Assignment 3.1: Trusted Platform Module (TPM)

Joshua A. Wortz

University of San Diego

CSOL-510

Prof. Hoffman

05/25/2026

Platform Configuration Registers

Platform Configuration Registers (PCRs) are dedicated hardware registers within the Trusted Platform Module (TPM) 2.0 designed to store integrity measurements in a shielded location. These registers do not permit the direct overwriting of data; instead, they utilize a one-way cryptographic "extend" operation. This operation is mathematically represented as $PCR\,new = H(PCR\,old \;//\; new_hash)$, where the current register value is concatenated with a new measurement (hash) and then re-hashed. This mechanism ensures that the PCR maintains a cumulative, immutable digest of the system's software state, in which any modification to a single component in the sequence fundamentally alters the final digest.

In cloud-native environments, PCR 10 is specifically leveraged for the Linux Integrity Measurement Architecture (IMA). As system components such as kernel modules or executables are loaded, their hashes are recorded in an append-only Measurement List (ML) and extended into PCR 10, anchoring the runtime software state to the hardware (Bin Emran et al., 2025).

Table 1: Standard TCG Chain of Trust Measurements

Boot Component	Measurement Action	PCR Mapping
Firmware / BIOS	Initial hardware and low-level code hash	PCR 0
Bootloader	Hash of the OS loader and configuration	PCR 4
Operating System	Kernel image and boot parameters	PCR 8
Runtime (IMA)	Hashes of executables and critical files	PCR 10

Measured Boot

Measured Boot establishes a verifiable chain of trust through transitive measurements, starting at power-on from the Core Root of Trust for Measurement (CRTM). In this sequence, each component hashes the subsequent stage before execution: the BIOS hashes the bootloader, and the bootloader hashes the kernel. These digests are recorded in the TPM PCRs, creating a cryptographic audit trail that prevents the undetected execution of tampered or unauthorized code.

As a systems researcher, it is vital to distinguish between "Static" boot measurements and "Runtime" measurements. Static measurements capture the immutable platform state up to the OS load. Conversely, runtime measurements, facilitated by the IMA, provide a continuous record of every file or binary accessed during operation, stored in an append-only measurement list (ML) that is periodically verified against the hardware-rooted PCR 10 value (Bin Emran et al., 2025).

Sealed Storage

The TPM provides confidentiality for sensitive data via "Sealed Storage," a mechanism that binds cryptographic keys to specific platform states (PCR values). Access to encrypted content is granted ("unsealed") only if the current PCR values match the policy defined at the time of initial encryption. If the platform integrity is compromised, the PCR values will deviate, resulting in a total loss of access to the encrypted content, a critical consideration for hardware-rooted key management.

Confidentiality requirements differ significantly between "data in transit" and "stored data." While data in transit emphasizes short-lived latency and frequent key renegotiation, stored data requires persistence and protection against offline attacks. It is critical to note that encryption alone provides confidentiality, not integrity; if the ciphertext is altered, standard decryption may not detect the change (University of San Diego, n.d.). Furthermore, modern block ciphers utilize distinct structures: Triple DES (TDEA) employs a 64-bit block size and a 48-round Feistel structure. In contrast, the Advanced Encryption Standard (AES) utilizes a 128-bit block size and a substitution-permutation network (VideosCoursera, 2012). Cryptographic implementations must also distinguish between deterministic encryption (where identical blocks produce identical ciphertext) and probabilistic encryption, which uses nonces or Initialization Vectors (IVs) to ensure uniqueness and protect against structural data leakage (Cryptography for Everybody, 2022). Finally, hardened implementations must account for side-channel attacks, such as cache timing, which target the execution environment rather than the algorithm itself (University of San Diego, n.d.).

Network Attestation

Network Attestation, or Remote Attestation (RA), is the technical workflow where a "Prover" (the node being challenged) provides cryptographic evidence of its integrity to a "Verifier." This process is essential for ensuring that only nodes in a known-good state are permitted to access network resources. In Kubernetes-based deployments, the Keylime framework is a primary implementation for managing this continuous trust (Bin Emran et al., 2025).

The Six-Step Remote Attestation Process:

1. **Nonce-based Challenge:** The Verifier sends a unique, random nonce to the Prover to mitigate replay attacks.
2. **PCR Quoting:** The Prover's TPM generates a "quote" of specific PCR values. This quote includes a digital signature over the PCRs and the nonce, signed by the TPM's Attestation Key (AK). This prevents "Attestation Simulation" attacks.
3. **ML Retrieval:** The Prover retrieves the current IMA Measurement Log (ML), which contains granular file hashes.
4. **Transmission:** The Prover transmits the Integrity Report (Quote + ML + Nonce) to the Verifier over a secure channel.
5. **Digest Re-computation:** The Verifier re-computes the hashes in the ML to ensure they match the quoted PCR 10 value.
6. **Whitelist Comparison:** The Verifier compares the validated ML entries against a "golden" allowlist of approved software.

Algorithm Agility

TPM 2.0 implements "Algorithm Agility," the ability to transition between cryptographic primitives (e.g., from SHA-1 to SHA-256 or future quantum-resistant hashes) without hardware replacement. This agility is critical for 5G Virtual Network Functions (VNFs), which serve as critical infrastructure in multi-vendor environments where a single compromised algorithm could undermine the entire system (Bin Emran et al., 2025).

The security of modern hash functions like SHA-256 relies on properties of preimage resistance and collision resistance. Finding a collision in SHA-256 is considered mathematically

"infeasible," with a probability of approximately 1 in 10^{40} (Gensler, 2018). Algorithm agility allows decentralized networks to maintain security by transitioning the entire system to a new hash function if an existing primitive's properties are significantly weakened.

Enhanced Authorization (EA)

Enhanced Authorization (EA) is a policy-driven mechanism for managing access that adheres to the Zero Trust principle of "never trust, always verify." This alignment with the CISA Zero Trust Maturity Model (ZTMM) addresses the pillars of Identity, Devices, and Workloads (Cybersecurity and Infrastructure Security Agency [CISA], 2023).

In Kubernetes environments, the Verifier assigns trust states to pods, such as the Access and Mobility Management Function (AMF) or User Plane Function (UPF), labeling them as Trusted, Untrusted, or Start. To achieve pod-level granularity, custom IMA templates utilize "cgpath" (cgroup paths) to bind specific measurements to Pod UIDs (Bin Emran et al., 2025). If the EA policy detects an unauthorized binary or a hash mismatch, it initiates a remediation phase. This involves "Persistence Removal," where the Kubernetes runtime evicts and restarts the compromised pod with a fresh Pod UID, effectively neutralizing the attacker's foothold and restoring a clean, verified state.

References

Bin Emran, A. N., Upadhyay, R., Paudyal, R., Donnan, L., & Wijesekera, D. (2025). TPM-based continuous remote attestation and integrity verification for 5G VNFs on Kubernetes. arXiv. https://arxiv.org/html/2510.03219v1

Cryptography for Everybody. (2022, July 14). Basics of cryptology – Part 24 (Cryptography – Block cipher modes of operation) [Video]. YouTube. https://www.youtube.com/watch?v=BasicsOfCryptologyPart24

Cybersecurity and Infrastructure Security Agency (CISA). (2023). Zero trust maturity model version 2.0. https://www.cisa.gov/sites/default/files/2023-04/zero_trust_maturity_model_v2_508.pdf

Gensler, G. (2018). Blockchain basics & cryptography [Video]. YouTube; MIT OpenCourseWare. https://www.youtube.com/watch?v=3.BlockchainBasics&Cryptography

University of San Diego. (n.d.). CSOL-510 Module 3: Key concepts and terms: Block cipher basics.

VideosCoursera. (2012, March 14). 3 1 What are block ciphers? 17 min [Video]. YouTube. https://www.youtube.com/watch?v=WhatAreBlockCiphers