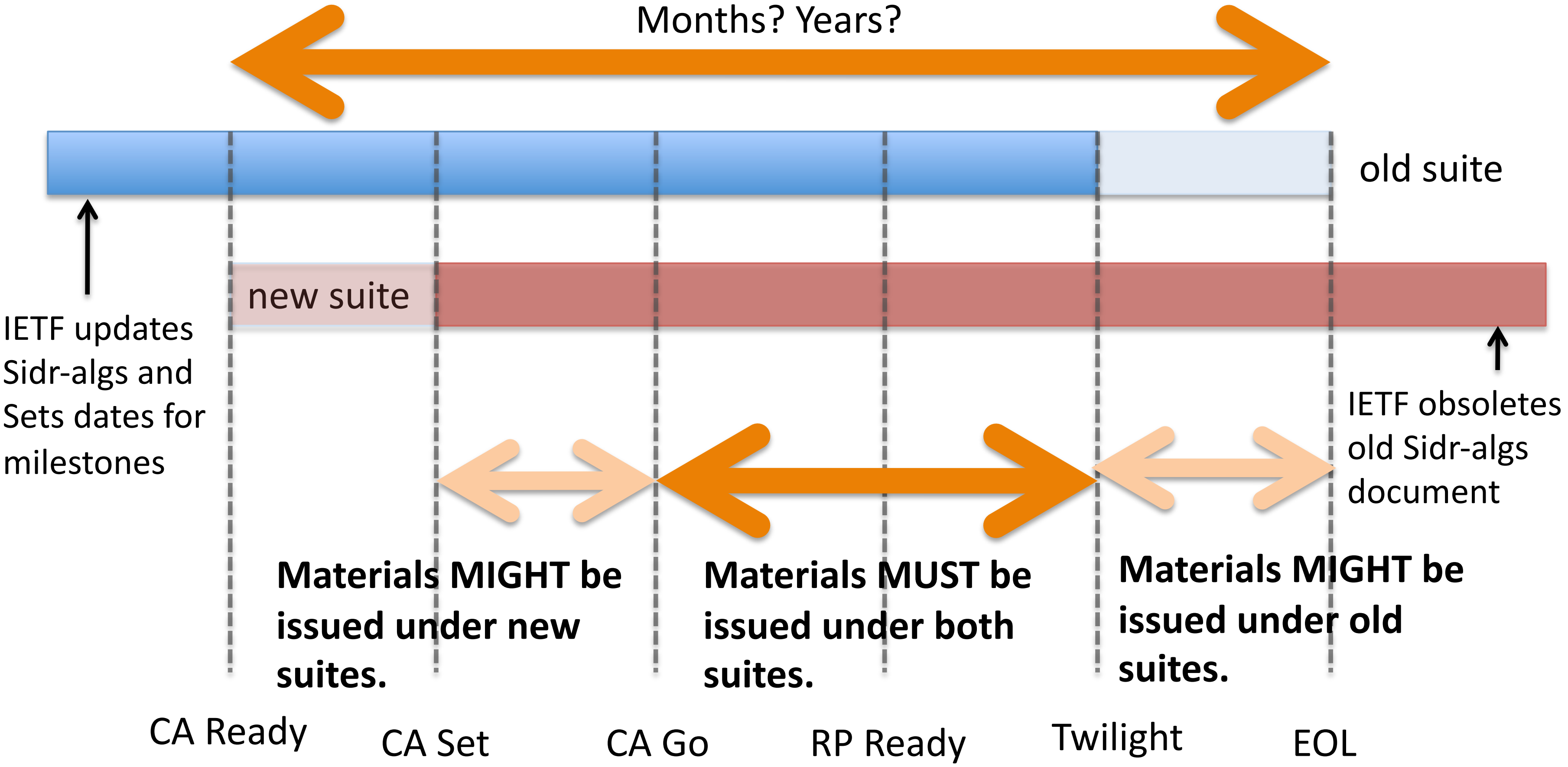
Stephen Kent

Sean Turner

Context

- SEC AD requested the SIDR WG to document a mechanism for algorithm migration in RPKI.
- We spent sometime reviewing key roll-over.
- The algorithm transition document will include:
  - A formal procedure to implement algorithm transition by updating rpki-algs document (and the CP?)
  - A set of steps/milestones to be achieved during the transition period.
  - Required behavior by CAs and RPs during migration
  - We will not have emergency mechanism for algorithm transition.

Algorithm migration (normal process):



Months? Years?

old suite

new suite

IETF updates Sidr-algs and Sets dates for milestones

IETF obsoletes old Sidr-algs document

Materials MIGHT be issued under new suites.

Materials MUST be issued under both suites.

Materials MIGHT be issued under old suites.

CA Ready

CA Set

CA Go

RP Ready

Twilight

EOL

Question 1: Transition Path

- Top-down only: we implement a mechanism where a child CA can transition to a new algorithm only if its parent CA has already transitioned.
- Laissez faire: any CA can begin using the new algorithm suite at any point in the PKI hierarchy (if the parent can execute Proof of Possession).

Do we want to support only top-down algorithm transitions?

Answer 1

We recommend pursuing only the top-down option.

Reasoning:

- It complies with the requirements from SEC-AD.
- It may be simpler as it may avoid the need for cross-algorithm certificates (e.g., new-in-old)
- If cross-algorithm certificates are needed, we still need support for PoP from parent CA (“CA Ready”).
- It seems likely that the smaller number of higher tier entities will be ready for algorithm migration sooner than lower tiers (e.g., ISPs)

Question 2

Do we want to support multiple signatures in the CMS objects (ROAs & manifests)?

(See Section 3 draft-ietf-sidr-rpki-algs)

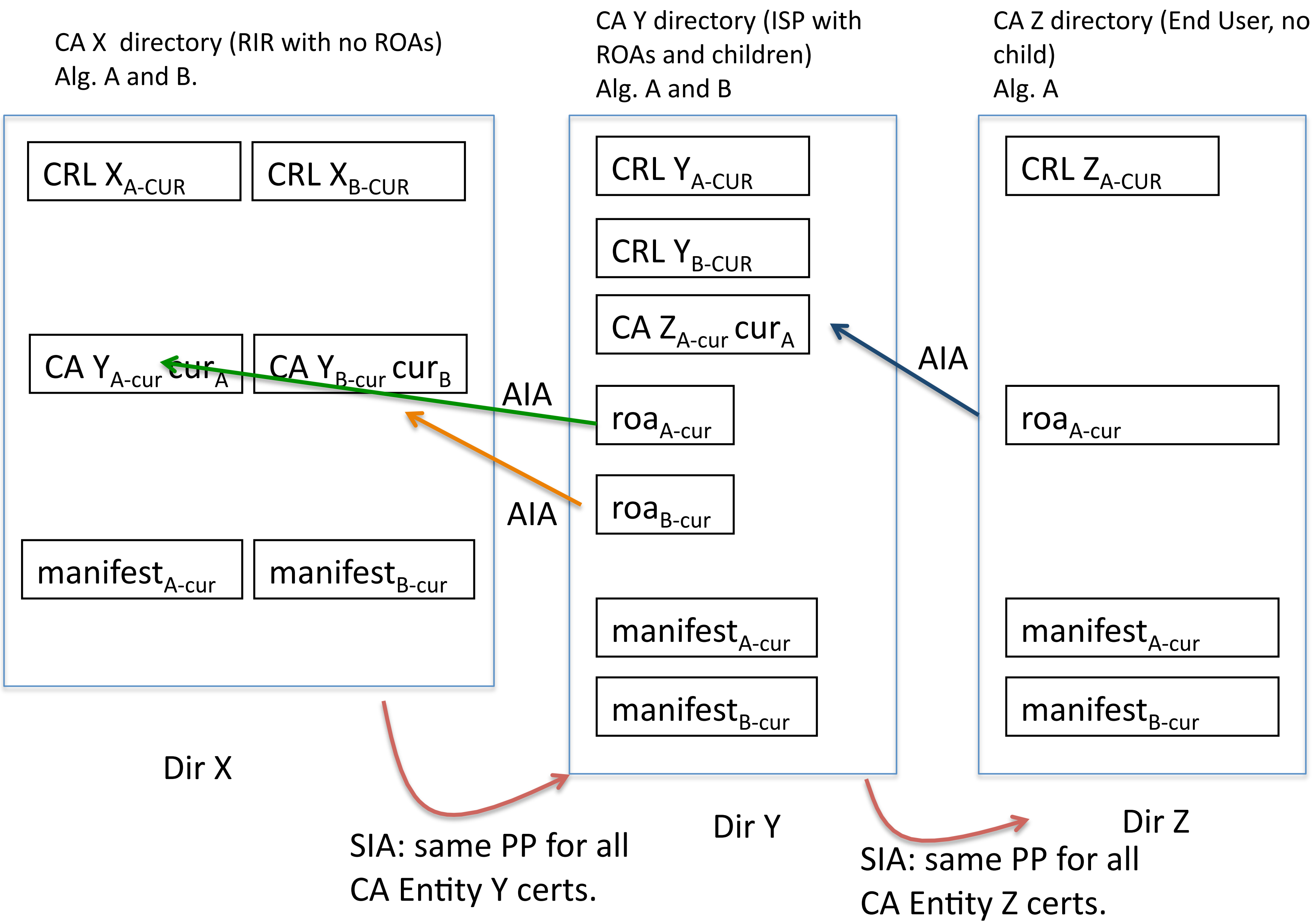
Support for multiple signatures may reduce:

- duplication of signed objects in repositories
- complexity of CA software
- complexity of repository operation

Answer 2

- We don’t recommend multi-signature objects
  - Certificates and CRLs MUST contain only a single signature, so only CMS objects are in play here
  - Spec would have to change
  - CAs may elect to publish separate, single signature objects anyway, so RPs have to accommodate
  - A multi-signature manifest is ambiguous, unless we modify the format to add an algorithm ID

ALG. ROLL-OVER (without KEY ROLL OVER) Repository Structure with single signature ROAs and Manifests.



CA X directory (RIR with no ROAs)
Alg. A and B.

CRL $X_{A\text{-}CUR}$

CRL $X_{B\text{-}CUR}$

CA $Y_{A\text{-}cur}$ cur_A

CA $Y_{B\text{-}cur}$ cur_B

manifest$_{A\text{-}cur}$

manifest$_{B\text{-}cur}$

Dir X

CA Y directory (ISP with ROAs and children)
Alg. A and B

CRL $Y_{A\text{-}CUR}$

CRL $Y_{B\text{-}CUR}$

CA $Z_{A\text{-}cur}$ cur_A

roa$_{A\text{-}cur}$

roa$_{B\text{-}cur}$

manifest$_{A\text{-}cur}$

manifest$_{B\text{-}cur}$

Dir Y

CA Z directory (End User, no child)
Alg. A

CRL $Z_{A\text{-}CUR}$

roa$_{A\text{-}cur}$

manifest$_{A\text{-}cur}$

manifest$_{B\text{-}cur}$

Dir Z

AIA

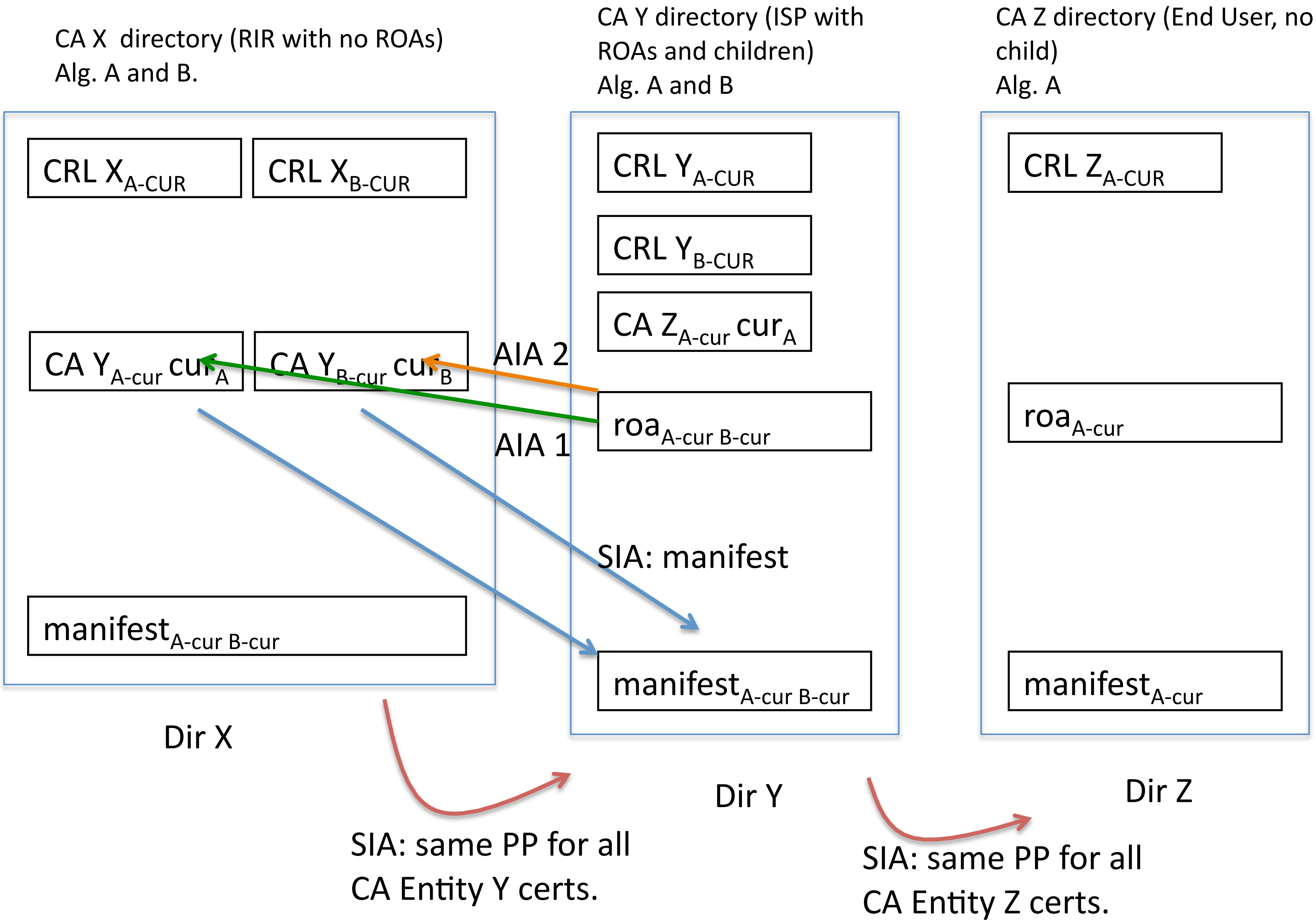
AIA

AIA

SIA: same PP for all
CA Entity Y certs.

SIA: same PP for all
CA Entity Z certs.

ALG. ROLL-OVER (without KEY ROLL OVER) Repository Structure with multiple signature ROAs and Manifests.



CA X directory (RIR with no ROAs)
Alg. A and B.

CRL $X_{A\text{-}CUR}$

CRL $X_{B\text{-}CUR}$

CA $Y_{A\text{-}cur}$ cur_A

CA $Y_{B\text{-}cur}$ cur_B

$manifest_{A\text{-}cur\ B\text{-}cur}$

Dir X

CA Y directory (ISP with
ROAs and children)
Alg. A and B

CRL $Y_{A\text{-}CUR}$

CRL $Y_{B\text{-}CUR}$

CA $Z_{A\text{-}cur}$ cur_A

$roa_{A\text{-}cur\ B\text{-}cur}$

AIA 2

AIA 1

SIA: manifest

$manifest_{A\text{-}cur\ B\text{-}cur}$

Dir Y

SIA: same PP for all
CA Entity Y certs.

CA Z directory (End User, no
child)
Alg. A

CRL $Z_{A\text{-}CUR}$

$roa_{A\text{-}cur}$

$manifest_{A\text{-}cur}$

Dir Z

SIA: same PP for all
CA Entity Z certs.

Question 3: Validation during transitions

- There are some interesting scenarios:
  - A prefix is validated using one algorithm suite (and stored in local cache) and the RP later receives a ROA signed using another algorithm set.
  - Two ROAs (identical content), each signed using different algorithm suites; one validates, the other does not. What does an RP do?

Comment: During transition both algorithm sets should be treated as equals.

Any thought on these issues or other validation issues during algorithm transition?

Summary

- Do we want to support only top-down algorithm transitions?

- Do we want to support multiple signatures in CMS signed objects?

- Any validation issue that should be addressed in transition document?

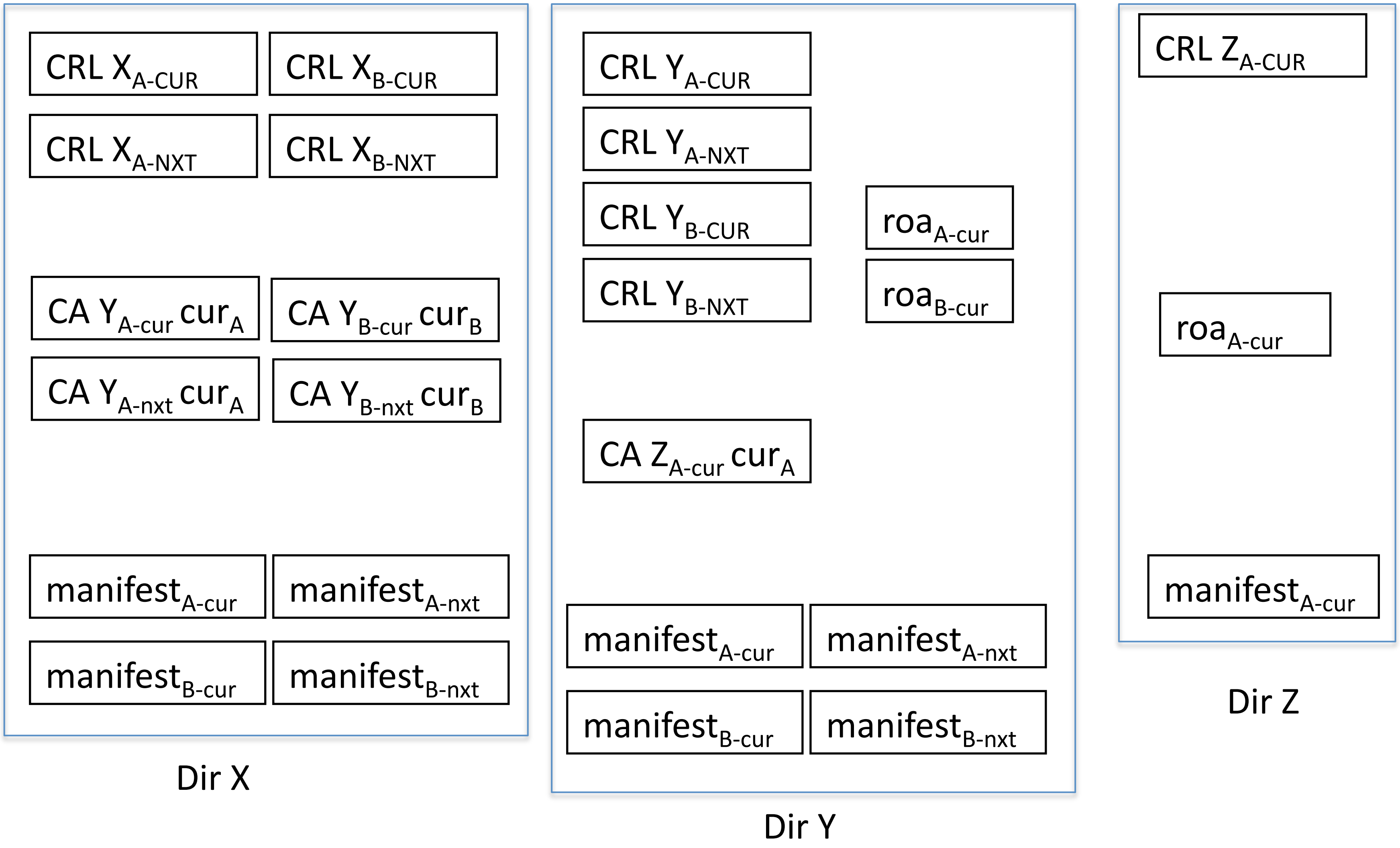
Backup

ALG. ROLL-OVER and KEY ROLL OVER Repository Structure with single signature ROAs and Manifests.

CA X directory (RIR with no ROAs)
Alg. A and B. Key Roll-over

CA Y directory (ISP with ROAs and children)
Alg. A and B. Key Roll-over

CA Z directory (End User, no child)
Alg. A . No key Roll-over



CRL $X_{\text{A-CUR}}$

CRL $X_{\text{B-CUR}}$

CRL $X_{\text{A-NXT}}$

CRL $X_{\text{B-NXT}}$

CA $Y_{\text{A-cur}}$ cur_{A}

CA $Y_{\text{B-cur}}$ cur_{B}

CA $Y_{\text{A-nxt}}$ cur_{A}

CA $Y_{\text{B-nxt}}$ cur_{B}

$\text{manifest}_{\text{A-cur}}$

$\text{manifest}_{\text{A-nxt}}$

$\text{manifest}_{\text{B-cur}}$

$\text{manifest}_{\text{B-nxt}}$

Dir X

CRL $Y_{\text{A-CUR}}$

CRL $Y_{\text{A-NXT}}$

CRL $Y_{\text{B-CUR}}$

$\text{roa}_{\text{A-cur}}$

CRL $Y_{\text{B-NXT}}$

$\text{roa}_{\text{B-cur}}$

CA $Z_{\text{A-cur}}$ cur_{A}

$\text{manifest}_{\text{A-cur}}$

$\text{manifest}_{\text{A-nxt}}$

$\text{manifest}_{\text{B-cur}}$

$\text{manifest}_{\text{B-nxt}}$

Dir Y

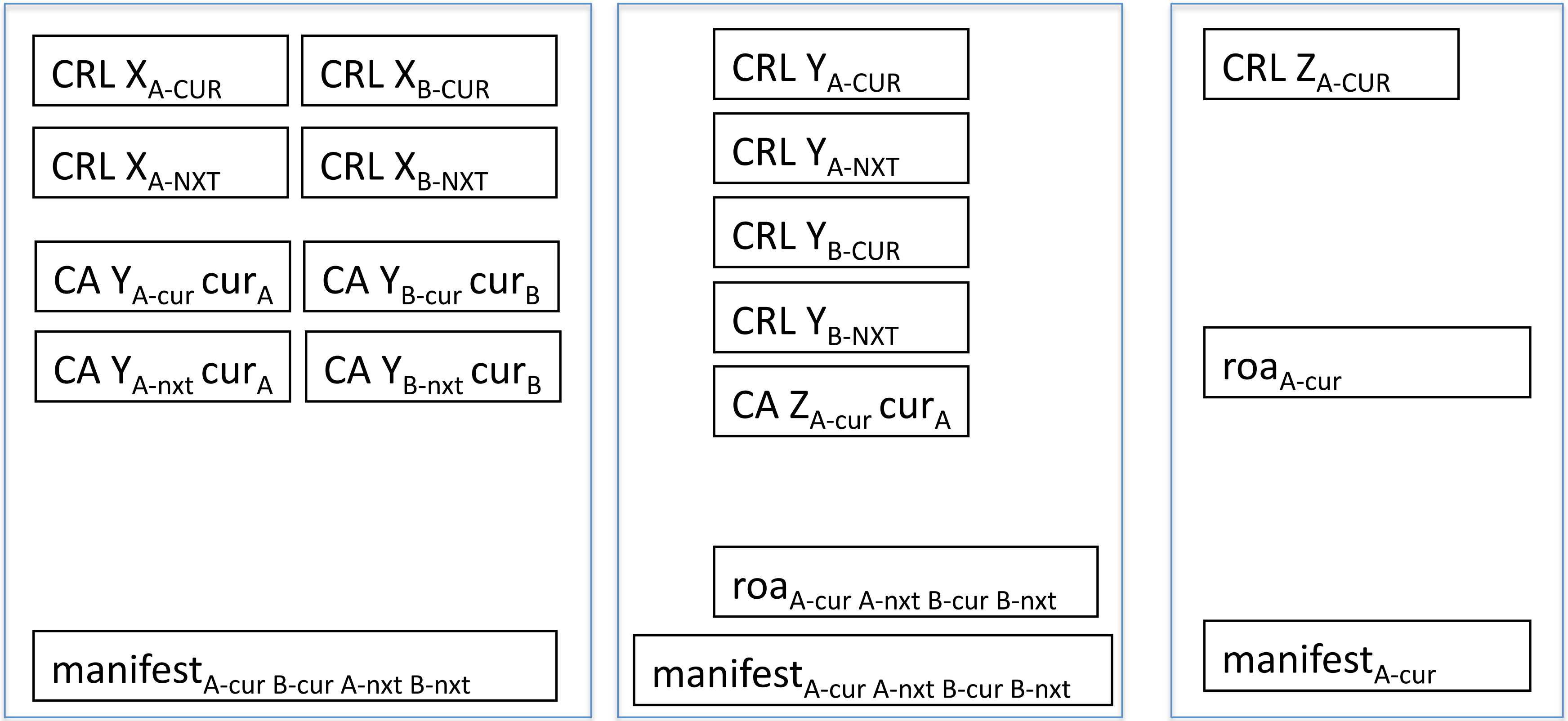
CRL $Z_{\text{A-CUR}}$

$\text{roa}_{\text{A-cur}}$

$\text{manifest}_{\text{A-cur}}$

Dir Z

ALG. ROLL-OVER and KEY ROLL OVER Repository Structure with multiple signature ROAs and Manifests.



CA X directory (RIR with no ROAs)
Alg. A and B. Key Roll-over

CRL $X_{A\text{-}CUR}$

CRL $X_{B\text{-}CUR}$

CRL $X_{A\text{-}NXT}$

CRL $X_{B\text{-}NXT}$

CA $Y_{A\text{-}cur}$ cur_{A}

CA $Y_{B\text{-}cur}$ cur_{B}

CA $Y_{A\text{-}nxt}$ cur_{A}

CA $Y_{B\text{-}nxt}$ cur_{B}

$manifest_{A\text{-}cur\ B\text{-}cur\ A\text{-}nxt\ B\text{-}nxt}$

Dir X

CA Y directory (ISP with
ROAs and children)
Alg. A and B. Key Roll-over

CRL $Y_{A\text{-}CUR}$

CRL $Y_{A\text{-}NXT}$

CRL $Y_{B\text{-}CUR}$

CRL $Y_{B\text{-}NXT}$

CA $Z_{A\text{-}cur}$ cur_{A}

$roa_{A\text{-}cur\ A\text{-}nxt\ B\text{-}cur\ B\text{-}nxt}$

$manifest_{A\text{-}cur\ A\text{-}nxt\ B\text{-}cur\ B\text{-}nxt}$

Dir Y

CA Z directory (End User, no
child)
Alg. A . No key Roll-over

CRL $Z_{A\text{-}CUR}$

$roa_{A\text{-}cur}$

$manifest_{A\text{-}cur}$

Dir Z