Certificate Repository Structure

draft-huston-sidr-repos-struct-00.txt

Basic Model

- 'named objects make other named objects'
  - Natural parent-child relationship in process
- Follow the hierarchy
  - Nodes (ie directories) and Objects (files)
  - Information management advantages
    - Local sub-trees can be independently managed
    - ‘what made me’ & ‘what do I make’ easy to find/combine into global information base
- Use g(AKI) and g(SKI) for names, CRL, '*IA'
  - Algorithmic naming, not real-entity. Tied to public key of certificate
    - (very low) risk of hash collision
    - Avoids ‘real world name’ politics

Hierarchy model

- Rooted at each Trust Anchor (TA)
  - Trust anchor self signed
  - In repository name model, AKI == SKI
  - (don’t just trust self-signed or AKI==SKI, TA must be externally defined to be valid)
- For each certificate issued
  - Sub-dir at ‘node level’ (for products of certificate) named by SKI of certificate. Sub-dir contents:
    - produced certs, CRL *(for CA certs)*
    - Signed objects *(for EE certs)*
  - Name structure stable across certificate re-issuance
    - if public key remains constant.

Trust Anchors (self-reference)

Canonical file name: `pvpjvwUeQix2e54X8fGbhmdYMo0-2F.cer`

Trust Anchor

Ipv4 10.0.0.0/8
192.168.0.0/16

serial: 2F

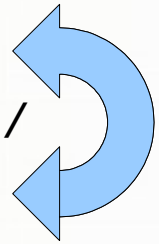
g(AKI): `pvpjvwUeQix2e54X8fGbhmdYMo0`

g(SKI): `pvpjvwUeQix2e54X8fGbhmdYMo0`

CRLdp: `rsync://repository.apnic.net/APNIC/pvpjvwUeQix2e54X8fGbhmdYMo0/pvpjvwUeQix2e54X8fGbhmdYMo0.crl`

AIA: `rsync://repository.apnic.net/APNIC/pvpjvwUeQix2e54X8fGbhmdYMo0`

SIA: `rsync://repository.apnic.net/APNIC/pvpjvwUeQix2e54X8fGbhmdYMo0`



Products of TA

Canonical file name: `DLAl5E2IJgSdp2syO9gvEeptpsI-51.cer`

Cert made by TA

Ipv4 10.0.0.0/8
192.168.0.0/16

serial: 51

g(AKI): `pvpjvwUeQix2e54X8fGbhmdYMo0`

g(SKI): `DLAl5E2IJgSdp2syO9gvEeptpsI`

CRLdp: `rsync://repository.apnic.net/APNIC/pvpjvwUeQix2e54X8fGbhmdYMo0/DLAl5E2IJgSdp2syO9gvEeptpsI/DLAl5E2IJgSdp2syO9gvEeptpsI.crl`

AIA: `rsync://repository.apnic.net/APNIC/pvpjvwUeQix2e54X8fGbhmdYMo0`

SIA: `rsync://repository.apnic.net/APNIC/pvpjvwUeQix2e54X8fGbhmdYMo0/DLAl5E2IJgSdp2syO9gvEeptpsI`

Products of (products of TA)

Canonical file name: zGJyRYzO3n7rcGV_hH-Hmn68OPY-505.cer

Cert made by Cert, made by TA

Ipv4 10.0.0.0/8

serial: 505

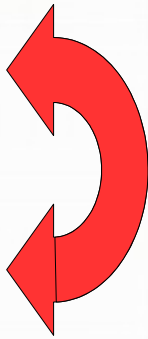
g(AKI): DLAl5E2IJgSdp2syO9gvEeptpsI

g(SKI): zGJyRYzO3n7rcGV_hH-Hmn68OPY

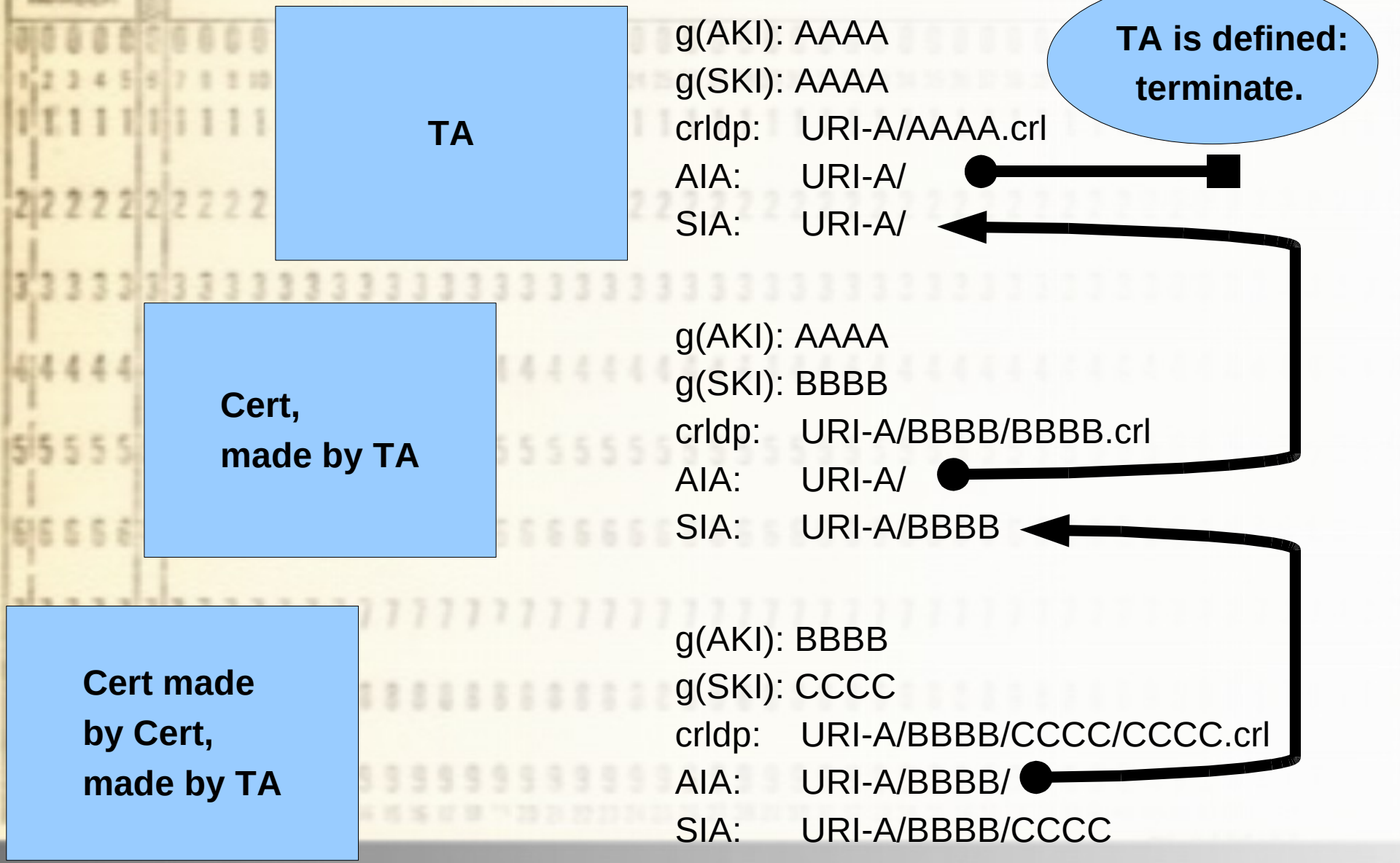
CRLdp: rsync://repository.apnic.net/APNIC/
pvpjvwUeQix2e54X8fGbhmdYMo0/
DLAl5E2IJgSdp2syO9gvEeptpsI/
zGJyRYzO3n7rcGV_hH-Hmn68OPY/
zGJyRYzO3n7rcGV_hH-Hmn68OPY.crl

AIA: rsync://repository.apnic.net/APNIC/
pvpjvwUeQix2e54X8fGbhmdYMo0/
DLAl5E2IJgSdp2syO9gvEeptpsI

SIA: rsync://repository.apnic.net/APNIC/
pvpjvwUeQix2e54X8fGbhmdYMo0/
DLAl5E2IJgSdp2syO9gvEeptpsI
zGJyRYzO3n7rcGV_hH-Hmn68OPY



path discovery to TA



Object-Name model

- For a CA
  - Your parent's SKI became your AKI.
  - Your SKI becomes your children's AKI
- Sha1 hash over ASN.1 of public key
  - Extremely low collision risk
  - g(ski) or g(aki)+<serial> unique for any given CA
  - g(ski) alone: all re-issues with same public key hash to same location (serials not involved)
    - No name change with normal certificate renewal
  - Public key change == complete namespace rollover (unavoidable)
- Represented by 'url friendly' modified base64 representation, <64 chars per cert instance

Hierarchy Implications

- Inter-RIR registration/transfers explicit in repository certificate, naming model
- Clean separation of address-types in hierarchy
  - Experimental separated from Historical from Current RIR policy from …
- Clean model for independent sub-trees
  - Provide publication point in RSYNC: URL space
  - Irrespective of local naming policy, cert identity in repository is deterministic
  - No risk of collision when uploaded into global repository structure

Modified-Gutmann Alg: g(ski)

- Use of base64 Specified in rfc4387
  - Reduces 160-bit sha1 to 27 char Base64 encoding.
    - URLs not workable in filestore contexts due to presence of '/' character.
    - '+' character interfered with URL parsing
- I-D.josefsson-rfc3548bis
  - Base64 transform, '/' and '+' replaced by '-' and '_'
    - No size increase, URL friendly, filestore friendly
  - Can be implemented as a post-process on Base64 transform, or as a native function
- Resulting object names are at least 26 chars long
  - Plus serial (up to 20 chars of HEX) plus syntactic sugar for filename.ext purposes)
  - Under 64-char limits for older FS (but clearly over 8.3)
  - Not 'mandatory' to implement as dir/file store, but useful

Why not certificate names?

- These are not identity certs
  - No applicability in browsers, webservers, email signing
  - Identity checks for issuance still required but now local to issuer, no global context
    - Unless driven by other needs
  - Can use ‘shadow cert’ process to derive certificates with apparent name for business purposes
- Avoids any consideration of entity name collision
  - in certificate namespace, encodings, field choices
  - directory name model is inherently complex
- Survivable across future models of address transfer/trading
- “it’s the crypto, not the name which secures”
  - Proof of knowledge of private key authorizes signed outcomes, not the name on the certificate in this model

Operations models

- Authenticated copy
  - TA **must be** sourced out-of-band.
  - Fetch repository via rsync
    - Crl from CRLdp of TA
    - Repository contents via AIA of TA
  - Top-down walk to validate all objects published at TA
    - Recurse for independent SIA/CRLdp
- Local sub-repository
  - Requires at LEAST path back to issuing TA to perform validation

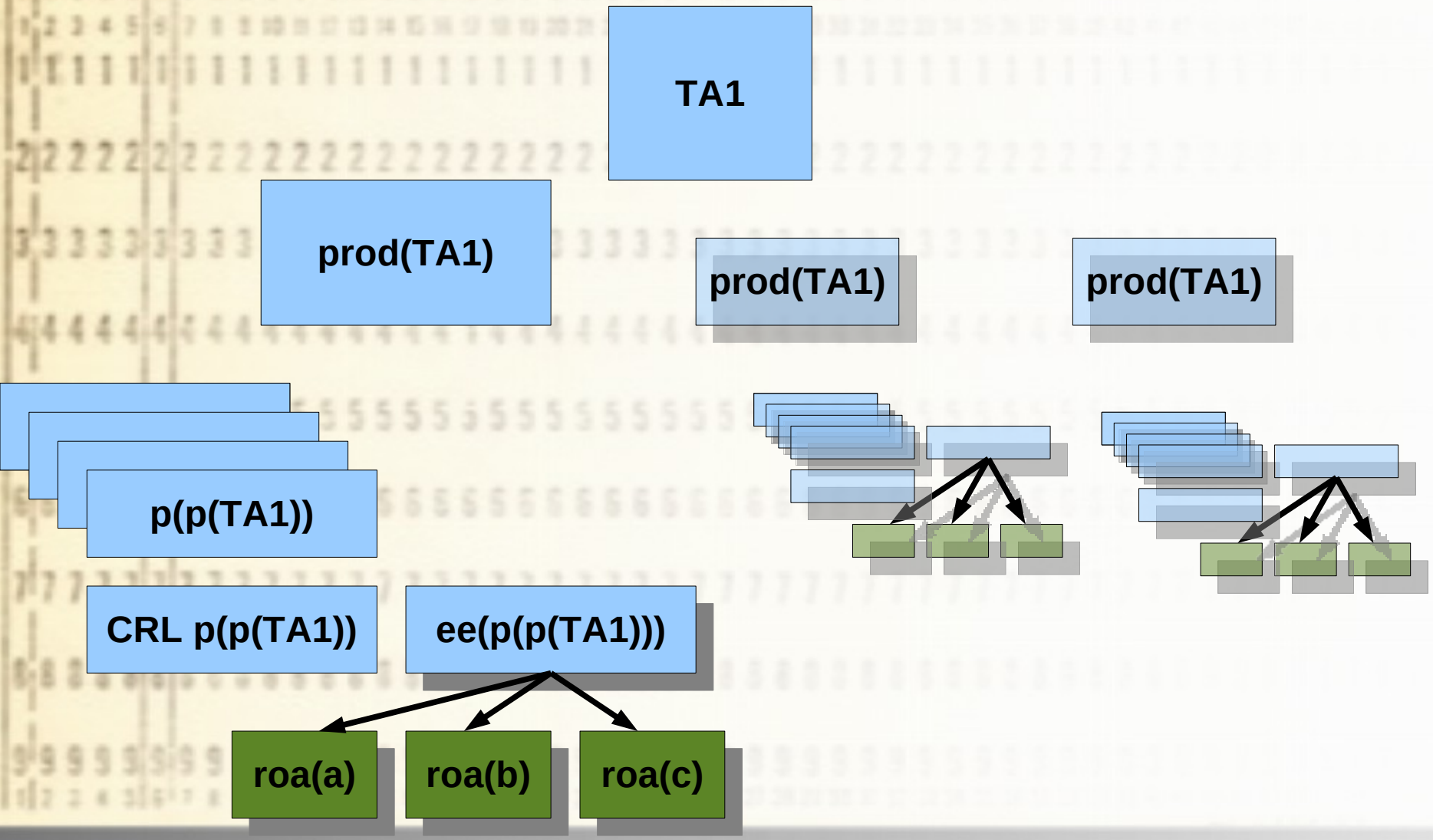
Why Rsync?

- Avoid inventing new protocol
- Desireable features
  - Can fetch single objects, trees
  - Byte efficient (only differing blocks)
  - No fetch of unchanged objects
- Downsides
  - May be expensive on server
  - Lax formal specification

What do we get?

- Deterministic, simple naming
- Objects always have a known place in hierarchy
- EE certs, CRLs, 'products' nest cleanly inside 'producer' namespace
- Clean separation of certs by delegation (right back to TA == 'root')

Natural hierarchies form



...but you inherit multiple TA