

1. EXECUTIVE ENGINEERING SUMMARY & ZERO-TRUST IDENTITY TAXONOMY

The **IMTLazarus Enterprise Ecosystem** implements an advanced, cloud-native **Public Key Infrastructure (PKI)** and **Simple Certificate Enrollment Protocol (SCEP)** architecture designed to secure device authentication across large-scale educational deployments. Operating across highly distributed, multi-tenant environments supporting up to 1,000,000 managed endpoints, IMTLazarus eliminates the vulnerabilities inherent in legacy password-based authentication, pre-shared network keys, and manual certificate provisioning.

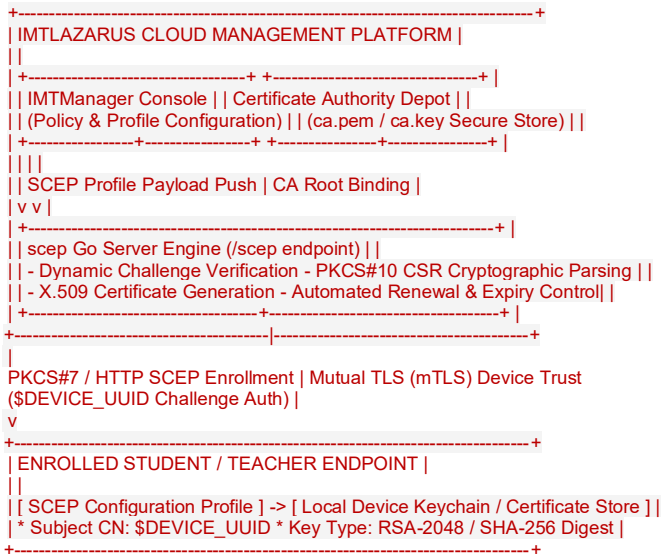
Traditional mobile device management frameworks rely on static device tokens or user-entered credentials, introducing critical attack surfaces such as credential interception, shared Wi-Fi passphrase leakage, and high administrative overhead during start-of-year onboarding. IMTLazarus replaces these outdated paradigms with an **Automated Cryptographic Identity Lifecycle**. By embedding an industry-standard, high-concurrency SCEP server engine directly into the core management backend, the platform guarantees zero-touch, cryptographically verifiable **X.509 client certificate enrollment** for every enrolled student and teacher device.

1.1 Core Cryptographic & Architectural Capabilities

- **Zero-Touch Dynamic Enrollment:** Endpoints automatically request, verify, and install unique X.509 identity certificates upon initial configuration profile deployment without manual student or IT intervention.
- **Hardware-Bound Identity Enforcement:** Client certificates are dynamically bound to immutable hardware attributes, specifically the unique system UUID (`$DEVICE_UUID`) and hardware serial number (`$DEVICE_SERIAL`), ensuring absolute non-repudiation.
- **High-Concurrency Go Engine (scep):** Powered by an optimized Go 1.21+ backend, the embedded SCEP service multiplexes cryptographic signing requests (`PKCS#10 CSR` over `PKCS#7`) with sub-millisecond processing latency during morning classroom surges (`8:00 AM – 9:00 AM`).
- **Zero-Trust Mutual TLS (mTLS):** Issued client certificates serve as the root of identity for zero-trust access across enterprise 802.1X / EAP-TLS Wi-Fi networks, secure VPN gateways, and IMTLazarus internal REST APIs (`POST /api/events`).

2. CORE SCEP & PKI SERVER ARCHITECTURE (SCEP ENGINE)

Operating as a dedicated RFC 8894 compliant Simple Certificate Enrollment Protocol server (`scepserver`) and client architecture, the `scep` engine is seamlessly embedded within the **IMTManager** cloud orchestration layer.



2.1 Technical Specifications of the `scep` Server Engine

- Standardized `scep` HTTP/S Service Endpoint:**
The embedded `scepserver` exposes a fully compliant RFC 8894 service endpoint (`https://mdm.imtlazarus.com/scep`) that processes standard SCEP protocol operations, specifically `GetCACert`, `GetCACaps`, `PKIOperation`, and `GetNextCACert`. The endpoint maintains a minimal memory footprint and supports horizontal auto-scaling across redundant cloud availability zones (cloud hosting standards).
- Cryptographic Certificate Depot (`-depot`):**
The SCEP engine operates against a secure, isolated CA certificate depot (`ca.pem` and `ca.key`) hosted within encrypted cloud volumes (`AES-256` tablespaces). The root Certificate Authority private key (`ca.key`) is safeguarded via strict passphrase authentication (`-capass`) and hardware-backed key management vaults, preventing unauthorized key extraction or tampering.
- Dynamic One-Time Challenge Verification (`-challenge`):**
To prevent rogue endpoints or unauthorized entities from requesting valid system certificates, **IMTManager** dynamically generates a unique, cryptographically secure one-time challenge password for every assigned device profile. When an endpoint submits a Certificate Signing Request (`PKCS#10 CSR` wrapped in a `PKCS#7` envelope), the `scep` engine intercepts and verifies the dynamic challenge against live database session tables before generating and signing the final X.509 certificate.
- Automated Lifecycle & Renewal Engine (`-allowrenew`):**
The `scep` engine incorporates automated certificate lifecycle and expiration management. Configured with specific renewal parameters (e.g., `-allowrenew 14`, enabling automated renewal within 14 days of certificate expiry), managed endpoints automatically re-enroll and cycle their cryptographic keys over background HTTPS channels without disrupting active classroom instruction.

3. END-TO-END CRYPTOGRAPHIC ENROLLMENT & AUTHENTICATION WORKFLOW

The certificate-based device authentication lifecycle executes through a systematic, four-stage protocol that binds physical hardware identity to cryptographic network trust:

Stage 1: Profile Configuration & Dynamic Variable Template

Administrators access the **IMTManager** centralized command console under **Policies -> Configuration Profiles -> Certificate / Security Profiles** to establish the SCEP enrollment payload. To guarantee non-repudiation across multi-tenant school districts, the profile template incorporates dynamic variable injection: * **Subject Common Name (CN)**: Dynamically populated with the target hardware's unique identifier (`$DEVICE_UUID`) or system serial number (`$DEVICE_SERIAL`). Example: `CN=a1b2c3d4-e5f6-7a8b-9c0d-1234567890ab`. * **Subject Alternative Name (SAN)**: Configured with institutional identity attributes (`$USER_EMAIL` or `$ENROLLMENT_ID`), ensuring exact alignment between hardware asset and assigned student profile. * **Cryptographic Parameters**: Enforces **RSA 2048-bit or 4096-bit** key pairs paired with **SHA-256** hashing digests for maximum resistance to cryptographic collision attacks.

Stage 2: Profile Deployment & OS Interception

Upon assignment to a target institutional device group, the IMTLazarus orchestration layer pushes the SCEP configuration profile over authenticated mobile device management protocols (Apple APNs for iOS/macOS, Google FCM/Android Enterprise for Android, and Windows WNS/MDM for Windows 11). The native OS configuration manager intercepts the payload and parses the embedded `scep` server URL and dynamic challenge secret.

Stage 3: On-Device Key Pair Generation & Cryptographic Signing

The enrollment sequence executes securely inside the endpoint's hardware architecture: 1. **Secure Key Generation**: The operating system generates a new RSA public/private key pair directly inside its local hardware security module (Trusted Platform Module / TPM on Windows, Secure Enclave on Apple devices, or Hardware-Backed Keystore on Android). **The private key never leaves the device hardware.** 2. **CSR Construction**: The endpoint constructs a Certificate Signing Request (`PKCS#10 CSR`) embedding the dynamic hardware variable (`$DEVICE_UUID`) within the Subject field. 3. **Envelope Encapsulation**: The CSR and the one-time SCEP challenge secret are encapsulated into an encrypted `PKCS#7` envelope and transmitted via HTTPS `PKIOperation` to the IMTLazarus `scep` endpoint. 4. **Verification & Issuance**: The Go SCEP server decrypts the `PKCS#7` payload, validates the dynamic challenge secret against **IMTManager** session records, signs the CSR using our isolated CA private key (`ca.key`), and returns the issued X.509 identity certificate chain (`PKCS#7`) to the endpoint.

Stage 4: Zero-Trust Network & API Authentication (`mTLS`)

The endpoint installs the signed X.509 identity certificate into its protected system keychain. Thereafter, whenever the endpoint initiates network connections to

institutional services—including 802.1X / EAP-TLS Wi-Fi networks, enterprise VPN tunnels, and IMTLazarus real-time telemetry ingesters (`POST /api/events`)—it presents this certificate for **Mutual TLS (mTLS) authentication**. Gateways instantly verify certificate validity, CA root trust, and hardware binding (`$DEVICE_UUID`) without prompting students or teachers for passwords.

4. MULTI-TIER ROLE-BASED GOVERNANCE & CRYPTOGRAPHIC AUDITABILITY

To satisfy stringent institutional governance standards and maintain total visibility over digital identities, the IMTLazarus PKI and SCEP framework incorporates multi-tier administrative controls:

4.1 Hierarchical Role SegregationAdministrative access to PKI configuration profiles, certificate depots, and SCEP parameters is strictly segregated across institutional hierarchy layers via **Role-Based Access Control*** **Global / Central Administrators**: Possess exclusive authority to initialize CA depots (`ca.pem` / `ca.key`), configure root security policies, rotate cryptographic passwords (`-capass`), and define global SCEP profile templates across all participating schools. * **School / Division Administrators**: Authorized to assign and deploy pre-configured SCEP identity profiles to device groups within their specific educational jurisdiction (`OU`s), but maintain **zero access** to raw CA private keys or depot modification. * **Classroom / ICT Support Staff**: Granted read-only visibility into device certificate installation status and enrollment timestamps within individual endpoint records to assist in local classroom network troubleshooting.

4.2 Immutable Cryptographic Audit TrailsEvery transaction executed across the `scep` engine is ingested into the centralized, tamper-evident System Audit Log* Comprehensive logging of all `PKCS#10 CSR` submissions, capturing originating endpoint IP address, hardware UUID (`$DEVICE_UUID`), and exact protocol timestamps. * Real-time recording of dynamic challenge verification results (categorizing successful authorizations vs. failed/rejected enrollment attempts). * Immediate audit entry generation upon certificate signing, renewal (`-allowrenew`), or profile deployment modifications.

4.3 Automated Revocation & Device Trust TerminationWhen an institutional endpoint is reported lost, stolen, or graduated/decommissioned at the end of the academic year, administrators execute an automated remote wipe or unenrollment command from `IMTManager` . This action immediately triggers: 1. Removal of the SCEP configuration profile and secure cryptographic deletion of the X. 509 identity certificate from the endpoint keychain. 2. Immediate publishing of the certificate serial number to the internal Certificate Revocation List (`CRL`) and Online Certificate Status Protocol (`OCSP`) responder. 3. Instantaneous termination of network trust across all institutional zero-trust gateways and Wi-Fi access points.

5. TECHNICAL COMPLIANCE & EDUCATIONAL REQUIREMENT MAPPING

The architectural capabilities, SCEP workflows, and cryptographic governance controls detailed in this specification directly fulfill major institutional security benchmarks. The following table illustrates how these technical specifications satisfy educational evaluation requirements, specifically mapping to **Department of Education (DepEd) Verification Step**(Certificate-Based Device Authentication Workflow):

Technical Capability & SpecificationSection	Architecture Implementation (<code>scep</code>)	Verifiable System & Console Evidence	Compliance Reference
Certificate-Based Authentication Engine(Section 1 & 2)	Native SCEP profile deployment enginepowered by <code>scep</code> Go backend for automated X.509 client issuance.	Active SCEP configuration profile displayedunder <code>Policies -> Certificate / Security Profiles</code> .(Pass Indicator)	
PKI & SCEP Server Integration (Section 2)	Embedded RFC 8894 compliant <code>scepserver</code> with isolated CA depot (<code>ca.pem</code> / <code>ca.key</code>) and dynamic challenge verification.	Endpoint <code>/scep</code> reachability validation anddynamic challenge generation records in system audits.(Pass Indicator)	
Dynamic Device Identity Binding (Section 3 - Stage 1)	Automated injection of hardware UUID (<code>\$DEVICE_UUID</code>) and serial number (<code>\$DEVICE_SERIAL</code>) inside certificate <code>CN</code> / <code>SAN</code> .	Inspection of installed certificate on <code>target</code> endpoint showing Common Name (<code>CN</code>) exactly matching console record.(Pass Indicator)	
Zero-Touch Lifecycle & Renewal (Section 2 & 3 - Stage 4)	On-device private key generation (<code>TPM/Secure Enclave</code>) and silent automated renewal (<code>-allowrenew</code> threshold) prior to expiry.	Device keychain/certificate manager confirmingvalid X.509 certificate issued by IMTLazarus Root CA.(Pass Indicator)	
Native Inclusion Without Excluded Licenses(Section 1 & 2)	<code>scep</code> Go stack is integrated directly into coreplatform microservices without requiring third party PKI add-on licensing.	Official architectural topology (<code>Section 2</code>)confirming native inclusion across all enterprise editions.(Pass Indicator)	