LEDGER

hardwear.io
Hardware Security Conference and Training

OneKey is all it takes

The misuse of Secure Components in Hardware Wallets

Michaël Mouchous – Karim Abdellatif – 06.02.2023

WHO ARE WE?

LEDGER DONJON

Improve the security level of the cryptocurrency ecosystem

Michaël Mouchous

Karim Abdellatif

Thanks to Olivier Hériveaux and all other Ledgers

CONTEXT OF EVALUATION

Laser Fault Injection

- Mid-cost bench creation
  Ledger Blog - Mounting a low-cost laser bench

OneKey Mini (August 2021)

- Secure Memory: ATECC 608A
  Blackbox Laser Fault Injection on a Secure Memory - Olivier Heriveaux - SSTIC 2020
- MicroController Unit: STM32F4
  blog.ledger.com/Unfixable-Key-Extraction-Attack-on-Trezor

Easy Analysis

- Open Source Firmware
  github.com/OneKeyHQ/firmware/tree/mini
- Fork from Trezor project
  Addition of Secure Memory

OUTLINE

Hardware description

MCU Memory Dump

Focus on PIN

Laser Fault Injection on ATECC

Conclusion

HARDWARE DESCRIPTION

HARDWARE ARCHITECTURE

SECURE MEMORY – ATECC

Secure Authentication IC by MicroChip

3 revisions: 508A, 608A, and 608B

- 508A and 608A are **deprecated**

Features

- AES128 / SHA256
- ECDSA / ECDH
- Secure Boot Support
- Two Monotonic Counters
- Random Number Generator
- Unique 72 bits SerialNumber

SLOTS

16 Slots used as 32 bytes entries

Keys, Certificates or **User Data**

Protection Levels

Configuration **immutable** after init

Slot’s content optionally **immutable**

PROTECTION LEVELS

R/W Protection	Description
PUBLIC	**Read** / **Write** accepted
PROTECTED$_{KeyId}$	**Nonce** **GenDig(KeyId)** $TempKey \leftarrow SHA_{256}(Nonce, SN, Key_{KeyId})$ **Read** / **Write** granted $C \leftarrow XOR(Data, TempKey)$
PRIVATE	**Read** / **Write** refused

IsSecret

- Slot must be **PRIVATE** or **PROTECTED**$_{KeyId}$
- Internal encryption *AES*

ATECC STORAGE – ONEKEY MINI SLOTS CONFIGURATION

#	Name	Purpose	Protection levels		
			IsSecret	Read	Write
0	PRIMARY PRIVATE KEY	[...]	YES	PRIVATE	PUBLIC
1	**IO PROTECT KEY**	**Protect usage of other slots**	**YES**	**PRIVATE**	**PRIVATE**
2	USER PIN	Stores hash of PIN	YES	PRIVATE	$PROTECTED_1$
3	PIN ATTEMPT	SHA context initialisation	YES	PRIVATE	PRIVATE
4	COUNTER MATCH	Incremented on each PIN verif	NO	PUBLIC	$PROTECTED_1$
5	LAST GOOD COUNTER	Last successful PIN verif	NO	PUBLIC	$PROTECTED_1$
6	**USER SECRET**	**Seed mnemonics**	**YES**	**$PROTECTED_1$**	**$PROTECTED_1$**
7	USER STATE	State machine (PIN set, device initialized, seed strength)	NO	PUBLIC	$PROTECTED_1$
8	DEVICE CERT	[...]	NO	PUBLIC	PUBLIC

MCU – OTP MEMORY

16 Blocks of 32 bytes

Initial value is **FFFFFFFF**...

Fuse writing: bits can only be **unset**

Can be locked to be **immutable**

Protected against reading

MCU – OTP MEMORY LAYOUT

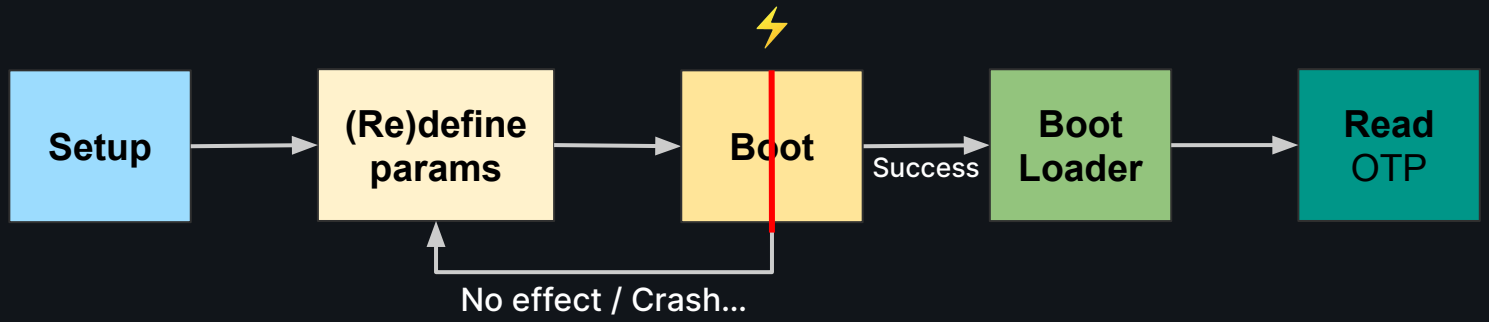
#	Name	Purpose	Source
3	RANDOMNESS	Hardware entropy	MCU - random32()
7	MCU SERIAL	MCU Serial Number	External
8	ATECC SERIAL	ATECC Serial Number	ATECC - GetSerial command
9	**IO PROTECT KEY**	**ATECC Protection Key**	**ATECC - Random command**
10	ATECC INIT PIN	PIN reset value	ATECC - Random command
11	ATECC MIX PIN	Salt for PIN Hashing	ATECC - Random command
13	ATECC CONF VER	Configuration version of ATECC	MCU - Hardcoded "0.0.1"
14	FLASH ENC KEY	AES Encryption key for external flash	MCU - random32()
15	CPU INFO | PRE FIRMWARE	Informations during initialization	External source value

1st vulnerability

MCU – OTP DUMP

MCU – OTP DUMP – ATTACK DESCRIPTION

Motivation: Retrieve **`IO PROTECT KEY`** value from OTP
Method: Perturb boot phase to grant access of bootloader, then OTP
Previously performed by Karim Abdellatif on various versions of MCU
blog.ledger.com/Unfixable-Key-Extraction-Attack-on-Trezor



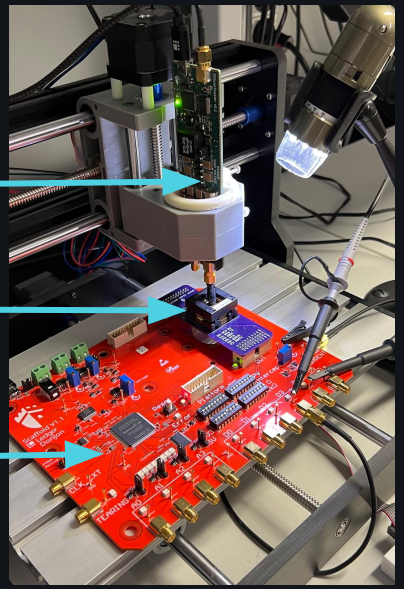
⚠️ Reducing read protection level erases Flash Memory ⚠️
We don't care because **`IO PROTECT KEY`** is stored in **OTP**

MCU – OTP DUMP – SETUP

SiliconToaster
EM Injection Probe

MCU

Scaffold
Communication board
Current measurement
Signals generation



EMFI setup

Power consumption during boot phase

[SiliconToaster: A Cheap and Programmable EM Injector for Extracting Secrets]
[Scaffold - Donjon hardware tool for circuits security evaluation]

hardwear.io
Hardware Security Conference and Training

MCU – OTP DUMP – RESULT

0 1 2	UNUSED	ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
3	RANDOMNESS	19a361efa384f4e094ea5fe29614cf13ca7b95040c1997348e4b07cd7292de3c
4 5 6	UNUSED	ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
7	MCU SERIAL	4d4930355730313230323131323239303631393536303030333731350000000000 .M.I.0.5.W.0.1.**2.0.2.1.1.2.2.9**.0.6.1.9.5.6.0.0.0.3.7.1.5........
8	ATECC SERIAL	0123d71b4d302793eeffffffffffffffffffffffffffffffffffffffffffffff
9	**IO PROTECT KEY**	**ffff0000ffff0000ffff0000ffff0000ffff0000ffff0000ffff0000ffff0000**
10	ATECC INIT PIN	ffff0000ffff0000ffff0000ffff0000ffff0000ffff0000ffff0000ffff0000
11	ATECC MIX PIN	ffff0000ffff0000ffff0000ffff0000ffff0000ffff0000ffff0000ffff0000
12	UNUSED	ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
13	ATECC CONF VER	302e302e31000000ffffffffffffffffffffffffffffffffffffffffffffffff 0 . 0 . 1
14	FLASH ENC AES KEY	c7499ccd82f554d5d883ace3fb3a0e82cf2048d503f3805fcaa60325fad74a36
15	CPU INFO	0000000000000000000000000000000000000000000000000000000000000000

MCU – OTP DUMP – CHECK THE KEY ON ATECC

1. Check the content of Slot 1
   a. Get `Nonce` and generate *rand*
   b. Computation of `M` ← SHA_{256}*(Nonce, Key, rand, SN)*
   c. `CheckMac(Slot:1, Chall:rand, Mac:M)` ⇒ OK!

2. Read content of Slot 6
   a. Get `Nonce`
   b. `GenDig(KeyId:1)`
   c. Compute *TempKey* ← SHA_{256}*(Key, SN, Nonce)*
   d. `C` ←`Read(Slot:6)`
   e. `P` ←`C` *XOR* `TempKey` ⇒ Value of the seed!

`ffff0000...` value **IS** the data contained in Slot 1

1st vuln: Impossible to patch: **SLOT 1** and **OTP** are **LOCKED**

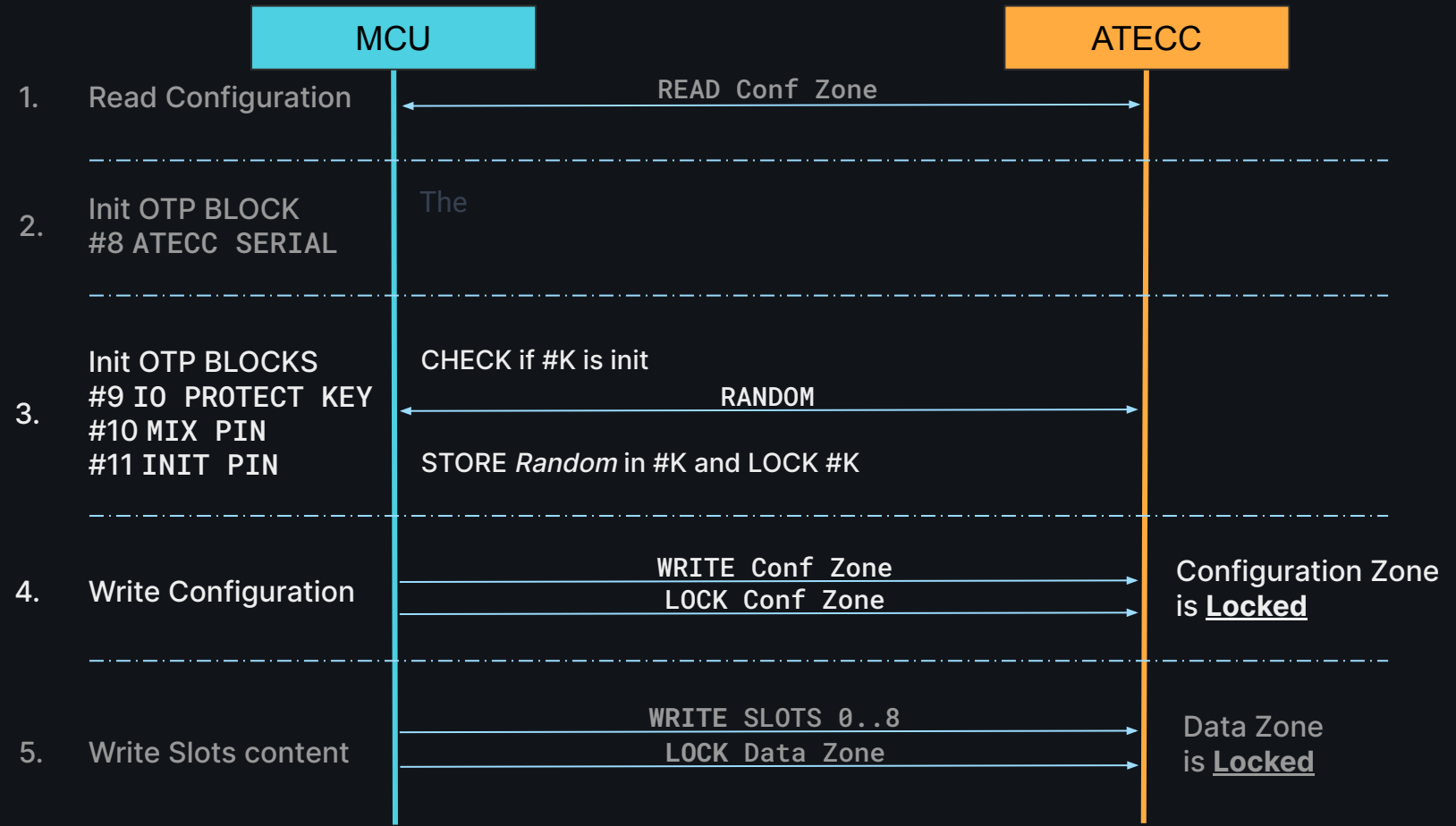
MCU – OTP DUMP

July 2022 – Buy another OneKey Mini

7	MCU SERIAL	4d4930355730313230323230353138313030383135303030383632360000000 .M.I.0.5.W.0.1.**2.0.2.2.0.5.1.8**.1.0.0.8.1.5.0.0.0.8.6.2.6.....
8	ATECC SERIAL	0123f95c26ce71a9eeffffffffffffffffffffffffffffffffffffffffffffff
9	**IO PROTECT KEY**	94f0fd15b00d95c9b20df0cd54aa60f3ff57d93a16a79116ba572a3fd62eb322
10	ATECC INIT PIN	267fdeb31cd5147761439a89864149bb741b146bbac14d5fe4a63b371148e30b
11	ATECC MIX PIN	c32492acff275f6fc85f9cae12f6f147b85b44ed3704eaaa48472ab9718fe5fd
13	ATECC CONF VER	302e302e31000000ffffffffffffffffffffffffffffffffffffffffffffffff 0 . 0 . 1
14	FLASH ENC AES KEY	f3e92bbaed99f43bcfaf4595a17df2affa5b20b3030cae45128ea88170a751a9
15	CPU INFO	0000000000000000000000000000000000000000000000000000000000000000

IO PROTECT KEY: true random? 🤯

ATECC x MCU – INITIALIZATION PHASE



ATECC – DATASHEET SPECIFICATIONS

Random Command

The Random command generates a random number for use by the system.

Random numbers are generated via the internal NIST 800-90 A/B/C random number generator.

Prior to the Configuration zone being locked, the RNG produces a value of 0xFF, 0xFF, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0x00 to facilitate testing.

1st ROOT CAUSE FOUND!

BROWSING ON GITHUB

× **chore(factory):add device retest function**

lihuanhuan committed on May 9, 2022 commit 34aebe4

Lock of Configuration Zone **has moved up**, prior Random commands

Commit date: **May 9, 2022**

18 legacy/atca/atca_api.c

```
@@ -129,6 +129,15 @@ void atca_config_init(void) {
129 129    atca_assert(atca_read_config_zone((uint8_t *)&atca_configuration),
130 130                "get config");
131 131
    132 +  if (atca_configuration.lock_config == ATCA_UNLOCKED) {
    133 +    atca_assert(atca_write_config_zone((uint8_t *)&atca_init_config),
    134 +                "set config");
    135 +
    136 +    atca_assert(atca_lock_config_zone(), "lock config");
    137 +    atca_assert(atca_read_config_zone((uint8_t *)&atca_configuration),
    138 +                "get config");
    139 +  }
    140 +
132 141    memcpy(serial_no, atca_configuration.sn1, ATECC608_SN1_SIZE);
133 142    memcpy(serial_no + ATECC608_SN1_SIZE, atca_configuration.sn2,
134 143           ATECC608_SN2_SIZE);
@@ -190,15 +199,6 @@ void atca_config_init(void) {
190 199                        FLASH_OTP_BLOCK_608_SERIAL * FLASH_OTP_BLOCK_SIZE),
191 200           sizeof(pair_info_obj));
192 201
193     -  if (atca_configuration.lock_config == ATCA_UNLOCKED) {
194     -    atca_assert(atca_write_config_zone((uint8_t *)&atca_init_config),
195     -                "set config");
196     -
197     -    atca_assert(atca_lock_config_zone(), "lock config");
198     -    atca_assert(atca_read_config_zone((uint8_t *)&atca_configuration),
199     -                "get config");
200     -  }
201     -
202 202    if (atca_configuration.lock_value == ATCA_LOCKED) {
203 203      return;
```

DIFFERENCES BETWEEN TWO WALLETS

1st OneKey Mini's **MCU SERIAL**

```
4d49303557303132303231313232393036313935363030303337313500000000
.M.I.0.5.W.0.1.2.0.2.1.1.2.2.9.0.6.1.9.5.6.0.0.0.3.7.1.5........
```

Initialisation date: **December 29, 2021** (*before* the code change)

2nd OneKey Mini's **MCU SERIAL**

```
4d49303557303132303232303531383031303038313530303038363236000000000
.M.I.0.5.W.0.1.2.0.2.2.0.5.1.8.1.0.0.8.1.5.0.0.0.8.6.2.6........
```

Initialisation date: **May 18, 2022** (*after* the code change)

2nd ROOT CAUSE FOUND!

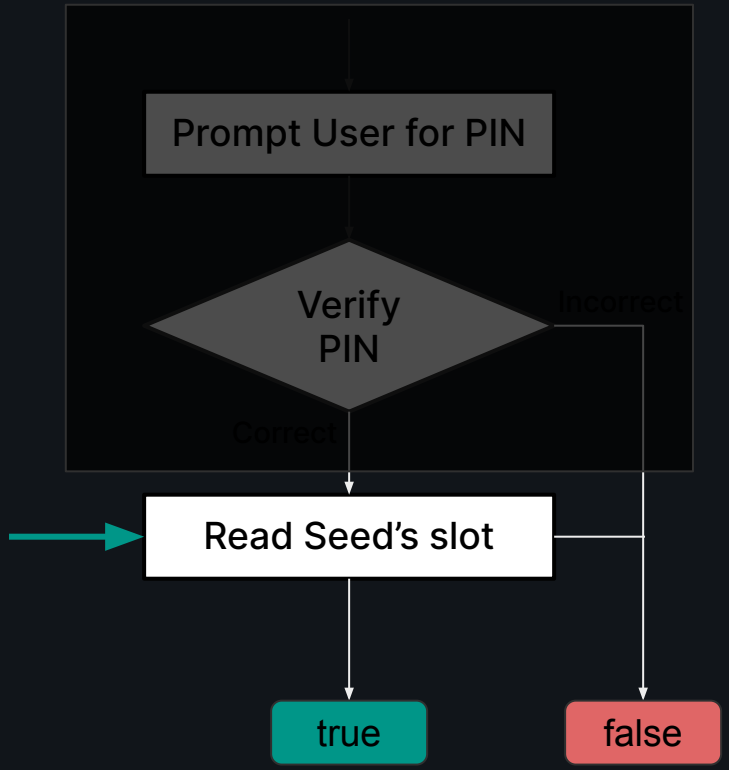
2nd vulnerability

FOCUS ON PIN

HARDWARE WALLETS - PURPOSE OF PIN

- Secret Value, only known by the user, not by the device
- Limited tries before wiping seed (3, 5, 10)
- Seed should be impossible to get without the value of PIN (encryption...)

ONEKEY MINI – EXPORT THE SEED




```
bool se_export_seed(uint8_t *seed) {
    uint8_t pin[32] = {0};
    pin_cacheGet(pin);

    atca_pair_unlock();
    if (ATCA_SUCCESS == atca_mac_slot(SLOT_USER_PIN, pin)) {
        if (ATCA_SUCCESS == atca_read_enc(
            SLOT_USER_SECRET, 0, seed,
            pair_info->protect_key,
            SLOT_IO_PROTECT_KEY)) {
            return true;
        }
    }
    return false;
}
```

ATECC – PASSWORD CHECKING FEATURE

Password Checking

Many applications require a user to enter a password to enable features, decrypt stored data, or perform some other task.

If the device determines that the correct password has been entered, then the device can use this fact to optionally release a secondary high entropy secret.

Current Configuration	Recommended Configuration
PIN hash stored in SLOT 2 SLOT 6 configuration: - ReqAuth: `true` - AuthKey: `0x01`	PIN hash stored in SLOT 2 SLOT 6 configuration: - ReqAuth: `true` - AuthKey: `0x02`

2nd vuln: Impossible to patch: Configuration Zone is **LOCKED**

3rd vulnerability

LASER PERTURBATION ON ATECC

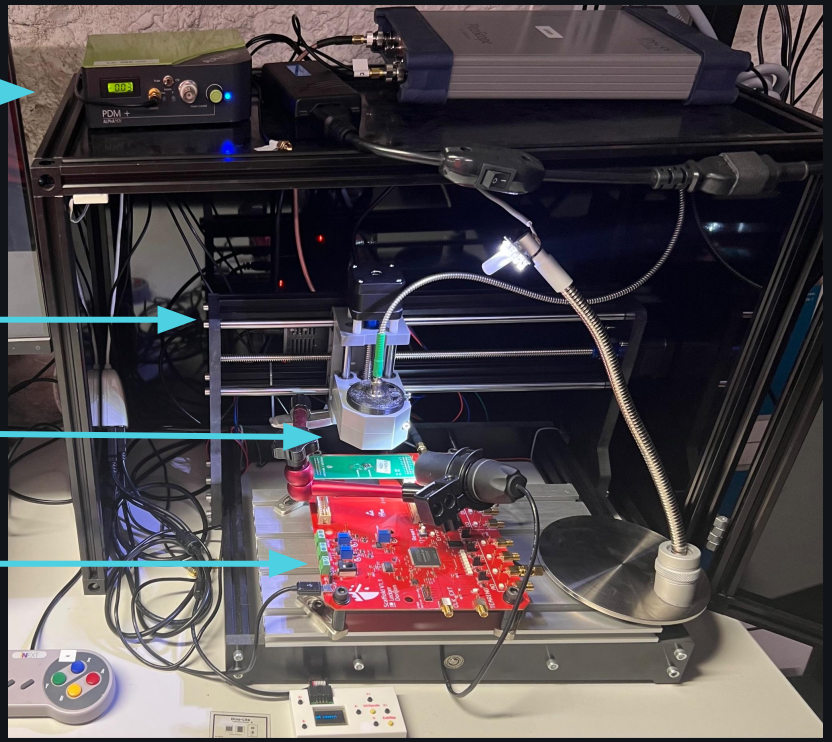
FAULT INJECTION – SETUP

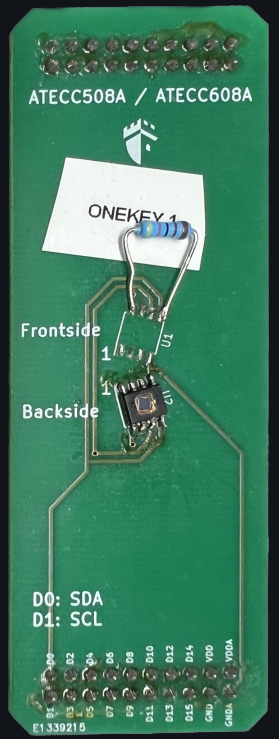
Alphanov PDM+

Low-priced X/Y/Z actuator

Low-priced optics

Scaffold comm board





hardwear.io
Hardware Security Conference and Training

FAULT INJECTION – ATECC CONFIGURATION

Motivation: ATECC608A / ATECC608B Security level comparison

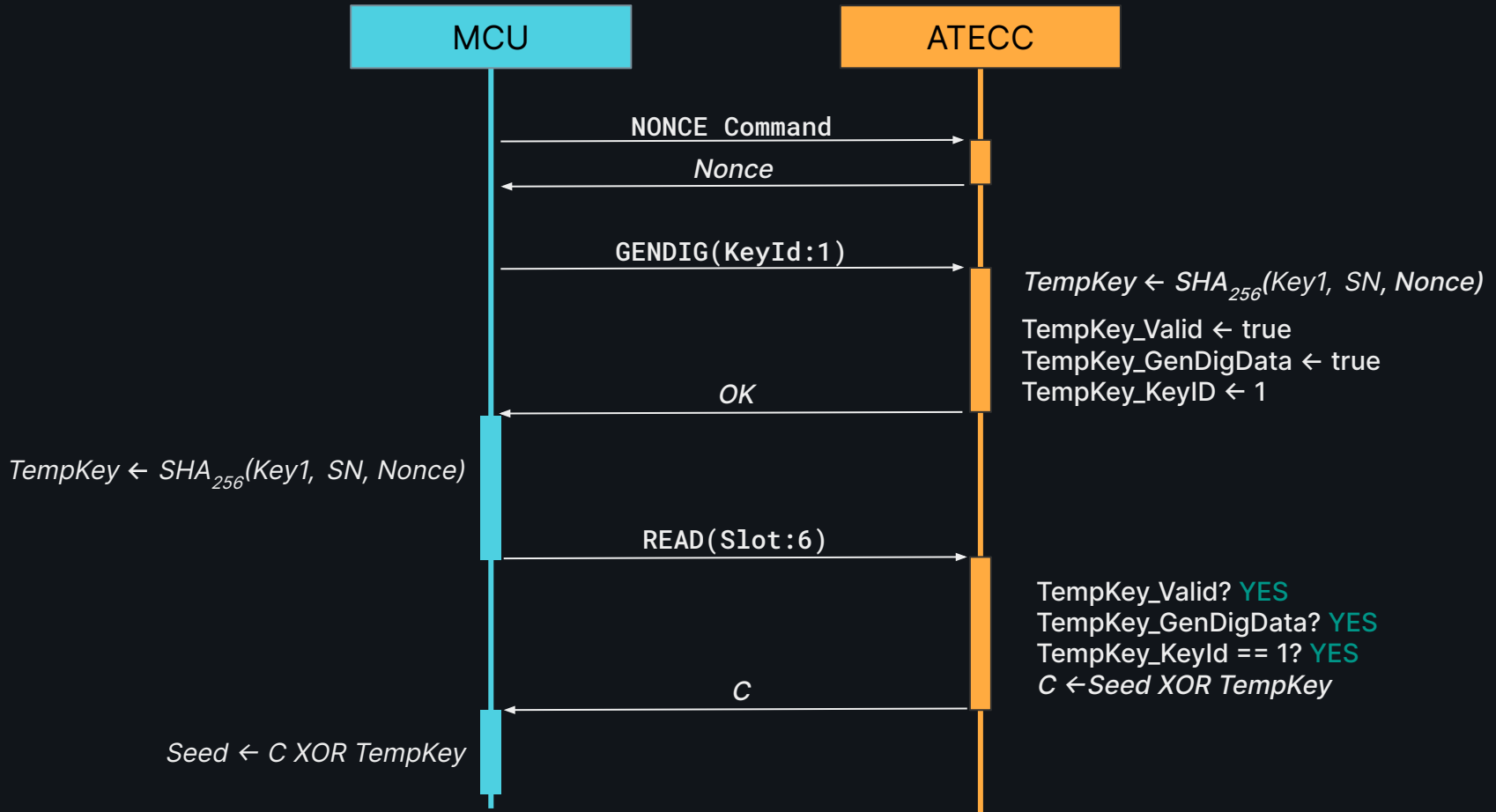
Open Samples

Same configuration as original but

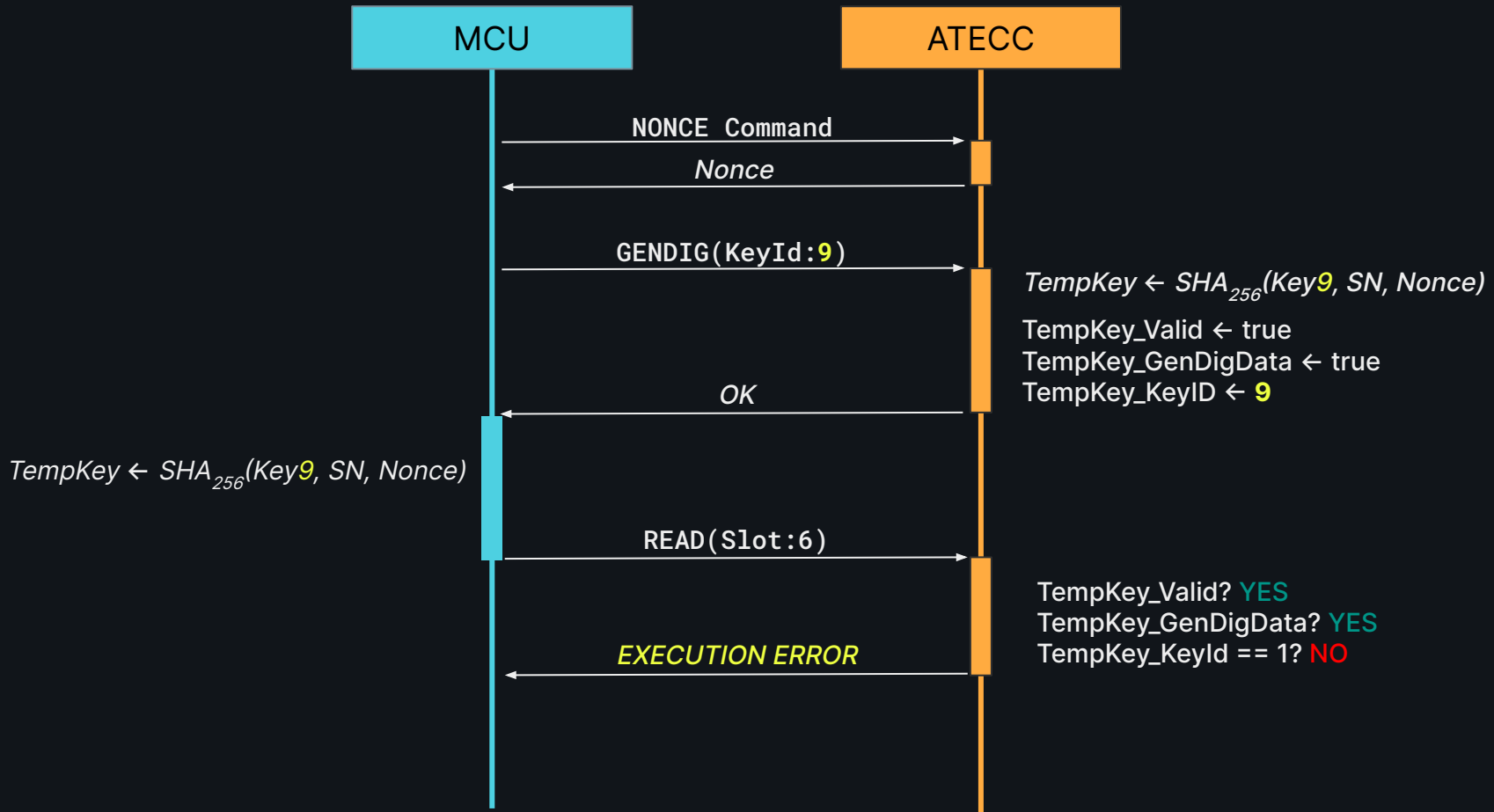
- Slot 1 protection level **is lowered to PROTECTED$_{10}$**
- Slot 1 (`IO PROTECT KEY`) is **known**
- Slot 9 and Slot 10 are set with **known values**

Attack scenario:
Perturb `READ` operation of SLOT 6 (**PROTECTED$_1$**) without knowledge of SLOT 1, using SLOT 9 instead

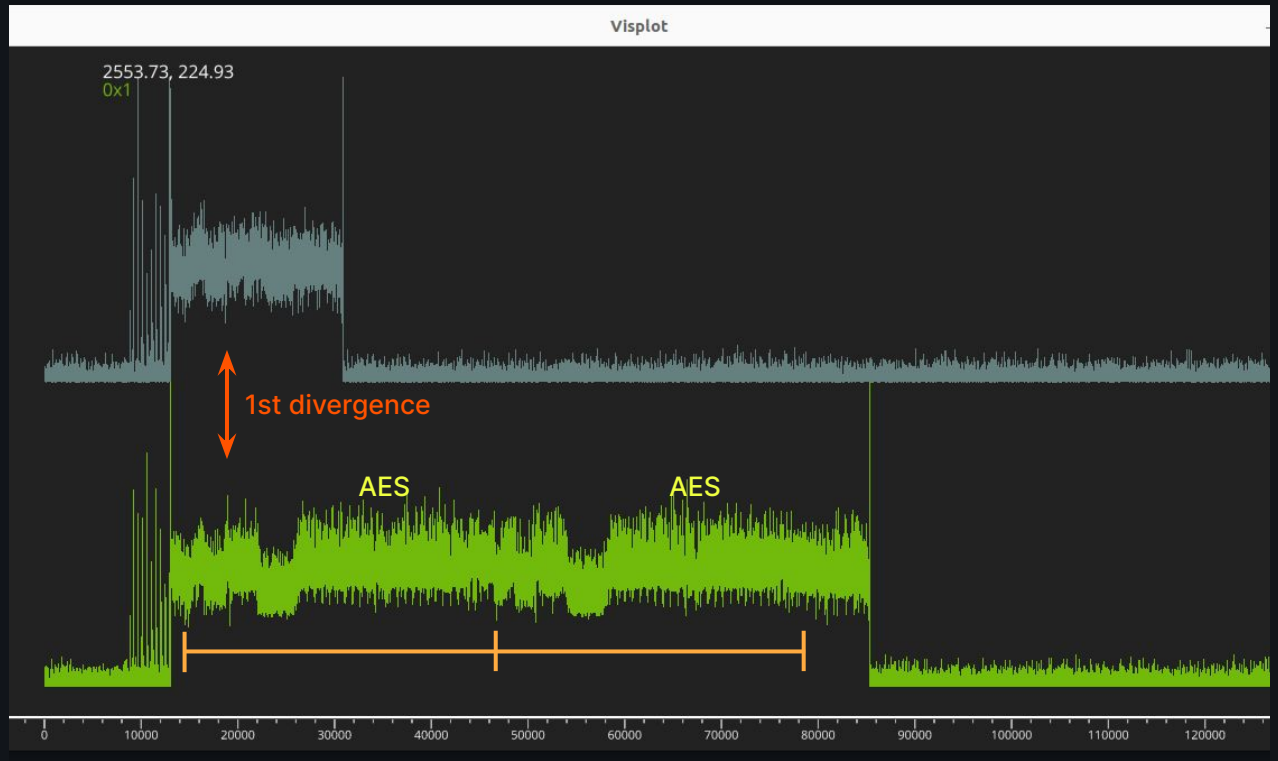
READ PROTECTED SLOT 6 – NORMAL EXECUTION



READ PROTECTED SLOT 6 – FORBIDDEN READ



READ PROTECTED SLOT 6 – CONSUMPTION CURVES

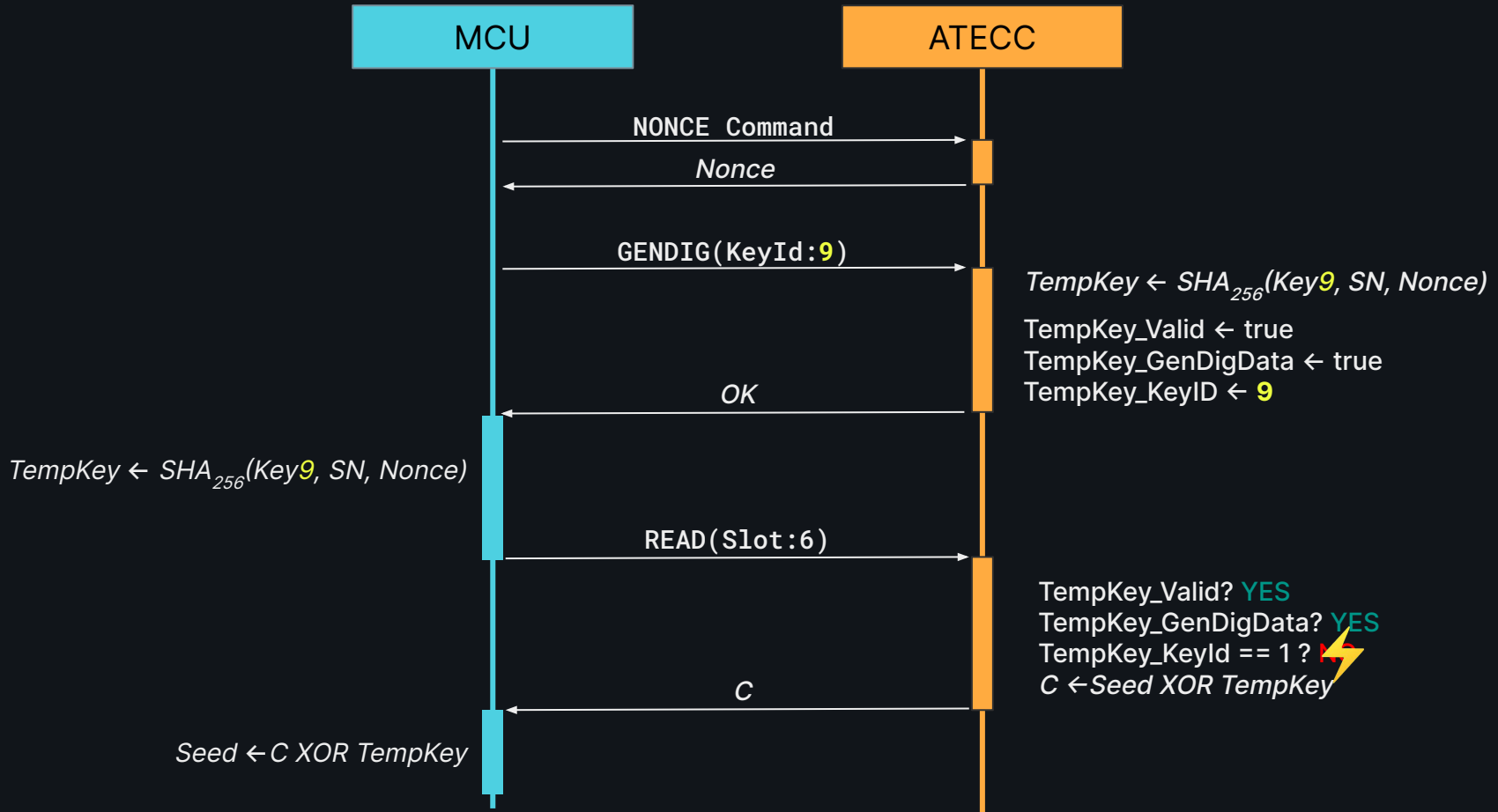


Forbidden Read
EXECUTION ERROR

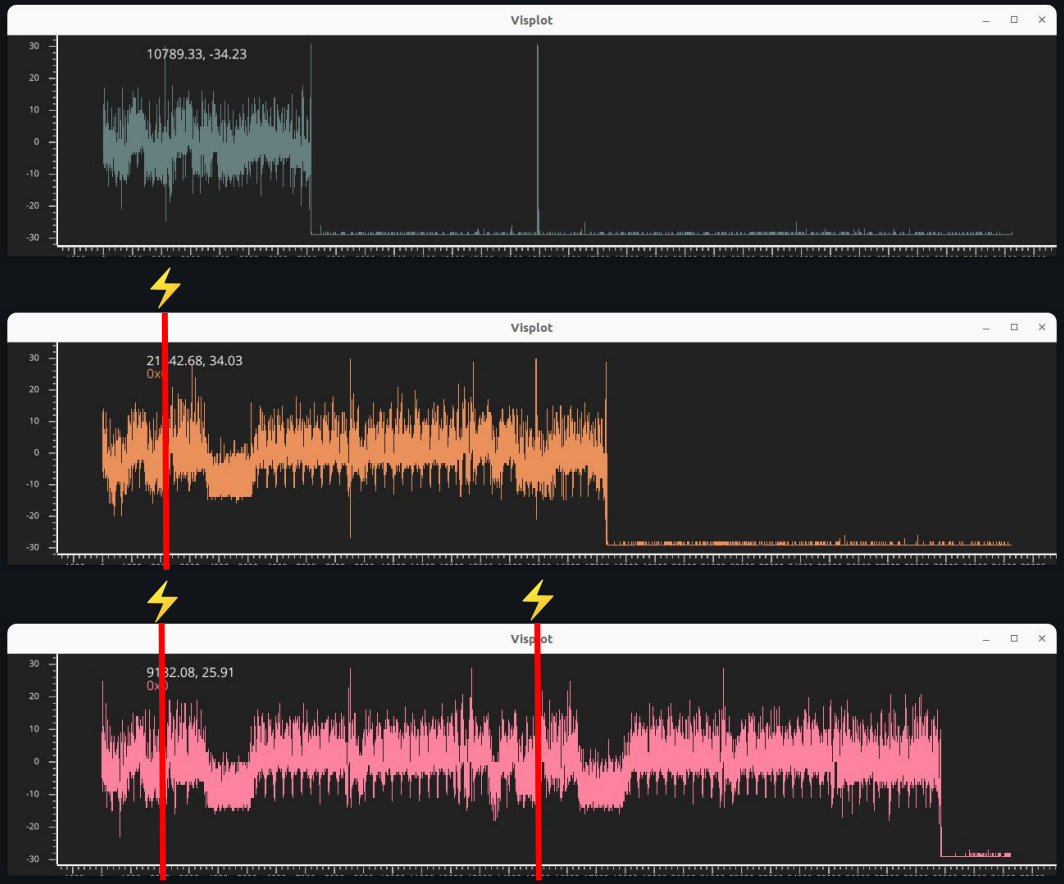
Granted Read
Seed XOR TempKey

hardwear.io
Hardware Security Conference and Training

READ PROTECTED SLOT 6 – FORBIDDEN READ – PERTURBED



READ ENCRYPTED SLOT 6 – CONSUMPTION CURVES



READ PROTECTED SLOT 6 – OBSERVED BEHAVIORS

No perturbation: Forbidden read response

Other Errors

No response

- Without data change
- With data corruption of Slot 6, Slot 1, Slot 9 or Configuration Zone

Successful execution of Read command

- Wrong content of Slot 6
- Correct content of Slot 6

🟢 Successful attack
🟠 Unsuccessful but recoverable
🔴 Unsuccessful and destructive

READ PROTECTED SLOT 6 – REVISIONS COMPARISON

	ATECC608A	ATECC608B
No Perturbation	●○○○○	●●●●○
Errors / No Response	●●○○○	●●●○○
Corruption of Slot	●○○○○	●●●●○
Successful attack	●●●○○	●○○○○

ATECC608B: success rate is reduced drastically
(time desync, high chances of data corruption)

3rd vuln: Keep using deprecated ATECC608A

CONCLUSION

A SERIE OF DESIGN ERRORS

1. Wrong usage of `Random`
   - **Misunderstanding** of Secure Device usage
2. No PIN verification actually needed
   - **Misconfiguration** of existing Secure Device feature
3. Usage of **deprecated** revision

But also:

4. Usage of STM32 OTP to store the `IO PROTECT KEY`
   - Security Design is **badly thought**
5. Fixed value of `IO PROTECT KEY`
   - **No Security Patch Plan** designed in case of compromission

TAKEAWAYS

All OneKey Mini before May 2020 are instantaneously hackable

Other OneKeys can be attacked within 1 day

No patch possible (OTP used and Configuration locked)

Needs Hardware upgrade to mitigate hardware attacks

NOTES ON DISCLOSURE

OneKey team has been contacted several times during second semester 2022

- Description of findings
- Recommendations

No proactive answer from them after 6 months

Design not modified since

Other products with same architecture

Using Secure Components is a great idea...

but only if you understand it properly

Thank you for your attention

Questions?