US Coast Guard Academy (USCGA1) eCTF 2026

Team: Tyler Bissett, Phillip Kuznetsov, Caden Waters, Daniel Yi
Advisor: Mohamed Elwakil, PhD
Department of Electrical Engineering and Computing, U.S. Coast Guard Academy

Introduction

This poster outlines a secure Hardware Security Module (HSM) design for the 2026 MITRE Embedded Capture the Flag (eCTF) competition by the United States Coast Guard Academy team (USCGA1). The system implements a permissioned file storage and transfer solution for ChipCorp Inc., a fictional chip manufacturer seeking to protect proprietary fabrication data. Built on the TI MSPM0L2228 microcontroller, the design uses Hardware Security Modules (HSMs) to store, manage, and transfer files with group-based permission controls. Each HSM is provisioned with a unique PIN and permissions (Read, Write, Receive) governing which file operations it may perform. Our design philosophy prioritizes simplicity and correctness, using wolfCrypt for authenticated encryption, message authentication, and key derivation.

System Setup

This section details the architecture of the HSM system. As shown in Figure 1, the system operates across a Customer Site and an Air Gapped Facility. HSM devices connect to host computers via a Management Interface (USB-serial) and to neighboring HSMs via a Transfer Interface (UART).

The HSM is implemented on a TI MSPM0L2228 microcontroller (Cortex-M0+). It stores up to 8 files in flash memory with group-based permission controls. Each HSM is provisioned at build time with a PIN, a permission set, and deployment secrets derived from the Global Secrets. The Management Interface connects the HSM to a host computer via USB-serial. Users interact with the HSM through organizer-provided host tools to list, read, and write files. PIN-protected commands require the correct PIN before execution. The Transfer Interface connects two HSMs via UART for cross-board operations. One HSM enters listening mode while the other initiates an interrogation (to list available files) or a receive (to transfer a file). The Host Computer runs the organizer-provided, read-only host tools that issue commands to the HSM. The Bootloader is a secure component provided by the organizers that verifies firmware integrity before execution. The bootloader uses the File Allocation Table (FAT) at address 0x3A000 and is not in scope for attack.

Our design adds MAC verification on all transferred files. The Host Computer runs the organizer-provided, read-only host tools that issue commands to the HSM. The Bootloader is a secure component provided by the organizers that loads encrypted firmware images using cryptographic digests from the File Allocation Table (FAT) at address 0x3A000. The bootloader is not in scope for attack.

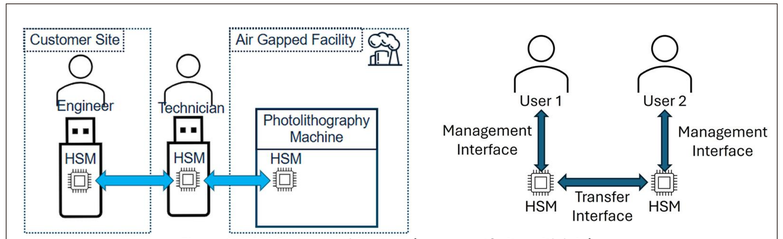

Figure 1. HSM System Architecture (source: ectfmitre.gitlab.io)

Functional Requirements

The functional requirements cover System Build and HSM Operations. The Build Environment uses Docker for dependency management. During Build Deployment, gen_secrets.py generates Global Secrets, which may include cryptographic key material, seeds, entropy, or other data needed for the design. The Build HSM stage compiles firmware for each HSM using its unique PIN (exactly 6 lowercase hexadecimal characters) and permission string (e.g., 1234=RWC:4321=R--), which defines its access to specific file groups. The secrets_to_c_header.py script packages the Global Secrets into a C header for firmware compilation. All HSMs sharing the same Global Secrets form a Deployment.

The HSM firmware implements six commands. List Files is PIN-protected and returns metadata for all stored files. Read File is PIN-protected and retrieves file contents if the HSM has read permission for that file's group. Write File is PIN-protected and stores a file to a slot if the HSM has write permission; if the slot already contains a file, it is overwritten. Files persist across power cycles and the HSM supports up to 8 file slots. The File Allocation Table (FAT) at flash address 0x3A000 tracks file UUIDs, lengths, and addresses.

For cross-board operations, Listen is not PIN-protected and places the HSM in listening mode to handle one command from a neighbor. Interrogate is PIN-protected and queries a connected HSM to list files for which the local HSM has receive permissions. Receive is PIN-protected and transfers a file from a neighbor if the local HSM has receive permission for that file's group; all file data (name, UUID, group, contents) must be transferred. Timing requirements are: Device Wake 1s, List 500ms, Read/Write/Interrogate 1000ms, Receive 2000ms. Invalid PIN attempts incur a mandatory 5-second delay.

Security Requirements

The security requirements define what our design must protect against. These are tested during the Attack Phase, where other teams attempt to violate them by capturing five attack flags.

SR1 (Permission Enforcement): An attacker should not be able to perform any file action without a validly provisioned HSM with the permissions to perform that action on files belonging to that group. This covers read, write, and receive operations. The attacker should not be able to read files without read permission, create files without write permission, or receive files without receive permission. The receive permission also applies to interrogate responses.

SR2 (PIN Confidentiality): No PIN-protected action should be completed by a user without prior knowledge of the PIN. This includes confidentiality of the PIN itself. The HSM should not expose information about its PIN to any unauthorized user.

SR3 (File Authenticity): An HSM should not successfully receive any file that was not generated by another valid HSM with write permissions for that group. This includes protecting file integrity from being compromised in any way by an attacker.

Security Plan

This section describes how our team plans to satisfy the three Security Requirements. The attack scenario involves three HSMs: an Engineer (Design: RWC), a Technician (attacker-controlled; Update: --C, Calibration: RWC, Telemetry: R-C), and a Photolithography Machine (remote; Design: R-C, Update: R-C, Calibration: R-C, Telemetry: -W-). The attacker knows the Technician's PIN, has full physical access to the Technician and Engineer HSMs, and has only remote access to the Photolithography Machine.

SR1 - Permission Enforcement: Each file operation checks the HSM's permission table before proceeding. Read requires R, Write requires W, and Receive requires C. The permission table is embedded at build time and cannot be modified at runtime. For file transfers, the requesting HSM presents HMAC-SHA256 authenticated permission certificates so the responding HSM can verify the requester's identity and filter the file list accordingly.

SR2 - PIN Confidentiality: The PIN is never stored in plaintext. At build time, a random salt is generated and the PIN is processed through PBKDF2 to produce a stored verifier. At runtime, the user-supplied PIN is derived with the same salt and compared using constant-time comparison (bitwise OR accumulation) to prevent timing attacks. Failed attempts trigger a 5-second delay. PBKDF2's key stretching makes brute-force computationally expensive even if the verifier is extracted from flash.

SR3 - File Integrity and Authenticity: At build time, gen_secrets.py generates a 256-bit master key, a 256-bit HMAC key, and a 128-bit salt. Per-group encryption keys are derived from the master key and each group ID. Files are encrypted using AES-GCM and authenticated with HMAC-SHA256. On receive, the HSM verifies the HMAC before accepting the file. Each HSM receives keys only for groups where it has Write or Receive permissions. The Technician never receives Design group keys and therefore cannot decrypt, forge, or authenticate Design files.

These measures collectively defend against the five competition attack flags: Steal Design, Read Update, Read Design, Compromise Machine, and Backdoor Design.

Conclusion

The secure HSM system developed for the eCTF competition demonstrates a balanced approach to functionality, security, and implementation feasibility. Our design satisfies all functional requirements including the six HSM commands and strict timing constraints. Security requirements are addressed through three reinforcing layers: HMAC-authenticated permission enforcement (SR1), PBKDF2-derived PIN verification with constant-time comparison (SR2), and AES-GCM authenticated encryption with HMAC-SHA256 integrity checks for files at rest and in transit (SR3). The selective distribution of per-group cryptographic keys ensures that the attacker's Technician HSM never possesses the keys needed to access or forge Design files. Hardware protections including the MSPM0L2228's write-only KeyStore and AES acceleration defend against physical key extraction from the Engineer HSM.

Contact

Mohamed ELwakil
Department of Electrical Engineering and Computing,
U.S. Coast Guard Academy
mohamed.m.elwakil@uscga.edu

References

1. MITRE. (2026). Embedded capture the flag (eCTF) competition. https://ectf.mitre.org/
2.
3.
4.
5.
6.
7.
8.
9.
10.