V’CER
Efficient Certificate Validation in Constrained Networks

David Koissier, Patrick Jauernig, Gene Tsudik, Ahmad-Reza Sadeghi
Motivation: Enable PKI in Constrained Networks

- Anemic computation & storage abilities
- Limited bandwidth
- Long range, low power antennas
- Devices hibernating
- Intermittent connectivity
- Moving nodes
- Dynamic topology
- Strict power budgets
Motivation: Enable PKI in Constrained Networks

Anemic computation & storage abilities
Limited bandwidth
Dynamic topology
Strict power budgets
Long range, low power antennas

Intermittent connectivity
Devices hibernating

Trust establishment via Public Key Infrastructure (PKI) is hard under these conditions
Specifically, checking the up-to-date revocation status of certificates
Existing Revocation Checks Not Sufficient

Online Certificate Status Protocol (i.e. on-demand check)
Existing Revocation Checks Not Sufficient

Online Certificate Status Protocol (i.e. on-demand check)

unreliable, expensive communication
Existing Revocation Checks Not Sufficient

Online Certificate Status Protocol (i.e. on-demand check) unreliable, expensive communication

Certificate Revocation Lists CA
Existing Revocation Checks Not Sufficient

Online Certificate Status Protocol (i.e. on-demand check)

- unreliable, expensive communication

Certificate Revocation Lists

- heavy on storage and communication
Existing Revocation Checks Not Sufficient

Online Certificate Status Protocol (i.e. on-demand check)
unreliable, expensive communication

Certificate Revocation Lists
heavy on storage and communication

Modern alternatives to Lists (e.g. CRLite, Let's Revoke)

CA
Existing Revocation Checks
Not Sufficient

Online Certificate Status Protocol (i.e. on-demand check)
- unreliable, expensive communication

Certificate Revocation Lists
- heavy on storage and communication

Modern alternatives to Lists (e.g. CRLite, Let’s Revoke)
- Do not handle nodes missing updates well
Existing Revocation Checks Not Sufficient

Online Certificate Status Protocol (i.e. on-demand check)

Certificate Revocation Lists

Modern alternatives to Lists (e.g. CRLite, Let's Revoke)

CA

unreliable, expensive communication

heavy on storage and communication

Do not handle nodes missing updates well

Biggest Problem: Lots of calling home puts large demand on network
Our Solution: V’CER

- Flexible & lightweight revocation checks to enable PKI in constrained networks
- Epidemic dissemination of validation information
- Novel algorithms allowing devices to keep each other up-to-date
- Evaluation on a real satellite and large-scale simulation
V‘CER Design
(Simplified for Brevity)
Naïve Approach

- Merkle Tree aggregates set of active certs
V’CER Overview

Naïve Approach

- Merkle Tree aggregates set of active certs
- Tree root hash is validation information
**V’CER Overview**

**Naïve Approach**

- Merkle Tree aggregates set of active certs
- Tree root hash is validation information
- Proof of Inclusion: path from leaf to root
V’CER Overview

Naïve Approach

- Merkle Tree aggregates set of active certs
- Tree root hash is validation information
- Proof of Inclusion: path from leaf to root
- Proof of Inclusion proves non-revoked cert
Naïve Approach

- Merkle Tree aggregates set of active certs
- Tree root hash is validation information
- Proof of Inclusion: path from leaf to root
- Proof of Inclusion proves non-revoked cert

Problem:
On any changes (revocation / issuance), all Proofs of Inclusion become invalid
V’CER Overview

Naïve Approach

Validation Forest data structure*

V’CER Approach

*see paper for details
V’CER Overview

Naïve Approach

Tree Root

Validation Forest data structure*

Sparse Merkle Tree

V’CER Approach

*see paper for details

Deterministic structure
V’CER Overview

Naïve Approach

Tree Root

Validation Forest data structure*

Sparse Merkle Tree

Distributed Repair algorithms

V’CER Approach

*see paper for details

Deterministic structure

Collaborative updating
V’CER Example

proof of Inclusion

CA
V’CER Example

Cert got revoked!
Tree changed!
V’CER Example

new root

proof update

proof of Inclusion

root up-to-date

Cert got revoked! Tree changed!
V’CER Example

CA
V’CER Example

CA
V’CER Example

CA
V’CER Example
V’CER Example

Topology independent, epidemic spread for fast revocation enforcement
Preliminary: Sparse Merkle Tree

Most parts “empty”, can be omitted

All possible outputs of SHA256 as leaves, i.e., $2^{256}$ leaves

0xA4...81
0xA0...D4
0x7B...15

0x00...00
0x00...01
0x00...02

Ø
Ø
Ø
Ø
Ø
Distributed Repair of Proofs

Deterministic position in tree

Tree Root
Distributed Repair of Proofs

- Deterministic position in tree
- Tree Root
- Meet up-to-date node
- Shares its proof
Distributed Repair of Proofs

- Deterministic position in tree
- Tree Root
- Missing hash can be calculated
- Meet up-to-date node
- Shares its proof

Diagram showing a tree structure with nodes representing positions in the tree. The root node is identified as the central point from where hashes can be calculated. Each node shares its proof with the up-to-date node.
Use Case: Satellite Networks

• Space sector is growing dramatically 🚀

• Key Trend:
Large networks of small satellites (opposed to individual big ones)
Evaluation in Space

• Collaboration with the European Space Agency
• OPS-SAT:
  CubeSat open for anyone to register and upload experiments
• Arm Dual-Core Cortex-A9 @ 800MHz
• Experiments run on Java Framework
• V’CER open source proof-of-concept deployed and tested onboard
Evaluation: Large-Scale Simulation

• Up to 1 million nodes

• Each day certs get revoked and re-issued (~10% yearly)

• Each week freshly issued certs (~5% yearly)

• Each node randomly meets 5 nodes per hour (each encounter tries distributed repair)

• Percentage of nodes miss CA update (10%, 30%, or 50% of nodes miss update)
Results

Avoids >93% of outdated nodes to *call home* for update

<100KB communications overhead between nodes (per node per week)
Conclusion

- Merkle tree root hash as validation information
- Epidemic spread for fast revocation enforcement
- Proof of Inclusion as non-revoked proof
- Distributed repair algorithms allow up-to-date nodes to help outdated nodes
- Probabilistic approach: large share can be repaired
- Only uses contact messages: topology independent
Thank you!
Questions?

Contact me:
david.koisser@trust.tu-darmstadt.de