Policy Externalization in 5G-PKI: Enabling Scalable and Quantum-Ready IoT

Manish Paudel · Maryna Veksler · Kemal Akkaya
(paudelm· vekslerm · akkayak)@vcu.edu

ADVANCED WIRELESS & SECURITY LAB (ADWISE)
Virginia Commonwealth University • Richmond, VA

MAgiCS , 05/10/2026 Rome, Italy

VCU Computer Science
College of Engineering

ADVANCED WIRELESS SECURITY ADWISE

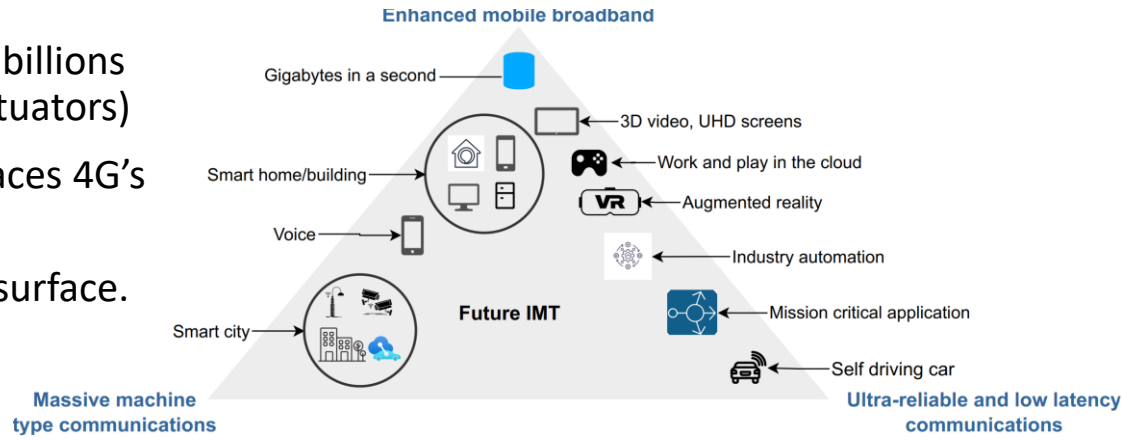
Overview

- Introduction
- Background
- Problem Statement
- Our Solution: External Policy Management (EPM)
  - System Architecture
  - Operational Protocols
- Security Guarantees
- Experimental Analysis
- Conclusion & Future Work

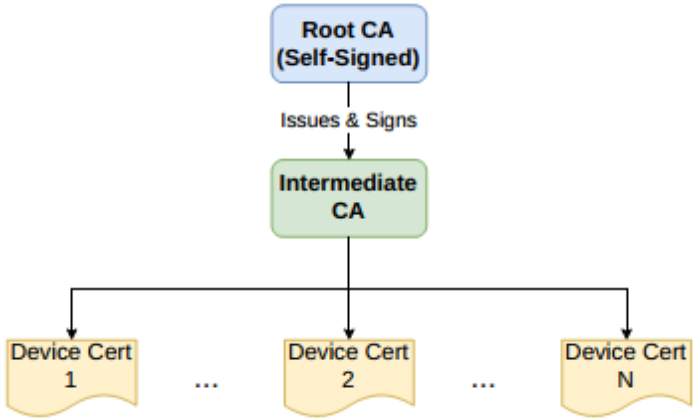
Introduction

The convergence of 5G, IoT, and Quantum Computing

- **The 5G-IoT**: 5G's mMTC case connects billions of unattended IoT devices (sensors, actuators)
  - Machine-to-machine model replaces 4G's human-centric design.
  - Increased overall security attack surface.
- **PKI:** 5G uses hierarchical X.509 PKI to authenticate every device
  - Root CA→Intermediate CA→Device certs
- **Converging Pressures:**
  - **Scale:** Billions of certs to manage
  - **Quantum threat:** Large PQC signatures.
  - **Resource Problem:** Limited IoT resource.
- **The Question:** Can today's PKI handle all certificate management including efficient revocation?



5G Use Case Services



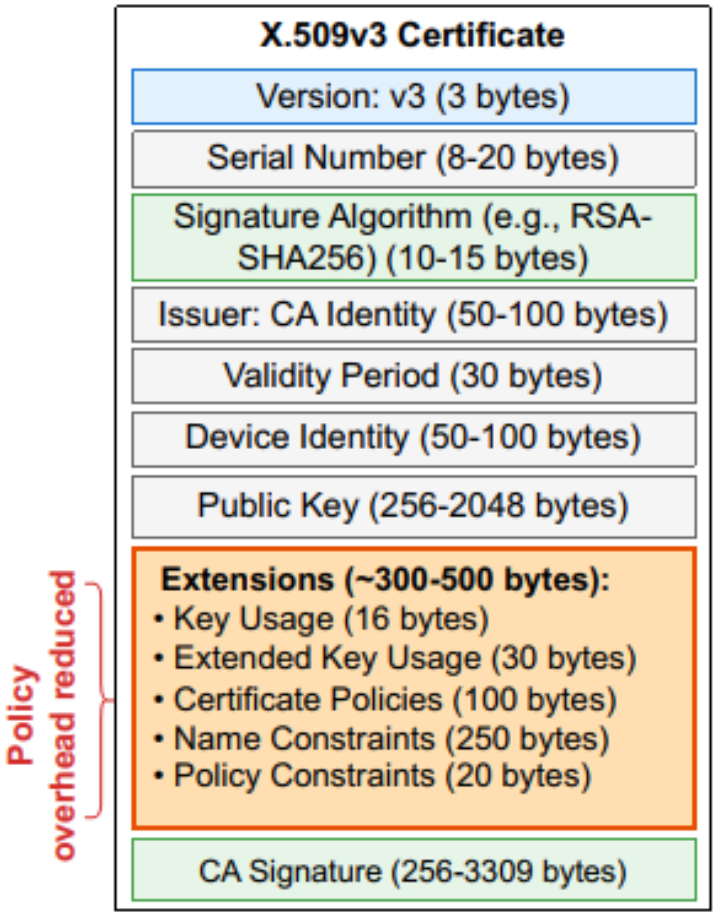
PKI

Background

X.509 Certificates · Embedded Policies · Revocation

X.509 Certificate Structure

- **X.509 v3 Certificates:** Binds cryptographic key → device identity via CA signature.
- **Policy Extensions:** Embedded directly
  - Key Usage, Extended Key Usage
  - Certificate Policies
  - Name Constraints, Policy Constraints
- **Adds 300-500 bytes of policy overhead per certificate**
- **Revocation Mechanism:**
  - CRL grows linearly (26B/entry)
  - OCSP adds latency



Background

PQC · 5G IoT

- **Quantum threat**
  - Shor's algorithm breaks RSA/ECC
- **NIST PQC Standards**
  - Significantly Larger signatures

Algorithm	Signature Size
RSA-2048	256 B
ECDSA P-256	64 B
ML-DSA	2,420 B
Falcon-512	666 B

- **Compounding Problem**
  - Larger PQC Signature
  - Existing 300-500 B of embedded policy overhead
  - Frequent reissuance from policy updates
- **PQC infeasible for resource-constrained IoT**

Problem Statement

Static Trust · Policy Rigidity · Revocation Cascade

Current PKI uses "Static Trust": Policies are cryptographically fused to identity in the certificate

Five concrete problems

- **P1- Policy Redundancy:** N identical devices each carry same 300-500 B policy →hundreds of MB of duplicate data
- **P2- Immutable Policy Coupling:** Any policy change invalidates the signature. A single update requires revoking + reissuing N certificates
- **P3- Linear CRL Growth:** ~26B per revocation →CRLs reach 260 MB at 10M revocations.
- **P4- Policy-Induced Revocations:** Routine updates trigger mass revocation cascades unrelated to security.
- **P5- PQC Integration:** Larger PQC signatures + policy bloat = bandwidth-prohibitive

Revocations	CRL Size	Download(1 Mbps)
10,000	260 KB	2.1 sec
100,000	2.6 MB	21 sec
1,000,000	26 MB	3.5 min
10,000,000	260 MB	35 min

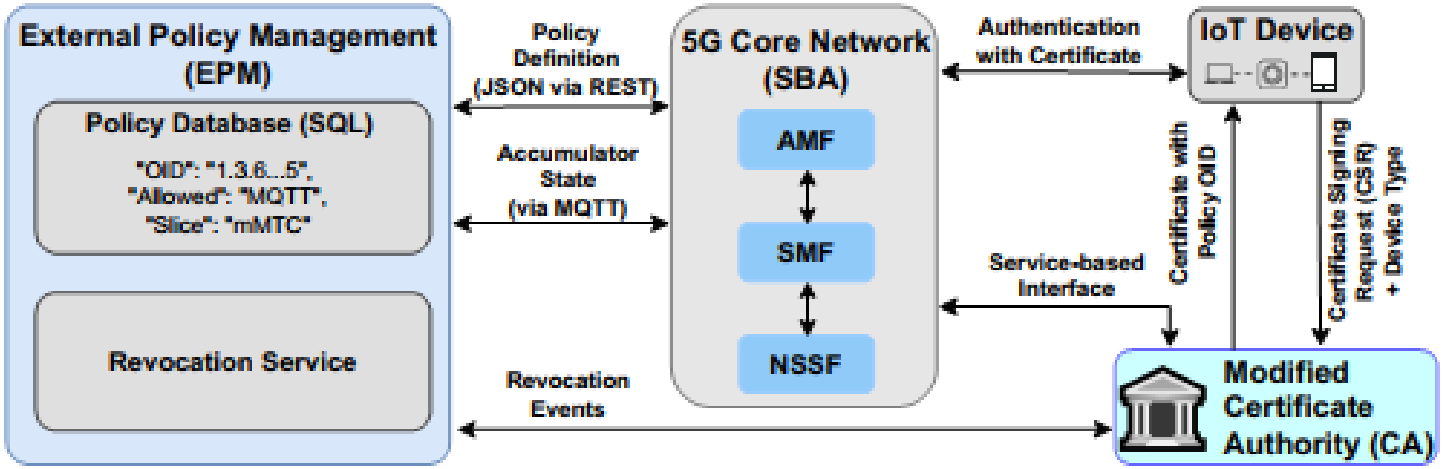
Goal: Achieve **policy agility** while remaining backward-compatible with X.509/TLS/3GPP

Our Solution

External Policy Management (EPM) Framework

Key Idea: Decouple policy from identity by replacing 300-500 B of policy extension with 80 B OID linked to a mutable policy in an external database.

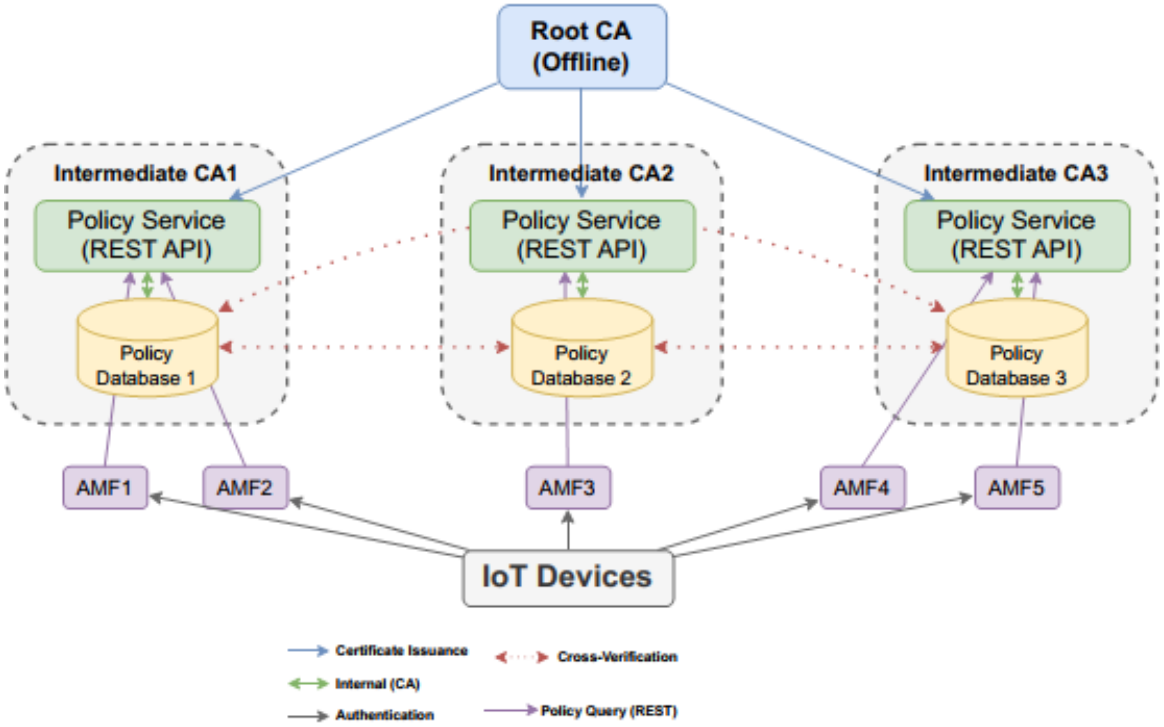
- **EPM Framework:** separates policy from cert
- **Regulated Revocation:** Eliminates policy driven revocations.
- **Practical PQC Integration:** Reclaims space, restricts large-cert transmission to enrollment + key compromise only
- **Implementation + Evaluation** on open-source 5G stack



- **New logical entities: EPM + Modified CA**
- **NFs query EPM via REST API with caching without TLS/hardware changes**

Deployment Topology

Multi-CA Deployment with Cross-Verification

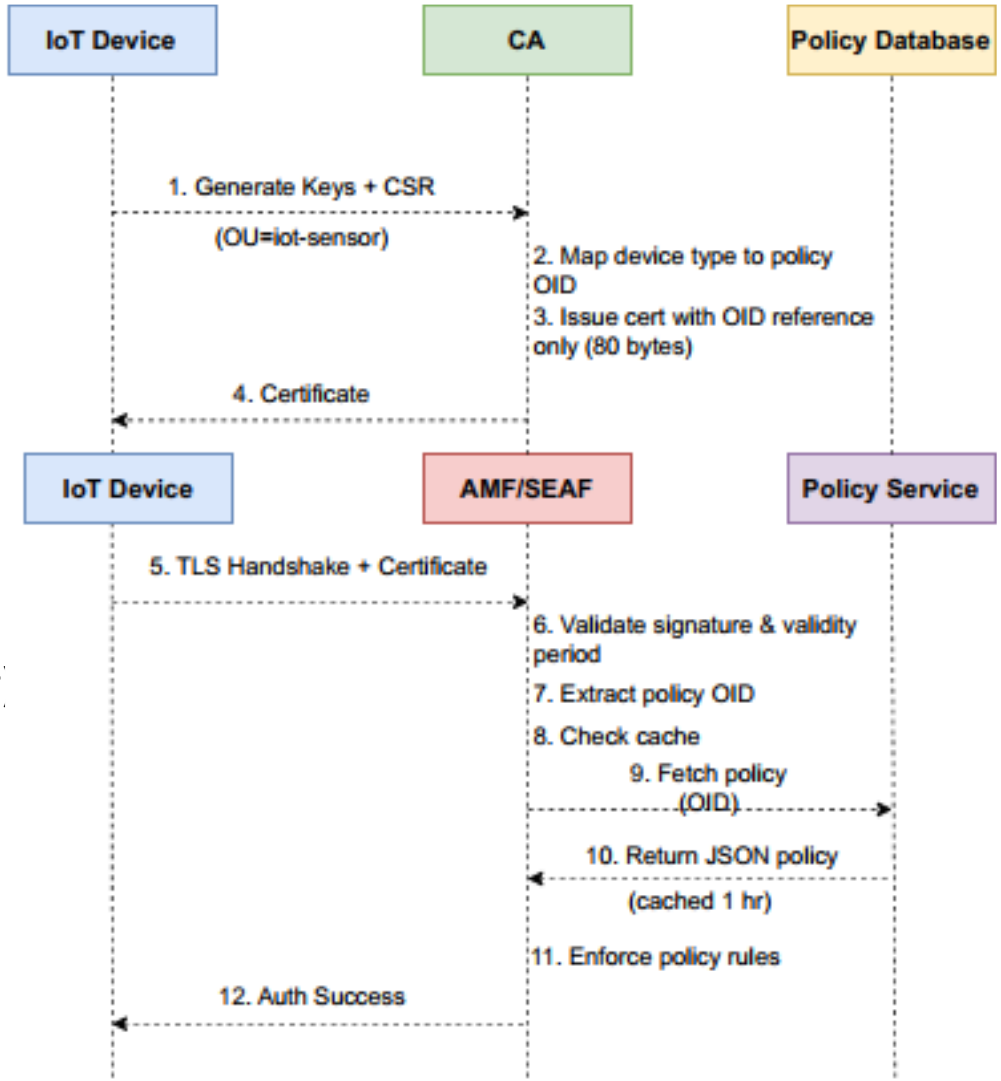


- **3 Intermediate CAs** with co-located policy DBs
- **Periodic cross-CA hash comparison** detects tampering (T1, T4)
- **Byzantine tolerance**: System secure if < k of n CAs compromised
- **Policy DB inherits CA's HSM**, physical security, audit logging

Our Solution

Operational Protocol 1: Enrollment + Authentication Flow

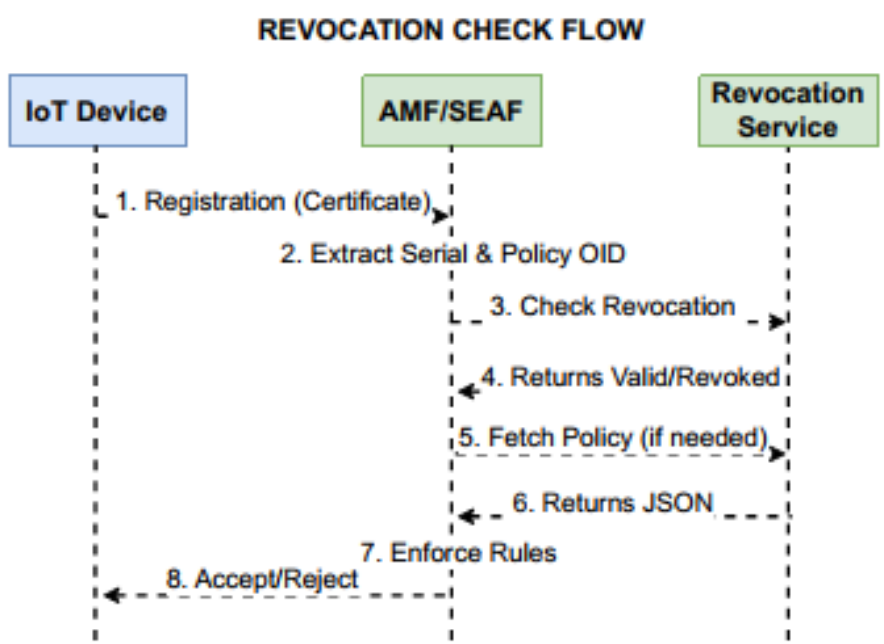
- Generate Keys + CSR
- Map device type → Policy OID
- Issue cert with OID reference only (80 B)
- Certificate (OU=iot-sensor)
- TLS Handshake + Certificate
- Validate signature & validity
- Extract policy OID
- Check cache
- Fetch policy by OID
- Return JSON policy (cache 1 hr)
- Enforce policy rules
- Auth Success ✓

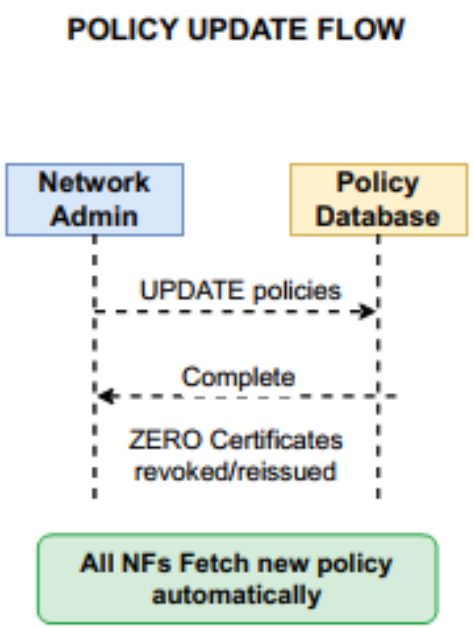


Our Solution

Operational Protocol 2: Revocation & Policy Updates

IoT Device → AMF → Revocation Service → Policy Service → Accept/Reject





- ❑ Revocation works with **CRL, OCSP, or Accumulators:** independent of our contribution.
- ❑ Only legitimate revocations (key compromise, decommissioning): no policy-induced cascades.

- Network Admin → Policy Database: UPDATE policies
- Complete in milliseconds
- **ZERO certificates revoked / reissued**
- All NFs fetch new policy automatically (after cache TTL)

Security Analysis

How EPM mitigates Threats

- **Threats are categorized from T1-T6 based on Dolev Yao Model**

Threat	Defense	Outcome
T1 / T4 - DB tampering / malware	k-of-n threshold signing + cross-CA hash check	✓ Detected in ~3.5 min
T2 - SQL Injection	Parameterized queries	✓ Blocked
T3 - Unauthorized policy update	Threshold k-of-n CA signatures required	✓ Blocked
T5 - Complete DB deletion	NF policy cache + DB restore from majority	✓ Service continues
T6 - MitM on policy retrieval	mTLS + per-policy CA signatures	✓ Blocked

- **Certificate integrity preserved**: same EUF-CMA signature scheme as traditional PKI.
- **Policy integrity**: A policy P is valid only if signed by ≥ k CAs.

Experimental Analysis

Implementation and Testbed

❑ Testbed

- **3 × Dell Precision Towers**
  Intel i7 · 64 GB RAM · 1 TB NVMe SSD
  1 × Laptop (UE)
- **Network**: 1 Gbps LAN
- **OS**: Ubuntu 20.04 LTS

❑ Implementation

- **Python 3.x** : CA, AMF, IoT components
- **liboqs** : PQC (CRYSTALS-Dilithium2)
- **SQLite** : policy storage (JSON)
- **pycryptodome** : cryptographic accumulators
- REST API + HTTP cache (TTL: 1 hr)

❑ Cryptographic Configurations

- **RSA-2048** (classical baseline)
- **ECDSA P-256** (elliptic-curve baseline)
- **ML-DSA (Dilithium2)** : NIST PQC
- Hash: SHA-256 across all schemes

❑ Three Experiments

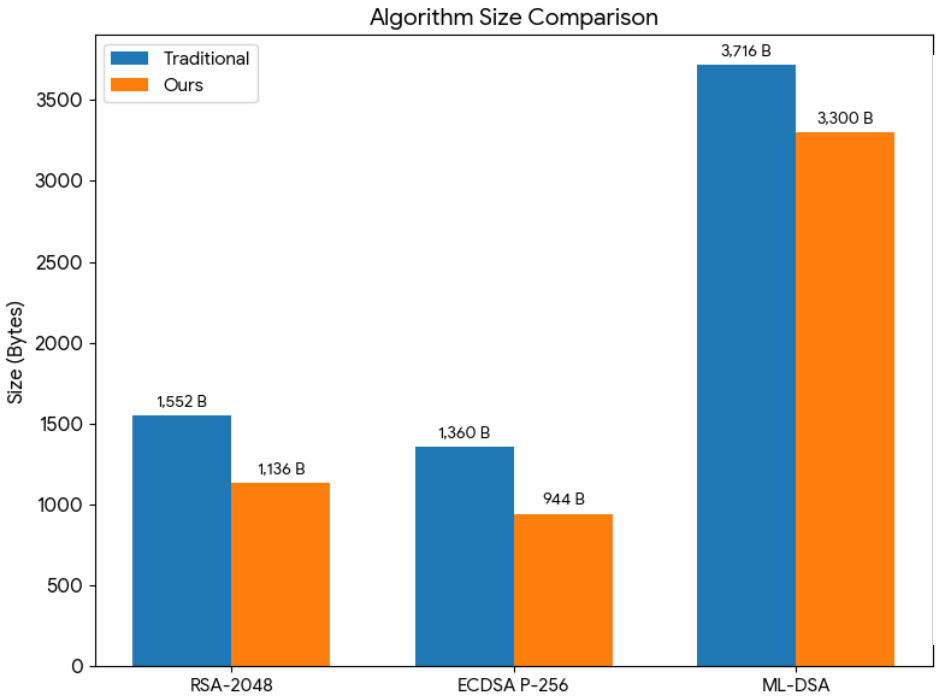
- **Certificate size** & transmission overhead
- **Authentication latency** (CRL vs. accumulator)
- **Policy update efficiency** (1K–1M devices)

❑ **Goal:** Identical implementation, fair head-to-head comparison vs. traditional PKI

Experimental Analysis

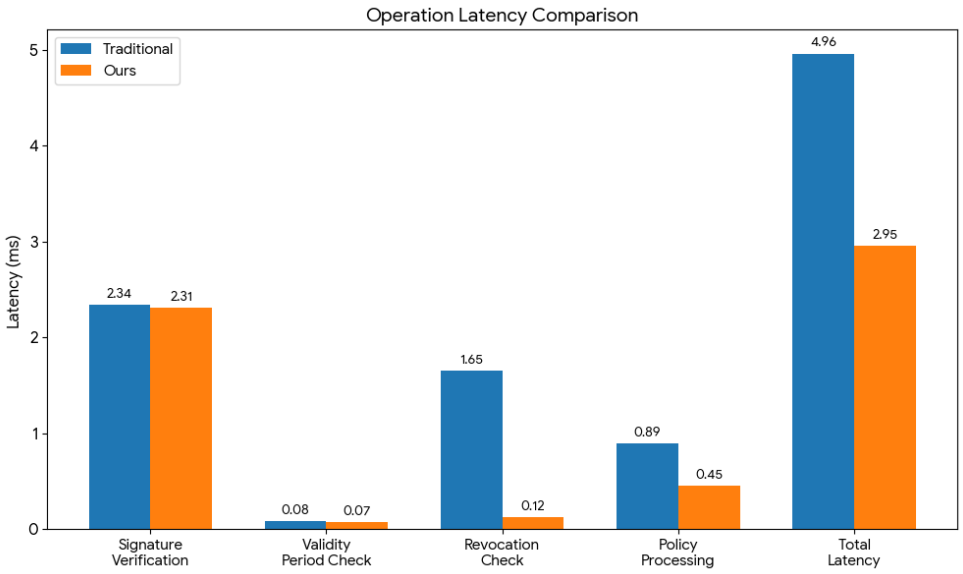
Certificate Size & Authentication Latency

1. Certificate Size



- Fixed 416B saving per certificate
- RSA → −26.8%
- ECDSA→ −30.6%

2. Authentication Latency



- End-to-End Auth: 40.5% faster
- Constant-size revocation proofs scale with fleet size

Experimental Analysis

Scalability and comparison · Where our work stands

"**14.3M× faster** at 1M devices" | "**0 certs revoked**" | "**0 downtime**"

Devices	Traditional	**Ours**	Speedup
1,000	2.3 min	9.7 ms	**14,319×**
10,000	23.2 min	9.7 ms	**143,505×**
100,000	3.9 hours	9.7 ms	**1.4M×**
1,000,000	**38.6 hours**	**9.7 ms**	**14.3M×**

Where Our Work Stands !

Approach	Policy Agility	Cert Size Reduction	PQC-Ready	Backward Compatible
Traditional X.509 PKI	✗	✗	✗	✓
Hierarchical PKI [20]	✗	✗	✗	✓
Lightweight PKI [33]	✗	Partial	✗	✓
Blockchain PKI [23]	✓	✗	✗	✗
Our EPM Framework	✓	✓	✓	✓

Conclusion

- **The Problem**
  5G-IoT PKIs embed policy in certs → cert bloat, revocation cascades, and PQC infeasibility at scale.
- **Our Solution**
  External Policy Management (EPM): replace embedded extensions with **80 B OID references** to a mutable policy DB.
- **The Results**
  30% smaller certificates · 40.5% faster authentication · 14.3M× faster policy updates (1M devices) · < 10 min recovery from compromise.
- **The Impact**
  Reclaims certificate space for PQC; eliminates policy-induced revocations; backward-compatible with X.509, TLS, and 3GPP standards.

Take Home Message: "Decoupling policy from identity is the missing primitive for scalable, quantum-ready 5G/6G PKI."

Future Work

- **PQC-Native Policy Signing**
  Sign policy documents with hash-based or lattice-based PQC; hybrid certificates during transition.
- **3GPP Standardization Path**
  Propose Certificate Policies extension semantics to 3GPP SA3 working group for 6G.
- **Distributed Policy Database**
  Replace centralized SQL with consensus-based replicated DB for higher availability
- **Formal Verification**
  Mechanized proofs of EPM security properties in Tamarin / ProVerif.

Acknowledgement

❖ *This research was funded by US NSF Grant No. 2147196*

Questions

QUESTIONS?