Replicated Secret Self Destruct USB

Electrical & Computer Engineering

Varun Siva, Alex Clemens, Danny Metzger

May 6th, 2025

Team 69 Introduction

Varun Siva
Computer Engineering
Class of 2025

Alex Clemens
Computer Engineering
Class of 2025

Danny Metzger
Computer Engineering
Class of 2025

OBJECTIVE

Problem

Many security issues with traditional flash drives:

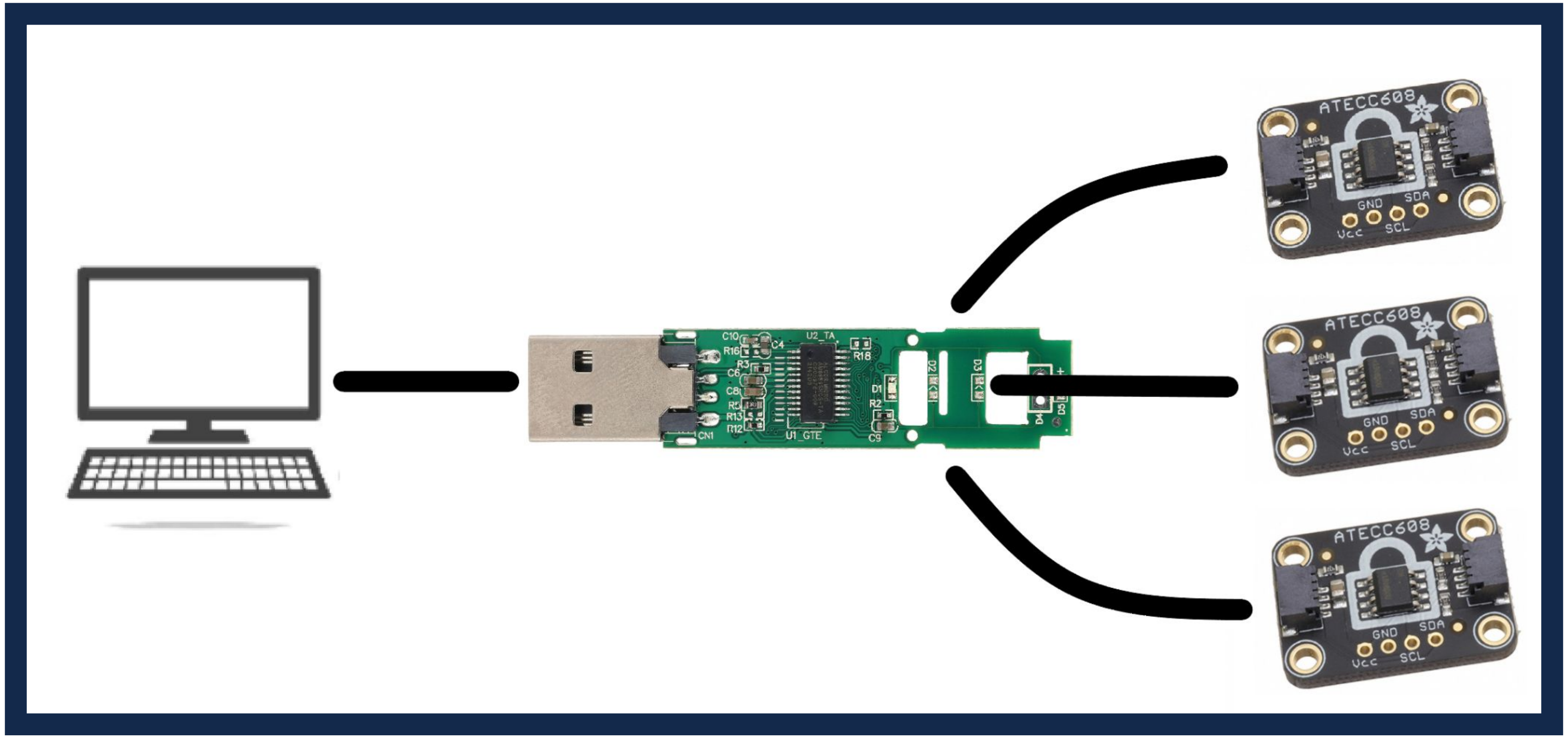
- Not designed for ultimate security despite storing sensitive data
    - ➔ Vulnerable to theft, loss, and unauthorized access
    - ➔ Software encryption can be bypassed via brute force & system exploits
    - ➔ Hardware encrypted drives still rely on passwords, lack tamper response
- About 90% of user passwords can be cracked within a few seconds

Solution

Our answer is a custom PCB flash drive with built-in hardware security:

- Uses replicated secret sharing to verify user
  - ➔ Encryption key split across 3 authentication cards
  - ➔ 2/3 authentication cards required to unlock data
  - ➔ No passwords required
- Includes tamper-resistant data deletion circuit
  - ➔ Triggers upon case removal or failed authentication
  - ➔ Operates even while disconnected from computer

Visual Aid



High Level Requirements

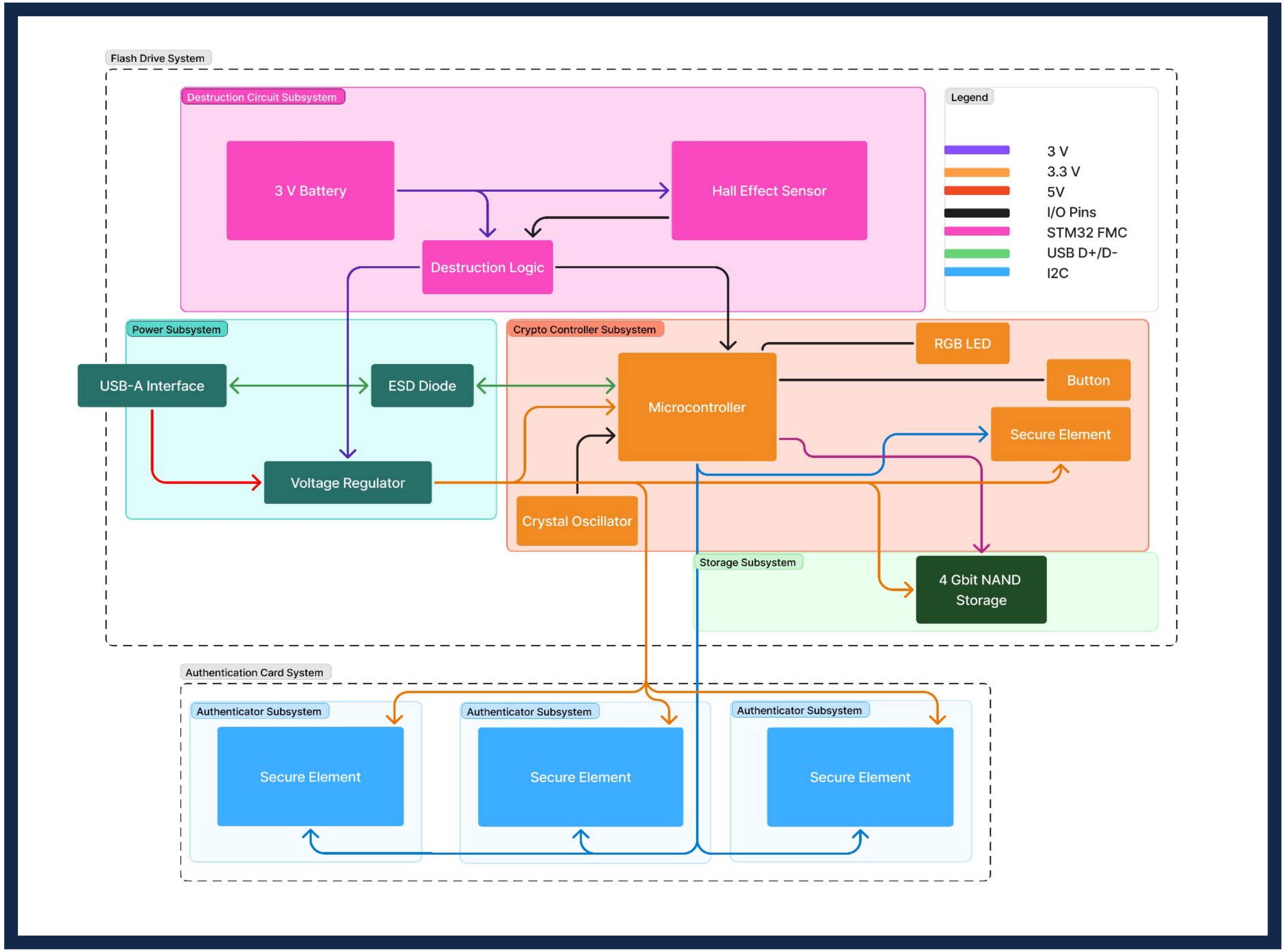
1. The flash drive must allow a maximum of 5 failed authentication attempts before triggering the self-destruct.

2. The flash drive must require at least 2 out of 3 physical authentication cards to decrypt the hidden partition.

3. The flash drive's various modes of encryption should all utilize at least 256-bit keys.

DESIGN

Original Block Diagram

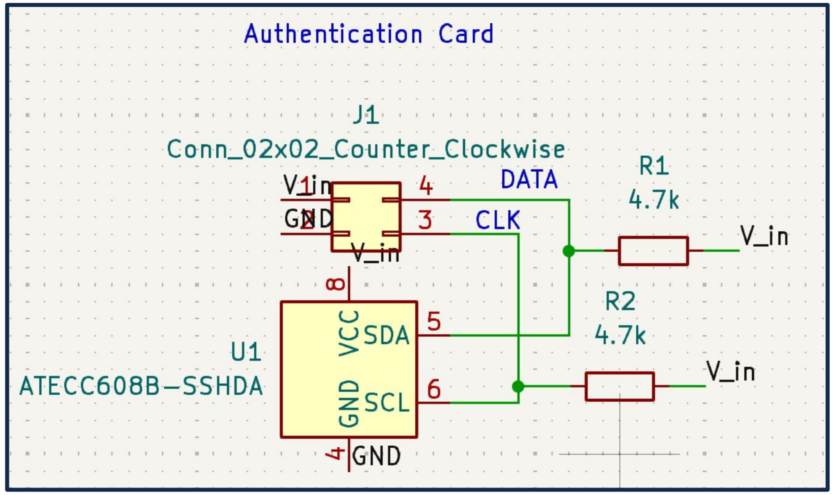


Authentication Subsystem

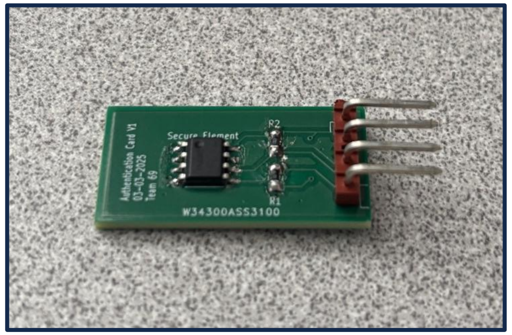
Securely verifies identity of the user

- Consists of 3 separate PCBs with ATECC608B Secure Elements, known as "authentication cards"
  - ➔ Each holds a cryptographic key share
  - ➔ Verifies its identity via I2C communication
- Communicates with Crypto Controller subsystem for auth verification
- Requires ⅔ cards connected via GPIO pins to unlock data

Each individual authentication card schematic



Physical Authentication Card



Cryptographic Security

Utilize AES-256 encryption to protect data stored on NAND flash

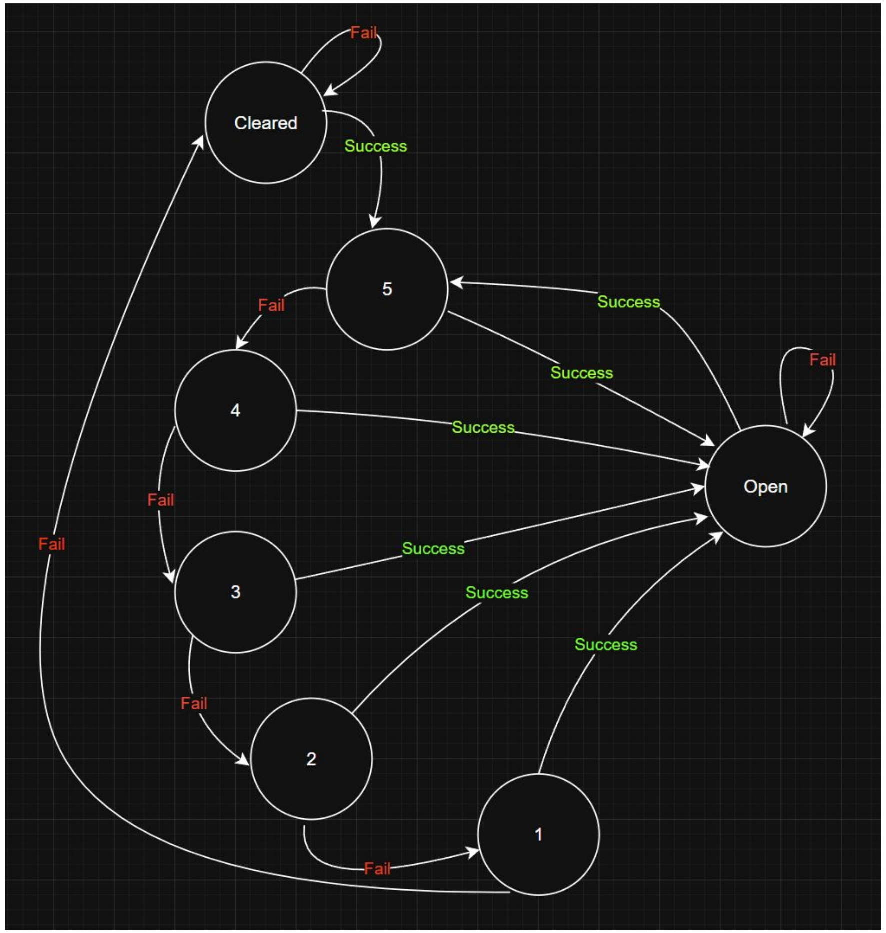
- Encryption key **K** split into three parts using XOR:

$$K = K0 \oplus K1 \oplus K2$$

- All three K values are required to decrypt data, and each authentication card holds a pair of the keys
  - ➔ Card1: Enc(K0,K1), Card2: Enc(K1,K2), Card3: Enc(K0,K2)
  - ➔ ⅔ cards necessary to form K
  - ➔ SHA-256 hash used to validate constructed K
- Secure erase triggered after 5 failed attempts

Cryptographic Security - FSM

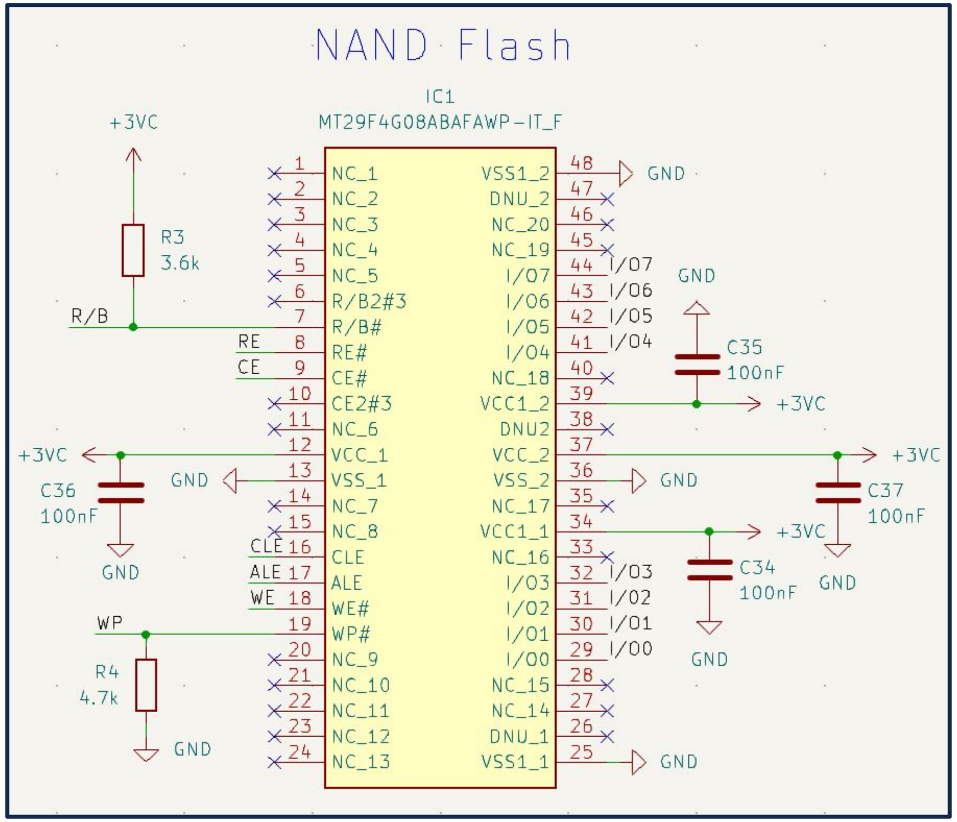
Success defined all three K values are available (⅔ cards present)



Storage Subsystem

In charge of handling the USB data storage and erasure

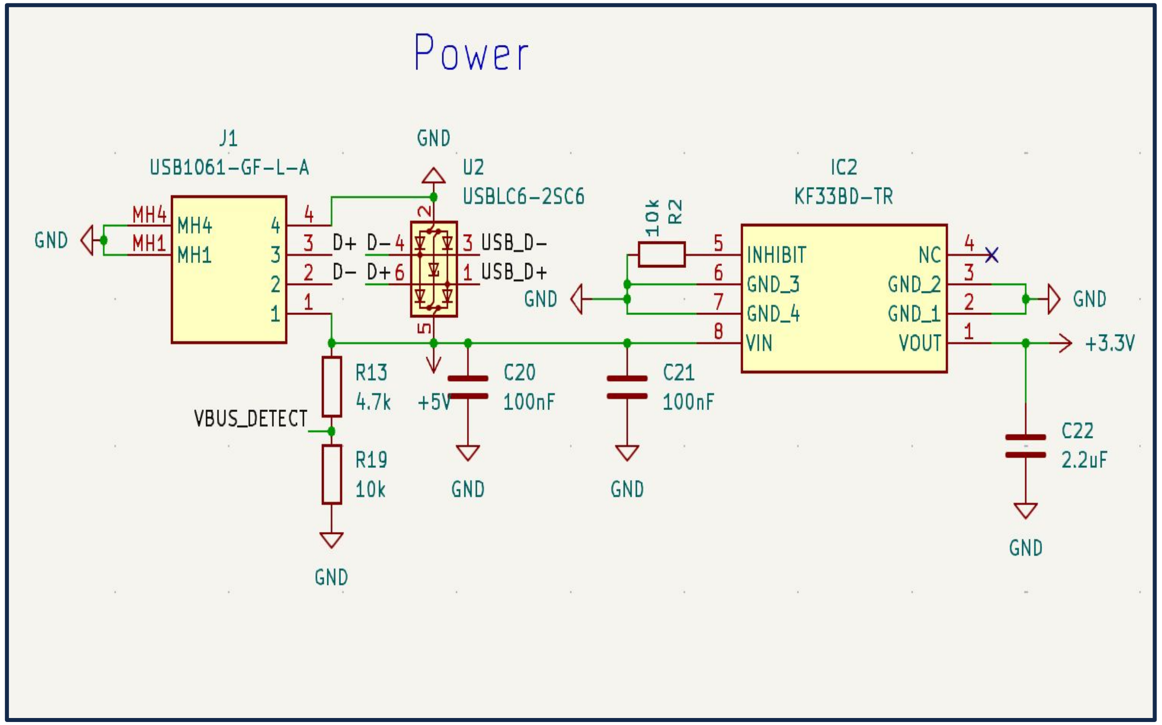
- Uses a 0.5 GB NAND flash chip to store encrypted data
  ➔ Data only accessible after authentication
- If USB is tampered or auth fails 5 times, microcontroller triggers the NAND to employ BLOCK ERASE data wipe
- Powered by 3.3V from USB or 3V from battery during tamper events



Power Subsystem

- Majority of systems powered by USB
  - ➔ 5V input converted to 3.3V via regulator
  - ➔ ESD diode to protect against static discharge

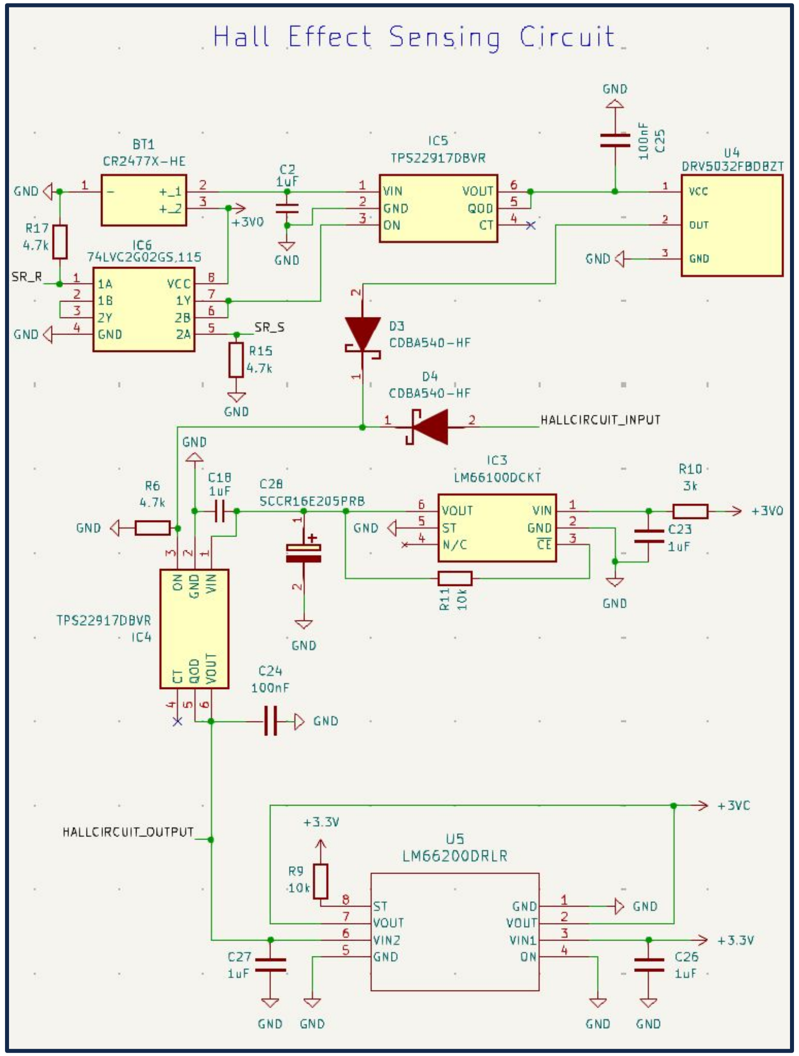
- Backup coin battery used when USB disconnected
  - ➔ Powers microcontroller & NAND **only** during tamper events
  - ➔ Enables data wipe when unplugged



Tamper Detection Subsystem

Ensures USB protection against physical tampering by triggering secure data wipe

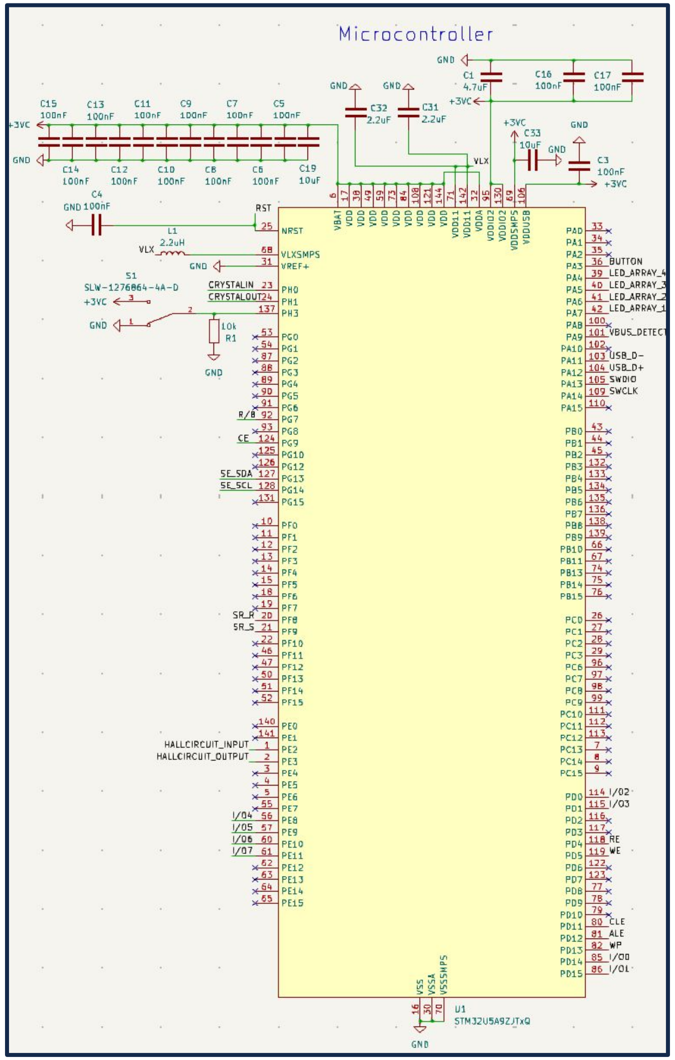
- Uses Hall Effect Sensor & internal case magnet to detect case removal
- Detection triggers data erasure sequence
  - ➔ Switches coin battery power to microcontroller and NAND flash
  - ➔ Signals for data wipe on NAND
- System only activated after key setup, avoids false triggers



Crypto Controller Subsystem

Controls the authentication process, USB communication, and NAND data control

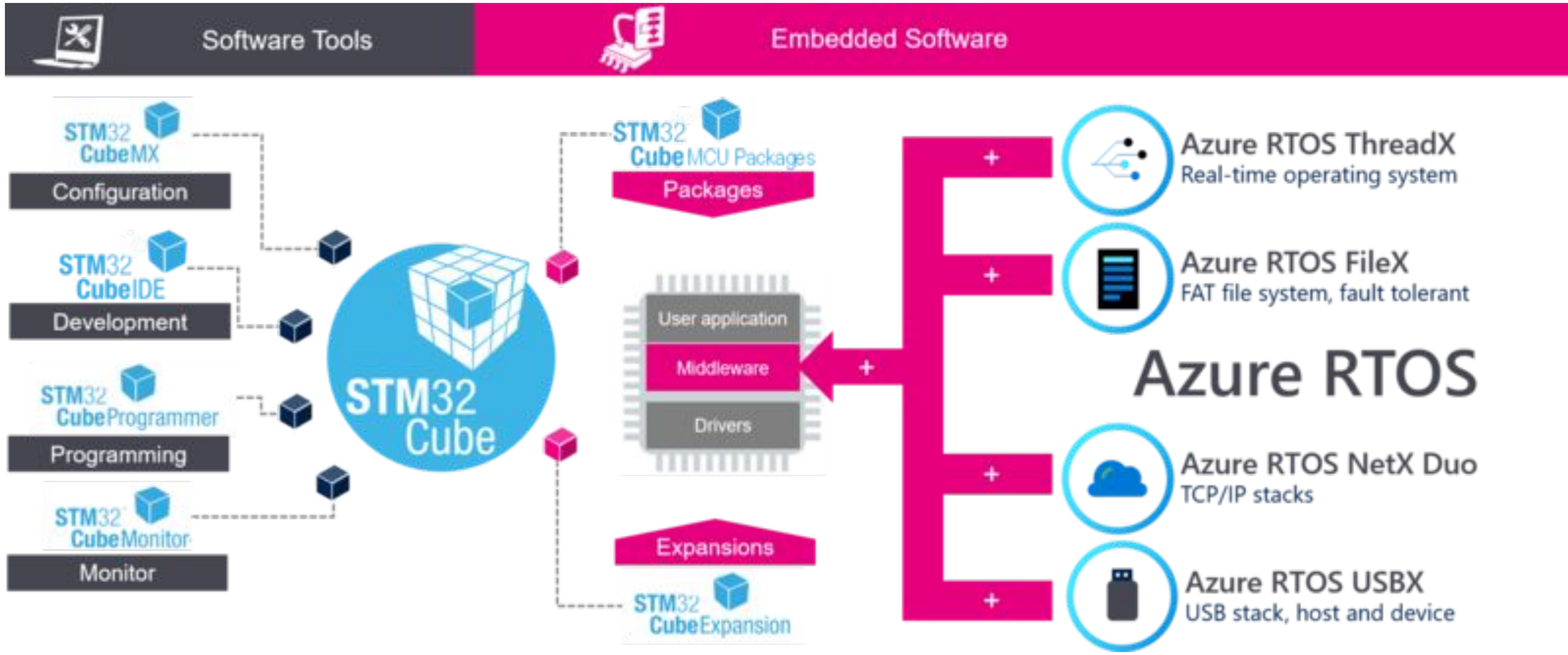
- Built around the STM32U5A microcontroller, which interfaces with 3 main peripherals
  - ➔ USB port for data I/O
  - ➔ NAND flash via flexible memory controller
  - ➔ Secure element and Auth Cards via I2C
- Manages data encryption and authentication
- Includes LEDs to showcase state and button to initiate authentication



Firmware & Software Libraries

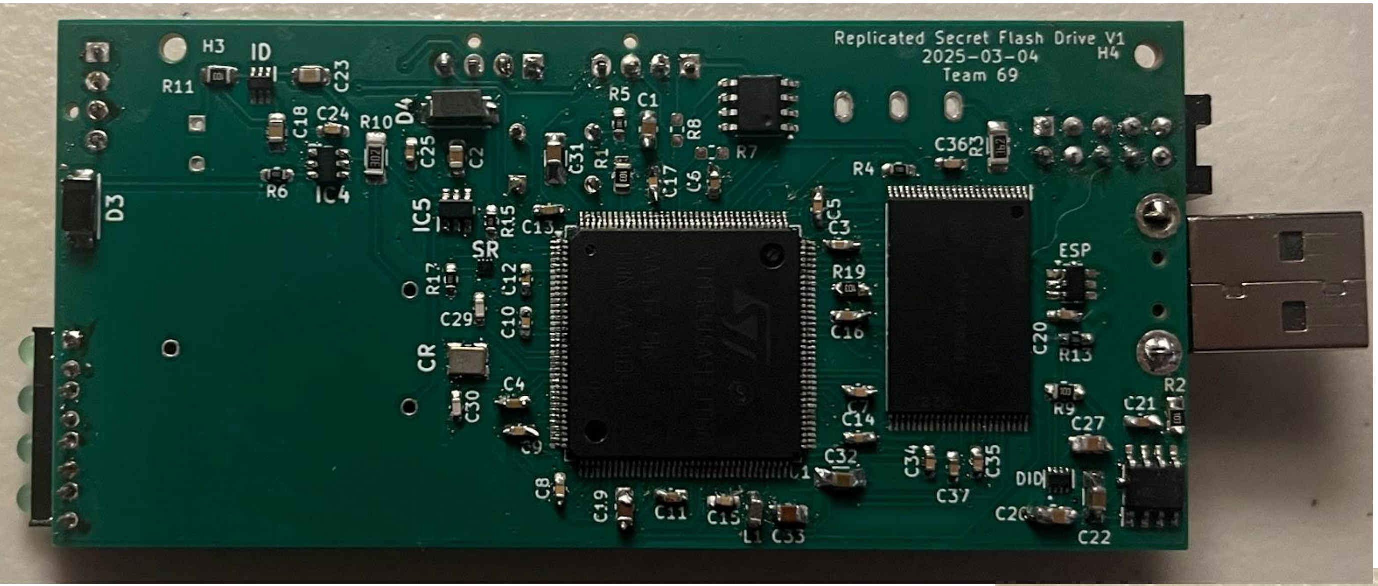
Used STM32 Cube IDE & four ST-based Libraries

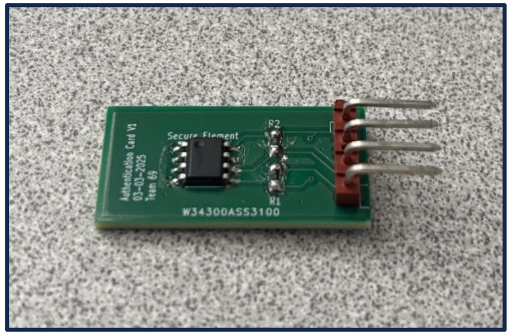
- ThreadX
  - Azure RTOS
  - Manages entire system
- USBX
  - Manages USB stack
  - Enumerates our USB as a MSC
- FileX & LevelX
  - LevelX handles low-level NAND operations
  - FileX handles formatting of drive

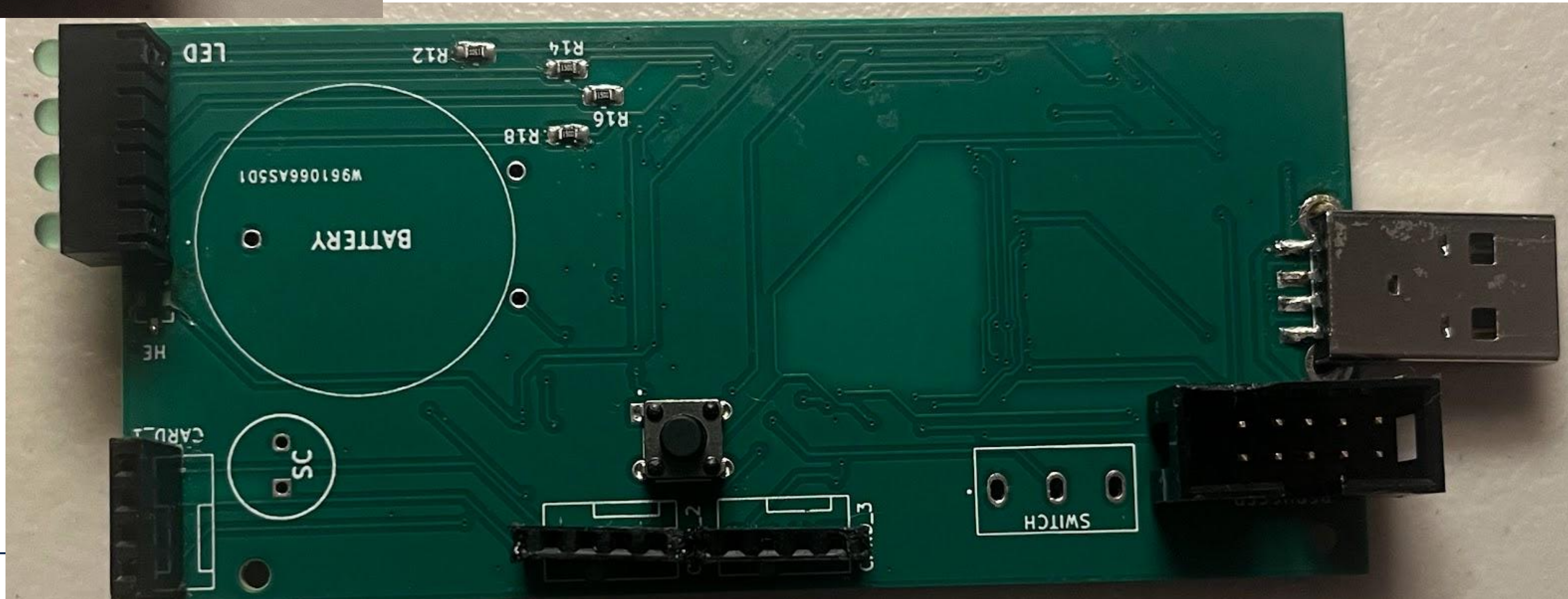


PROJECT BUILD

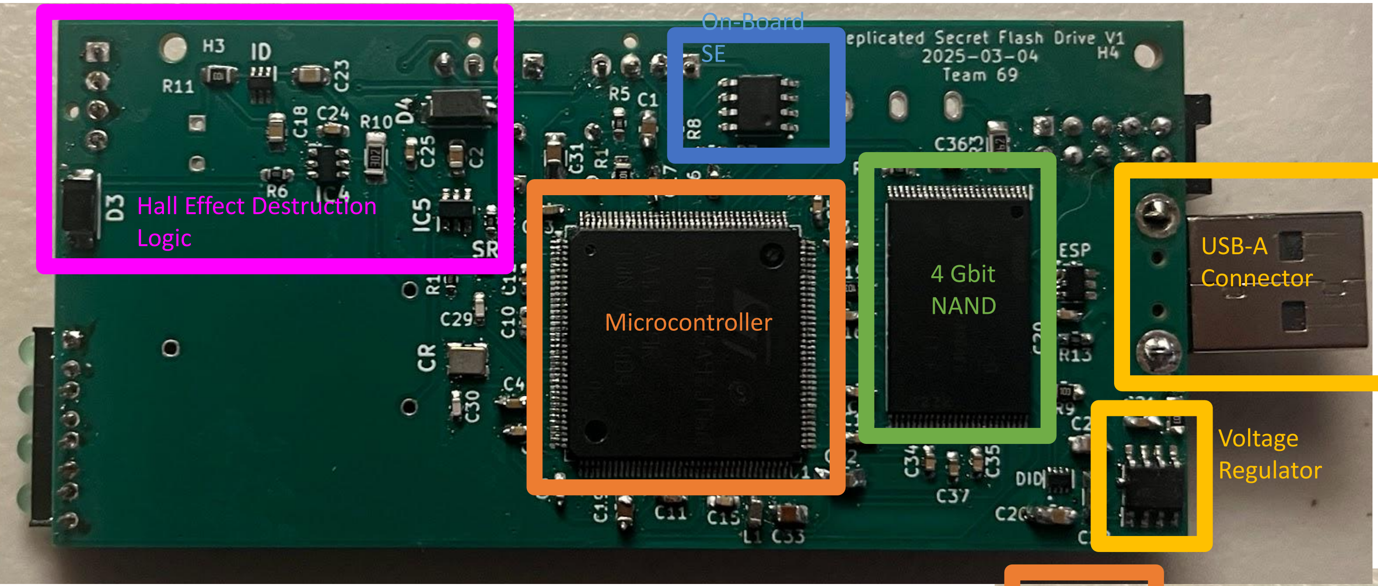
Project Build



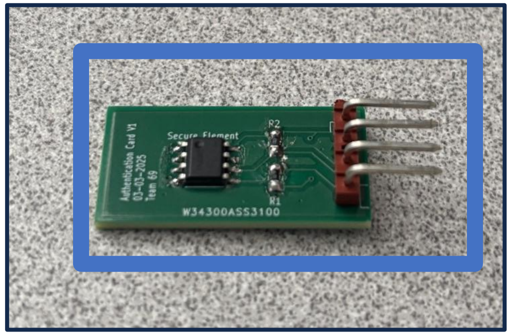




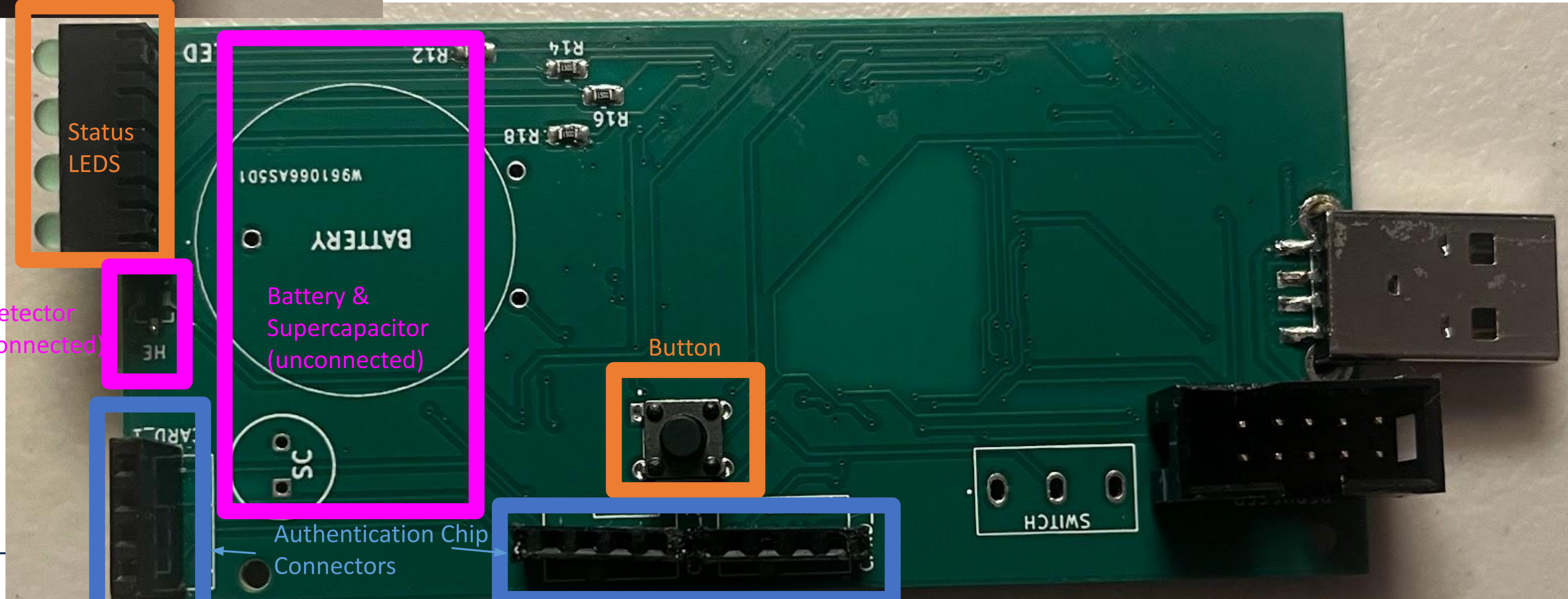
Project Build



Legend
- Tamper Detection Subsystem
- Crypto Controller Subsystem
- Storage Subsystem
- Power Subsystem
- Authentication Subsystem



Authentication Card



Crypto Controller - R&V

Requirements	Verified?	Reason (No) / Verification (Yes)
The microcontroller must be able to successfully transmit data in and out of the USB port	Yes	PCB enumerates as a MSC storage device when authenticated and plugged into a computer.
The microcontroller must communicate with the NAND flash using the Flexible Memory Controller	Yes	Data writted and read from the same addresses is the same.
The microcontroller must communicate with the secure elements using I2C.	Yes	Successful detection of all authentication cards when plugged in.
The LED must display the correct status when the button is pushed	Yes	Successful traversal and display of all FSM states.

Authentication Subsystem - R&V

Requirements	Verified?	Reason (No) / Verification (Yes)
All 3 authentication cards are able to be plugged into the USB via GPIO pins and initialized at the same time.	Yes	Initialization loop functions and encrypted values present on cards
Once initialized, the K-pair held in each authentication card cannot be altered or changed.Additionally, no further authentication cards can be initialized.	Yes	Initialization loop with one initialized card and two uninitialized cards does not unlock memory.
When connected to the USB PCB, the Authentication Card Secure Element is automatically prompted to send its K-pair via I2C communication.	Yes	Oscilloscope view of I2C shows correct transmission.

Power Subsystem - R&V

Requirements	Verified?	Reason (No) / Verification (Yes)
Must be able to regulate USB power to power components throughout the duration of connectivity to the computer.	Yes	Power draw is a stable 3.3V when connected to USB (verified through oscilloscope)
Proper ESD protection on USB Data Lines	Yes	Data lines still stable after multiple USB plug/unplug cycles.
Must be able to protect against variable changes in USB power input, as it may overvolt or draw too much current.	Yes	Regulator keeps voltage stable even with an overvoltage

Storage Subsystem - R&V

Requirements	Verified?	Reason (No) / Verification (Yes)
The NAND Flash correctly reads and writes data when the correct Authentication Cards are utilized.	Yes	MCU can store and retrieve data from NAND flash w/ authentication cards connected.
The NAND Flash contains only encrypted data, nothing that would be understandable without an encryption key.	Yes	Encrypted data received from NAND is unintelligible before decryption (verified in debugger).
All the valid data blocks stored on the NAND Flash are deleted once the destruction sequence is enacted with the physical tampering or incorrect Authentication Cards.	No	Ran out of time for demo, did get data blocks deleting after the demo.

Tamper Detection Subsystem - R&V

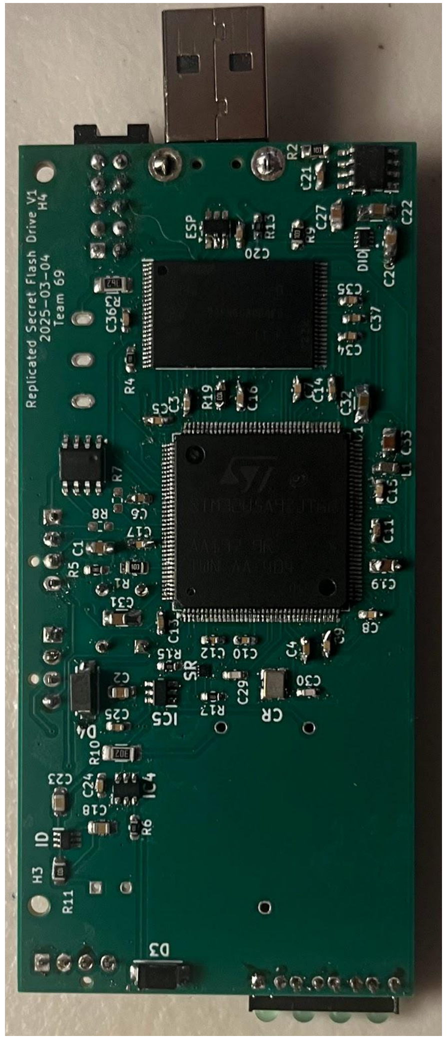
Requirements	Verified?	Reason (No) / Verification (Yes)
The signals sent from the Destruction Logic are not able to interface with the Microcontroller until the Cryptographic Keys are initialized.	No	No signals coming from the Destruction Logic.
Once the USB casing and magnet are removed, the Hall Effect Sensor stops sending its signal to the Destruction Logic.	No	Incorrect footprint for Hall Effect sensor ordered.
The Destruction Logic sends signals to the microcontroller to initiate data deletion and connect routing of the 3V coin battery to power the microcontroller and NAND memory.	No	No signals received, however routing was tested and functioned correctly.

CONCLUSIONS

Successes

Successfully created a secure flash drive

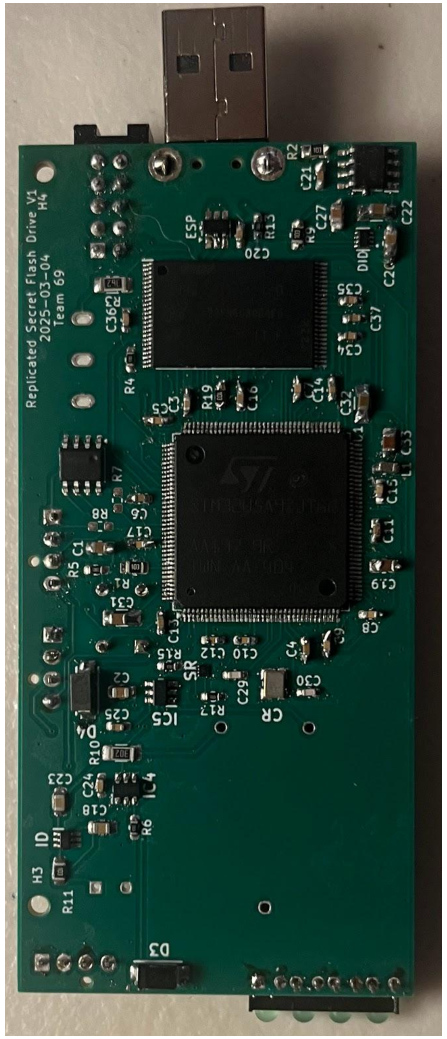
- Flash drive successfully interfaces with a computer
- Data storage is robust and can store and retrieve all data without corruption
- All data is successfully encrypted and decrypted
- Key exchange with secure elements functions correctly
- PCB functions correctly and no breadboard support needed



Aspects Not Completed

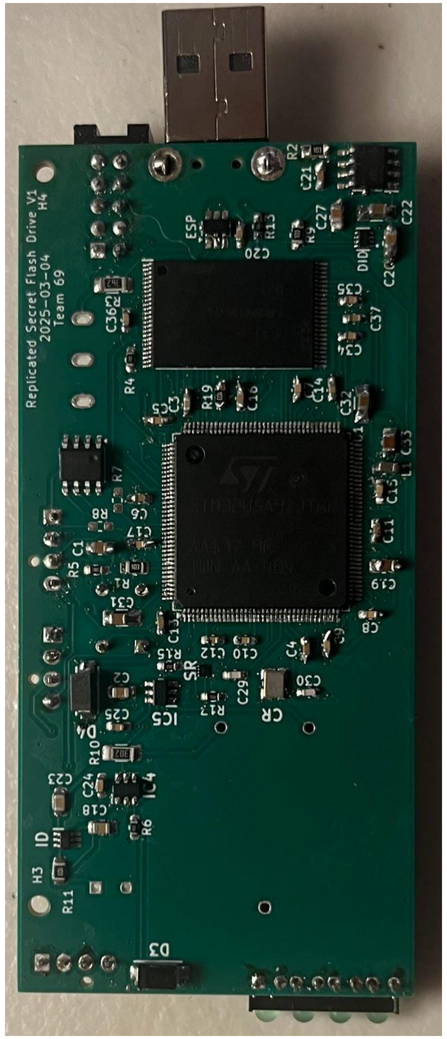
Fell short of our goal of absolute security

- No working tamper-detection subsystem
  - Attacker could theoretically connect to pins and brute force encryption
  - Need design of case to house magnet for Hall Effect Detection
- Didn't get data deletion working in time for demo
  - Did get it functioning shortly after



Challenges

- Soldering
  - 144-pin microcontroller and 48-pin NAND makes for a difficult solder
- Complexity of Microcontroller
  - Complex power supply
  - Required full RTOS to work with ST libraries
- Data Security of Secure Element
  - Working without a datasheet for secure element
  - In order to get datasheet we would need to travel to Chicago and sign an NDA



What We Learned

Designing a system like this from end-to-end was immensely educational

- Soldering skills vastly improved
- Find a way to breadboard before PCB implementation
- Check over your teammates work (especially hardware)
- Plan for tasks to take twice as much time as you think they will
- Lots of new embedded software knowledge
    - Interfacing this device with a computer was a really cool moment

Learning Moments

Three design decisions that would have reduced complexity

- Less complex microcontroller
  - Contained everything we needed for this project, definitely a bit overkill
  - Could split some functionality into different components
    - PHY Converter, used crypto functions on secure elements
- NOR storage instead of NAND storage
  - Less complex driver, less soldering, overall easier to integrate
  - Would result in essentially the same functionality
- Less secure Secure Element
  - Lots of headaches with no datasheet, essentially running blind

FUTURE WORK

Future Improvements

There are multiple design and production avenues we could explore to further enhance this project.

- Test and Develop the finished tamper detection circuit
  - ➔ Create 3D-printed encasing to finish
- Include new NAND/NOR flash with more storage
- Minimize size of actual PCB
- Designate a more fleshed out LED indicated state machine for user

QUESTIONS?