Tears for fears

Breaking an RFID counter

Pierre Granier
PhD-Student in Electromagnetic Cybersecurity

Jean-Joseph Marty
PhD in Cybersecurity

Rémy Delion
Research and development engineer

AMOSSYS

Presentation outline

Outline:

1. Introduction
2. Counter
3. Exploit
4. Results

AMOSSYS

Information Technology Security Evaluation Facility (ITSEF)

Software evaluation

Introduction

01.

Introduction - State of art[1]

Our goal was to reproduce state-of-the-art attack presented by Quarkslab in 2021 at SSTIC.

Quarkslab

- Targets NXP Cards
- Provides a methodology
- Provides tools

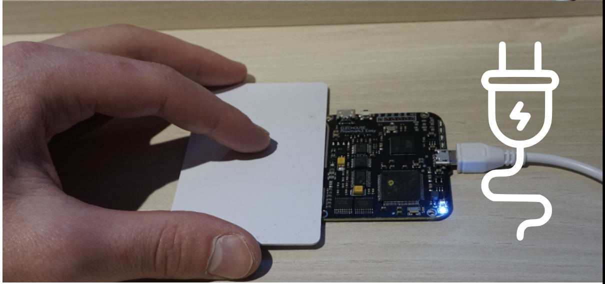
Our contribution

- Targets ST25TB* Cards
- Adapts their methodology

[1] C. Herrmann P. Teuwen. *EEPROM: It Will All End in Tears.*

Introduction - Tearing in a nutshell

Physical Removal
Cards designed to resist



Reader shutdown
What could go wrong ?

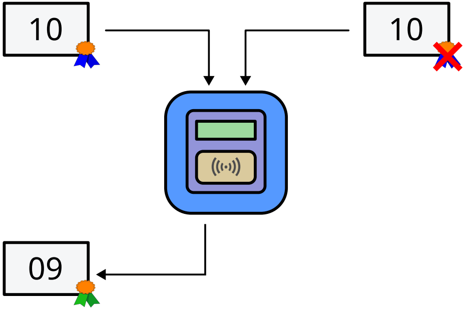
A Proxmark and Iceman's firmware is recommended

Introduction - ST25TB* Family

Address	ST25TB04K	ST25TB02K	ST25TB512-AC	ST25TB512-AT
[0:4]	Resettable OTP			Lockable EEPROM
[5:6]	Monotonic Counters			
[7:15]	Lockable EEPROM			
[16:23]	EEPROM	EEPROM		
[64:127]				
255	System OTP bits			
UID0	64 bits UID ROM D0 02 1F + serial	64 bits UID ROM D0 02 3F + serial	64 bits UID ROM D0 02 1B + serial	64 bits UID ROM D0 02 33 + serial
UID1				

Exploit tested on ST25TB512-AT, rest of the family confirmed vulnerable by ST.

Introduction - What is the impact ? [2]



Notes:

Cards are signed

Each machine can sign

Cards must be valid

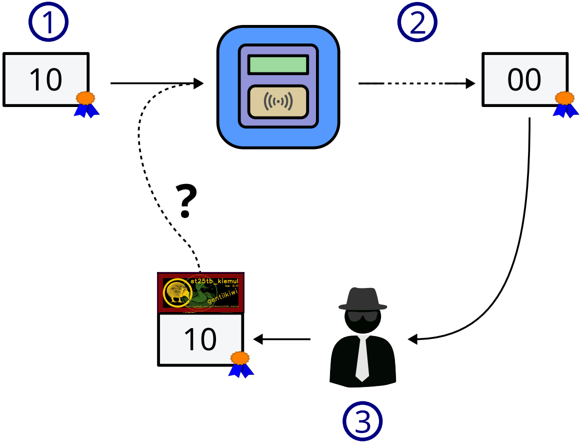
[2] Benjamin Delpy. *ST25TB series NFC tags for fun in French* public transports*, 2023.

Introduction - First attack

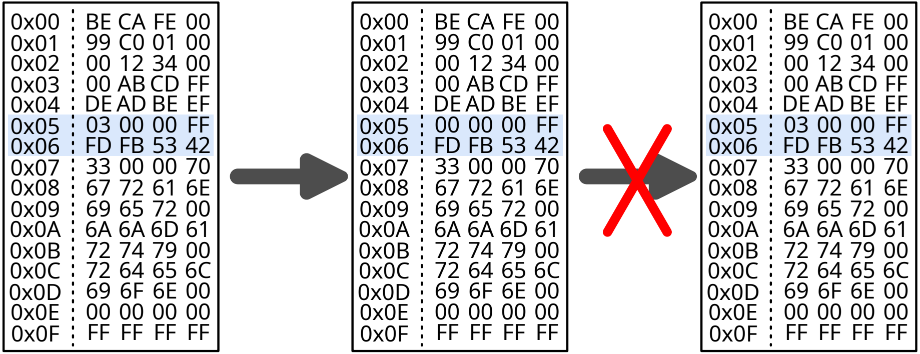
Benjamin Delpy:

1. Read the card
2. Use the card
3. Restore the original state

⟹ Using an emulator



Introduction - Not working



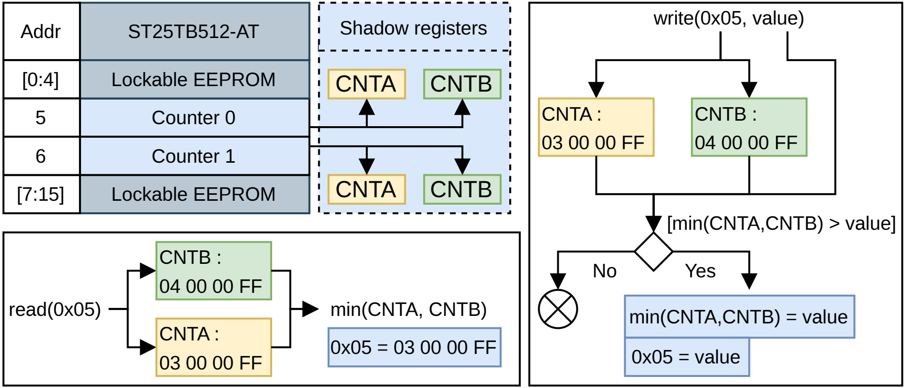
```
0x00  BE CA FE 00
0x01  99 C0 01 00
0x02  00 12 34 00
0x03  00 AB CD FF
0x04  DE AD BE EF
0x05  03 00 00 FF
0x06  FD FB 53 42
0x07  33 00 00 70
0x08  67 72 61 6E
0x09  69 65 72 00
0x0A  6A 6A 6D 61
0x0B  72 74 79 00
0x0C  72 64 65 6C
0x0D  69 6F 6E 00
0x0E  00 00 00 00
0x0F  FF FF FF FF
```

```
0x00  BE CA FE 00
0x01  99 C0 01 00
0x02  00 12 34 00
0x03  00 AB CD FF
0x04  DE AD BE EF
0x05  00 00 00 FF
0x06  FD FB 53 42
0x07  33 00 00 70
0x08  67 72 61 6E
0x09  69 65 72 00
0x0A  6A 6A 6D 61
0x0B  72 74 79 00
0x0C  72 64 65 6C
0x0D  69 6F 6E 00
0x0E  00 00 00 00
0x0F  FF FF FF FF
```

```
0x00  BE CA FE 00
0x01  99 C0 01 00
0x02  00 12 34 00
0x03  00 AB CD FF
0x04  DE AD BE EF
0x05  03 00 00 FF
0x06  FD FB 53 42
0x07  33 00 00 70
0x08  67 72 61 6E
0x09  69 65 72 00
0x0A  6A 6A 6D 61
0x0B  72 74 79 00
0x0C  72 64 65 6C
0x0D  69 6F 6E 00
0x0E  00 00 00 00
0x0F  FF FF FF FF
```

Impossible to reuse tickets because of the monotonic counters.

Introduction - How monotonic counters work ?



Addr	ST25TB512-AT
[0:4]	Lockable EEPROM
5	Counter 0
6	Counter 1
[7:15]	Lockable EEPROM

Logic based on observed counter behavior and patent : FR3103925B1

Introduction - EEPROM Tearing

Counter relies on EEPROM memories, EEPROM write are not atomic.

Write are block-wide (32 bits) and a two-step operation:

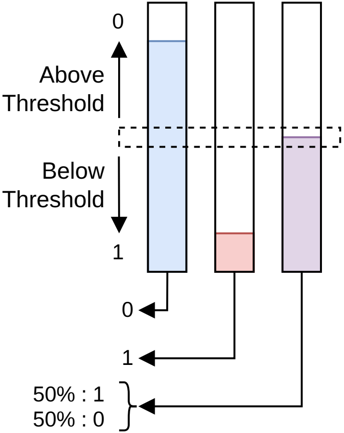
1. Erase everything (erased bits are logic 1)
2. Program bits (programed bits are logic 0)

Introduction - EEPROM Tearing

EEPROM cells are analogic.

A cell has a logic value of:

- **[0]** → Close to be full/programed
- **[1]** → Close to be emptied/erased
- **[?]** → **Weak bit**, probabilistic interpretation

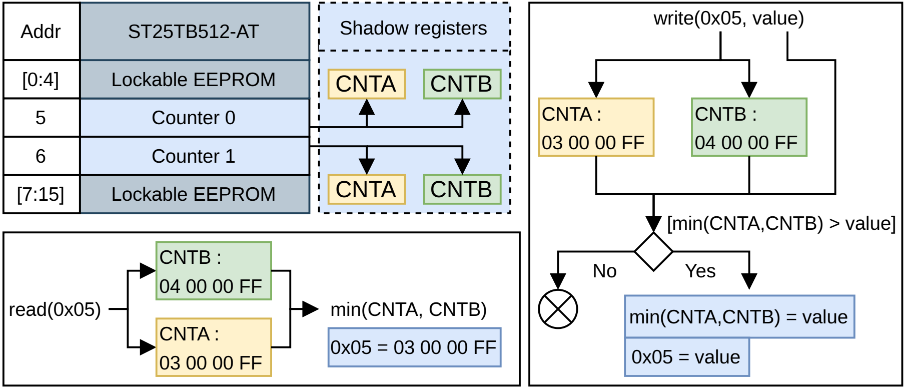


Counter

02.

Counter - Behavior

Counters use the following logic:



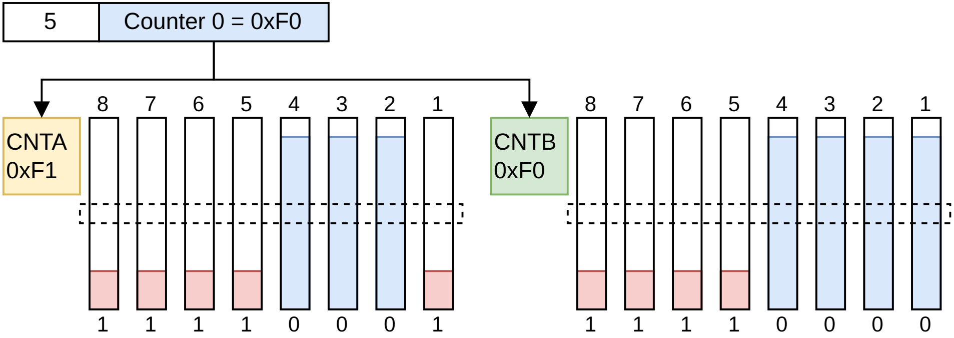
Addr	ST25TB512-AT
[0:4]	Lockable EEPROM
5	Counter 0
6	Counter 1
[7:15]	Lockable EEPROM

Based on observed counter behavior and patent : FR3103925B1

Counter - Architecture

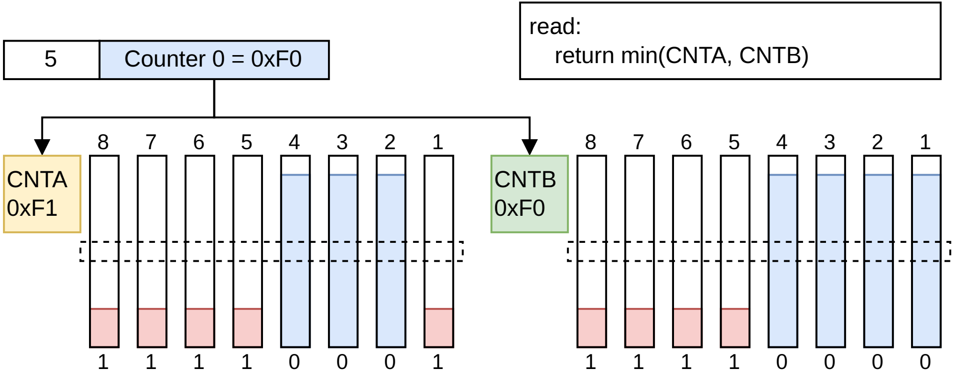
Counter architecture:

View truncated to first 8 bits, a bloc is 32 bits wide on ST25TB*.



Counter - Read example

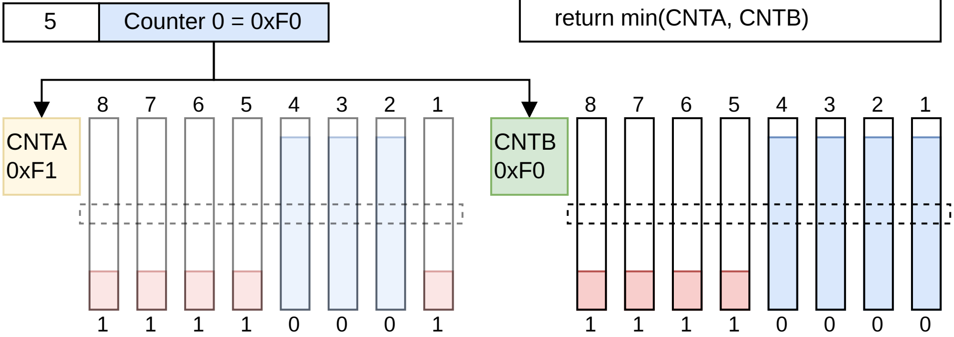
Example of a read on counter 5:



```
read:
    return min(CNTA, CNTB)
```

Counter - Read example

Example of a read on counter 5:

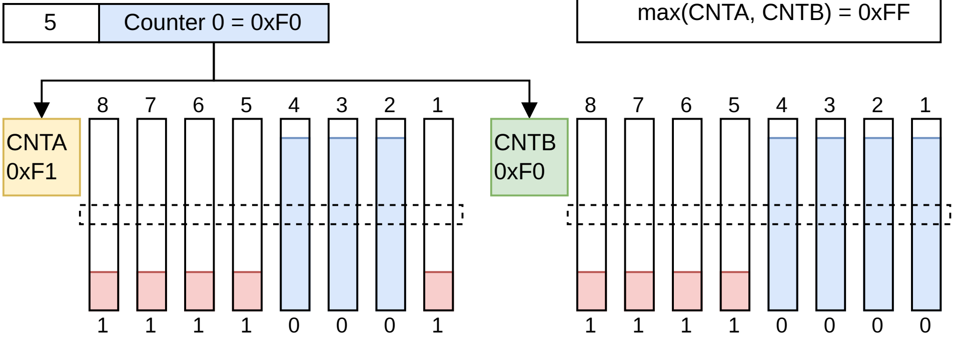


```
read:
    return min(CNTA, CNTB)
```

Counter - Failed write example

Example of a write(0xFF) on counter 5:

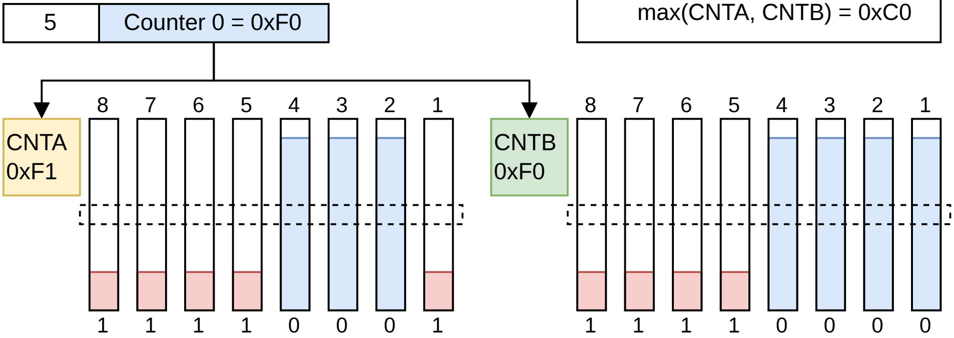
```
write (0xFF):
    if (0xFF < min(CNTA, CNTB)):
        max(CNTA, CNTB) = 0xFF
```



Counter - Write example

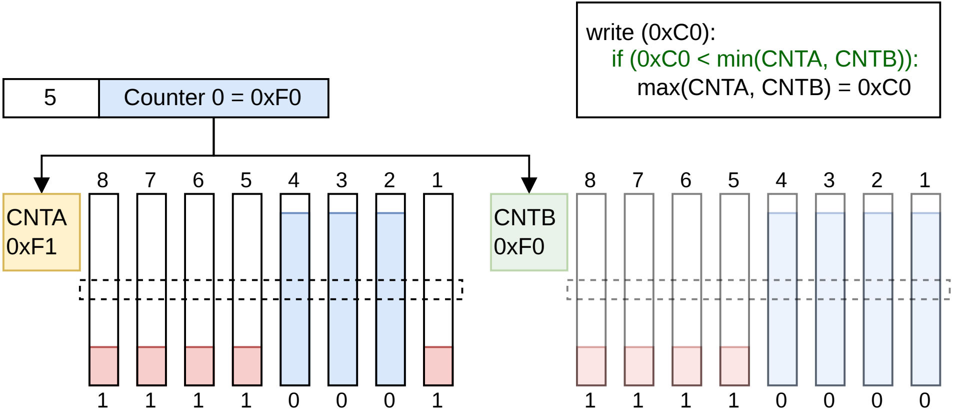
Example of a write(0xC0) on counter 5:

```
write (0xC0):
    if (0xC0 < min(CNTA, CNTB)):
        max(CNTA, CNTB) = 0xC0
```



Counter - Write example

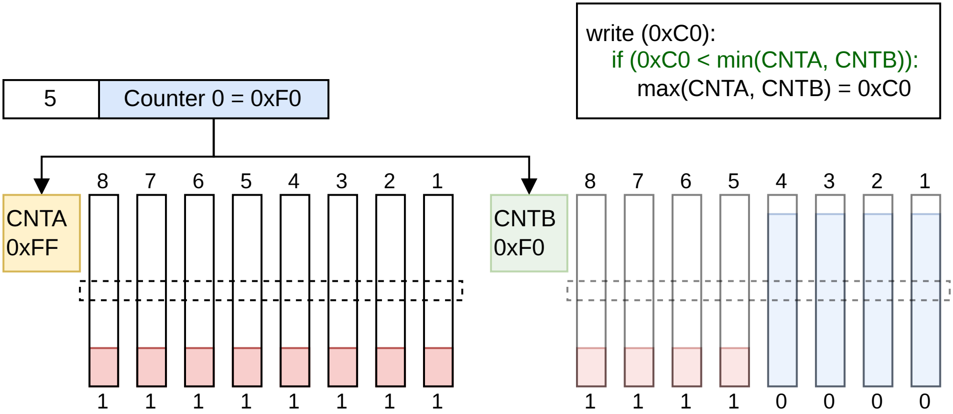
Example of a write(0xC0) on counter 5:



```
write (0xC0):
    if (0xC0 < min(CNTA, CNTB)):
        max(CNTA, CNTB) = 0xC0
```

Counter - Write example

Example of a write(0xC0) on counter 5:

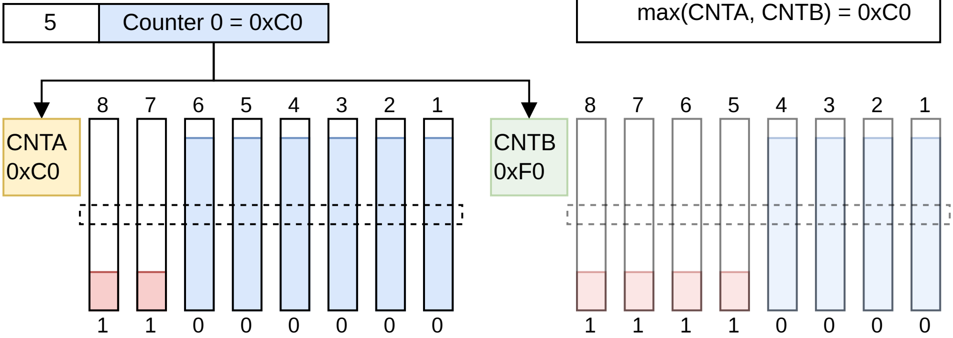


```
write (0xC0):
    if (0xC0 < min(CNTA, CNTB)):
        max(CNTA, CNTB) = 0xC0
```

Counter - Write example

Example of a write(0xC0) on counter 5:

```
write (0xC0):
    if (0xC0 < min(CNTA, CNTB)):
        max(CNTA, CNTB) = 0xC0
```



Counter - Write example

At no point in a single interrupted write the counter value is compromised.

We need a trick to write 0xFF in **both** shadow registers.

Exploit’s details

03.

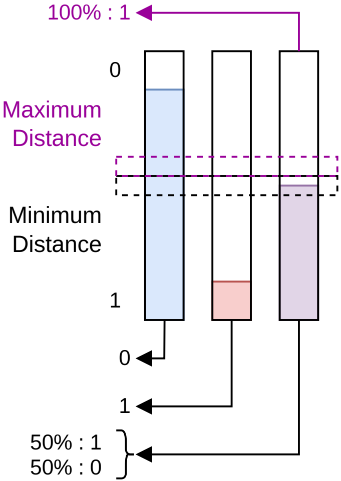
Exploit - Bit leveraging

Weak bit:

EEPROM cells close to the evaluation threshold have probabilistic interpretations.

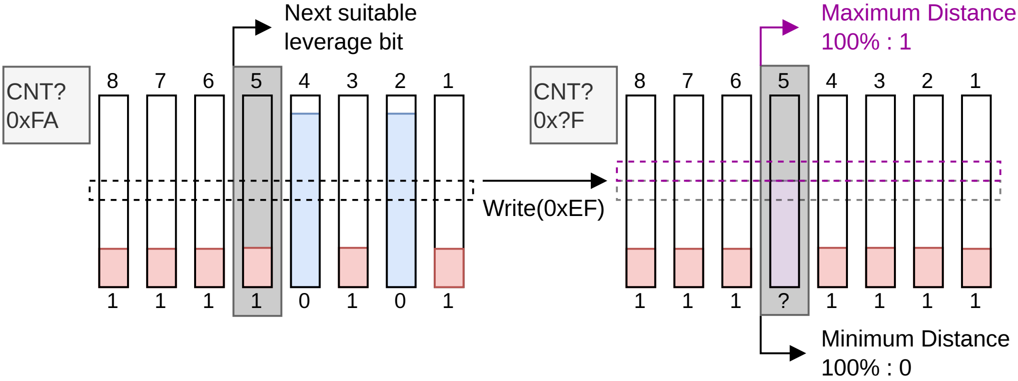
Distance dependency:

When a card is far from its reader, weak bits are more likely to be interpreted at 1.



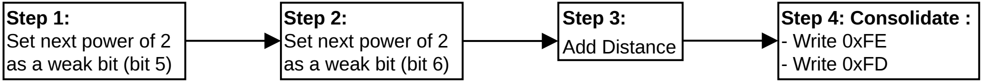
Exploit - Bit leveraging

Set next power of 2 at 0, and the rest to 1, shape a weak bit.



This kind of pattern is what we need to reset our counter.

Exploit - Exploit layout



Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

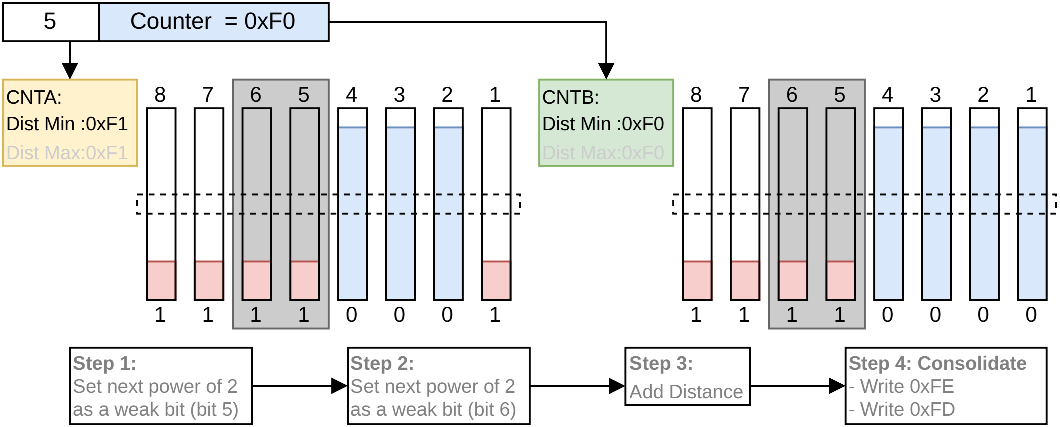
Step 3:
Add Distance

Step 4: Consolidate :
- Write 0xFE
- Write 0xFD

We use the previous gadget to control both sub-counter (Steps 1-2).

After adding distance we can write any arbitrary value and remove any weak bit (Steps 3-4).

Exploit - inital state



5 | Counter = 0xF0

CNTA:
Dist Min :0xF1
Dist Max:0xF1

8	7	6	5	4	3	2	1
1	1	1	1	0	0	0	1

CNTB:
Dist Min :0xF0
Dist Max:0xF0

8	7	6	5	4	3	2	1
1	1	1	1	0	0	0	0

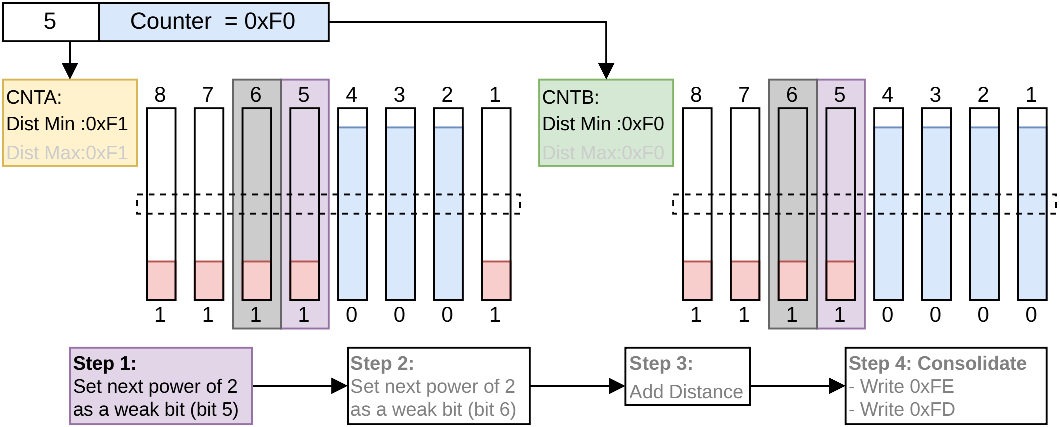
Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate
- Write 0xFE
- Write 0xFD

Exploit - first leverage



Counter = 0xF0

CNTA:
Dist Min :0xF1
Dist Max:0xF1

CNTB:
Dist Min :0xF0
Dist Max:0xF0

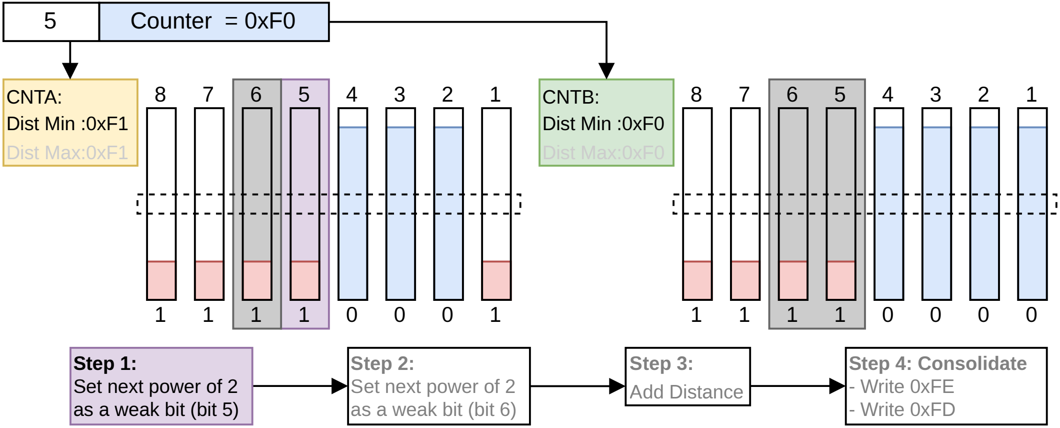
Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate
- Write 0xFE
- Write 0xFD

Exploit - first leverage



CNTA:
Dist Min :0xF1
Dist Max:0xF1

CNTB:
Dist Min :0xF0
Dist Max:0xF0

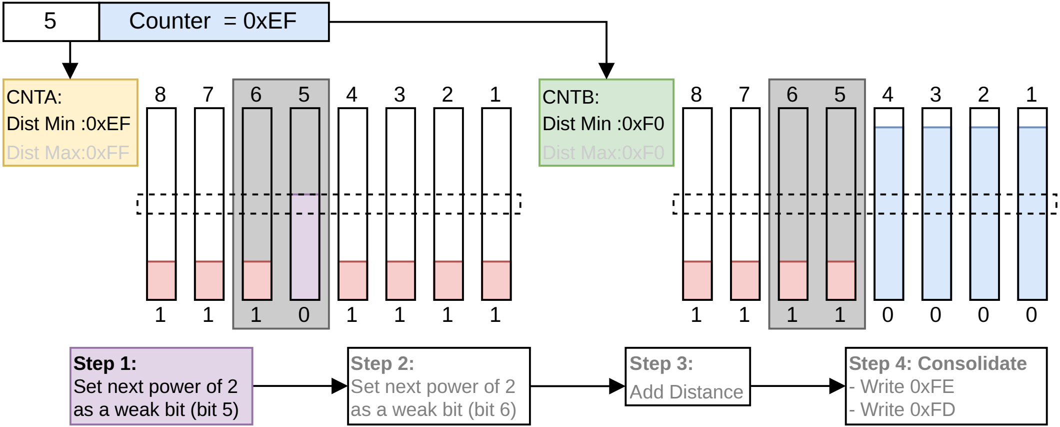
Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate
- Write 0xFE
- Write 0xFD

Exploit - first leverage



CNTA:
Dist Min :0xEF
Dist Max:0xFF

CNTB:
Dist Min :0xF0
Dist Max:0xF0

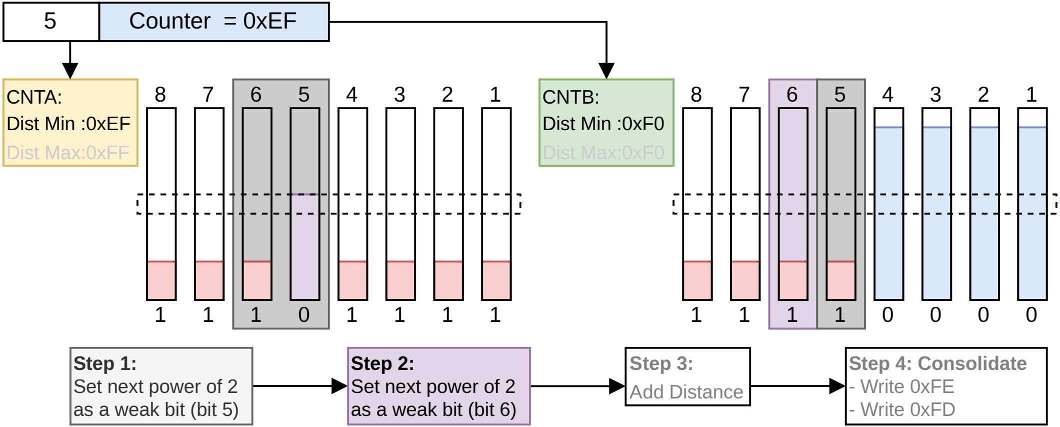
Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate
- Write 0xFE
- Write 0xFD

Exploit - second leverage



CNTA:
Dist Min :0xEF
Dist Max:0xFF

CNTB:
Dist Min :0xF0
Dist Max:0xF0

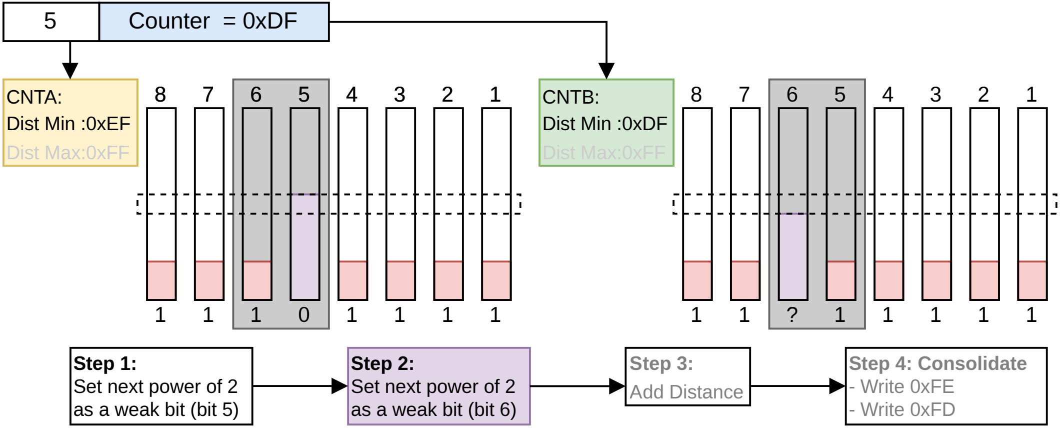
Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate
- Write 0xFE
- Write 0xFD

Exploit - second leverage



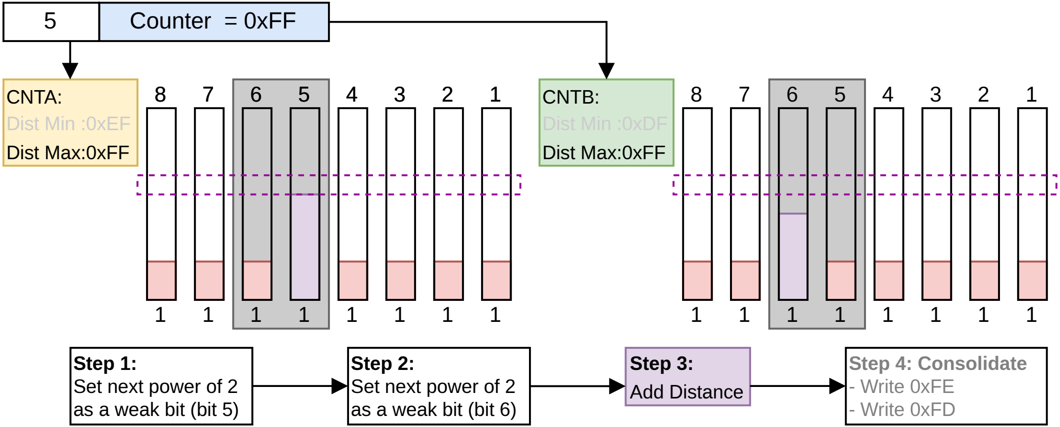
Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate
- Write 0xFE
- Write 0xFD

Exploit - consolidation



5 | Counter = 0xFF

CNTA:
Dist Min :0xEF
Dist Max:0xFF

CNTB:
Dist Min :0xDF
Dist Max:0xFF

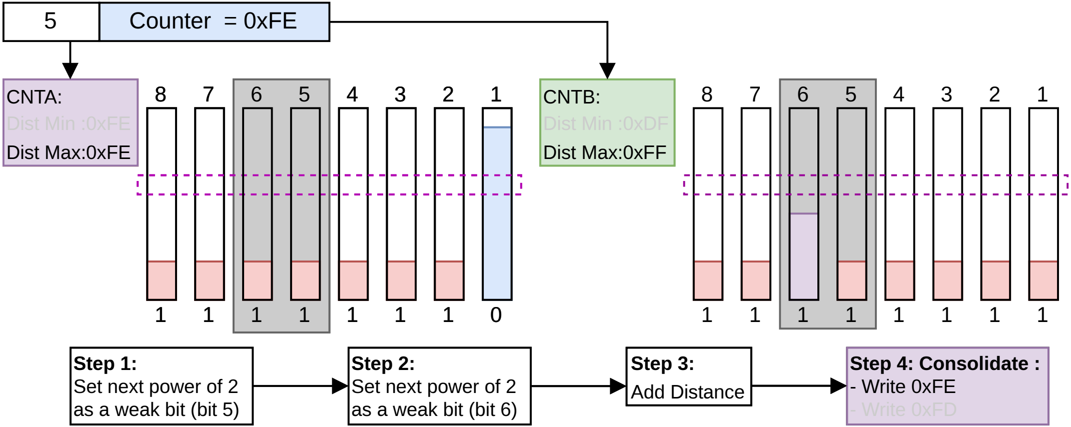
Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate
- Write 0xFE
- Write 0xFD

Exploit - consolidation



5

Counter = 0xFE

CNTA:
Dist Min :0xFE
Dist Max:0xFE

CNTB:
Dist Min :0xDF
Dist Max:0xFF

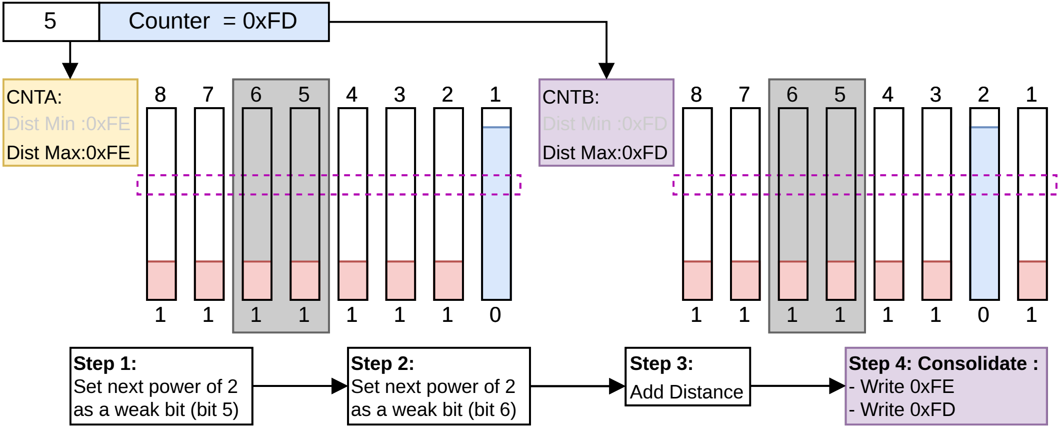
Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate :
- Write 0xFE
- Write 0xFD

Exploit - consolidation



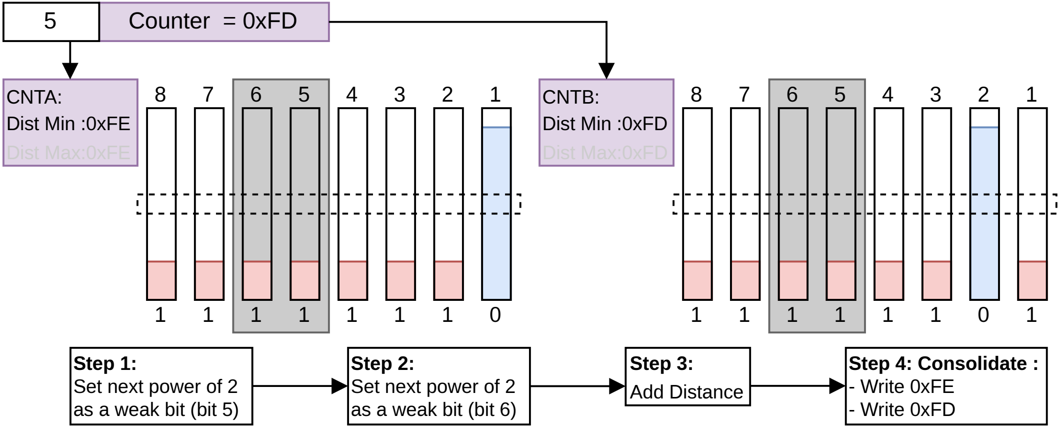
Step 1:
Set next power of 2
as a weak bit (bit 5)

Step 2:
Set next power of 2
as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate :
- Write 0xFE
- Write 0xFD

Exploit - consolidation



Step 1:
Set next power of 2 as a weak bit (bit 5)

Step 2:
Set next power of 2 as a weak bit (bit 6)

Step 3:
Add Distance

Step 4: Consolidate :
- Write 0xFE
- Write 0xFD

Results

04.

Results

Some pitfalls and odd things.

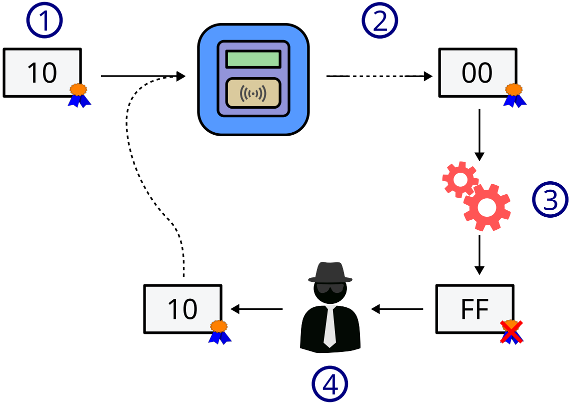
- Timings a lot shorter for counter EEPROM vs standard blocs.
- How quickly operations are done influence weak bits interpretations.

Exploit sources at : https://gitlab.com/SiliconOtter/tears4fears

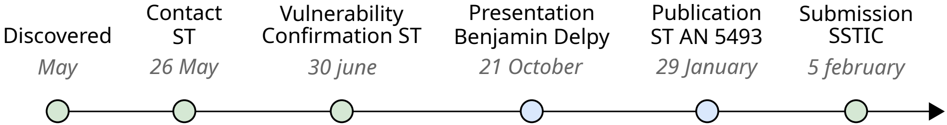
Second attack

New version:

1. **Read the card**
2. **Use the card**
3. **Increment the counters**
4. **Restore the original state**



Conclusion



Discovered
May

Contact ST
26 May

Vulnerability Confirmation ST
30 june

Presentation Benjamin Delpy
21 October

Publication ST AN 5493
29 January

Submission SSTIC
5 february

Don’t forget:

- Testing on production **is illegal**
- Might be detected
- Be responsible

Thanks for listening