A Publication-Point-Based Incremental Validation Procedure for RPKI Relying Parties

draft-su-sidrops-rpki-rp-incremental-validation-00

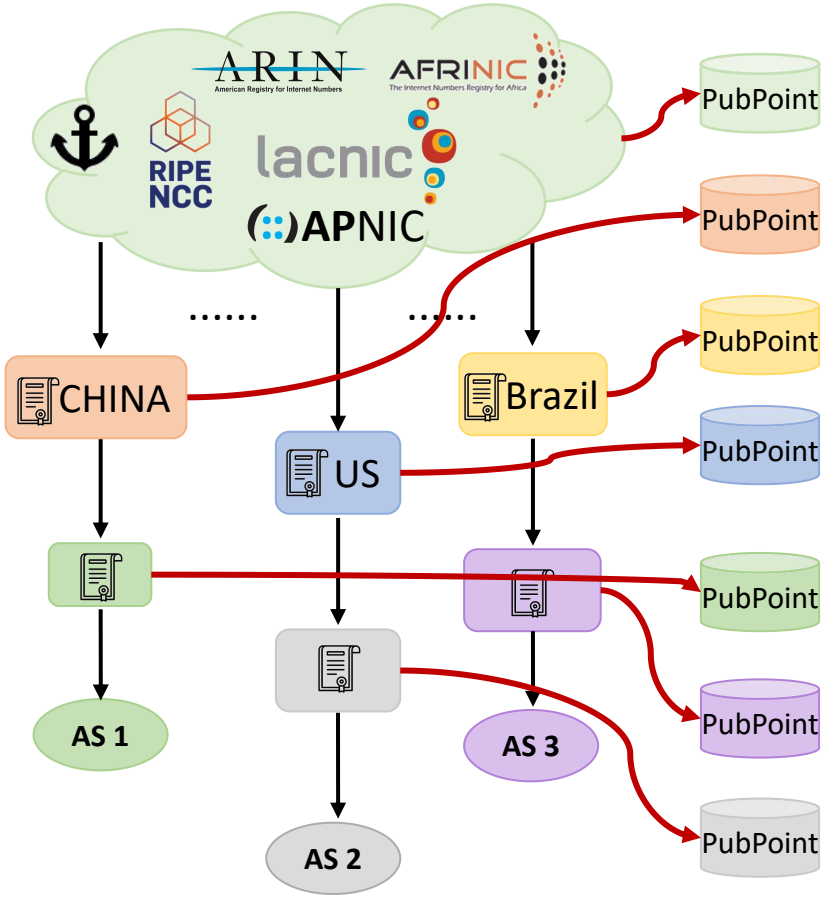
Yingying Su, Lancheng Qin, Dan Li

ZGC lab & Tsinghua university

July, 2026

Problem Statement & Key Idea

RPKI Follows Top-down Validation



Problem Statement

Validation, not data transfer, becomes the bottleneck.

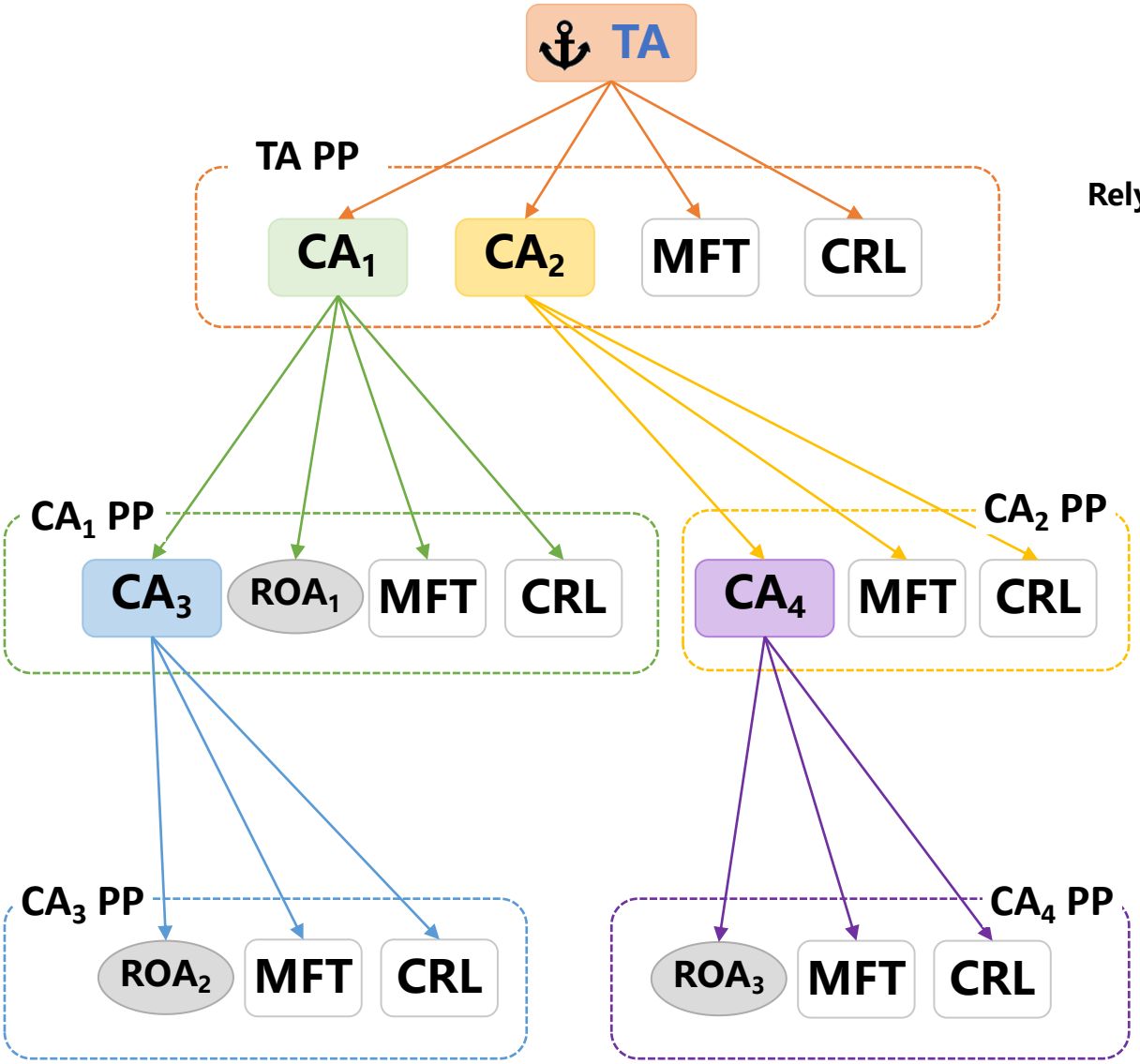
- RP repeats full top-down object validation, although most CA PPs and objects are unchanged during incremental updates.
- Repeating validation for objects whose validation inputs have not changed can introduce unnecessary validation overhead.

Core Idea

Reuse validation results when validation inputs are unchanged.

- Cache validation states for each publication point.
- Reuse CA PP cache states when relevant validation inputs are unchanged.
- Fall back to CA PP full validation when needed.

Validation Unit and CA Publication Point Cache





Relying Party

① validate TA PP , accept CA_1 & CA_2
② validate CA_1 PP , accept CA_3
③ validate CA_2 PP , accept CA_4
④ validate CA_3 PP
⑤ validate CA_4 PP

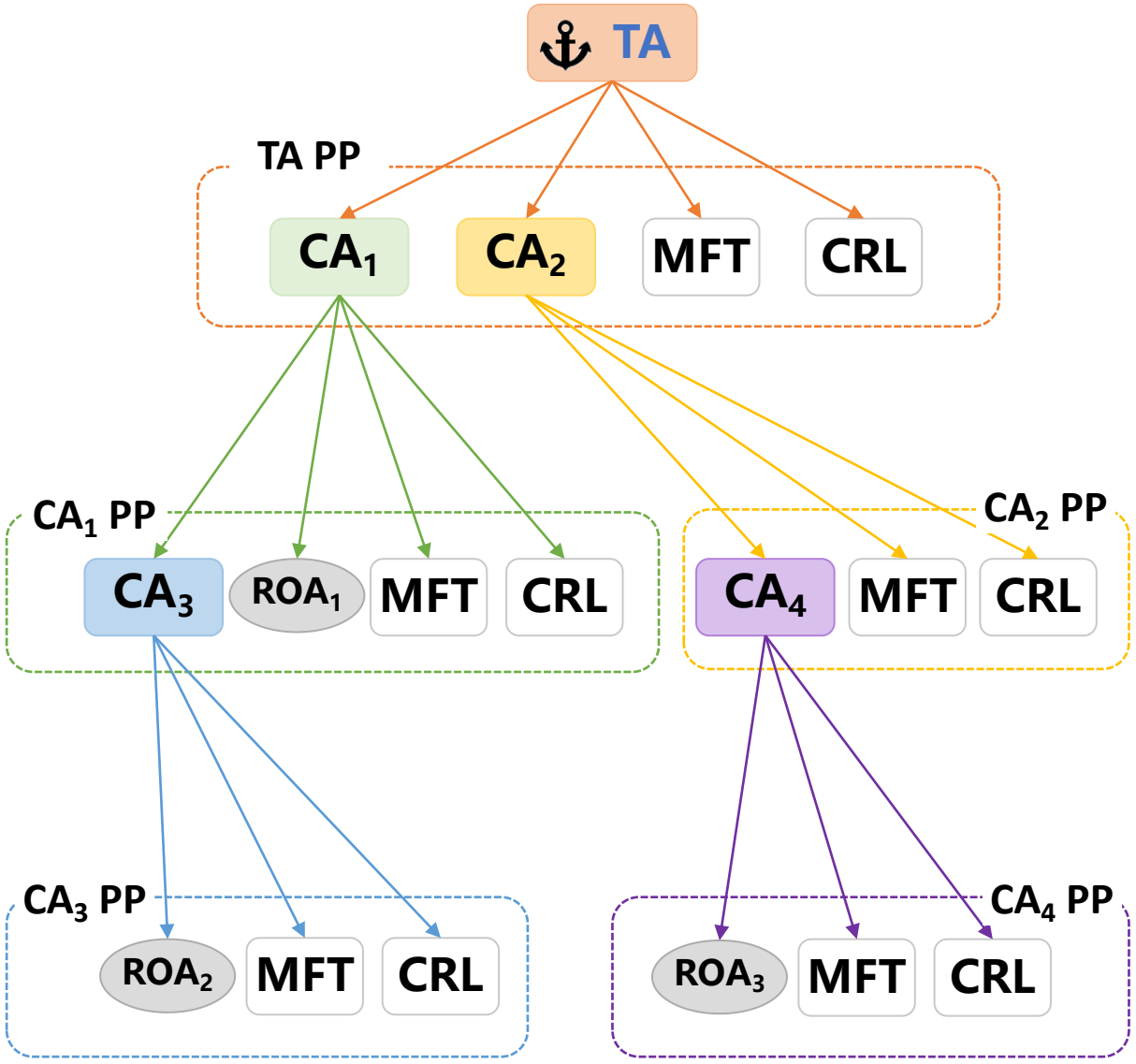
model RP validation as a function that repeatedly **ValidatePP(CA)**

CA PP Cache After Initial Full Validation

- **Key:** CA_ID = (CA certificate URI, hash(CA certificate object))
- **Value:** PublicationPointState (CA_ID)
- **PublicationPointState** {
  ca_effective_resources,
  manifest validation state,
  CRL validation state,
  per-object validation state,
  }

each **valid CA publication point** gets a **cache entry**

Incremental Validation Flow



PP Cache Example

TA →
ca_effective_resource,
mft: {uri,hash, thisUpdate, nextUpdate},
crl: {uri, hash, thisUpdate, nextUpdate},
objects: {(file_hash) → ObjectValidationState}

CA_1 → ...
CA_2 → ...
CA_3 → ...

RP Processes Pipeline

TA	CA_1	CA_2	CA_3	CA_4
hit	hit	hit	hit	miss

Core Procedure

1. New CA or changed CA certificate
→ full validation
2. Same CA but changed effective resources context
→ full validation
3. Same CA, same resources, same manifest and CRL
→ PP-level cache reuse
→ re-check validation time
4. Same CA, same resources, changed manifest
→ object-level cache reuse
→ re-check validation time and CRL status before reuse

Implement Results

RP Software	Snapshot	10min Delta avg	15min Delta avg	30min Delta avg	60min Delta avg
ours w/o cache	138s	94s	94s	95s	98s
ours w cache	**138s**	**47s**	**48s**	**49s**	**51s**
other RPs	162s ~ 564s	93s ~ 465s	97s ~ 469s	111s ~ 475s	113s ~ 490s

Key Findings

- 99.4% CA PP-level hit rate after a 15-minute interval.
- 49%–90% lower incremental update time compared with RP validation without the state cache.
- Incremental validation reduces repeated validation over unchanged PP state.

Summary & Discussion

Main Bullets

- Specifies an optional incremental validation procedure for RPKI RPs.
- Uses PP-level reuse first, object-level reuse when needed.
- Does not change RPKI object formats or validation semantics.
- Preserves the same validated output as full top-down validation.

Contact & Related Document

- Yingying Su (suyy@mail.zgclab.edu.cn)
- Lancheng Qin (qinlc@mail.zgclab.edu.cn)

- draft-su-sidrops-rpki-rp-incremental-validation

Thank you!