Gotta glitch 'em all!

10+ Years of War Stories Glitching Embedded Devices

riscure



AGENDA

Introduction

FI in the wild

Practical FI

Protections

Where is the problem?

```
const char *PIN = "19379148"

// Compare the PIN with the correct one
// Return true if different

bool comparePIN(char *pin) {

    for (int i=0; i<strlen(pin); i++) {
        if (pin[i]!=PIN[i])
            return true;
    }
    return false;
}
```

```
// Authenticate PIN.
// Return true if incorrect

bool pinAuthentication() {

    char pin[80];
    char tmp[80];
    bool result=false;


    printf("Please, introduce PIN\r\n");
    gets(pin);

    result = comparePIN(pin);

    if (result == false) {
        sprintf(tmp,"PIN correct\r\n");
    } else {
        sprintf(tmp,"PIN %s is
                        incorrect\r\
n",pin);
    }

    printf(tmp);
    return result;
}
```

Where is the problem?

```
const char *PIN = "19379148"

// Compare the PIN with the correct one
// Return true if different

bool comparePIN(char *pin) {

    for (int i=0; i<strlen(pin); i++) {
        if (pin[i]!=PIN[i])
            return true;
    }
    return false;
}
```

```
// Authenticate PIN.
// Return true if incorrect

bool pinAuthentication() {

    char pin[80];
    char tmp[80];
    bool result=false;


    printf("Please, introduce PIN\r\n");
    gets(pin);

    result = comparePIN(pin);

    if (result == false) {
        sprintf(tmp,"PIN correct\r\n");
    } else {
        sprintf(tmp,"PIN %s is
                        incorrect\r\
n",pin);
    }

    printf(tmp);
    return result;
}
```

Where is the problem?

```
const char *PIN = "19379148"

// Compare the PIN with the correct one
// Return true if different

bool comparePIN(char *pin) {

    for (int i=0; i<strlen(PIN); i++) {
        if (pin[i]!=PIN[i])
            return true;
    }
    return false;
}
```

IS IT NOW SECURE?

```
// Authenticate PIN.
// Return true if incorrect

bool pinAuthentication() {

    char pin[80];
    char tmp[80];
    bool result=false;


    printf("Please, introduce PIN\r\n");
    fgets(pin,sizeof(pin),stdin);

    result = comparePIN(pin);

    if (result == false) {
        sprintf(tmp,"PIN correct\r\n");
    } else {
        sprintf(tmp,"PIN %s is
                        incorrect\r\
n",pin);
    }

    printf("%s",tmp);
    return result;
}
```

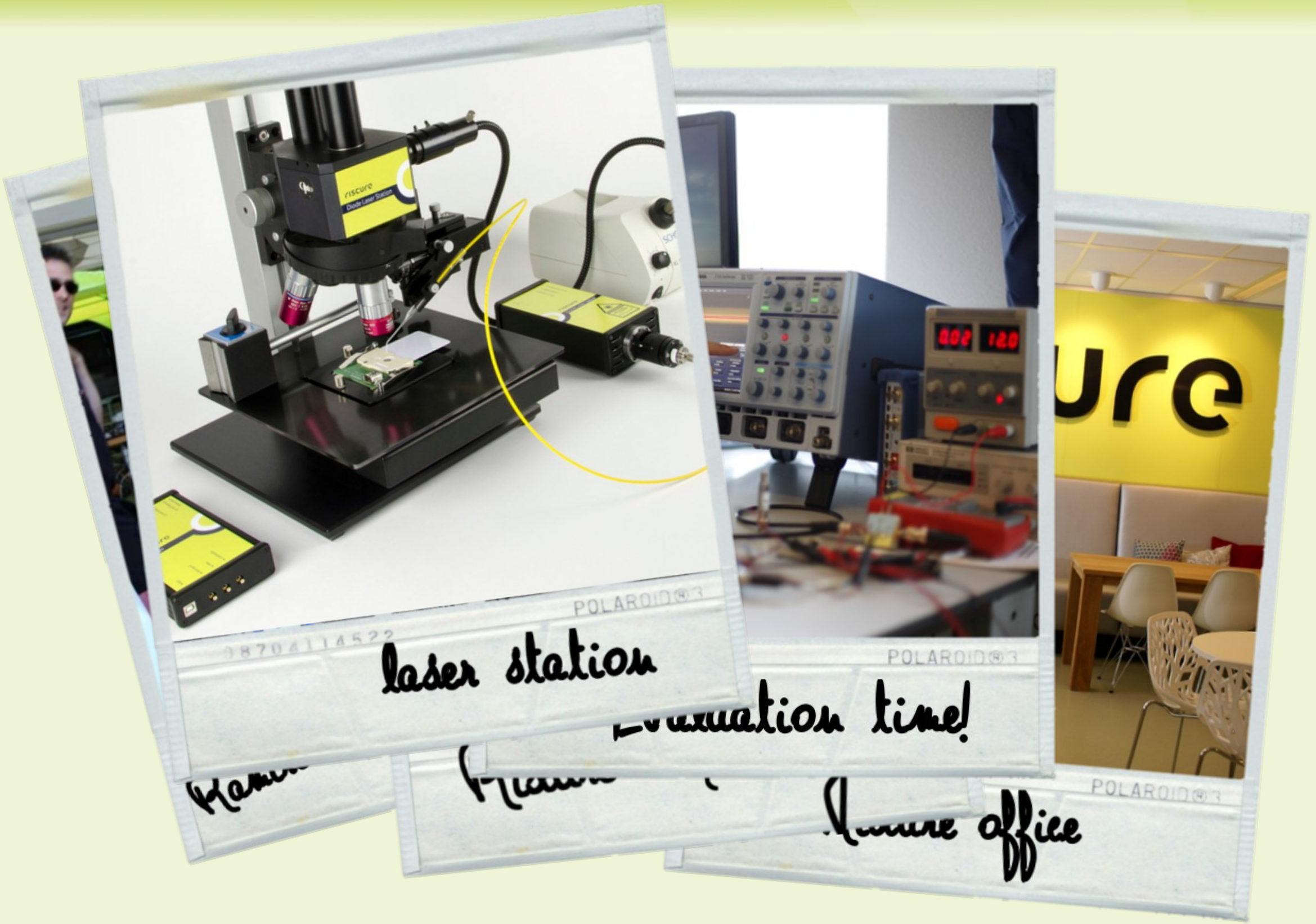
Where is the problem?

```
const char *PIN = "19379148"

// Compare the PIN with the correct one
// Return true if different

bool comparePIN(char *pin) {

    for (int i=0; i<strlen(PIN); i++) {
        if (pin[i]!=PIN[i])
            return true;
    }
    return false;
}
```

IS IT NOW SECURE?

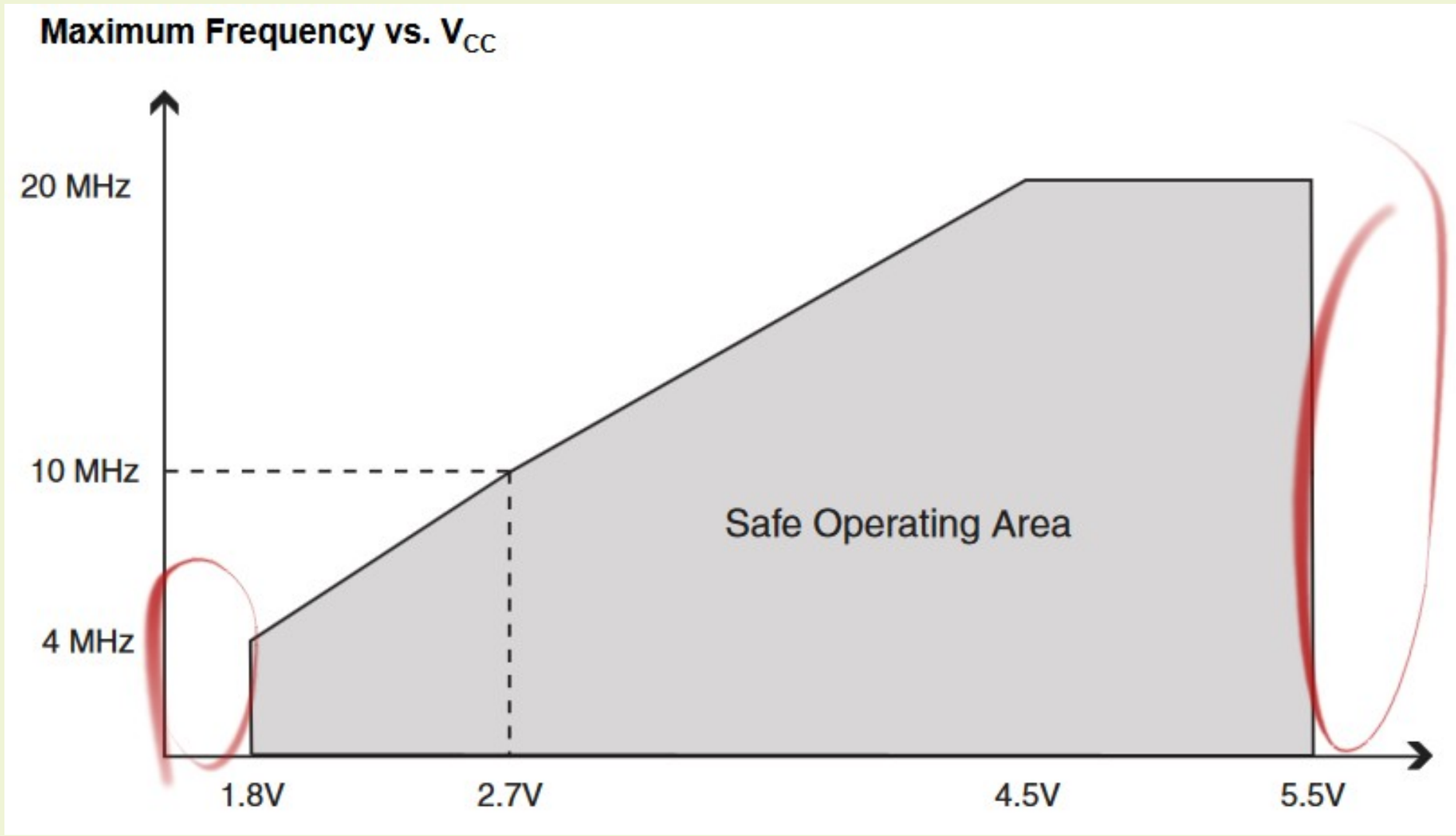
```
// Authenticate PIN.
// Return true if incorrect

bool pinAuthentication() {

    char pin[80];
    char tmp[80];
    bool result=false;


    printf("Please, introduce PIN\r\n");
    fgets(pin,sizeof(pin),stdin);

    result = comparePIN(pin);

    if (result == false) {
        sprintf(tmp,"PIN correct\r\n");
    } else {
        sprintf(tmp,"PIN %s is
                        incorrect\r\
n",pin);
    }

    printf("%s",tmp);
    return result;
}
```

Who am I?

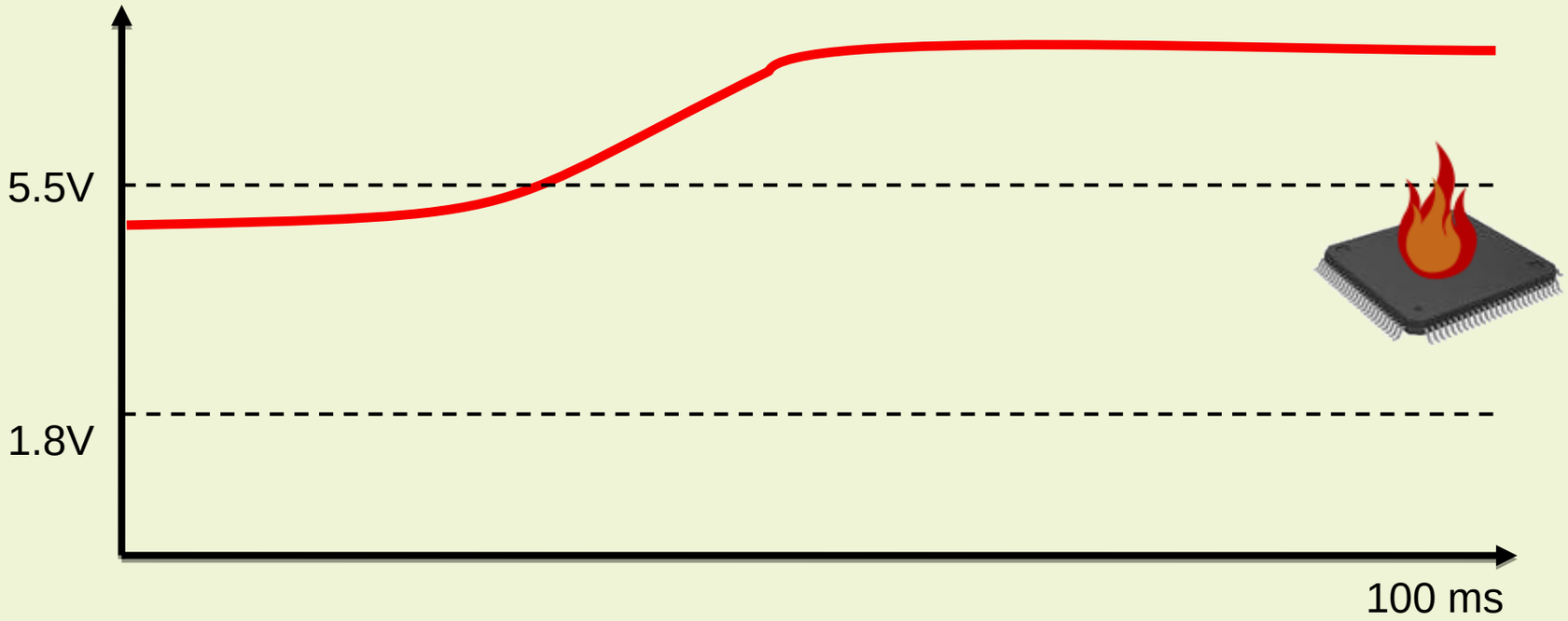

riscure
Diode Laser Station
POLARIDO
laser station
Evaluation time!
office

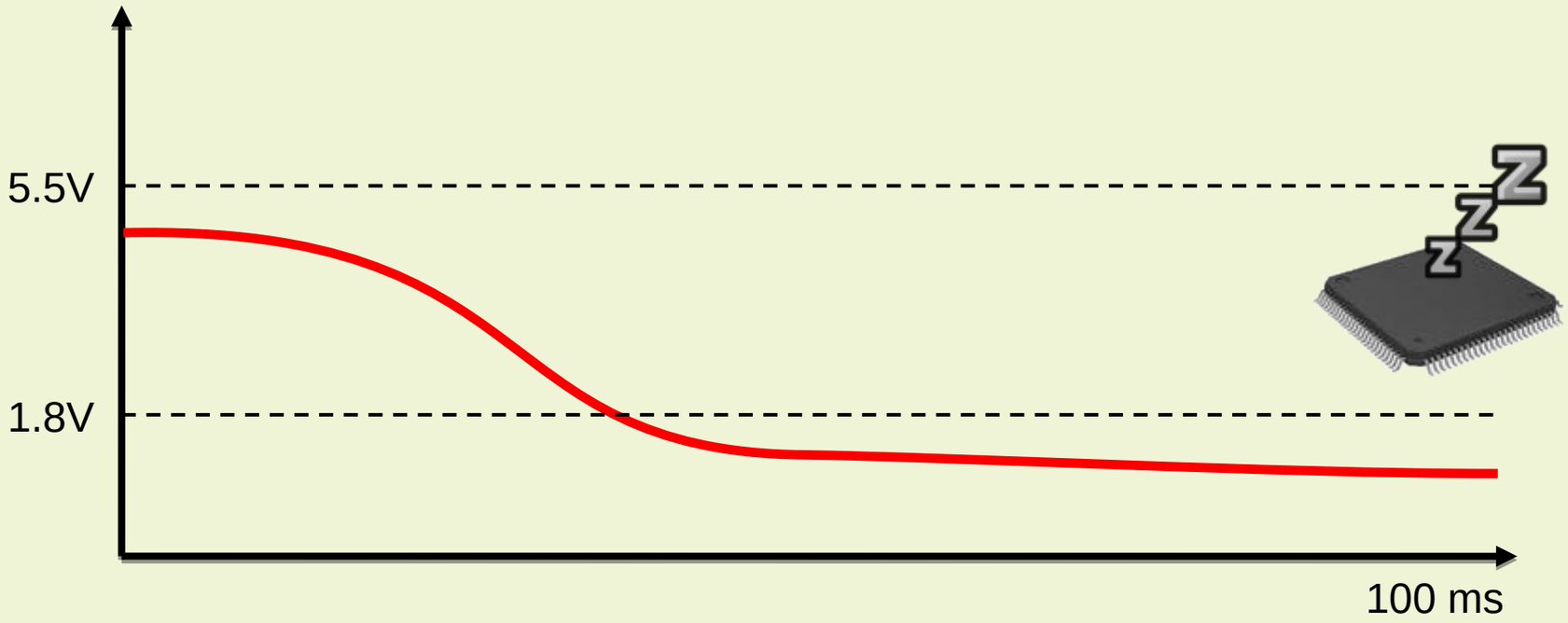
Demo

Fault Injection attacks

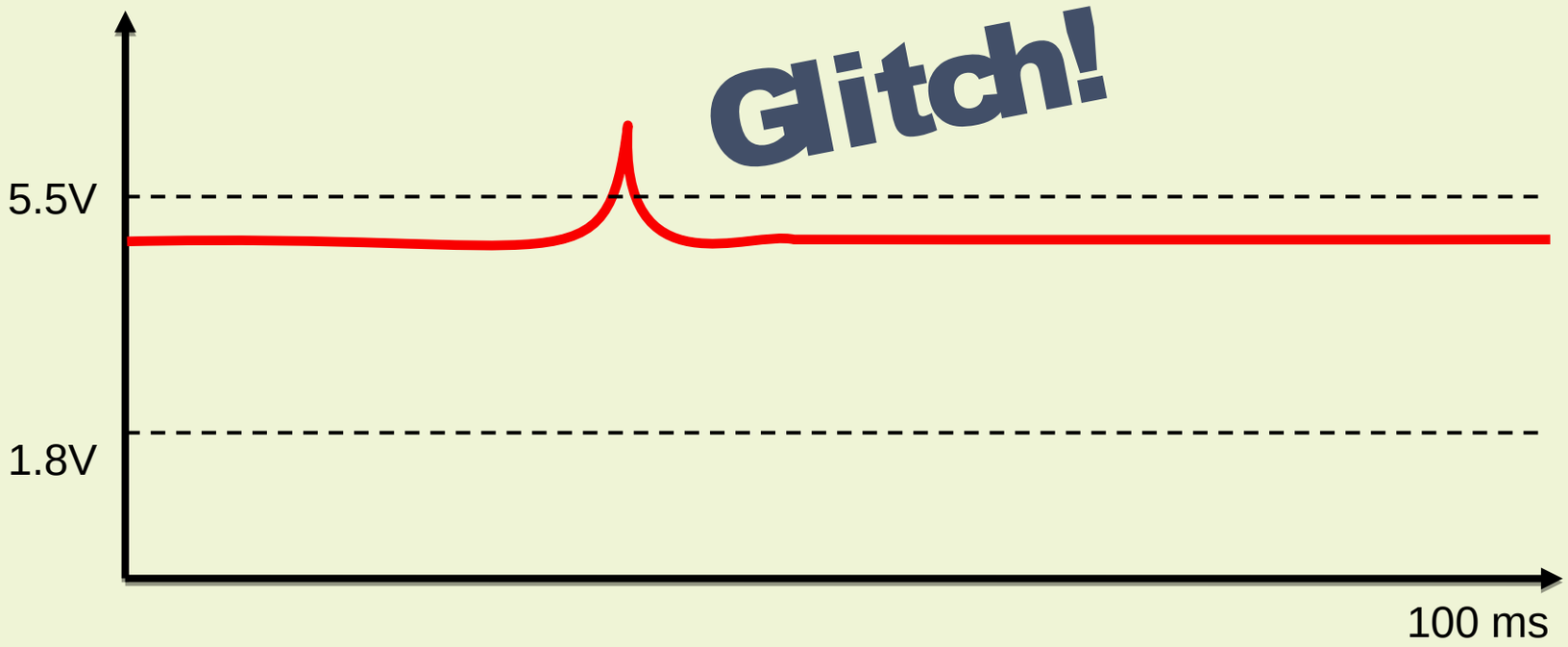


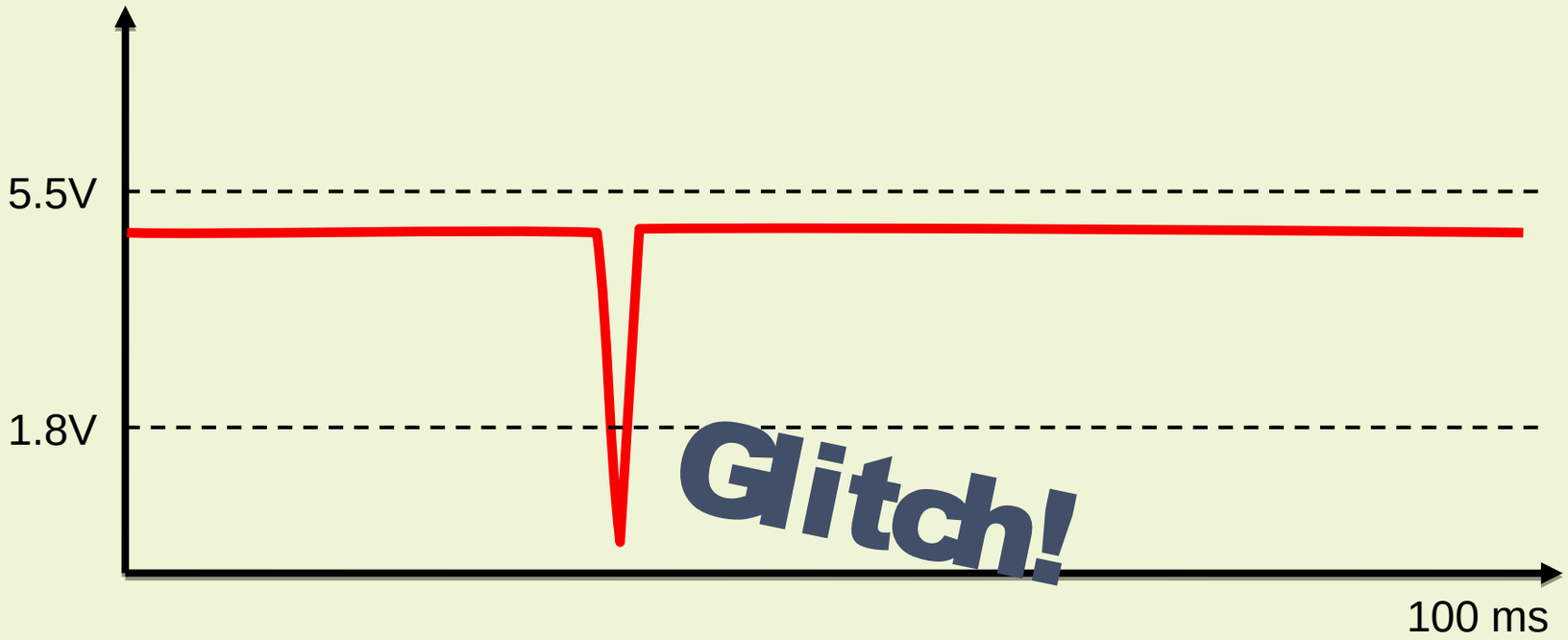
Voltage glitching



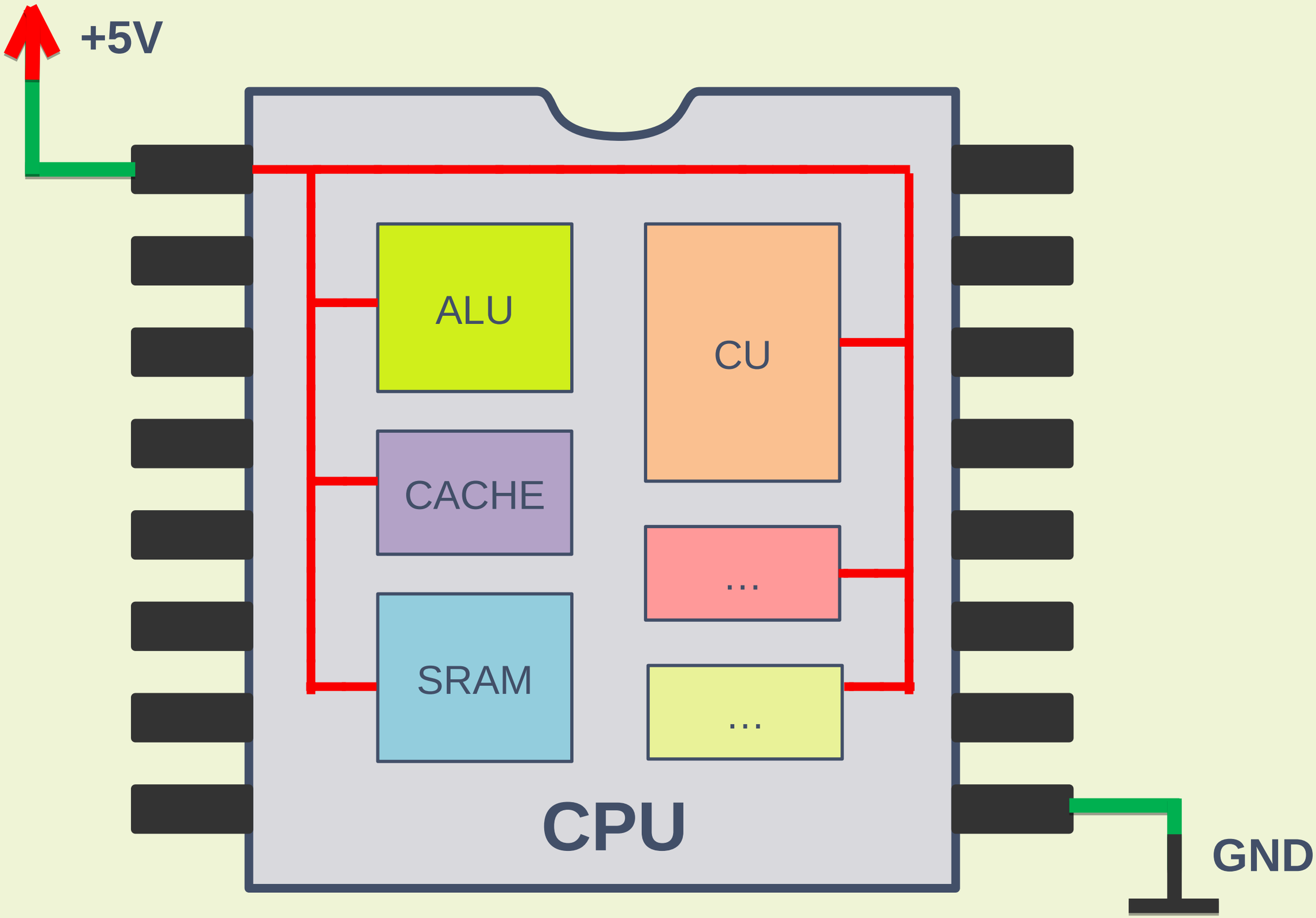


Voltage glitching

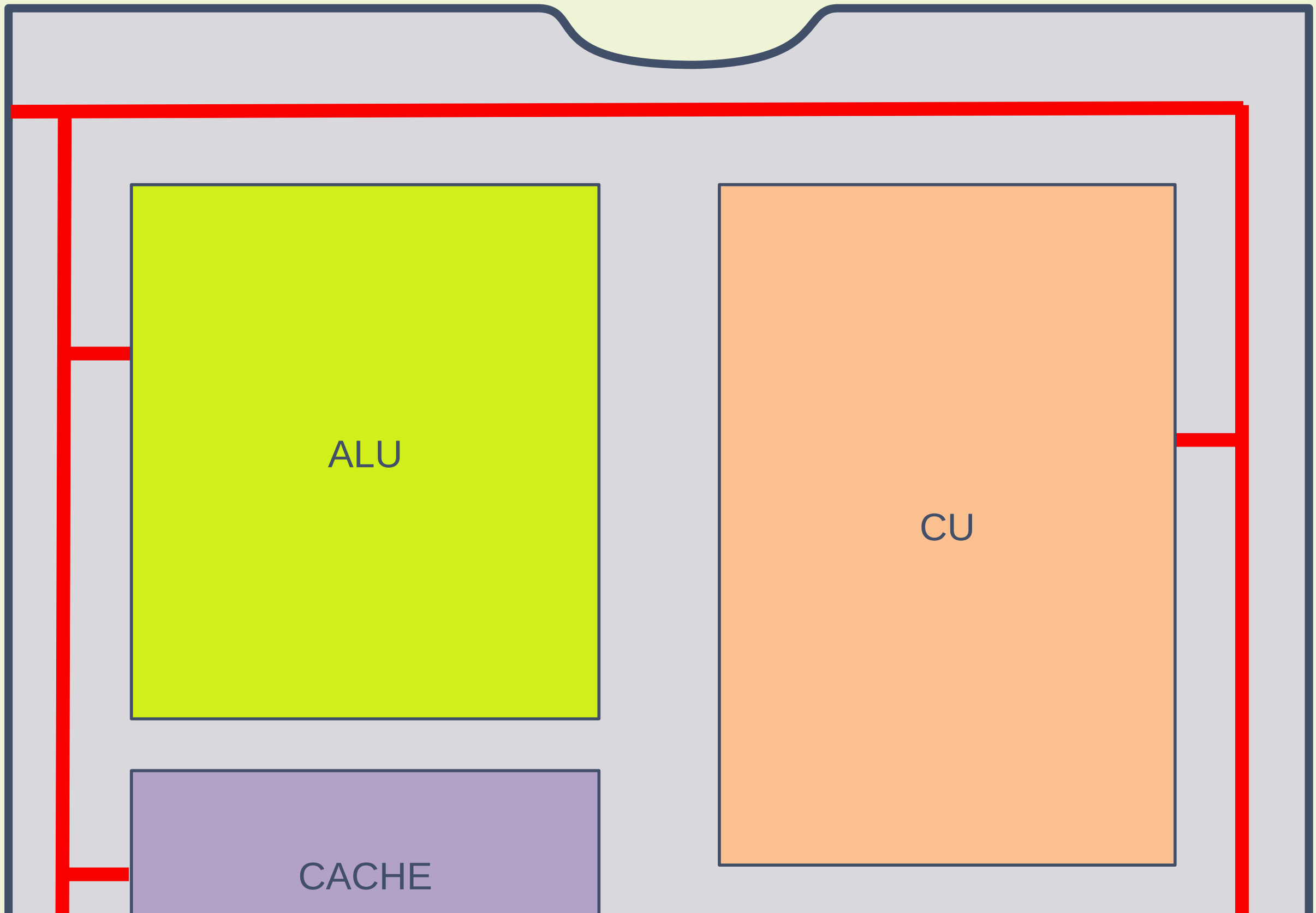




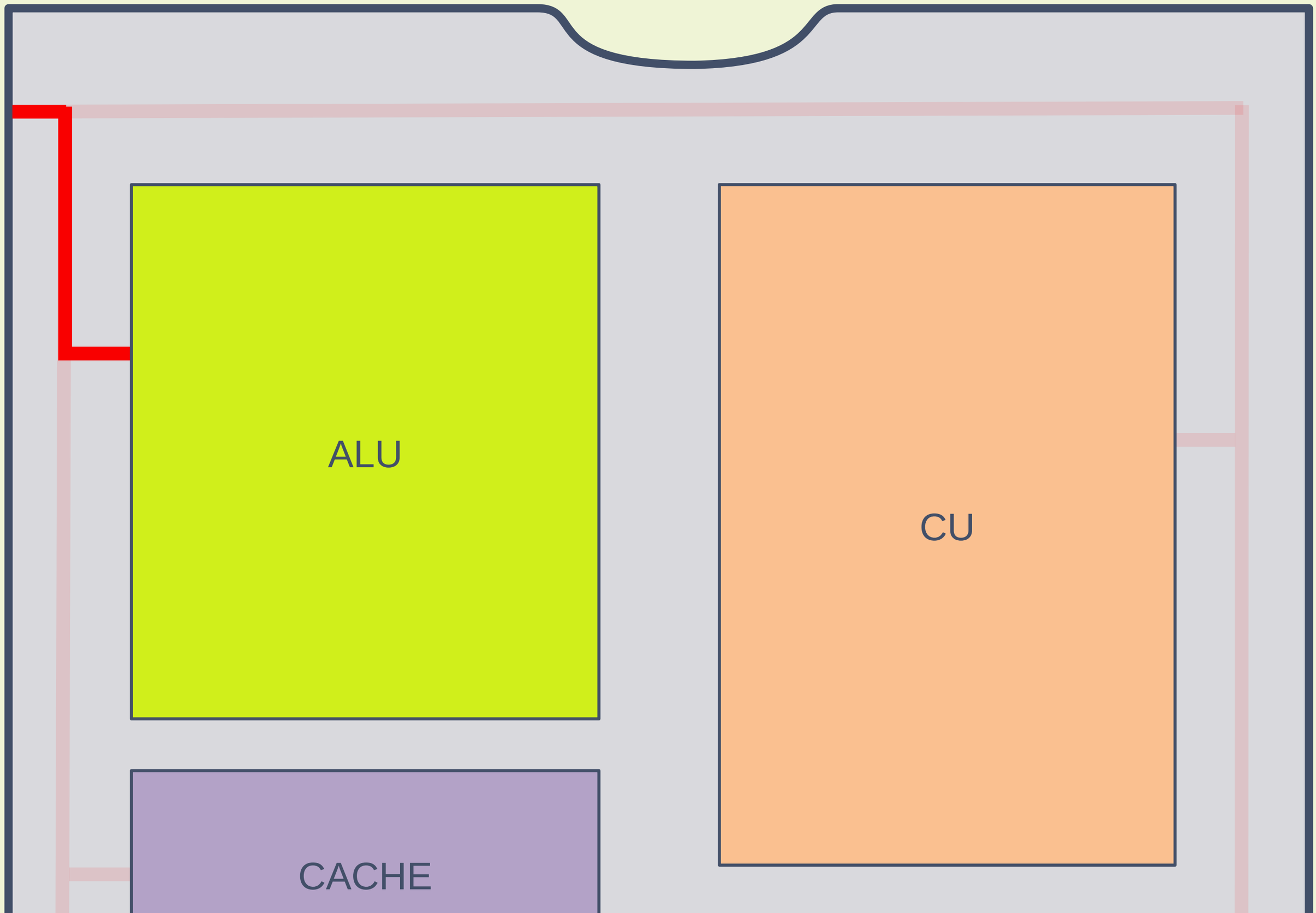
How FI works?



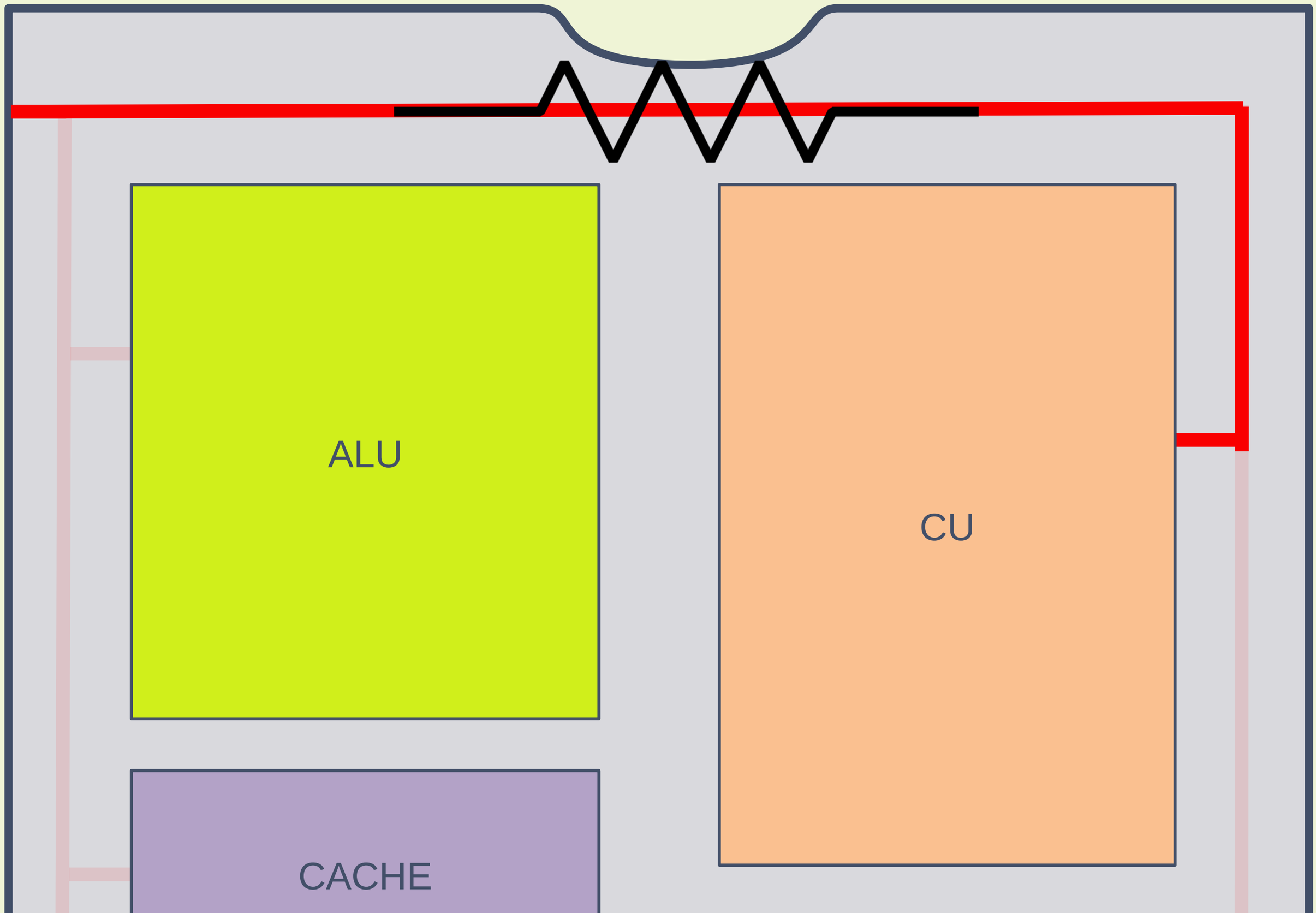
How FI works?



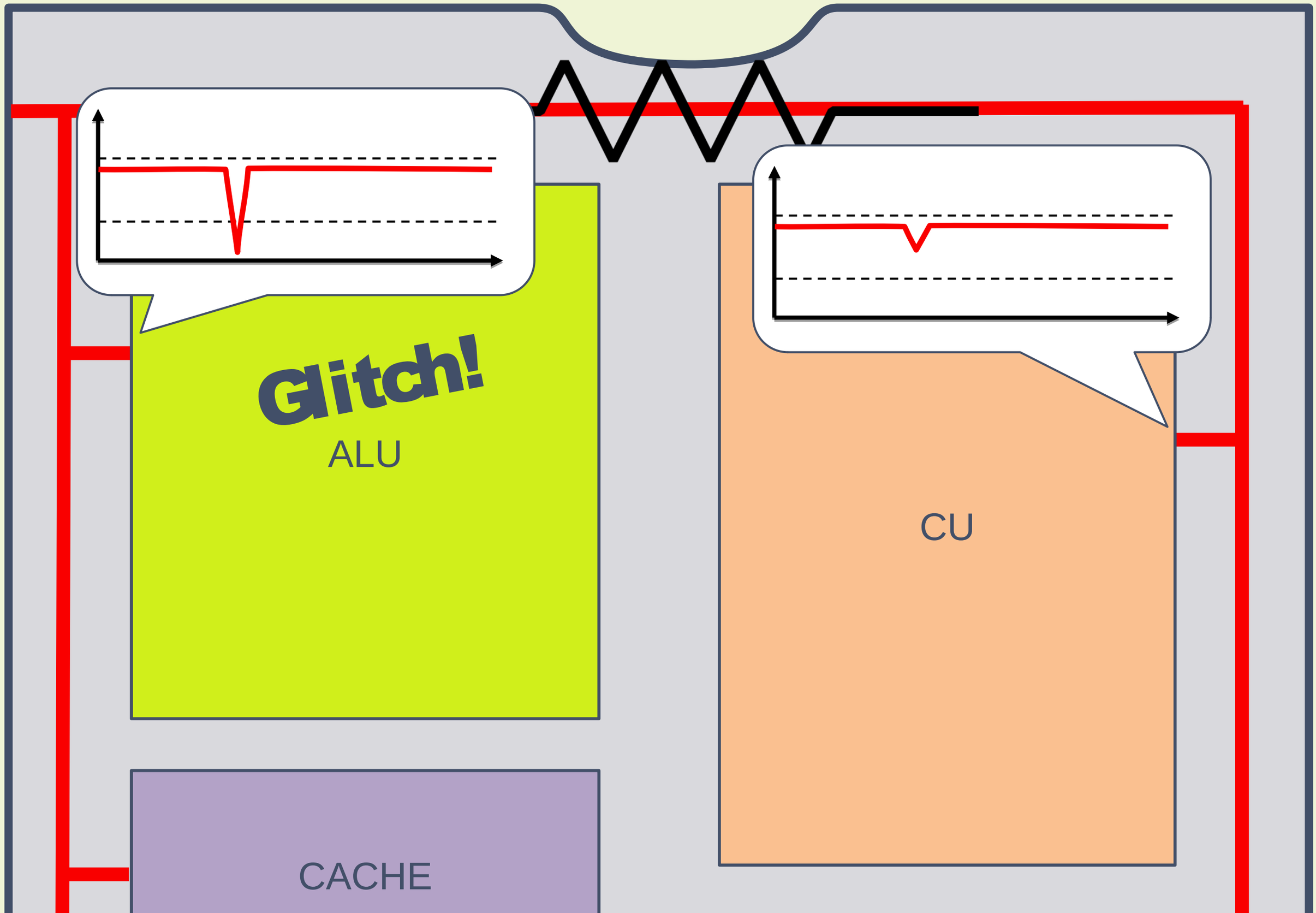
How FI works?



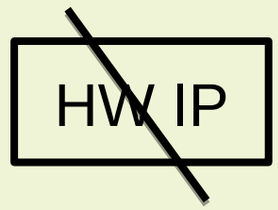
How FI works?



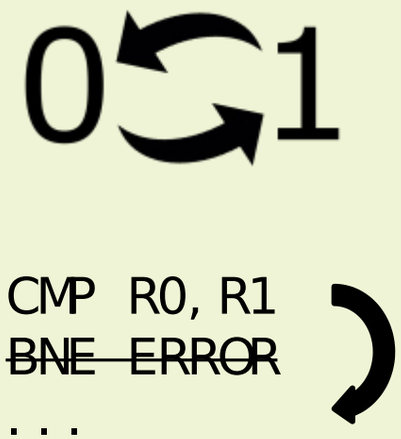
How FI works?



Effects of glitches

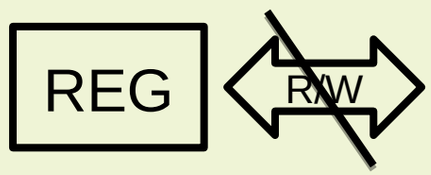


Disabling modules



Flipping bits

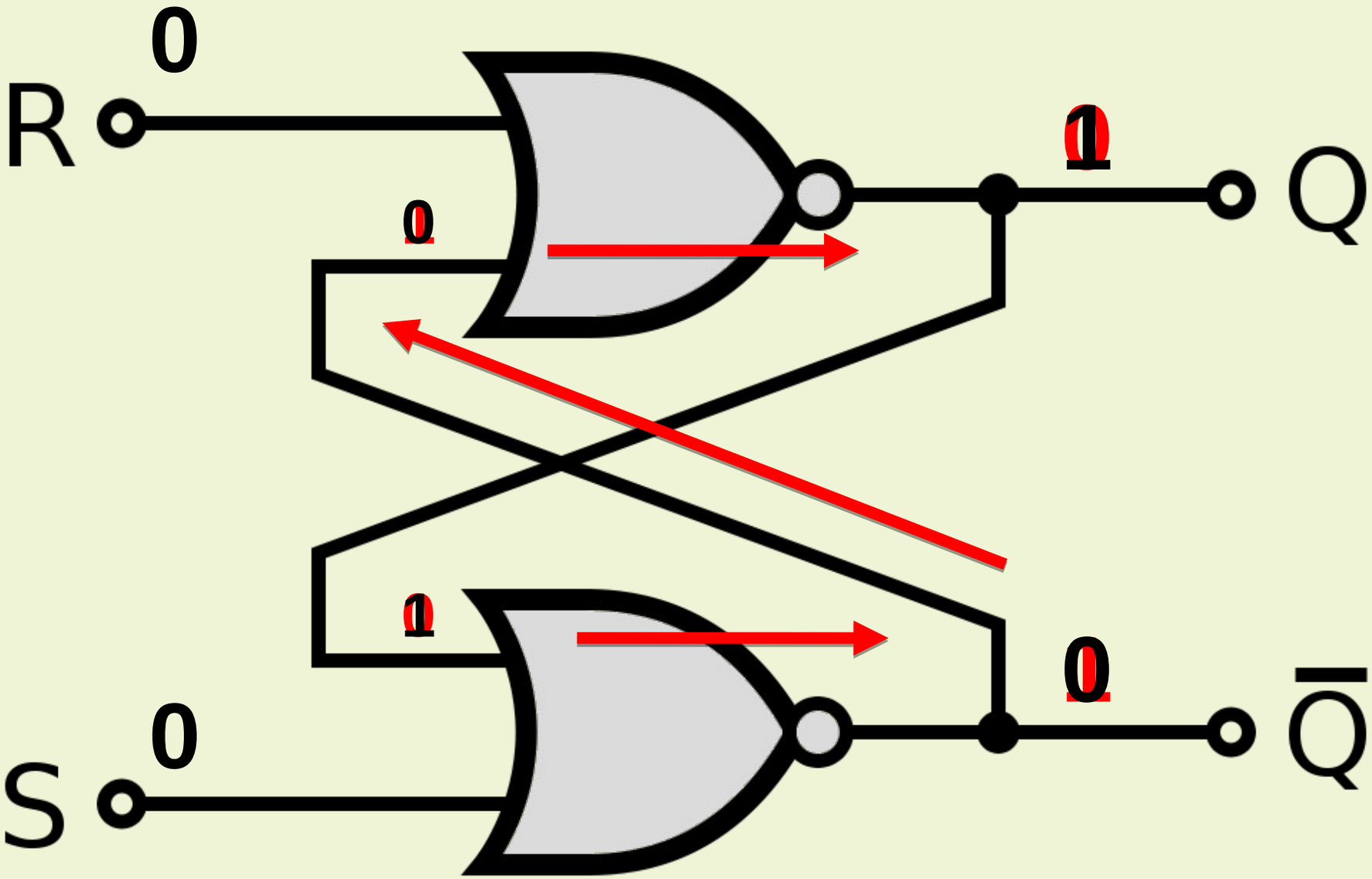
Skipping / corrupting instructions



Preventing R/W

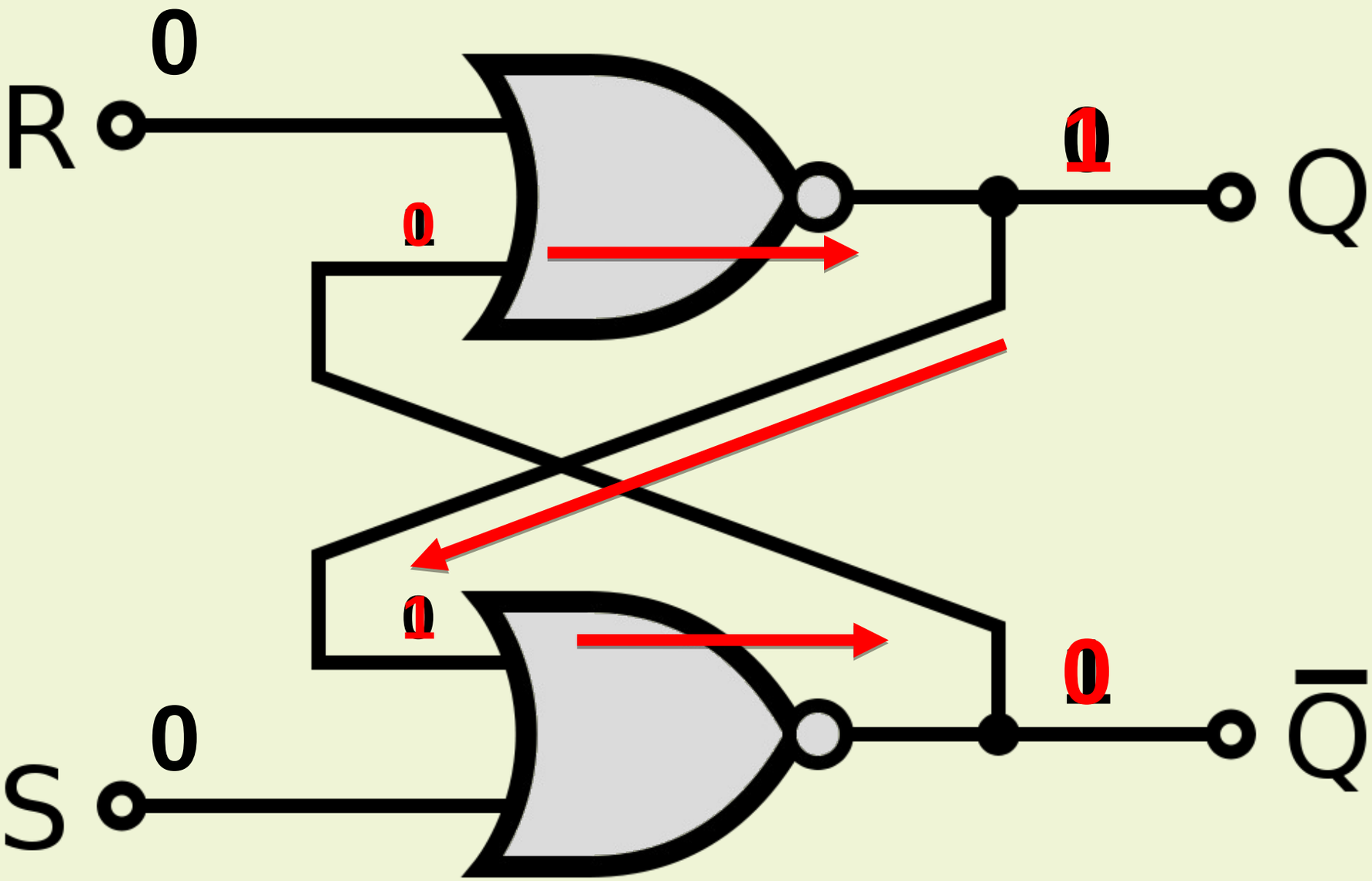
Flipping bits

$$1 \rightarrow 0$$



Flipping bits

$0 \rightarrow 1$



Skipping/corrupting instructions

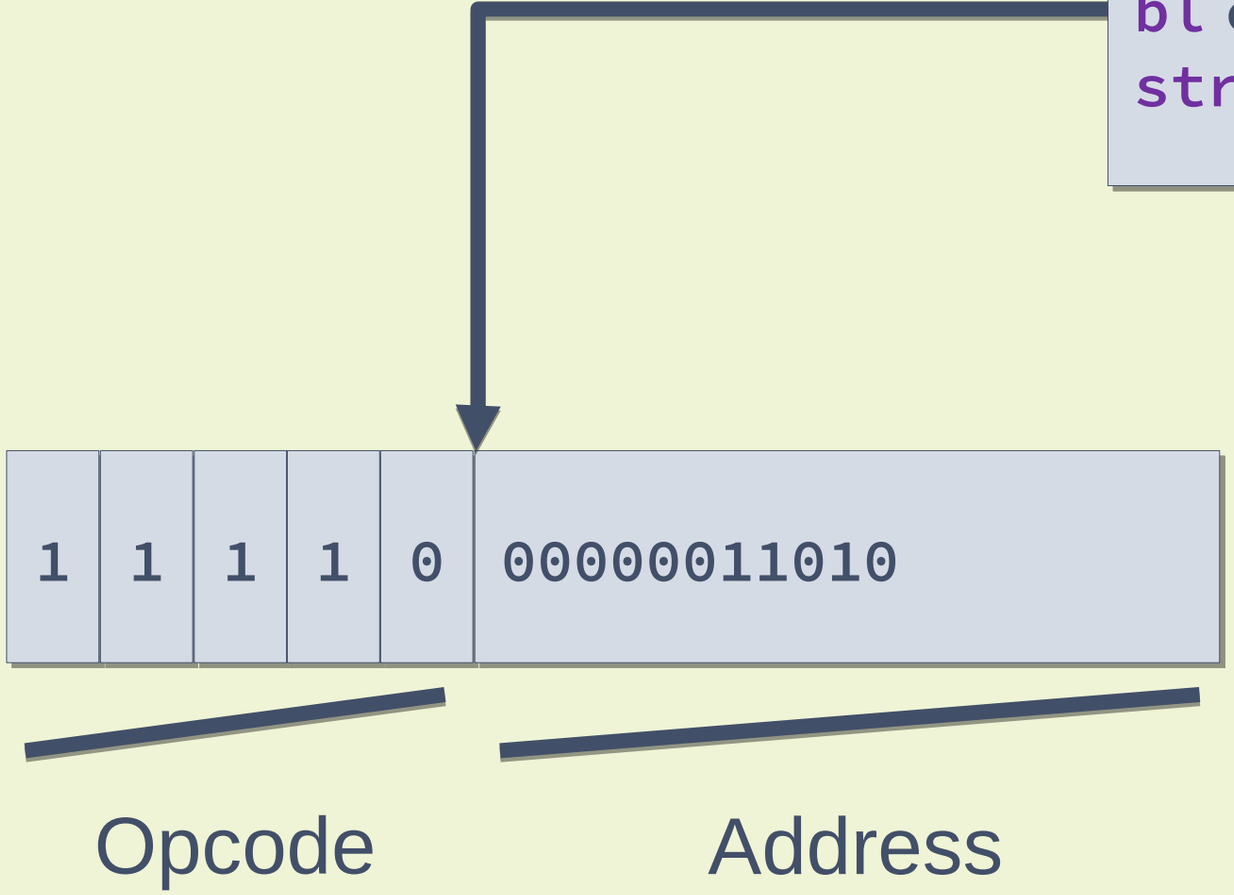
```
const char *PIN = "19379148"

// Compare the PIN with the correct one
// Return true if different

bool comparePIN(char *pin) {

    for (int i=0; i<strlen(PIN); i++) {
        if (pin[i]!=PIN[i])
            return true;
    }
    return false;
}
```

```
// Authenticate PIN.
// Return true if incorrect

bool pinAuthentication() {

    char pin[80];
    char tmp[80];
    bool result=false;


    printf("Please, introduce PIN\r\n");
    fgets(pin,sizeof(pin),stdin);

    result = comparePIN(pin);

    if (result == false) {
        sprintf(tmp,"PIN correct\r\n");
    } else {
        sprintf(tmp,"PIN %s is
                        incorrect\r\
n",pin);
    }

    printf("%s",tmp);
    return result;
}
```

Skipping/corrupting instructions

```
result = comparePIN(pin);
```

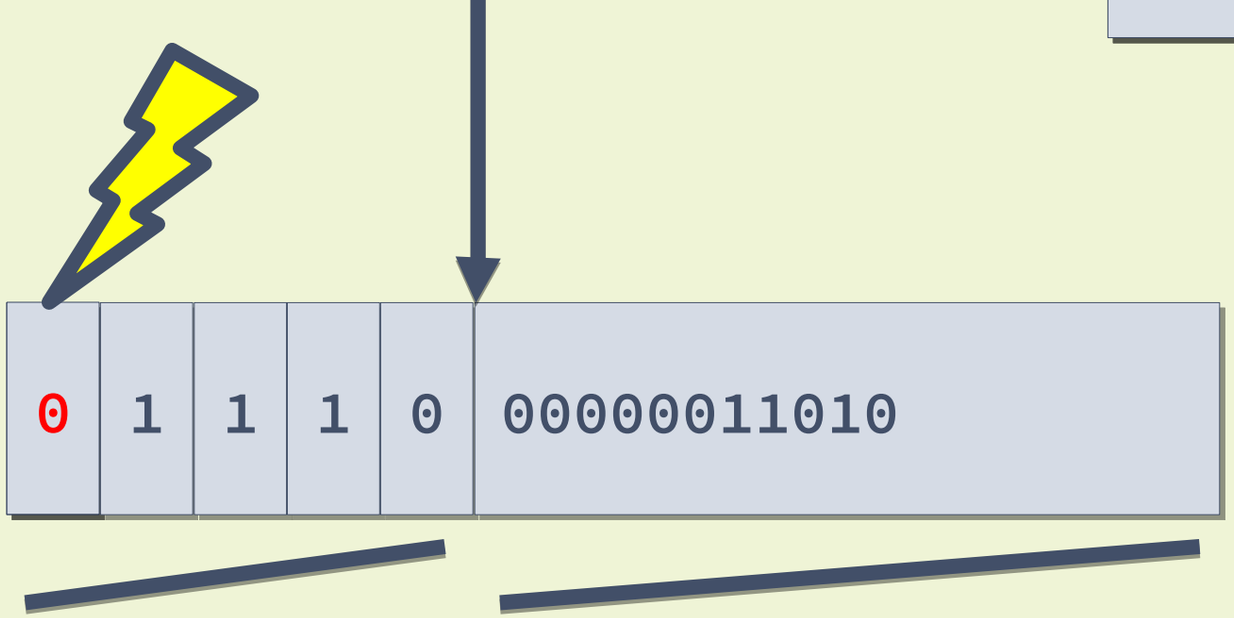
```
ldr   r0, [sp, #pin]
bl comparePIN
str   r0, [sp, #result]
```



Skipping/corrupting instructions

```
result = pin;
```

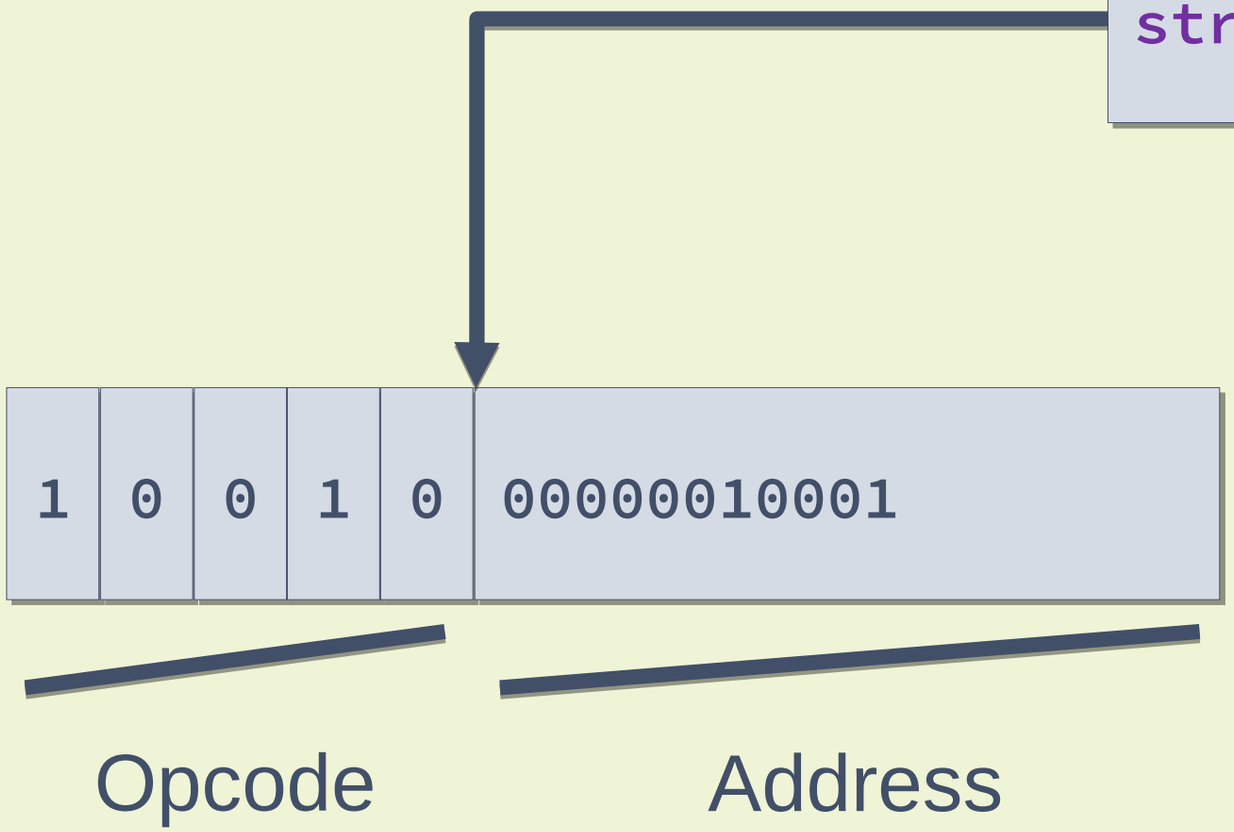
```
ldr  r0, [sp, #pin]
strb r2, [r3, #0]
str  r0, [sp, #result]
```



Skipping/corrupting instructions

```
result = comparePIN(pin);
```

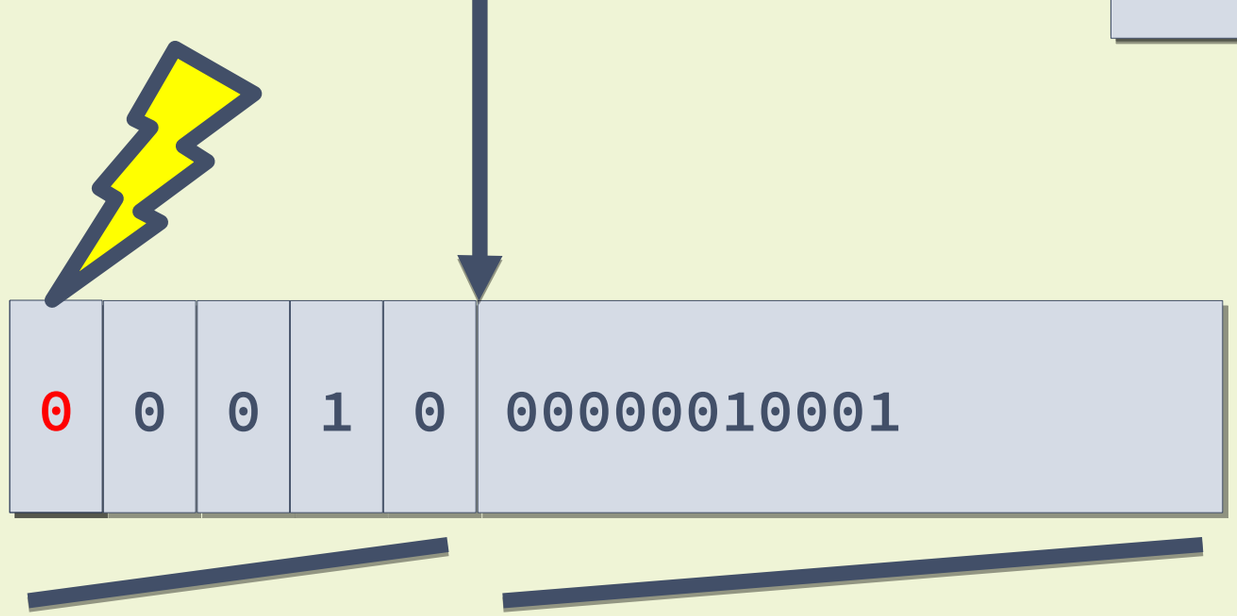
```
ldr   r0, [sp, #pin]
bl comparePIN
str   r0, [sp, #result]
```



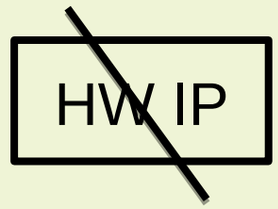
Skipping/corrupting instructions

```
comparePIN(pin);
```

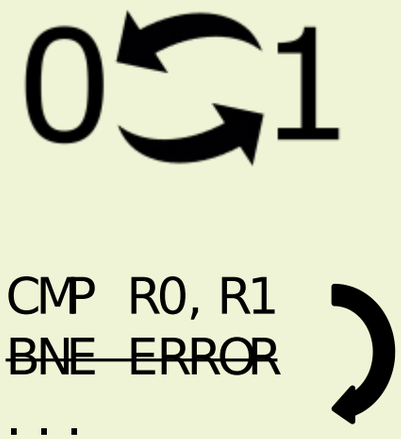
```
ldr  r0, [sp, #pin]
bl comparePIN
asrs r1, r2, #32
```



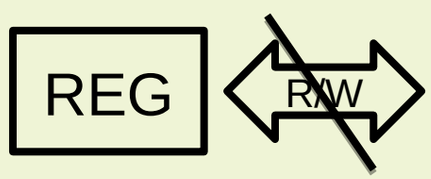
Effects of glitches



Disabling modules



Flipping bits



Skipping / corrupting instructions

Preventing R/W



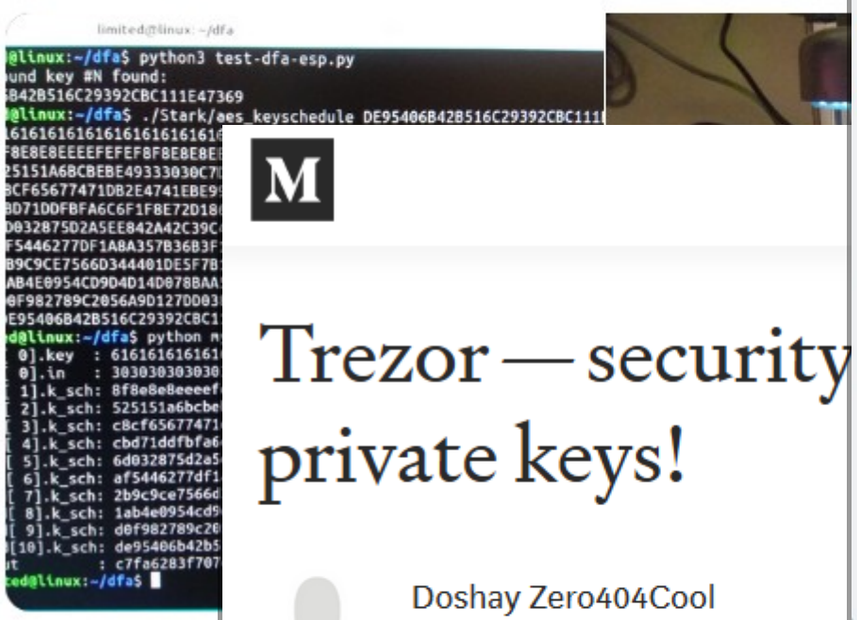
Chip Destruction

FI in the wild

FI is trendy!

LimitedResults
@LimitedResults

DFA successful on MbedTLS AES
esp32) #hardwarehacking #faulti
And nice analysis tools Jeangrey
@doegox thx man!



8:57 AM - 7 May

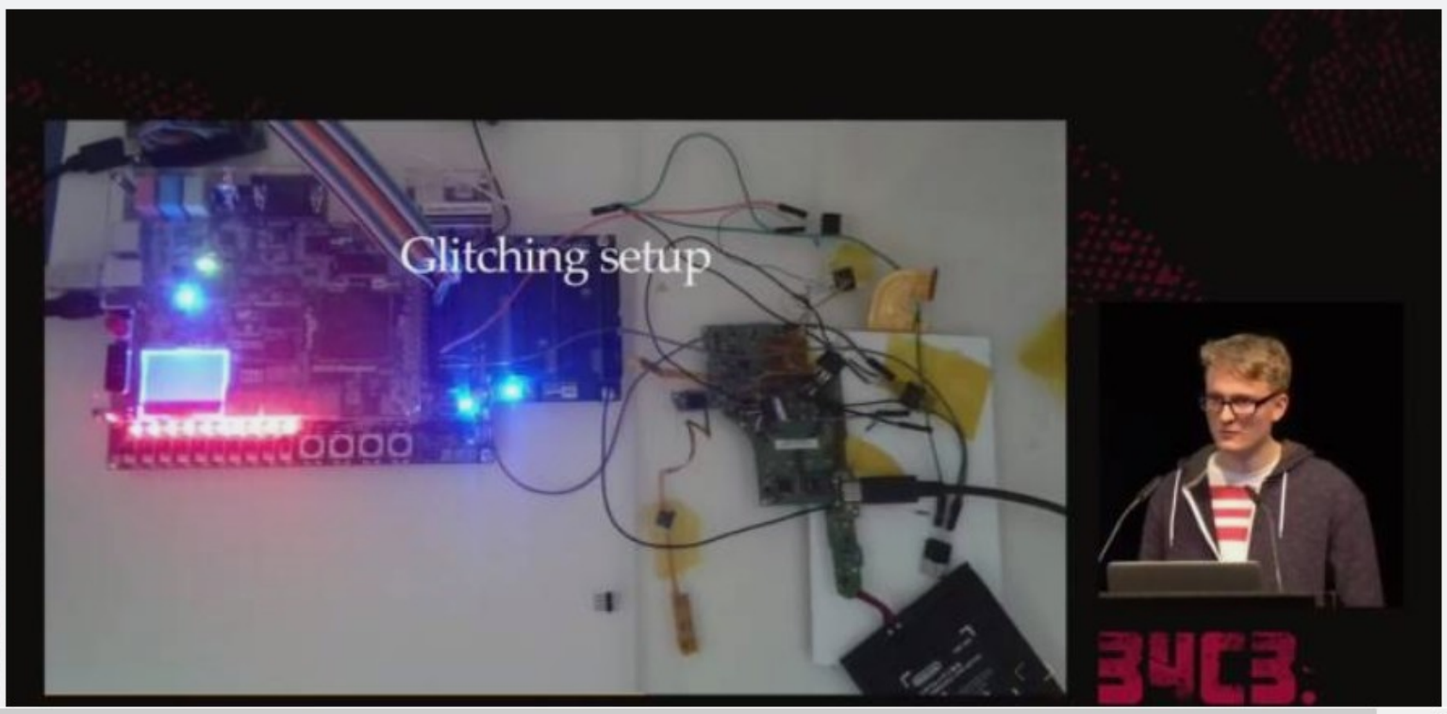
ars TECHNICA SUBSCRIBE SIGN IN

GAMING & CULTURE —

Hackers seem close to publicly unlocking the Nintendo Switch

Coders are rapidly finding exploits to open up Nintendo's new console.

KYLE ORLAND - 1/17/2018, 4:44 AM



M

Trezor — security
private keys!

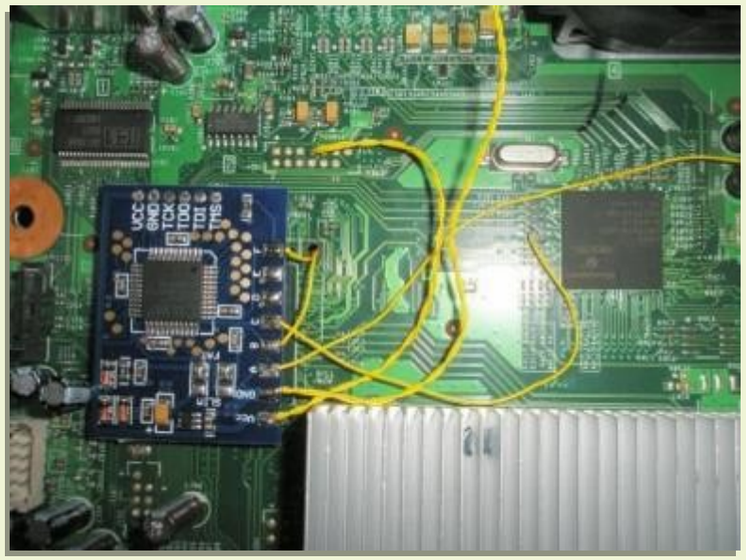
Doshay Zero404Cool
Aug 16, 2017 · 5 min read

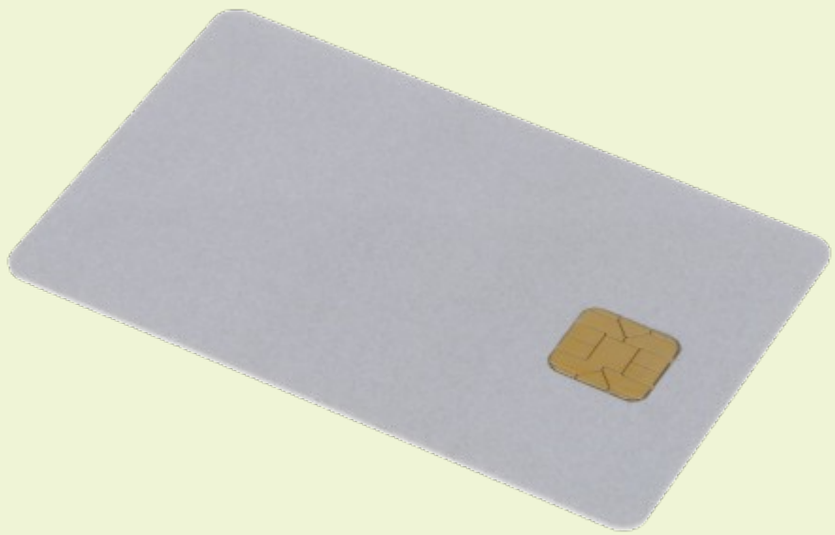
As we all know, a few weeks ago
stitches" attacks at DEFCON 25.
secure chips made by **STMicroelectronics**. The most important question everyone should ask is:

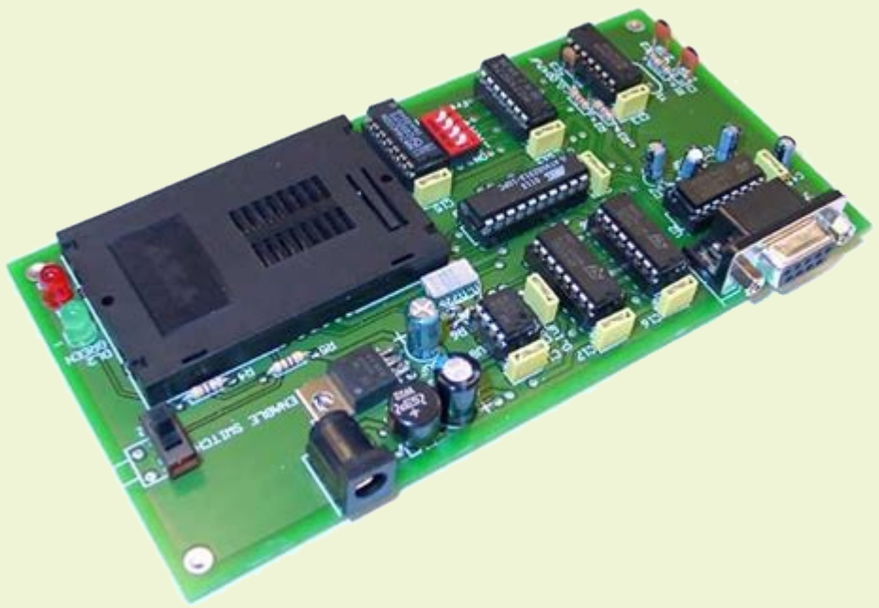
"Is the ST32F05 vulnerable to fault injection?"

But it has been there for years...









Age test!

What is the relation between these two?

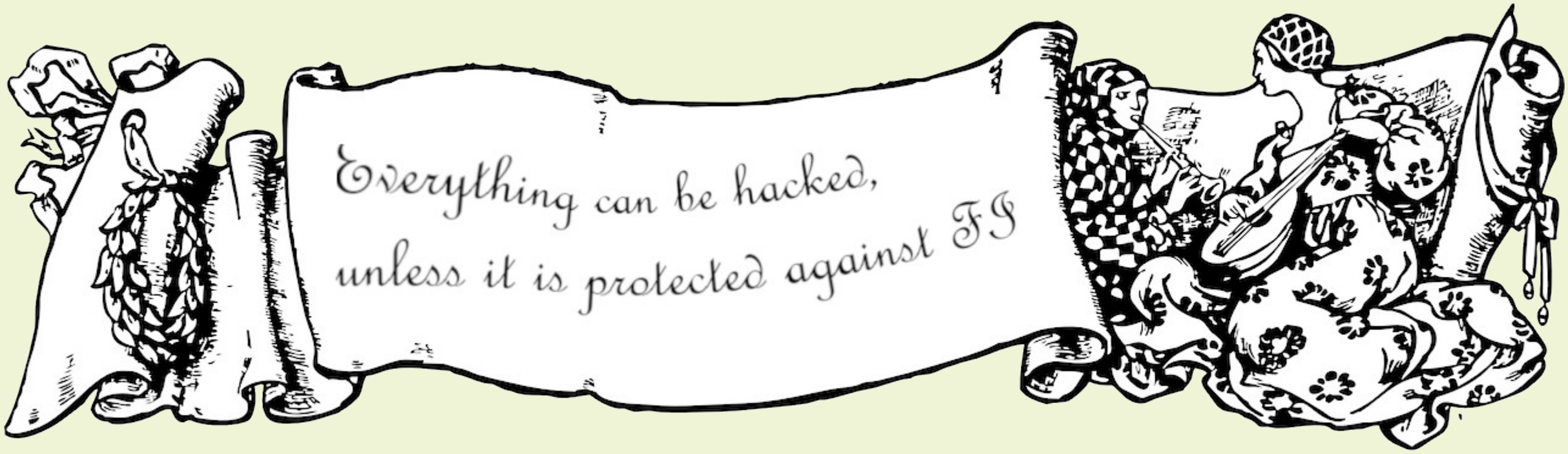




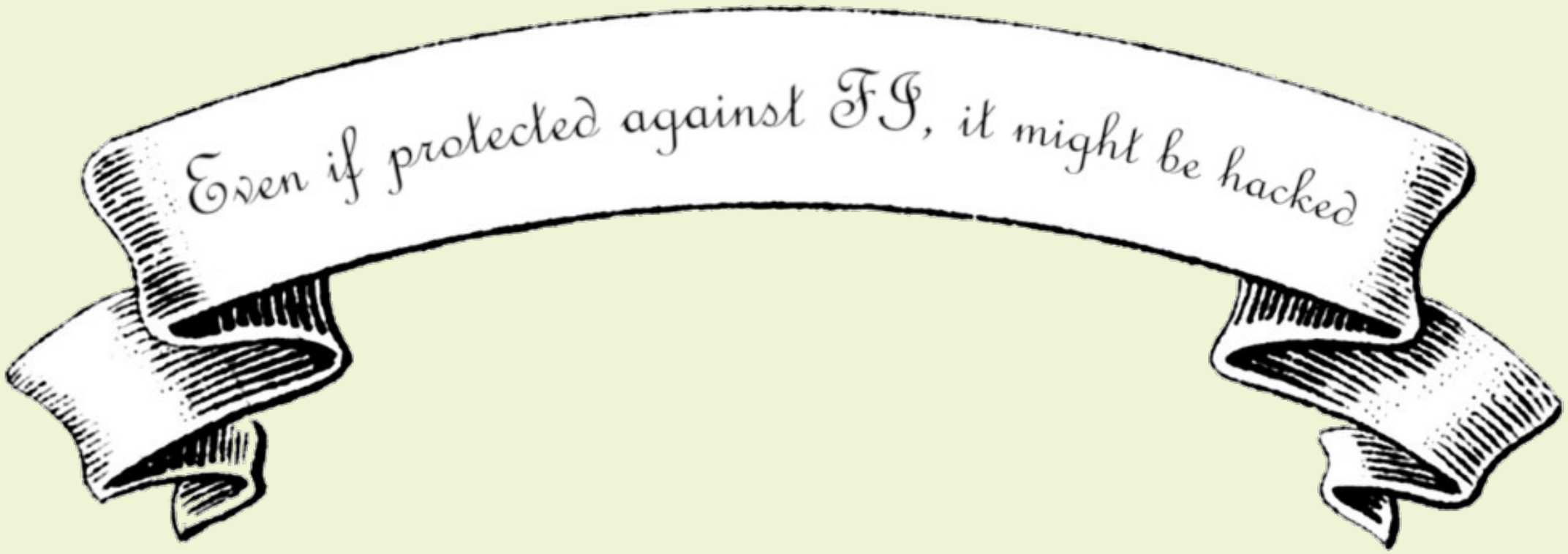
Don’t do this at home!

Practical FI

Fault Injection Rule #1



Fault Injection Rule #2



10+ Years Glitching

Bypassing authentication

Bypassing secure boot

Escalating privileges

Recovering crypto keys

Modifying security config.

Bypassing authentication

Bypassing authentications

```
const char *PIN = "19379148"

// Compare the PIN with the correct one
// Return true if different

bool comparePIN(char *pin) {

    for (int i=0; i<strlen(pin); i++) {
        if (pin[i]!=PIN[i])
            return true;
    }
    return false;
}
```

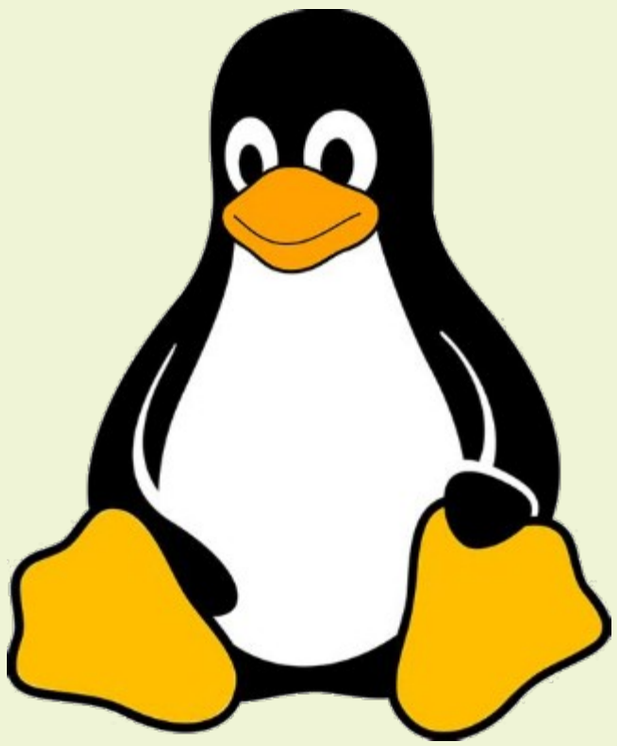
```
// Authenticate PIN.
// Return true if incorrect

bool pinAuthentication() {

    char pin[80];
    char tmp[80];
    bool result=false;


    printf("Please, introduce PIN\r\n");
    gets(pin);

    result = comparePIN(pin);

    if (result == false) {
        sprintf(tmp,"PIN correct\r\n");
    } else {
        sprintf(tmp,"PIN %s is
                        incorrect\r\
n",pin);
    }

    printf(tmp);
    return result;
}
```

Bypassing authentications



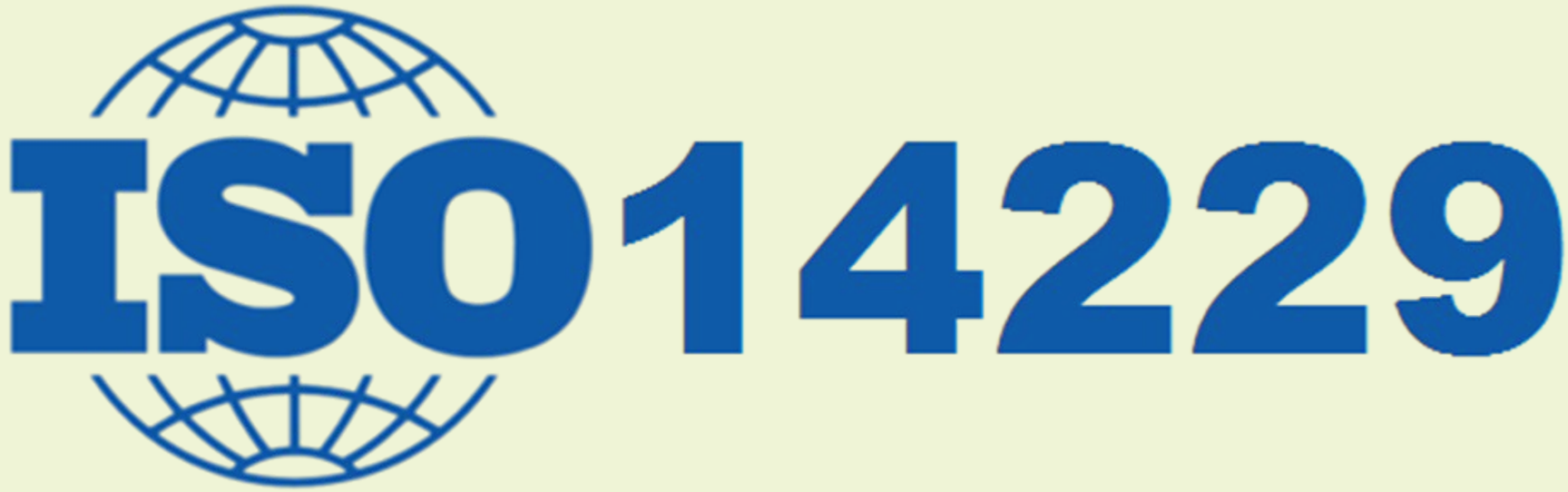








Unified Diagnostic Services

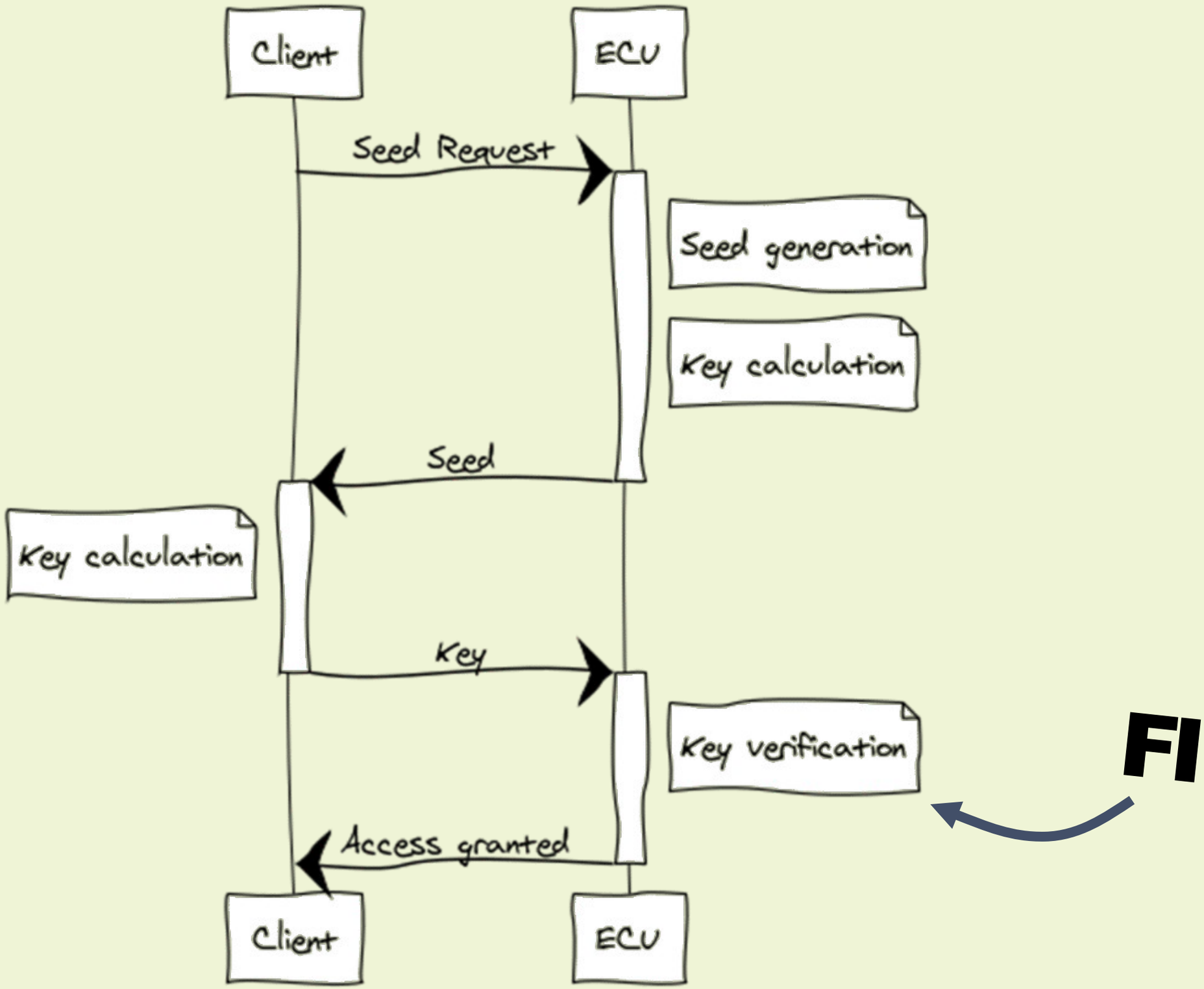


Unified

Diagnostic

Services

UDS authentication



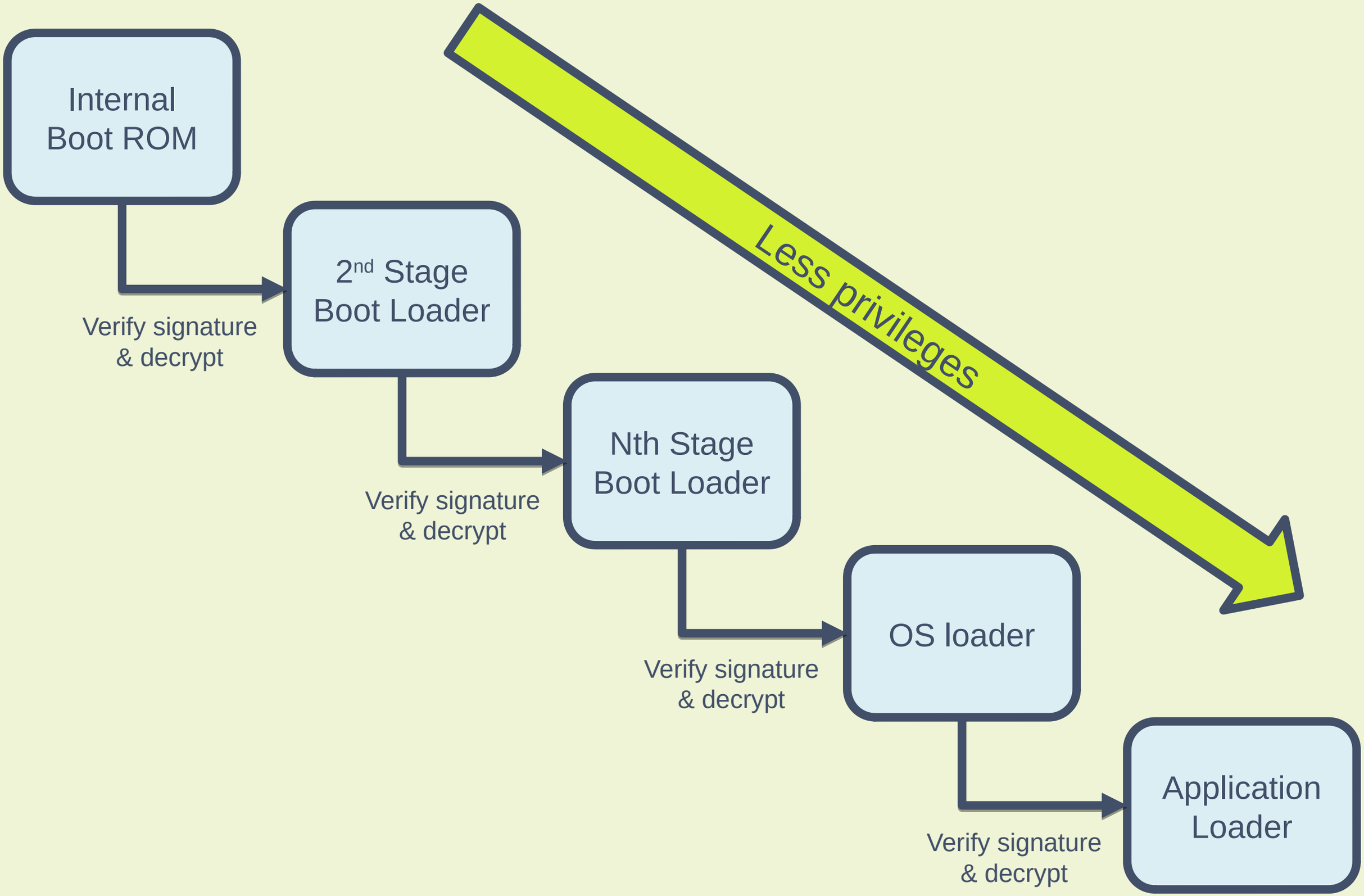
Bypassing UDS auth with FI

```
if (receivedKey != expectedKey)
        not_authenticated();
else
   authenticated();
```



Bypassing Secure Boot

Secure Boot


Internal
Boot ROM
Verify signature
& decrypt
2nd Stage
Boot Loader
Verify signature
& decrypt
Nth Stage
Boot Loader
Verify signature
& decrypt
OS loader
Verify signature
& decrypt
Application
Loader
Less privileges

Bypassing Secure Boot





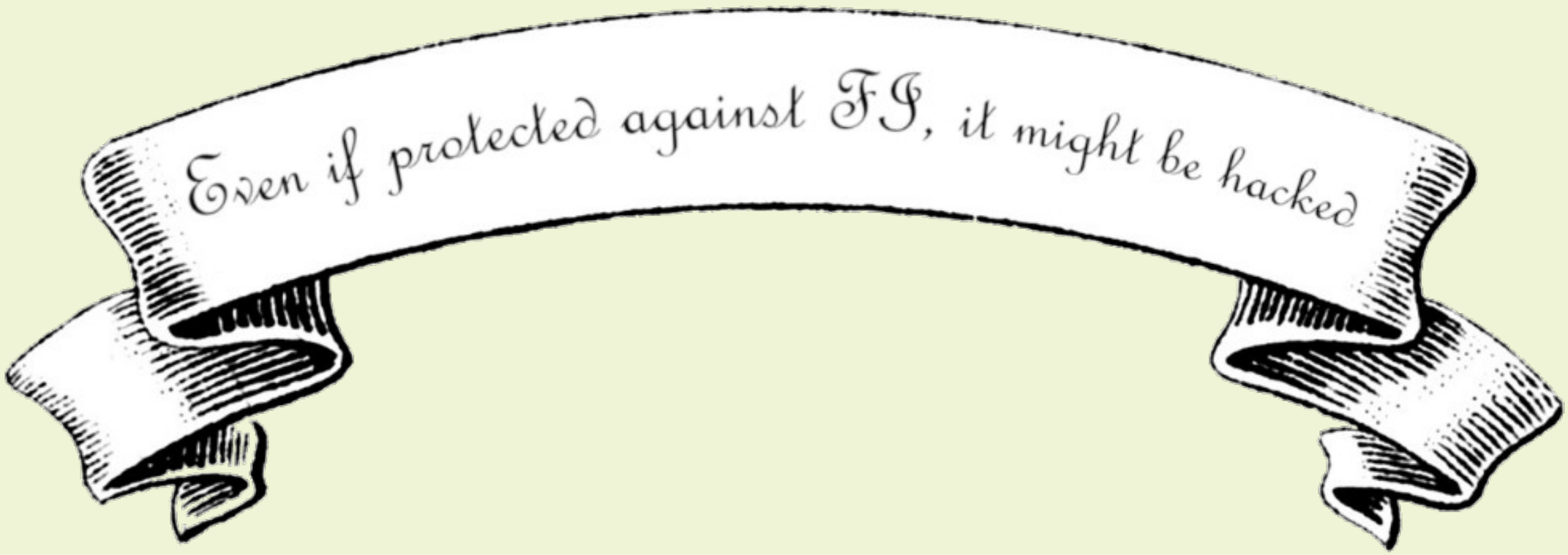


Bypassing Secure Boot

```
void boot() {

    void (*entryPoint)(void);
    ...

    // Load image to buffer
    uint32_t len = loadFlashImage(imgBuf)

    // Verify image
    if (verifyImage(imgBuf, len, sign,
                    rsaKey) == false) {
        goto reset;// Auth failed
    }

    // Jump to image
    entryPoint = imgbuf;
    (*entryPoint)();

    ...
}
```

```
//Verify image. Returns true is correct

bool verifyImage(uint8_t *image,
       uint32_t len, uint8_t *sign, rsaKey) {

    char *hash1, *hash2;
    ...

    // Calculate Hash
    calculateHash(image,len,hash1)

    // Verify signature
    verifySignature(sign, rsaKey, hash2)

    // Compare hashes
    if (memcmp(hash1, hash2, HASH_LEN) != 0)
        return false;
    else
        return true;

}
```

Do you remember the FI rule #2?

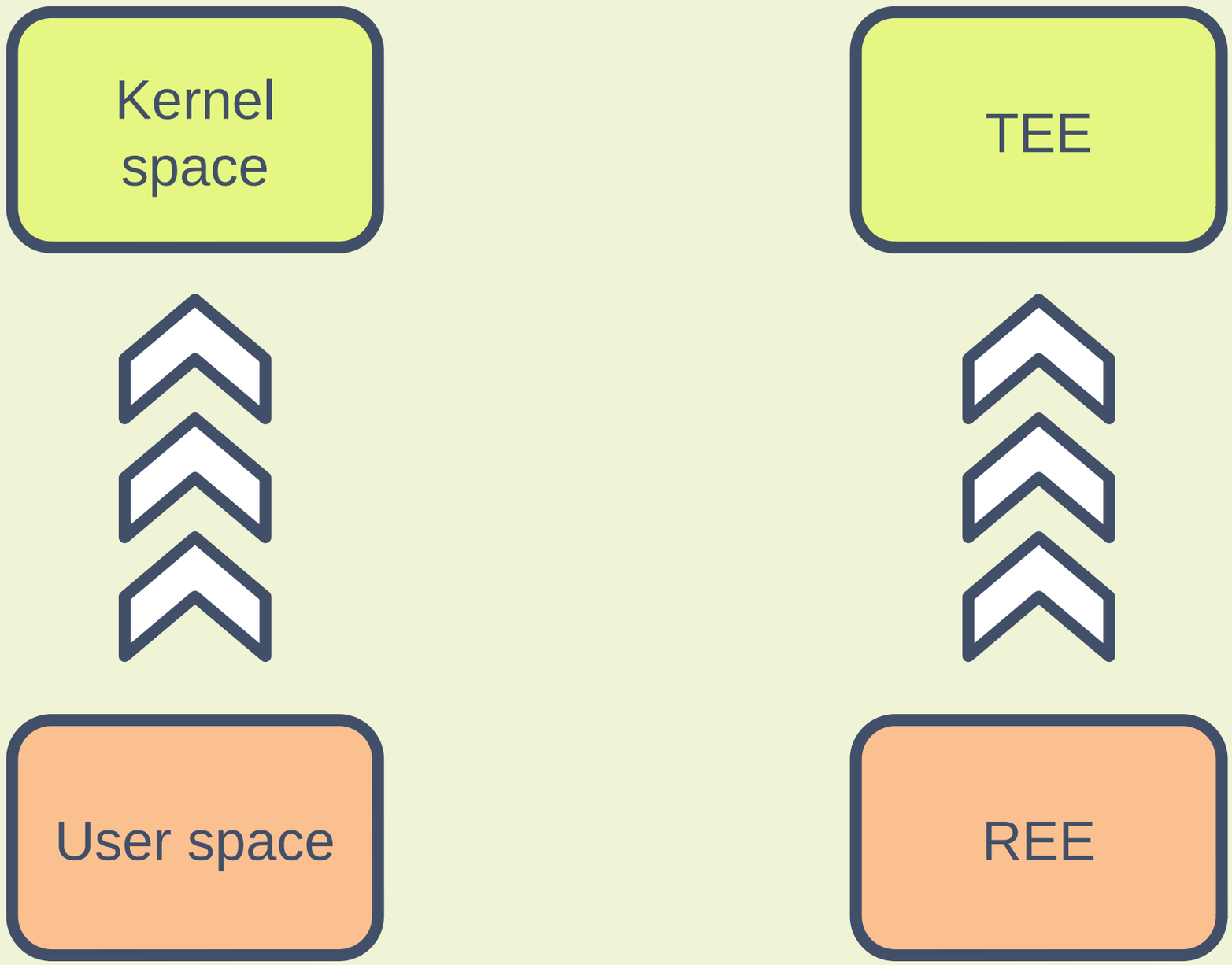


Protecting a Secure Boot can be a nightmare!

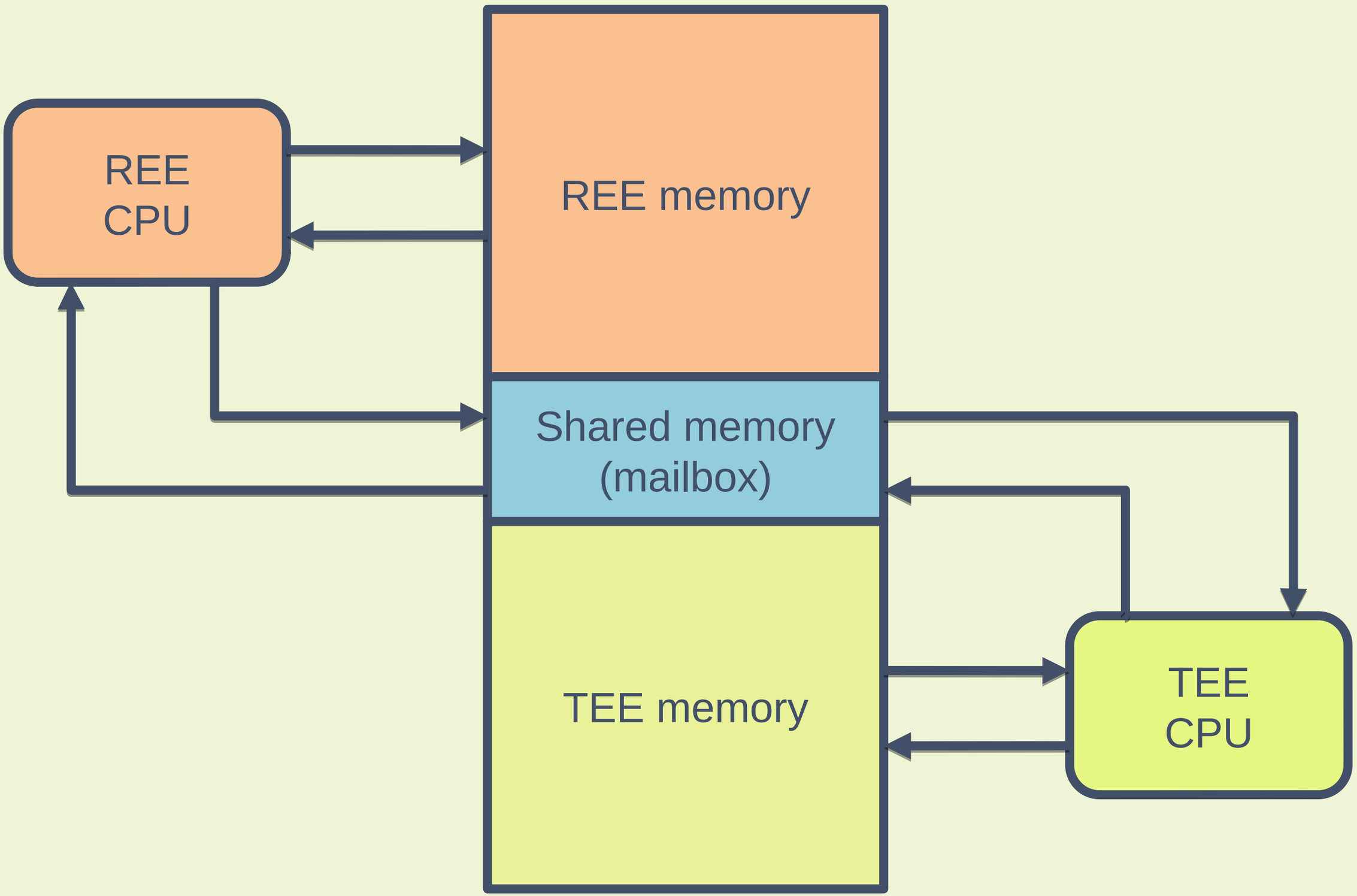
FISim Demo

Escalating privileges

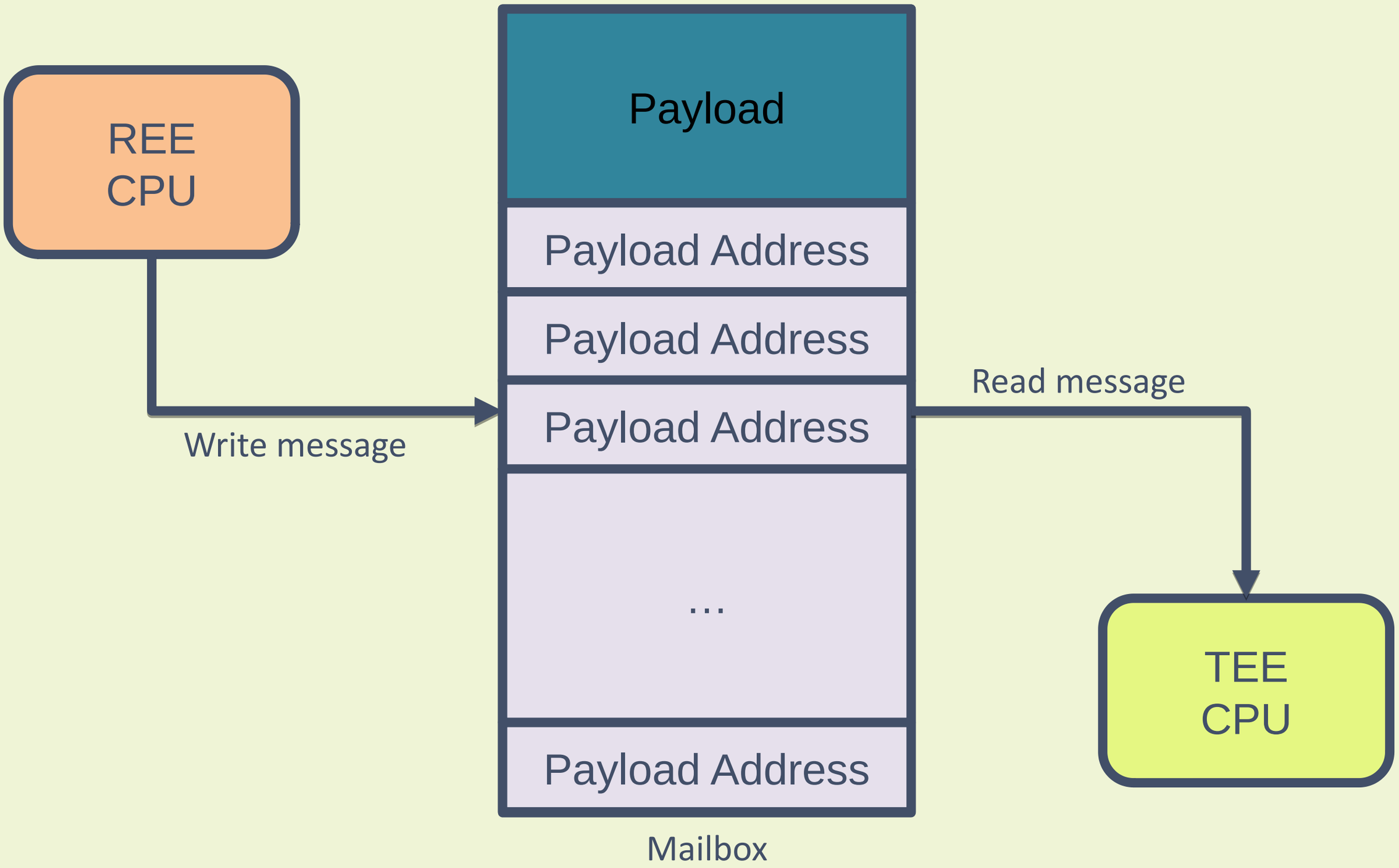
Escalating privileges



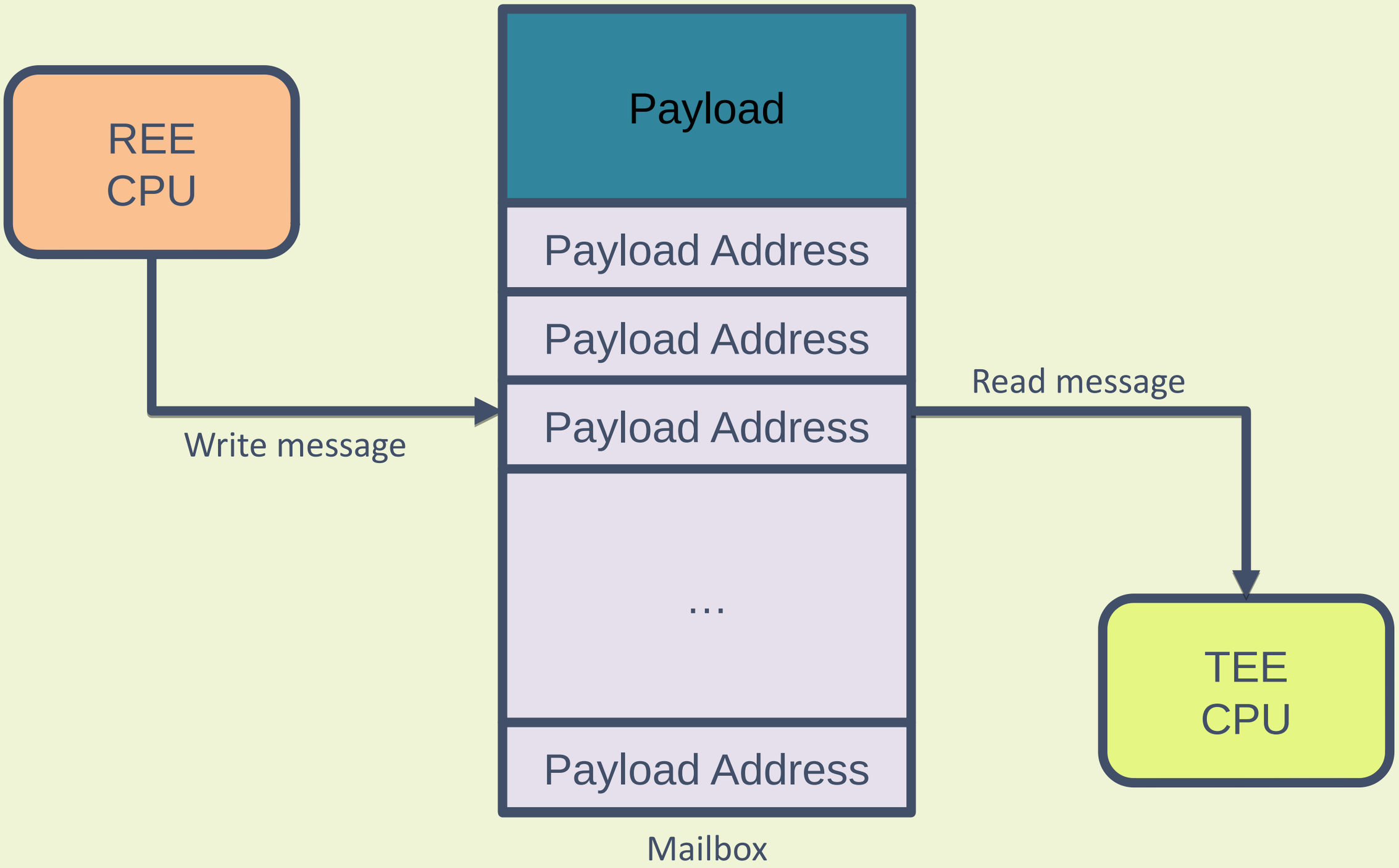
REE-TEE separation



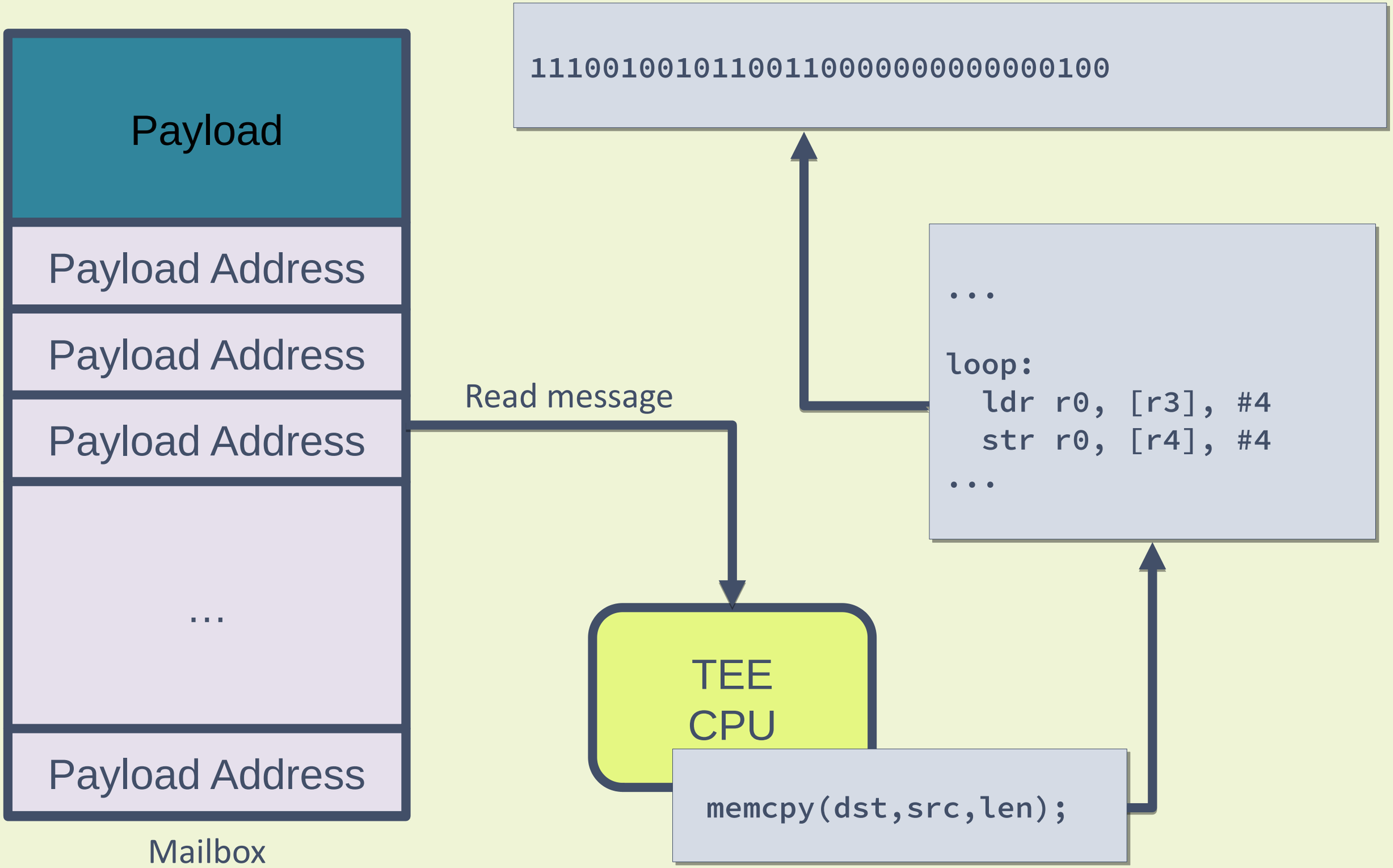
Wild Jungle Jump Attack



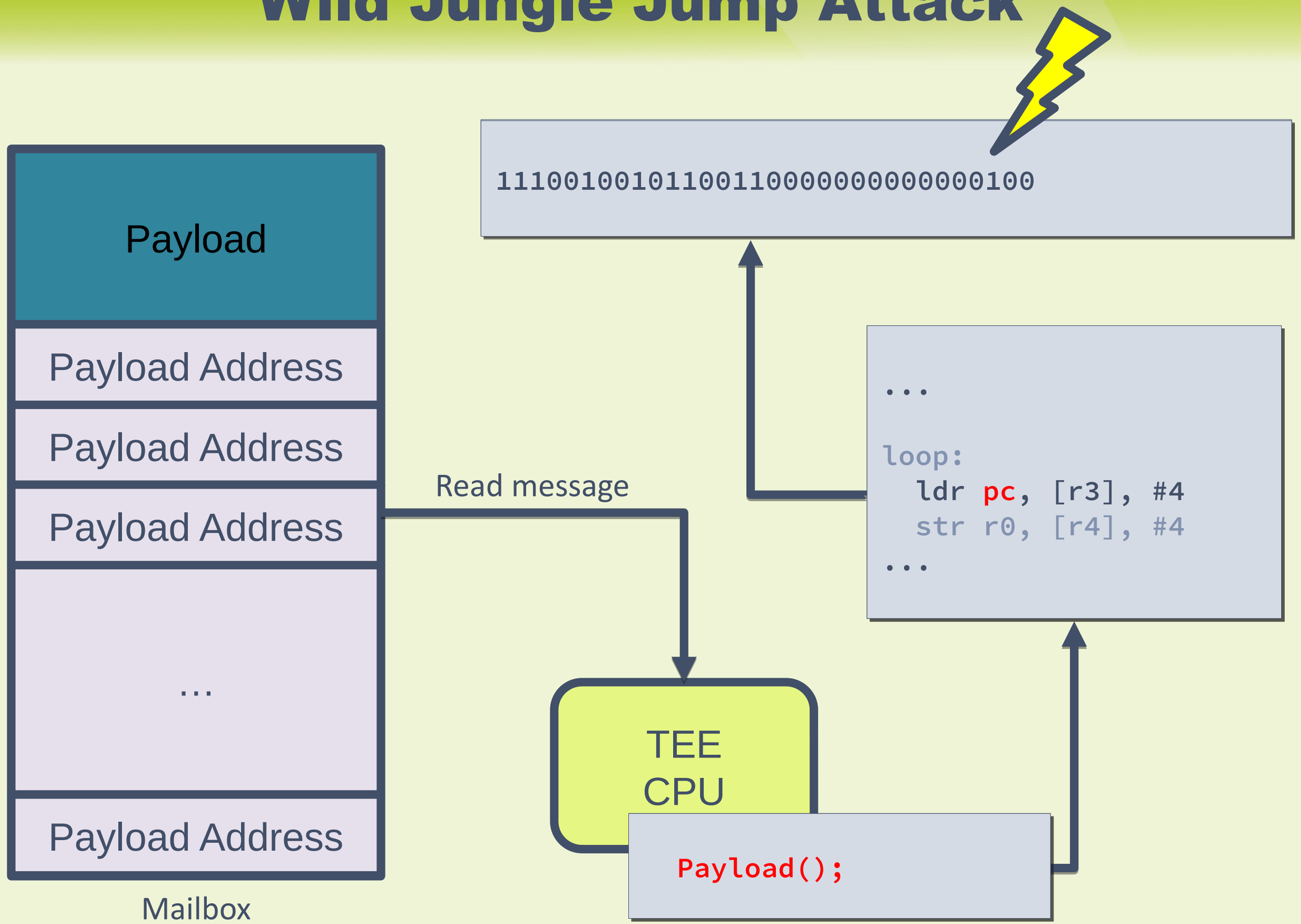
Wild Jungle Jump Attack



Wild Jungle Jump Attack

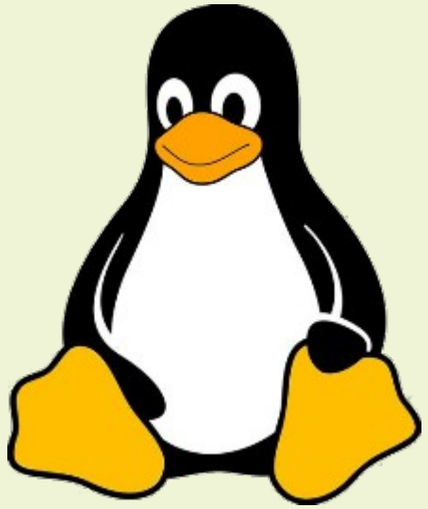


Wild Jungle Jump Attack



Escalating privileges









arm

TRUSTZONE

Recovering crypto keys

Recovering crypto keys

FAULT ANALYSIS

DFA

Differential Fault Anal

SFA

al Fault Analysis

DFIA

Differential Fault
Intensity Analysis

IFA

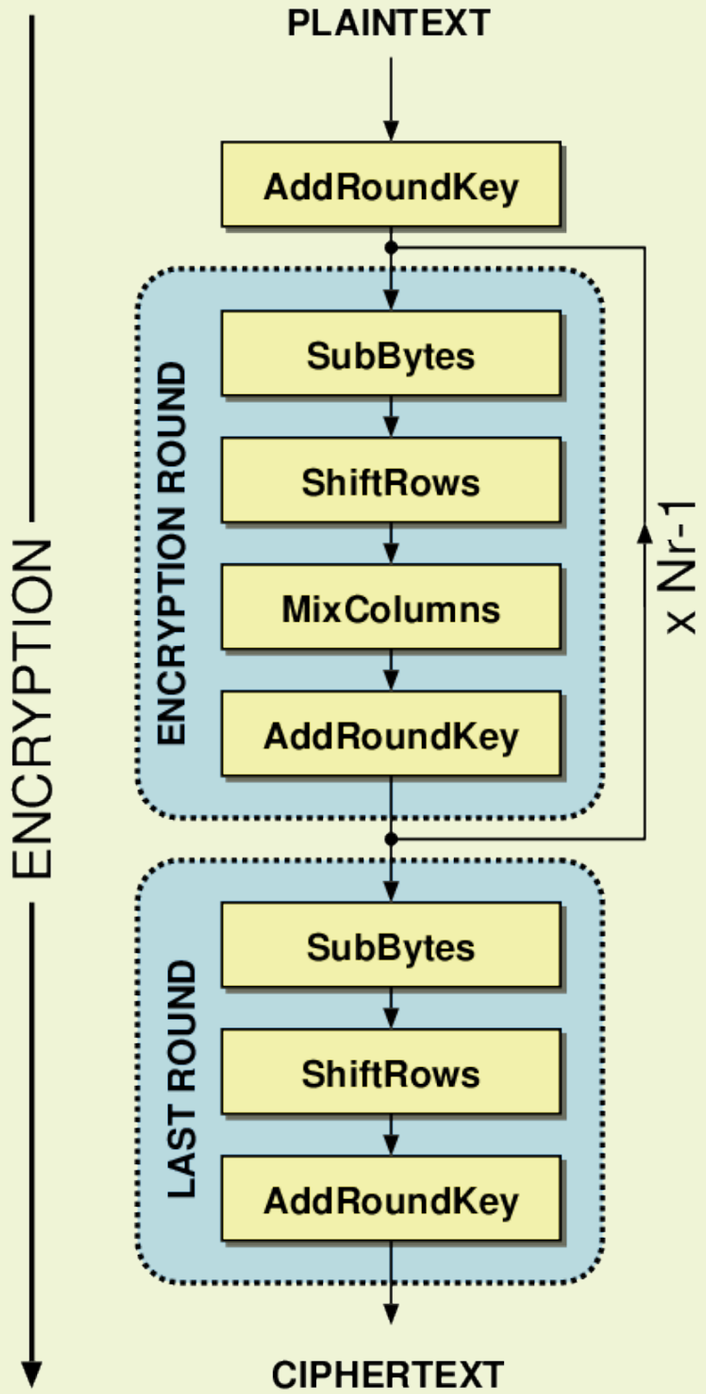
Ineffective Fault Analysis

Collis

SIFA

Statistical Ineffective
Fault Analysis

AES Fault Analysis

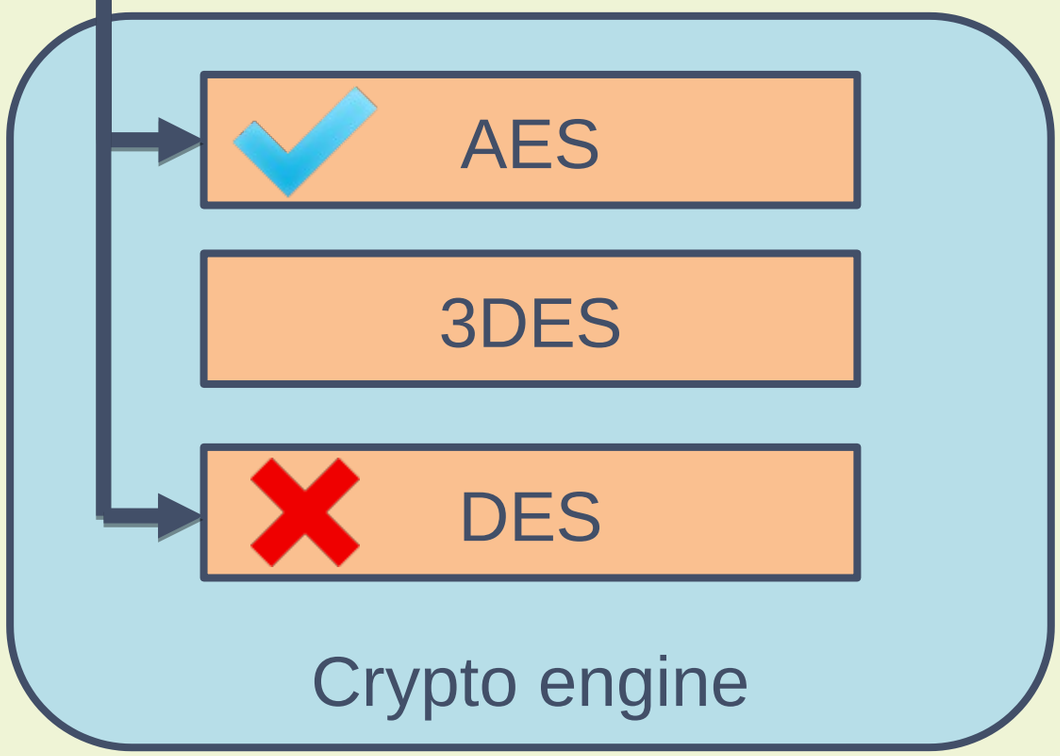


$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$C_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

FI on key attributes

Key	Attributes	
Key 1	AES	ENC
Key 2	3DES	DEC
Key 3	AES/3DES	ENC/DEC
…	…	…

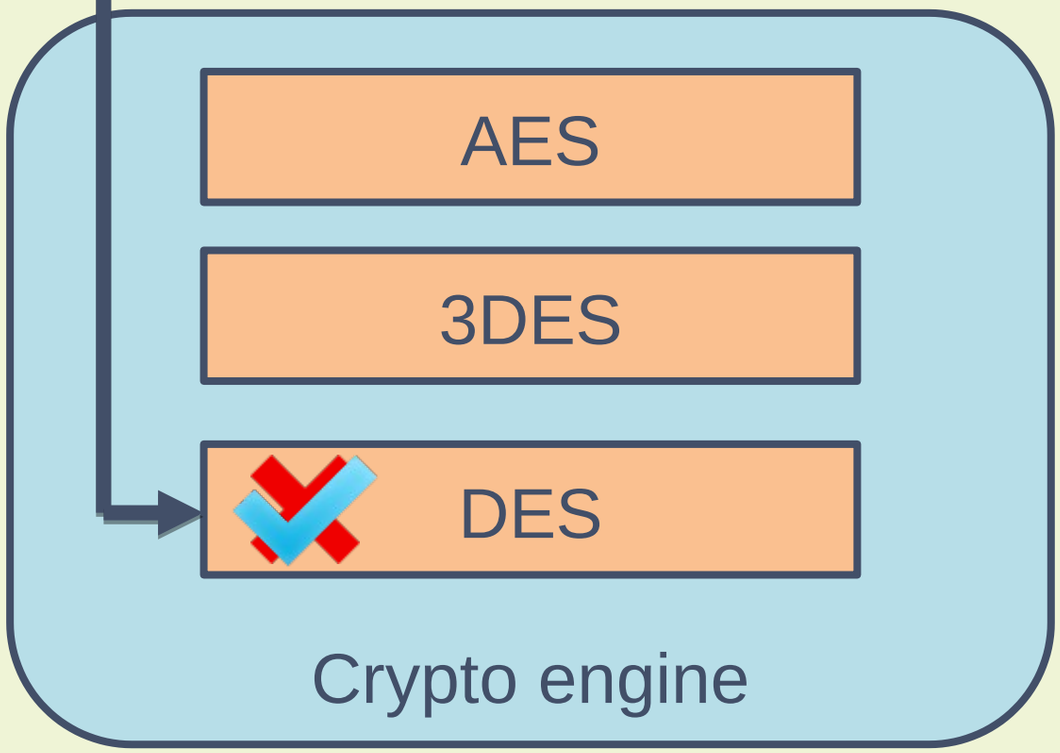
Keyslot



FI on key attributes

Key	Attributes	
Key 1	**DES**	ENC
Key 2	3DES	DEC
Key 3	AES/3DES	ENC/DEC
…	…	…

Keyslot



Recovering crypto keys









Modifying security configuration

Preventing the configuration

```
void boot() {

    void (*entryPoint)(void);
    ...

    // Load image to buffer
    uint32_t len = loadFlashImage(imgBuf)

    // Verify image
    if (verifyImage(imgBuf, len, sign,
                    rsaKey) == false) {
        goto reset;// Auth failed
    }

    // Jump to image
    entryPoint = imgbuf;
    (*entryPoint)();

    ...
}
```

```
// Read fuses and configure security modules

bool disableJtag = readFuse(DISABLE_JTAG);
bool enableMemoryScrambler =
            readFuse(ENABLE_MEM_SCRAMBLER);
bool enableTEE = readFuse(ENABLE_TEE);


if (disableJtag==true)
    closeJtag();

if (enableMemoryScrambler==true)
    configureMemoryScrambler();

if (enableTEE==true)
    configureTEE();
```

Localized Fault Injection

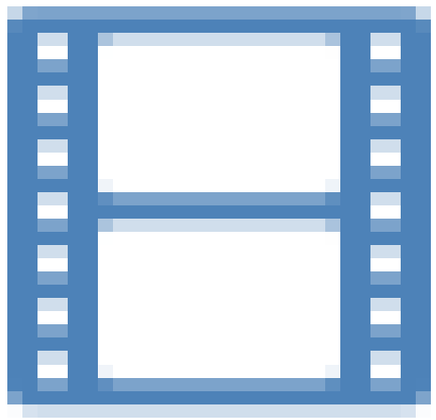


Electro-Magnetic FI

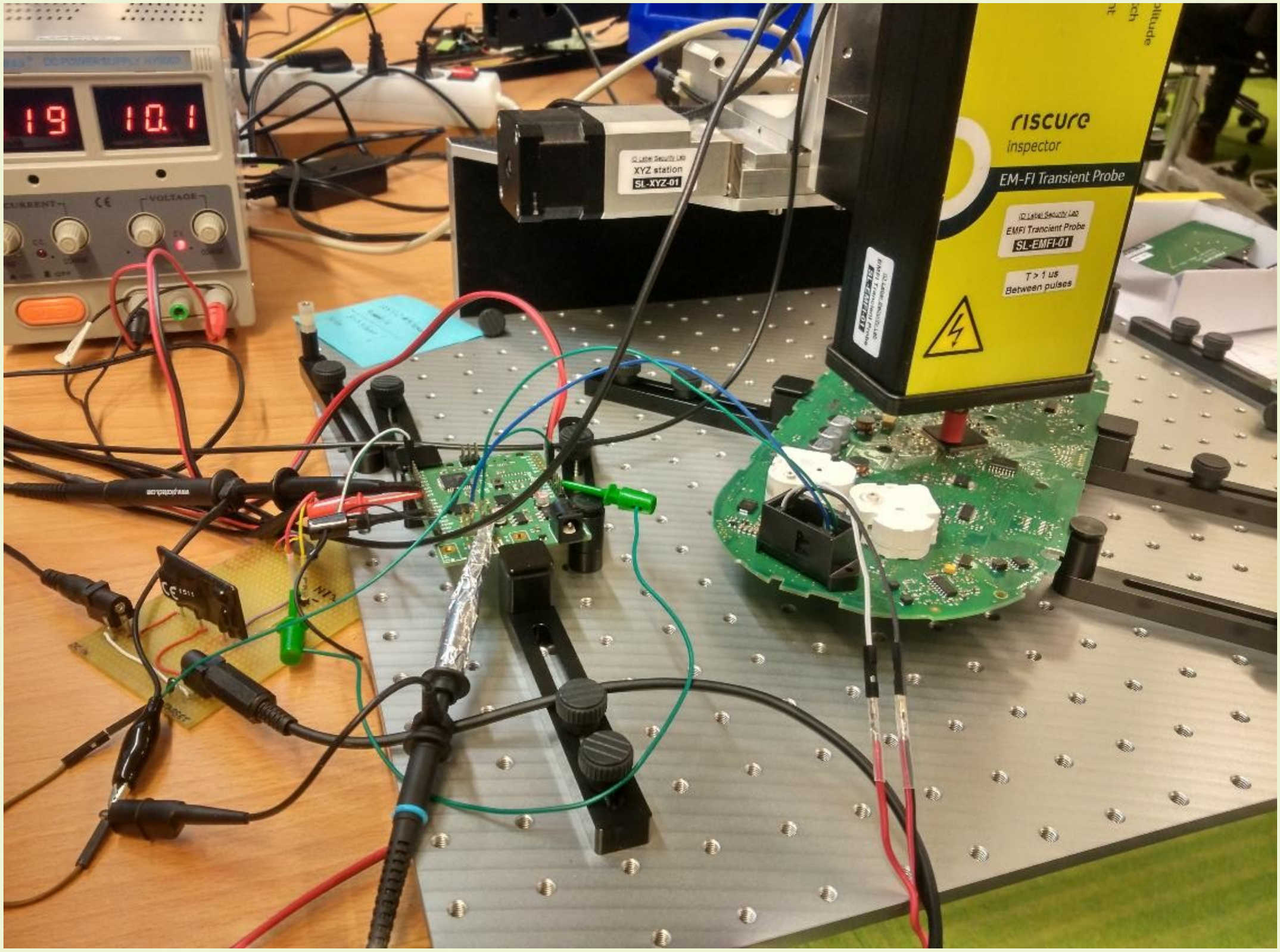


Laser FI

Laser Fault Injection

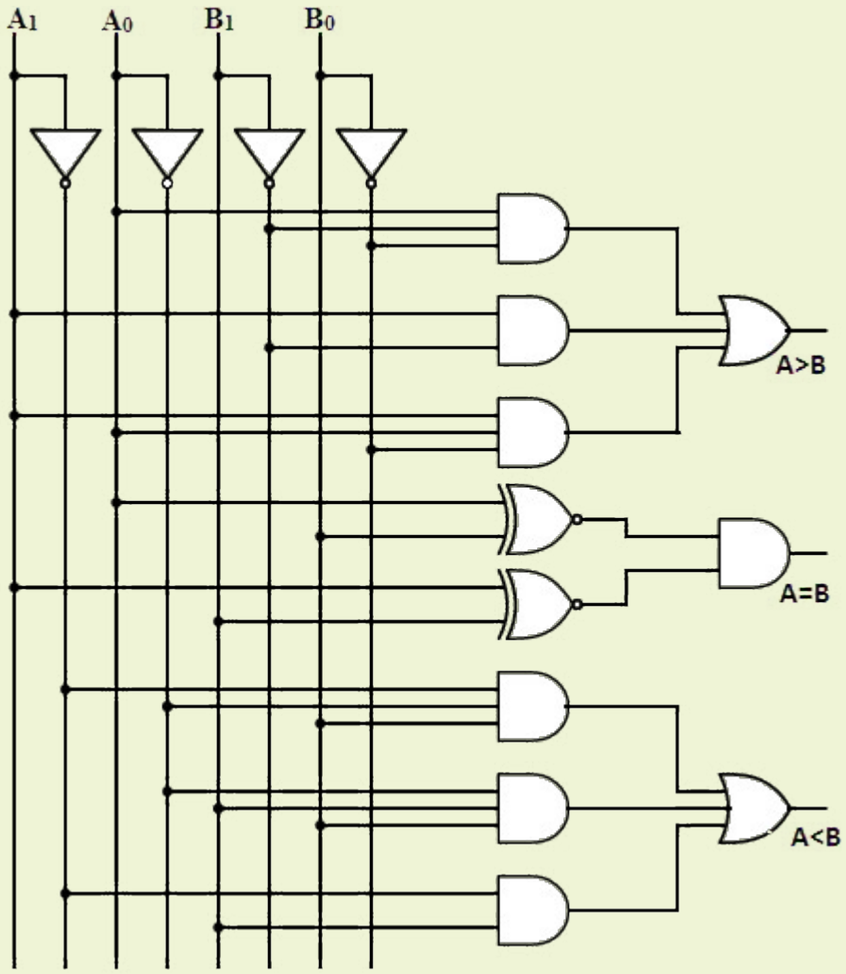


EM Fault Injection



Preventing FI

FI countermeasures



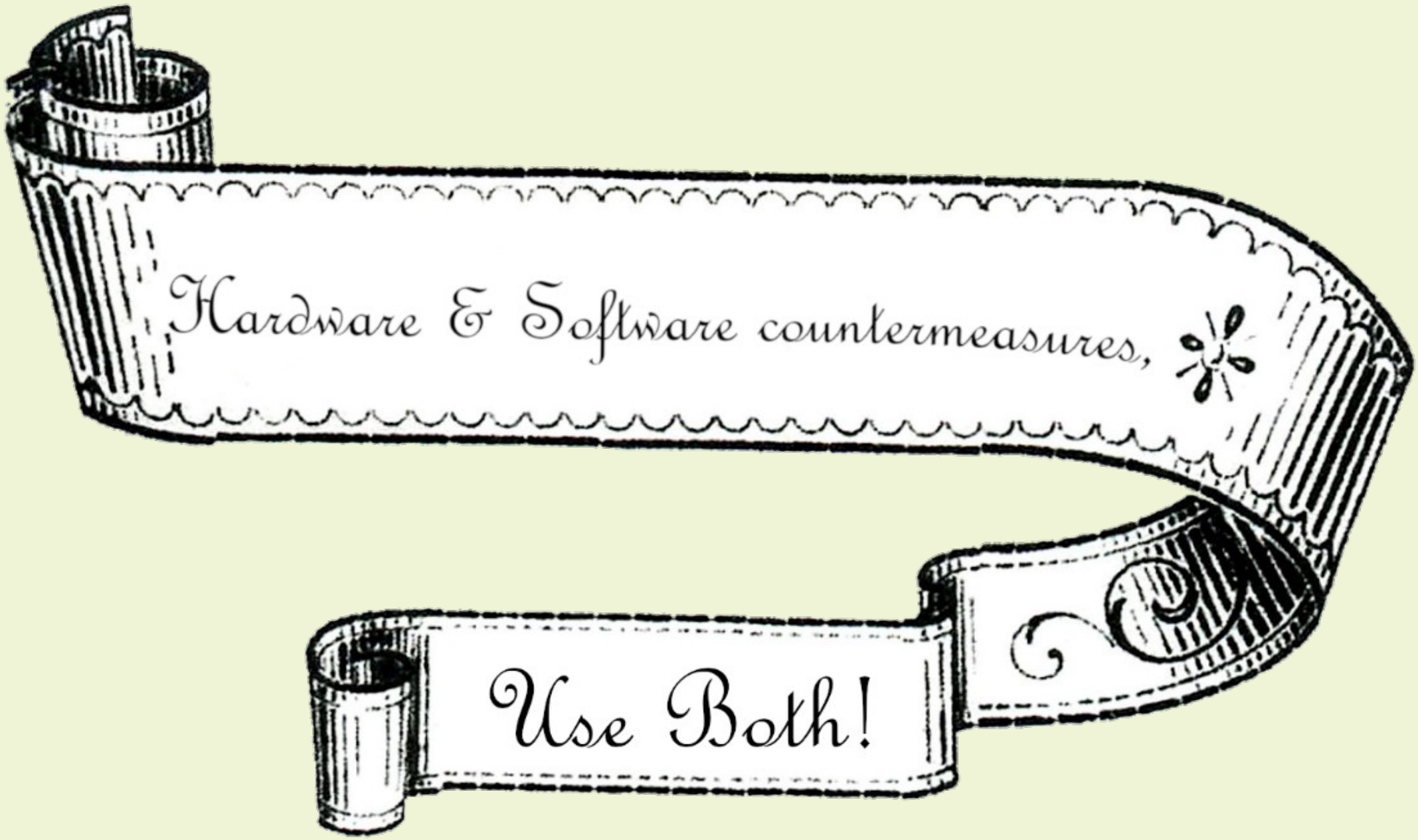
HW

VS

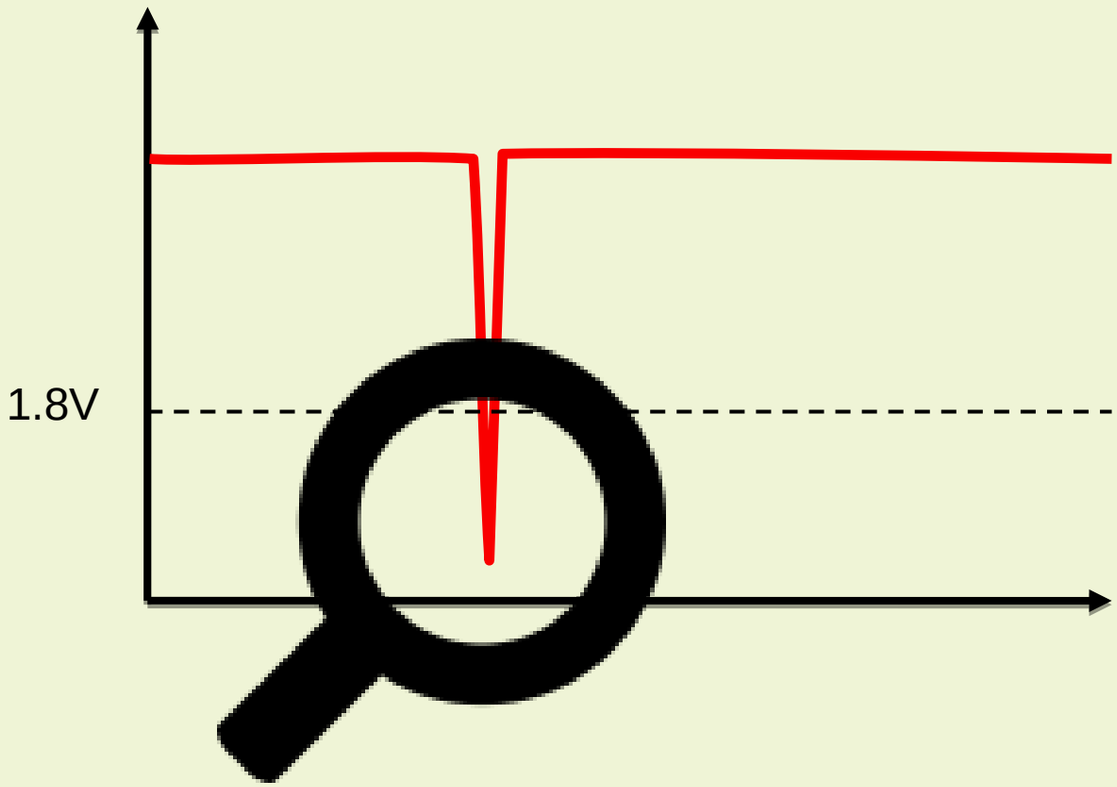
```
00000000                push    ebp
00000001                mov     ebp, esp
00000003                movzx   ecx, [ebp+arg_0]
00000007                pop     ebp
00000008                movzx   dx, cl
0000000C                lea     eax, [edx+edx]
0000000F                add     eax, edx
00000011                shl     eax, 2
00000014                add     eax, edx
00000016                shr     eax, 8
00000019                sub     cl, al
0000001B                shr     cl, 1
0000001D                add     al, cl
0000001F                shr     al, 5
00000022                movzx   eax, al
00000025                retn
```

SW

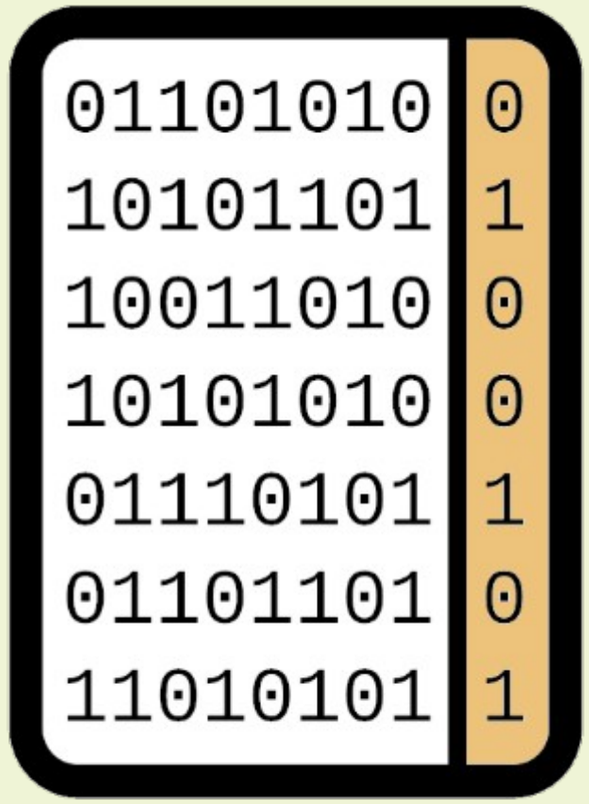
Fault Injection Rule #3



HW countermeasures



Glitch Sensors



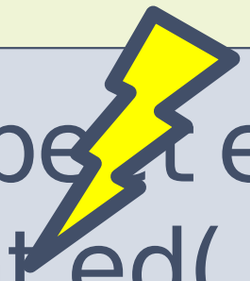
Redundancy

SW countermeasure #1

```
if (receivedKey != expectedKey)
        not_authenticated();
else
   authenticated();
```



SW countermeasure #1



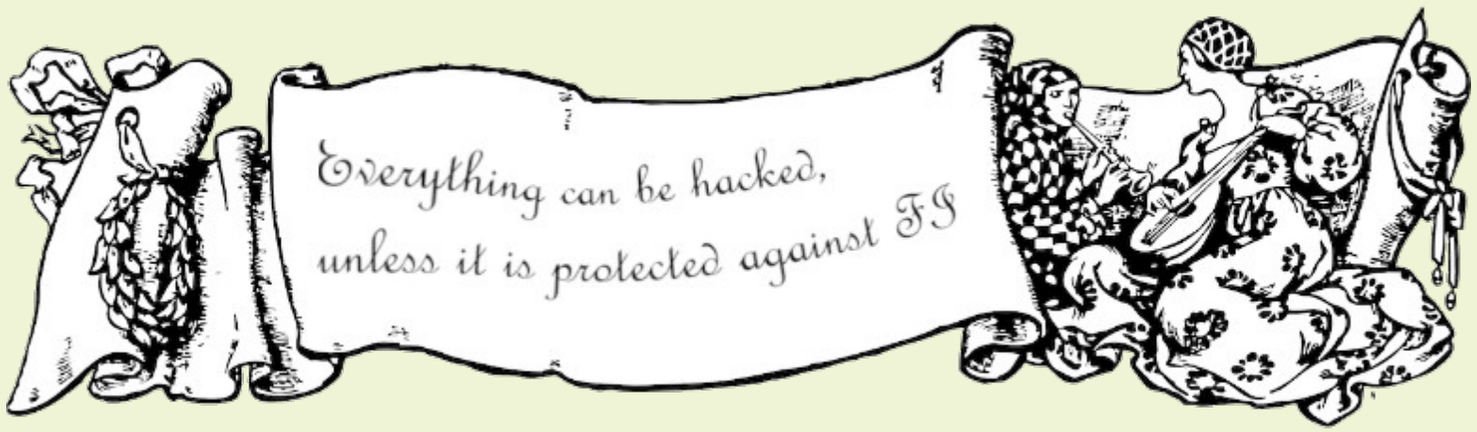
```
if (receivedKey != expectedKey)
        not_authenticated();

...
// Do something else
...

if (receivedKey != expectedKey)
        not_authenticated();
else
   authenticated();
```

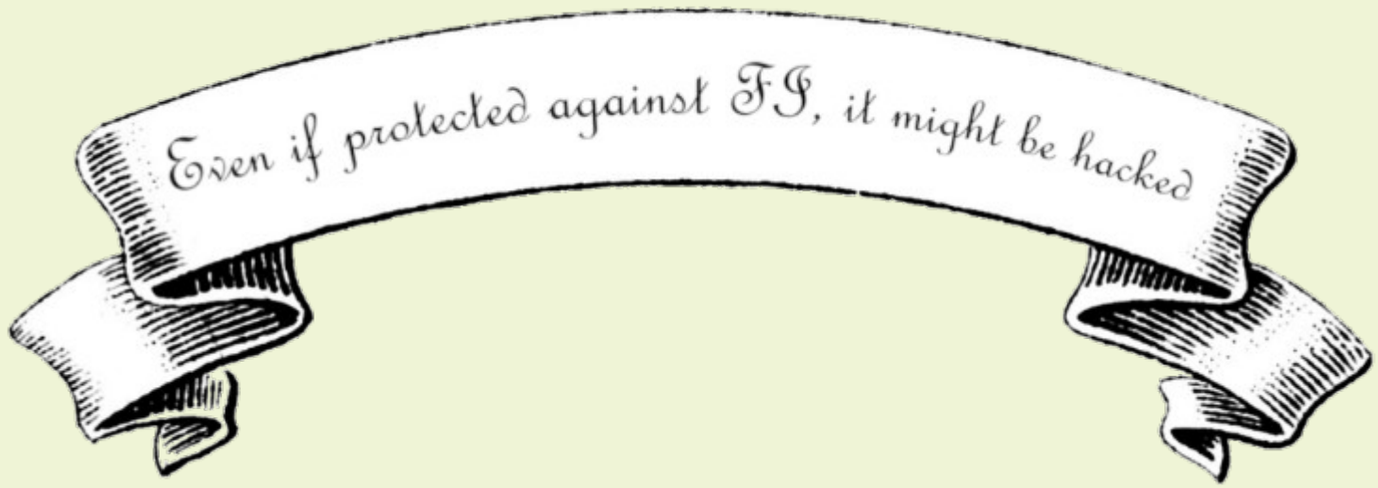
SW countermeasure #2

```
sleep(rand()); // Random delay

if (receivedKey != expectedKey)
        not_authenticated();
else
   authenticated();
```

Conclusions

Remember the rules!







More information:

Protecting software against FI
https://www.riscure.com/uploads/2018/11/201708_Riscure_Whitepaper_Side_Channel_Patterns.pdf

FI on UDS:
https://www.riscure.com/uploads/2018/06/
Riscure_Whitepaper_Fault_injection_on_automotive_diagnostic_protocols.pdf

Bypassing secure boot
https://www.riscure.com/uploads/2017/10/eu-16-Timmers-Bypassing-Secure-Boot-Using-Fault-Injection.pdf

Linux privileges escalation:
https://www.riscure.com/uploads/2017/10/
Riscure_Whitepaper_Escalating_Privileges_in_Linux_using_Fault_Injection.pdf

Optical FI
https://www.riscure.com/uploads/2017/09/Practical-optical-fault-injection-on-secure-microcontrollers.pdf

Wild Jungle Jump attack
https://www.riscure.com/uploads/2017/09/Controlling-PC-on-ARM-using-Fault-Injection.pdf

Practical DFA
https://www.slideshare.net/secret/K8jlDL4os1evrW

WE ARE HIRING!

Shanghai

Delft

(Netherlands)

San

Contact: Ramiro Pareja
pareja@riscure.com

Riscure China
2081, No.989, Changle Road
200031 Shanghai
China
Phone: +86 21 5117 5440
Wechat: RiscureChina

Riscure B.V.
Frontier Building, Delftechpark 49
2628 XJ Delft
The Netherlands
Phone: +31 15 251 40 90

www.riscure.com

Riscure North America
550 Kearny St.
Suite 330
San Francisco, CA 94108
+1 (650) 646 9979

inforequest@riscure.com