ECE 445

Spring 2025

Project Proposal

Shamir Secret Self-Destruct USB

Alex Clemens - clemens9
Danny Metzger - djm14
Varun Sivasubramanian - vsiva4

Introduction

Problem

Traditional flash drives pose a security risk when handling sensitive information like cryptographic keys, classified documents, or personal information. In some cases, a stolen or deleted drive can be recovered with some forensic analysis. Encrypting a flash drive does offer some protection, but you are often reliant on the software and an attacker could potentially brute force the password, resulting in vulnerable data at risk.

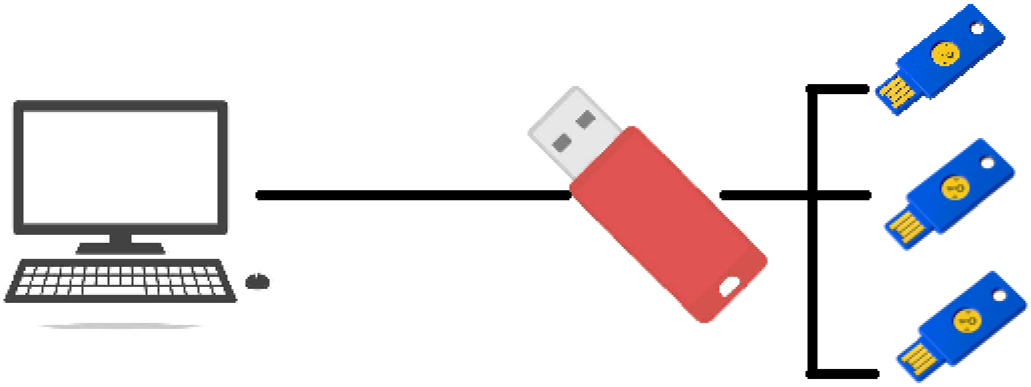
Software based security works for many cases, but could potentially be exploited or bypassed. To address security concerns, many users rely on hardware key solutions, like YubiKeys or TPM, to add an extra layer of security to their devices. Even these systems are not perfect, as access to the physical keys can cause a security breach. Many existing solutions do not provide a way to automatically destroy hardware in the event of tampering. A USB storage solution that combines these security premises could be useful.

Solution

Our solution is a custom PCB-based flash drive that implements a variety of security features through its hardware. The device will store its encryption key using Replicated Secret Sharing, requiring at least two out of three physical authentication cards connected before accessing the data. Through custom hardware, this flash drive will ensure that decryption can only occur with the correct physical authentication, deterring software attacks.

In addition, the flash drive will contain a tamper-resistant self-destruct circuit that triggers from multiple failed authentications, physical tampering, or probing of the key storage. This will permanently corrupt and destroy the stored data. There will be a small battery as part of the device, ensuring destruction even when the drive is unplugged. With all these features, this flash drive should provide high security to the user’s data.

Visual Aid

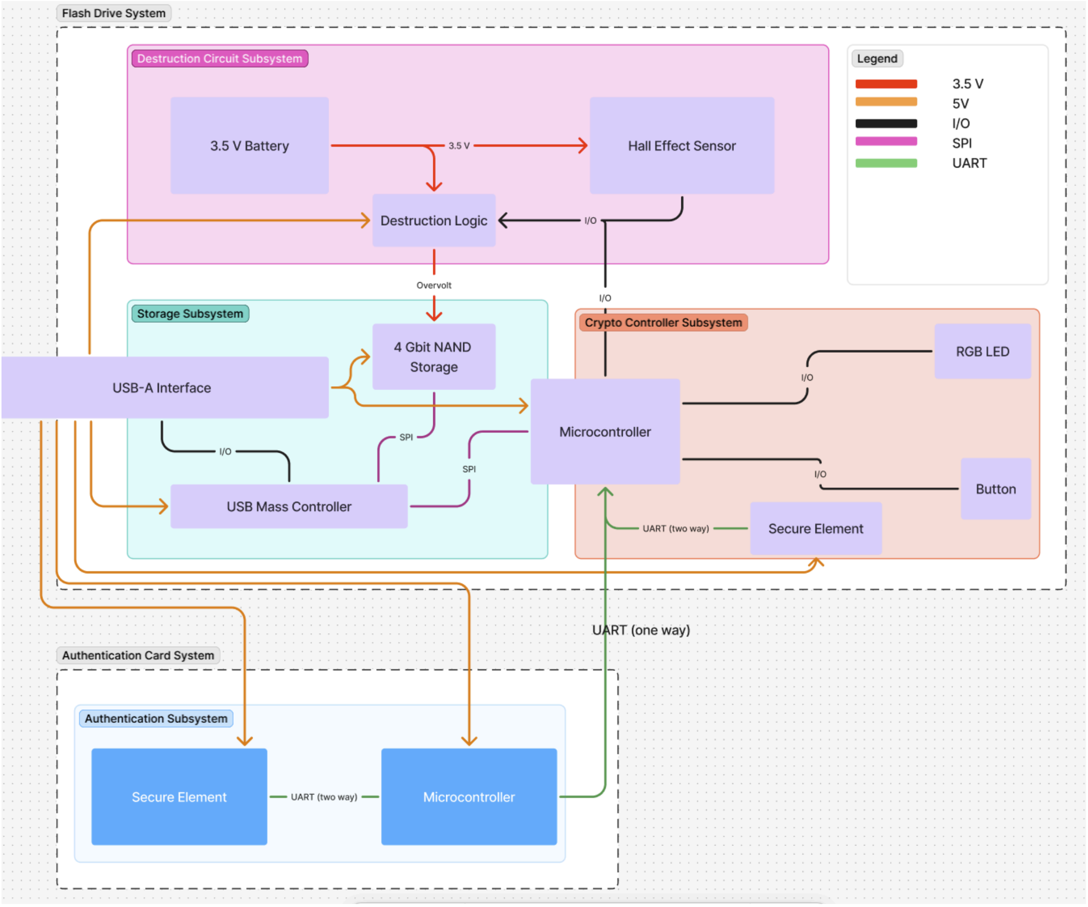


High-Level Requirements:

- The USB flash drive must allow a maximum of 5 failed authentication attempts before triggering the self-destruct.
- The flash drive must require at least 2 out of 3 physical authentication cards to decrypt the hidden partition.
- The flash drive should have at least 0.5 GB of storage.

Design

Block Diagram



Subsystem Overview

Authentication Card Subsystem

The Authentication Card Subsystem ensures secure, hardware-based authentication before granting access to the NAND storage. When two cards are plugged into the main device, they collectively provide the necessary key materials for decryption. This subsystem communicates with the Control Subsystem, sending or withholding the key shares that ultimately enable (or deny) access to encrypted data.

Storage Subsystem

The Storage Subsystem is responsible for handling all read/write operations to user data.. Decryption occurs only after the Control Subsystem authorizes it, at which point the mass storage controller can securely read and write to NAND. If a tamper or destruction event is triggered, this subsystem is physically overloaded to prevent data recovery.

Hardware Destruction Circuit Subsystem

The Hardware Destruction Circuit Subsystem guarantees irreversible data destruction if unauthorized access or tampering is detected.When triggered—either by a signal from the Control Subsystem (e.g., too many failed authentication attempts) or from direct enclosure tampering—this subsystem rapidly and permanently damages the storage subsystem

Control Subsystem

The Control Subsystem orchestrates operations across all other subsystems. An RGB LED displays system status (e.g., pending authentication, successful unlock, or warning states), and a physical button is used to confirm authentication after two valid authentication cards are present. By coordinating inputs from the Authentication Card Subsystem, enabling decryption in the Storage Subsystem, and issuing destruction signals to the Hardware Destruction Circuit Subsystem when necessary, the control subsystem serves as the decision-making core of the entire system.

Subsystem Requirements

Authentication Card Subsystem

The authentication card subsystem must be able to hold two parts of the encryption key in secure storage, with no way to extract the keys. The Secure Element holds the keys, while the STM32 microcontroller allows communication through its UART pins to the flash drive PCB. When plugged into the flash drive, the subsystem will receive power from the flash drive through additional pins and send an encrypted version of its keys to the flash drive’s microcontroller.

Storage Subsystem

The storage subsystem contains at least 0.5GB of NAND flash storage. The USB-A interface allows the drive to be plugged into a computer and also powers every component besides the destruction circuit subsystem. Like any traditional flash drive, the USB mass storage controller is necessary to allow a computer to interface with the NAND storage. The on-board microcontroller communicates with the mass storage as well to facilitate encryption and decryption.

Hardware Destruction Circuit Subsystem

The hardware destruction subsystem contains a 3.5 V battery that is responsible for destroying the NAND when tampering is detected. The battery must be able to provide 3V at 6µA to the hall effect sensor. Usually, the destruction logic is normally disconnected via a transistor. When the microcontroller or hall effect sensor triggers the logic, the battery's voltage is directly fed into the NAND flash, which is only rated for less than 2 V. As a backup, the USB power is also directed into the destruction logic, providing more voltage across the NAND.

Control Subsystem

The control subsystem requires logic to verify that the correct authentication devices are plugged into the flash drive. The Secure Element must store a hashed version of the constructed key and allow the microcontroller to compare an assembled key to the hash. The button must initiate the decryption sequence and the RGB indicates the system status. When the authentication cards are inserted, the microcontroller must reconstruct the key. If the key is incorrect 5 times, the microcontroller triggers I/O connected to the destruction logic.

Tolerance Analysis

One potential risk in our design is the possibility of the battery depleting and thereby disabling the Hall effect sensor, which is responsible for detecting physical tampering. In such a scenario, an attacker might theoretically open the flash drive, connect directly to the microcontroller's pins, and attempt to brute force the cryptographic keys. Although we intend to use a highly secure encryption scheme to mitigate brute force attacks, our core objective remains preventing sensitive data from ever falling into adversarial hands.

To address this concern, we plan to use a Hall effect sensor that operates on less than 6 µA at 3 V—such as the A3214 series from Allegro. At this low operating current and voltage, the sensor draws only about 18 µAh of power per hour, which corresponds to roughly 0.432 mAh per day. Because the Hall sensor is the only continuously active component in our tampering-detection circuit, even a relatively small 500 mAh battery could power it for around three years (500 mAh ÷ 0.432 mAh/day ≈ 3.17 years).

Additionally, the Hall effect sensor only requires power to **send** its output signal; it will still physically switch states when the magnetic field changes, even if unpowered. This means that if the battery depletes, the sensor can still register a tamper event. The next time the device is powered via the USB interface, the control subsystem will detect the changed sensor state and trigger NAND destruction if necessary. Given the sensor's extremely low power consumption and its ability to switch states without a continuous power source, we are confident this solution provides a high level of security even under adverse conditions.

Ethics & Safety

Our project potentially raises a few ethical and safety concerns that must be considered during the design, development, and production phases of our project. Some of these include the following.

Ethical Considerations

Honesty and Transparency

The IEEE tells us that we must “be honest and realistic in stating claims or estimates based on available data" (IEEE CoE Section 5). The USB device’s self-destructive feature is powerful but risky. Users must be fully aware of its existence, how it works, and the potential consequences of triggering it. Transparency is key to building trust and ensuring users understand what they’re dealing with.

In order to ensure this, we can provide clear documentation and warning messages on the device and in user instructions about the self-destruct mechanism and any potential consequences. This ensures that users are not blindsided by accidental data loss and can make informed decisions about using the device.

Contribution to Society

The ACM Code of Ethics says that innovations should "Contribute to society and human well-being, acknowledging that all people are stakeholders in computing" (ACM CoE Section 1.1). Our device contributes to the protection of sensitive data, which is in the public interest, especially given the increasing concerns over data breaches and privacy violations. However, as with any security device, there is a need to balance protection with usability.

We must make sure the device not only helps protect data but also takes into account user experience. By designing the self-destruct feature with careful safeguards and clear communication to the user, we can enhance both security and user confidence, improving the overall well-being of those who use it.

Safety Considerations

Secure Data Destruction

According to the IEEE, we must agree "To avoid harm to others, their property, reputation, or employment by false or misleading claims, and to avoid injuring others” (IEEE CoE Section 9).

In our case with our USB, we must ensure that this device safely deletes data and has safeguards to protect the data in case of accidental deletion or malicious deletion.

The self-destructing USB is intended to protect sensitive data, but if not properly designed, it could also lead to unintended data loss or harm on purpose by an attacker attempting to delete the data. While this is an inherent risk of having a super secure hard drive such as this, we could possibly necessitate the shamir secret to fail 3 times in a row to actually delete the data as a backup plan.

Testing for Reliability

The ACM requires one to "Ensure that software and hardware are developed and tested to meet the highest quality standards, and are fit for their intended purposes" (ACM CoE Section 2.1). In our case, reliable functioning of the device is vital, as the self-destruct mechanism could malfunction, leading to either data loss without authorization or failure to protect the data when needed.

In order to prevent this we perform rigorous testing (including edge cases and stress tests) to ensure the self-destruct mechanism works as intended under various conditions for each copy of the USB. We could test the majority of the copies we make to ensure the batch of PCBs was created as intended.

Regulations

Illinois Data Protection Laws

The Personal Information Protection Act (815 ILCS 530)
governs the protection of personal data in Illinois. It requires entities to implement reasonable security measures to protect sensitive data from unauthorized access or theft.

For our USB, we must ensure that the self-destruct mechanism securely erases sensitive information, such as passwords or other personally identifiable information (PII). If the data is not properly erased, or if there is a risk of recovery after destruction, this could violate state data protection regulations.

Other

Besides some of these ethical and safety concerns we must consider in relation to our project, it is also vital that we keep in line with the ethical concerns in relation to the operations of ECE 445. These rules we must adhere to include:

- **Honesty:** Ensure that we are honest about the development of our project and the data we collect as the semester progresses. We should not be faking or skewing data to make our project look more successful.
- **Record Keeping:** Be sure to document everything, including our failures, into our lab notebooks. Record keeping will also keep our intellectual property safe for the future.
- **Stealing/Plagiarism:** If we use the work or ideas of others, we must be sure to give credit and abide by fair use and copyright laws. Also must not try to pass off other's work as our own.
- **Other:** Treat our team members with respect, even through arguments or disagreements. Agree to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to properly credit the contributions of others.