Secure Firmware Update and Software Signing in Embedded Devices

Authors: Oscar Jiang, James Ni, Xin Qiu, Ting Yao, Lisa Yin

Speaker: Xin Qiu

PKI Center™, Aurora Networks™

www.pki-center.com

Agenda

ICMC26

Core Challenges

- Constraints, scale, and lifecycle realities

Architectural Building Blocks

- Root of trust, signing infrastructure, update verification

Lessons from Deployment

- Where systems fail and why

Conclusions

Core Challenges in Practice

Trust vs. Flexibility

- Immutable hardware roots constrain what can change later

Security vs. Constraints

- Embedded limits across storage, compute (crypto & policy), connectivity, recovery

Single Product vs. Fleet Scale

- Single-product assumptions fail across multi-vendor, product & long-lived fleets

DIY vs. Established Service

- Signing, key management, policy enforcement, and auditability require specialized expertise and sustained operational investment

Single-Product Thinking vs. Fleet-Scale Reality

What works in a limited deployment

- One product line, one boot flow, one signing key
- Tightly controlled team and release process

What breaks at fleet scale

- Inconsistent policies and key usages across vendors, products and firmware modules
- Temporary exceptions and workflow shortcuts become permanent

System implication

- Signing and update architecture must be designed for **multi-vendor**, **multi-product, long-lived fleets** from the start

Device Root of Trust & Chain Continuity

Hardware root of trust

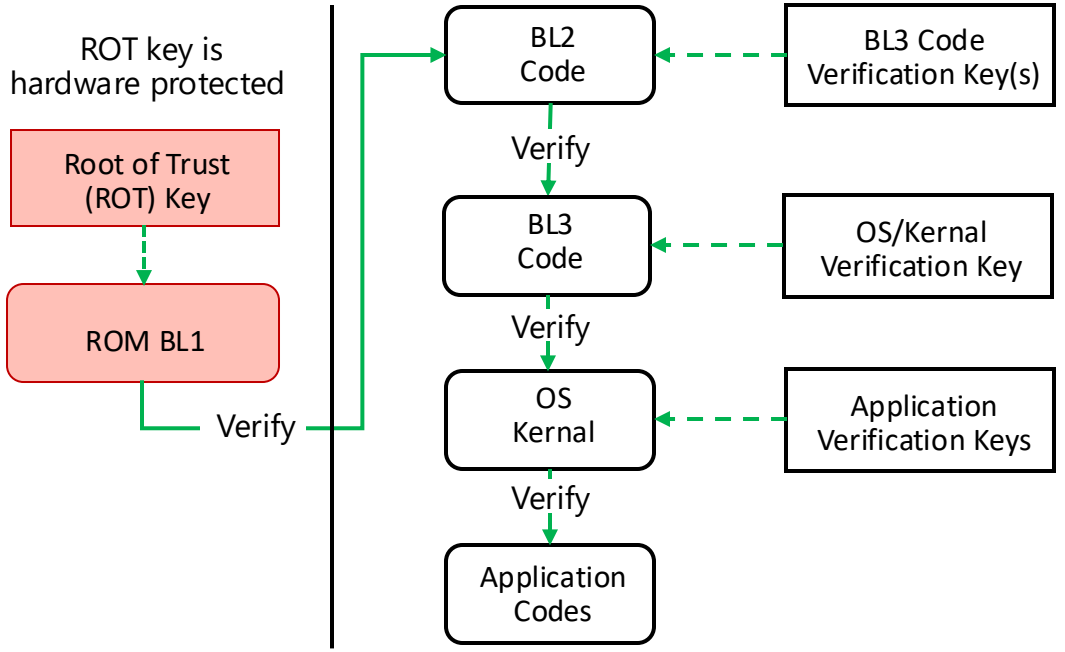
- ROM, secure element, etc.

Boot chain across firmware layers

- BL1 -> BL2 -> BL3 -> OS -> apps

Continuous validation across lifecycle

Typical boot & trust chain



Firmware Signing/Updates: Security Considerations

Enforcement Happens On-Device

- Update decisions are enforced by the device, not external systems
- Verification is chained to a hardware root of trust (RoT)

Tight Coupling to Hardware Root of Trust

- Verification keys / hashes are anchored in eFuse, ROM, or secure elements
- Software decisions are bound to immutable hardware state

Per-layer Signing Authority

- Bootloader, OS, application may require distinct keys
- Enables least-privilege trust across firmware layers (recommended)

Irreversibility and Risk with Firmware Updates

Mis-handled firmware may permanently disable a device

Recovery options are limited or non-existent in the field

Mistakes in signing policy translate directly into device failure

Securing Signing Lifecycle

Authenticated & Authorized Requests

- Every signing request is tied to a verified identity
- Policy-based authorization determines who can sign what

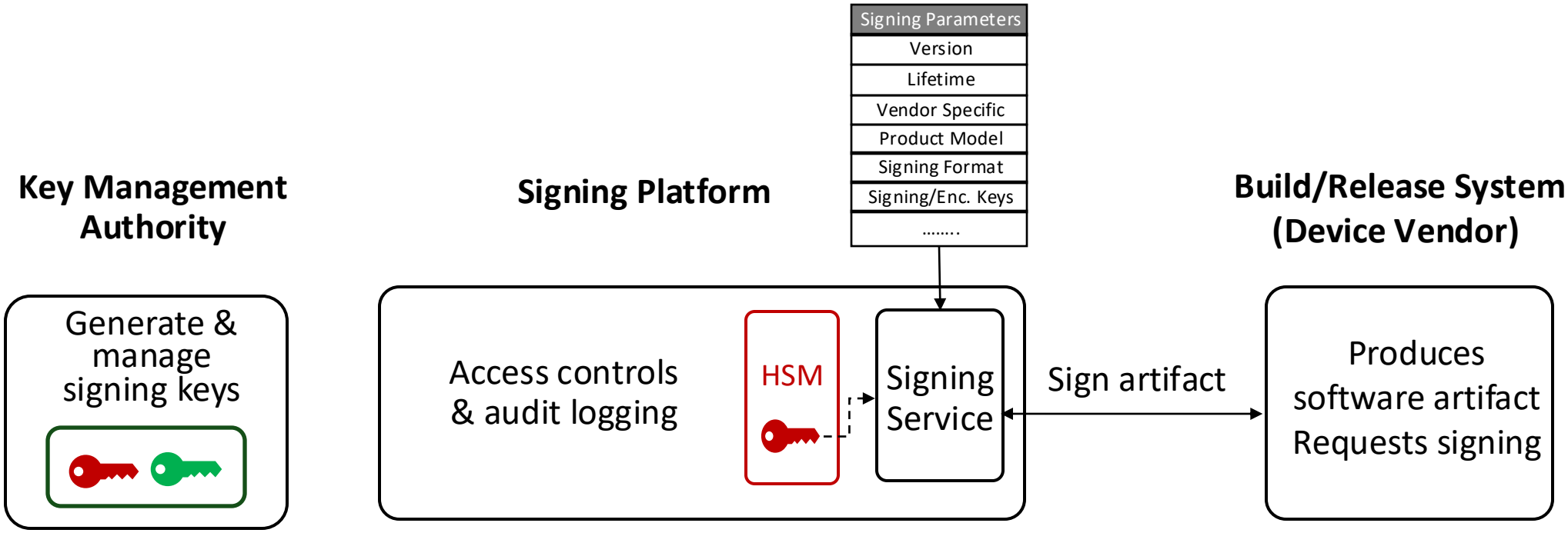
Separation of Duties & Key Protection

- Build system ≠ signing authority → controlled signing independent of CI/CD
- Signing keys remain protected inside HSM

Full Auditability & Traceability

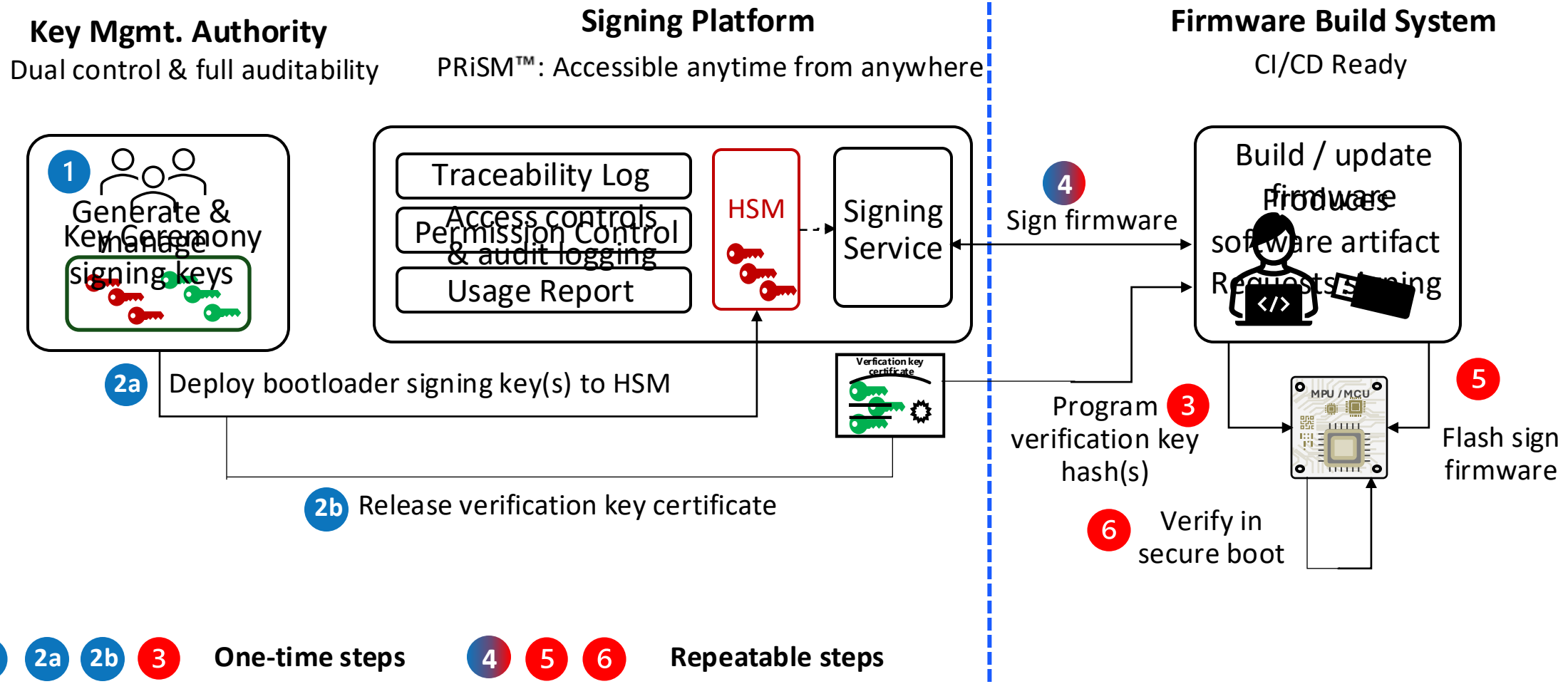
- Real-time logging of all signing actions
- Complete trace of who signed what, when, and under which policy, enabling compliance, accountability, and forensic analysis

Signing Architecture

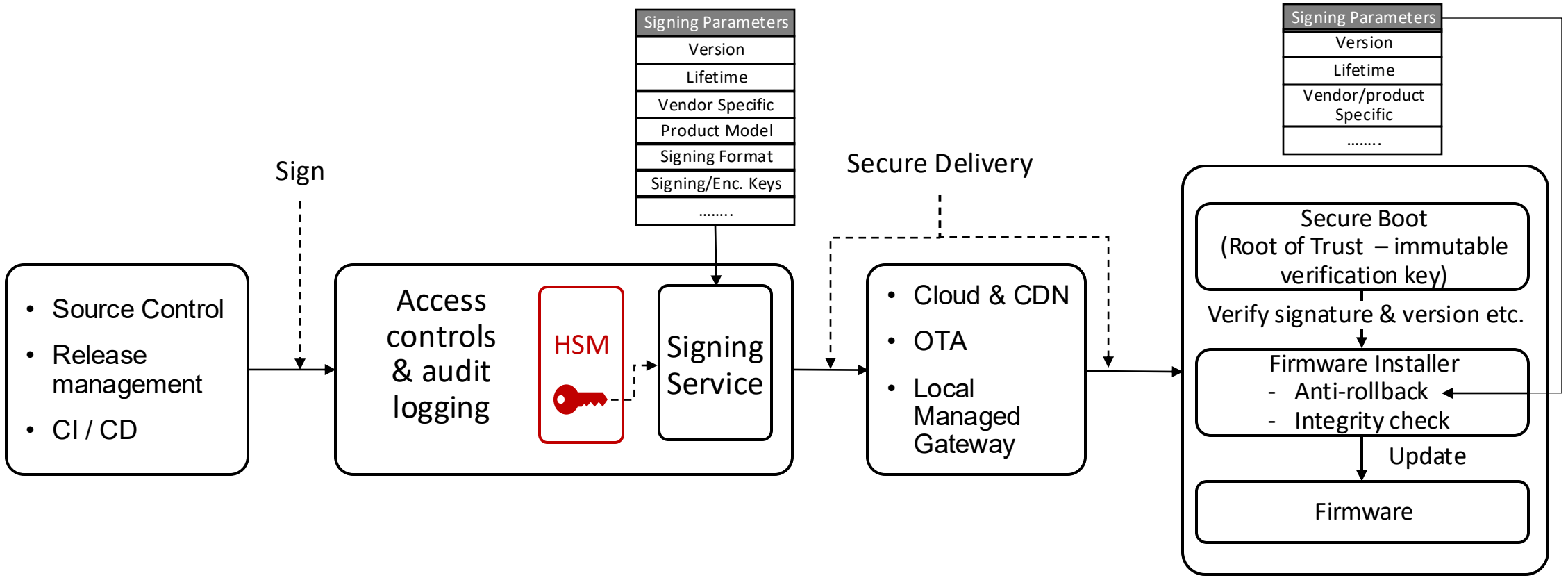


Compliance-grade signing infrastructure entails significant engineering and operational effort

Example Workflow



End-to-End Process and Details



 |

Key Lessons: Signing Process Failures

Key protection is a treated as a tooling problem

- Keys exposed via build systems, scripts, or ad-hoc automation
- HSMs alone are insufficient without controlled request paths

Uncontrolled access and overly broad signing authority

- Shared keys, weak permission models with broad signing privileges
- Lack of policy enforcement & multi-party control enables misuse

Insufficient provenance and intent binding

- Weak linkage between artifact, policy, and signing intent
- Cryptographically valid but mis-authorized images can propagate fleet-wide

Key Learned: Update System Failures

Inconsistent verification across layers

- Mismatched enforcement and policy check between bootloader, OS, updates, and application layers, breaking the trust chain

Gaps in rollback and staged validation

- Version checks and intermediate integrity verifications were not consistently enforced

Insecure recovery paths

- Recovery and bypass modes weakened normal security guarantees
- Failures and exploits propagated rapidly at fleet scale scale

Conclusions

Secure boot

- Establishes the immutable root of trust at power-on

Signing architecture

- Defines who can extend that trust, with separate authority for each firmware layer

Update mechanism

- Enforces it across the operational lifetime of the device

Together:

- All three must be designed as a cohesive system
- Provisioned correctly at manufacture, not retrofitted
- Stress-tested against real-world failure modes that emerge only at fleet scale

THANK YOU!

Q&A

xin.qiu@auroranetworks.com

PKI-center.com