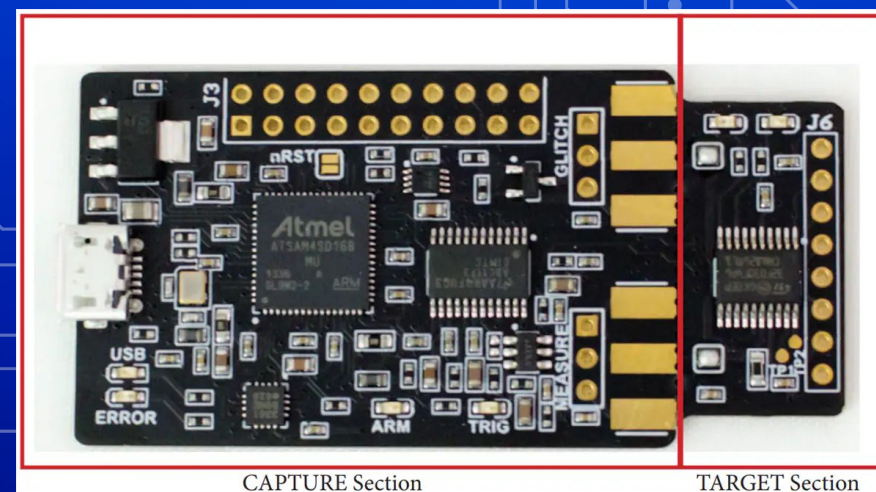




Introduction to Hardware (In)Security with the ChipWhisperer-Nano

Nathan Jones



<https://tinyurl.com/3kw99dfc>



Agenda

1. Introductions
2. How bad is it?
3. Introduction to the CW-Nano
4. Attacking a target
 - a. Side-channel attacks
 - b. Fault injection attacks
5. Closing thoughts



Agenda

1. Introductions
2. How bad is it?
3. Introduction to the CW-Nano
4. Attacking a target
 - a. Side-channel attacks
 - b. Fault injection attacks
5. Closing thoughts

Project 1: Getting a power trace

Project 2: Use power traces to identify individual bits

Project 3: Use DPA to recover an AES key

Project 4: Characterizing voltage faults

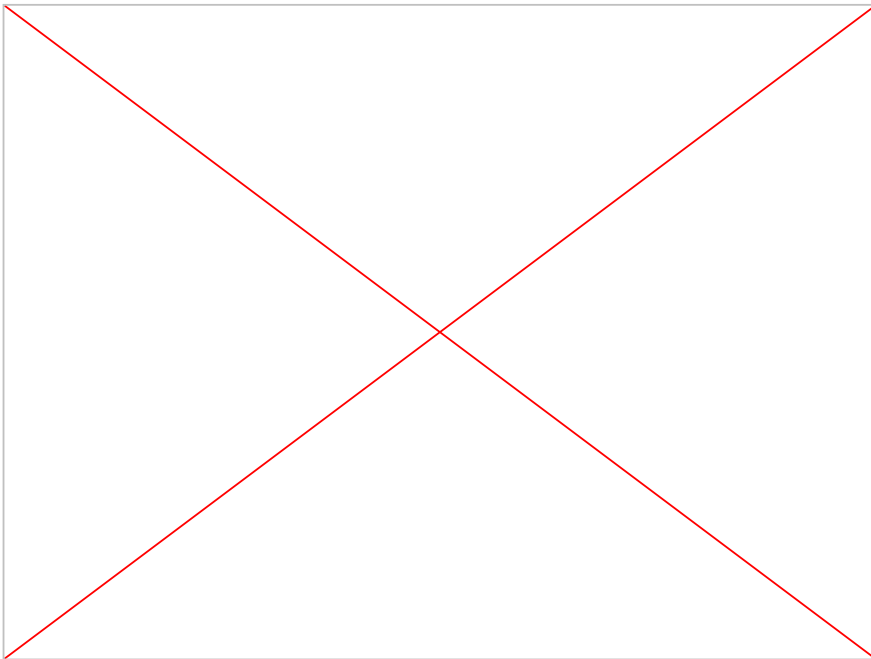
Project 5: Use voltage FI to bypass UART login

The *real* reason we're here today!

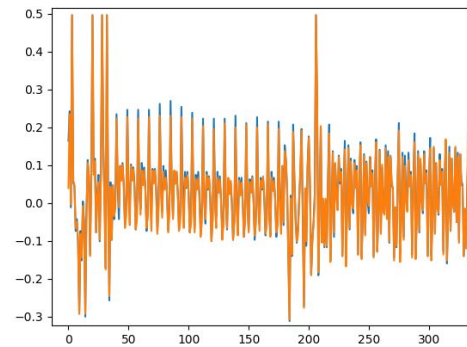
<https://tinyurl.com/3kw99dfc>



Differential Power Analysis (DPA) attack
on AES128 to recover private key

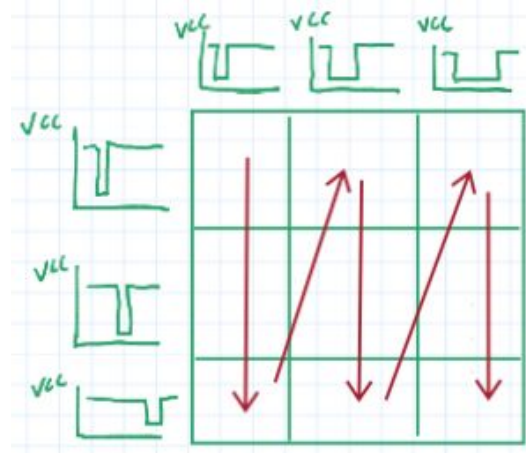
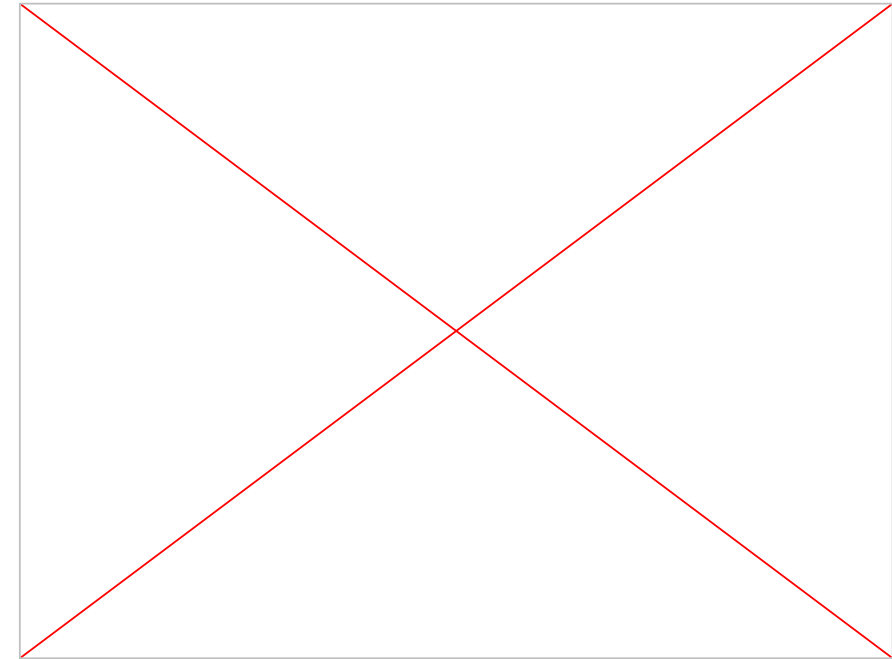


```
#define DEFAULT_KEY 0x2b,0x7e,0x15,0x16,0x28,
```



Voltage fault injection attack to bypass
UART password

```
for(cnt = 0; cnt < 5; cnt++){  
    if (pw[cnt] != passwd[cnt]){  
        passok = 0;  
    }  
}
```



[https://tinyurl
.com/3kw99dfc](https://tinyurl.com/3kw99dfc)



1

Introductions

Nathan Jones



DigiKey



https://
com/

**HARVEY
MUDD
COLLEGE**

EMBEDDED
SYSTEMS
SUMMIT

**NC STATE
UNIVERSITY**



Selected publications:

- "Inside the Voja4" (Hackaday Supercon)
- "OOP in C" (Embedded Online Conference)
- "You Don't Need an RTOS" (EmbeddedRelated.com)
- "Message Passing with Pointers" (Maker.io)

A fundamental problem

<https://tinyurl.com/3kw99dfc>



<https://tinyurl.com/65sk6fum>

The *only* solution I've come up with

<https://tinyurl.com/3kw99dfc>



– Ubuntu: –



<https://www.creativefabrica.com/product/ubuntu-i-am-because-we-are/>

A paradigm shift

<https://tinyurl.com/3kw99dfc>



- My goal today *isn't* to learn the most for myself, but to help *others* learn the most for themselves.
- Help can come in many forms
 - "Look at this cool thing I did! Let me teach it to you."
 - "Hey, you just did a cool thing! [or, Help, I don't get this!] Show me so that I can learn along with you."

–Ubuntu:–



<https://www.creativefabrica.com/product/ubuntu-i-am-because-we-are/>

Your turn!

<https://tinyurl.com/3kw99dfc>



- Introduce yourself to the people around you
- Ask:
 - What did you have for breakfast today?
 - What do you want to learn today?
 - What's the meaning of life?
- Also, **install the necessary software**
 - Visit chipwhisperer.readthedocs.io
 - Follow the installation instructions for
 - your OS
 - or just use VirtualBox (will require Firefox/Chrome)
 - Confirm setup using "ChipWhisperer Setup Test"

–Ubuntu:–



<https://www.creativefabrica.com/product/ubuntu-i-am-because-we-are/>

<https://tinyurl.com/3kw99dfc>



2

How bad is it?

Every device needs HW security

<https://tinyurl.com/3kw99dfc>



<https://tinyurl.com/4v5nmrf6>



<https://tinyurl.com/bdew33pb>

Threat capabilities

<https://tinyurl.com/3kw99dfc>

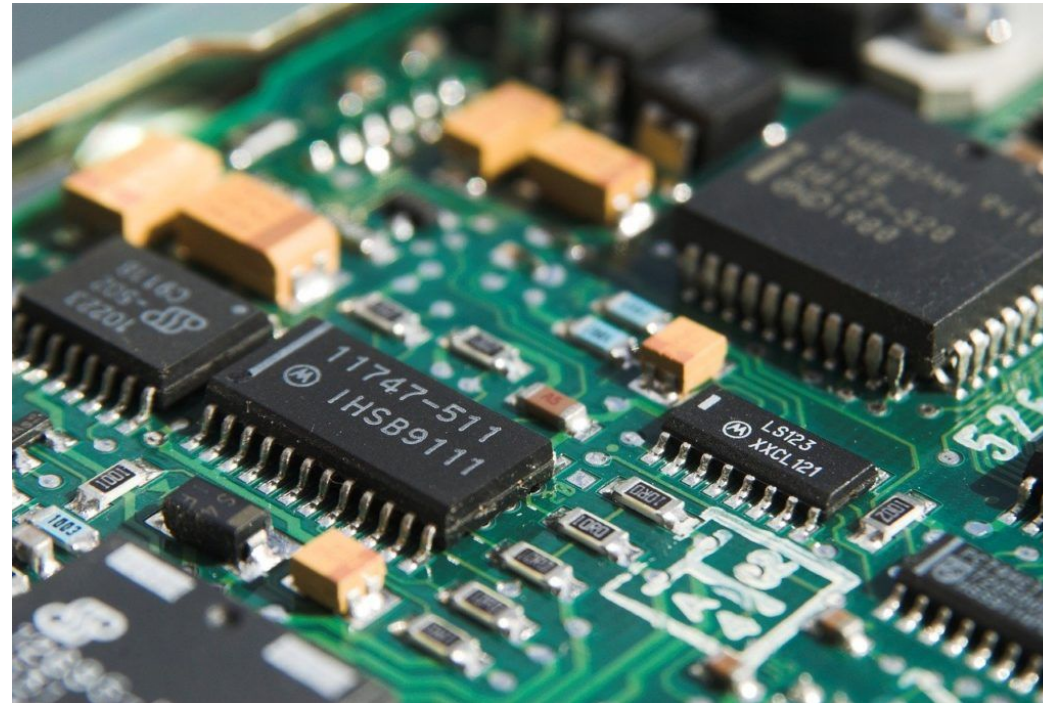


Step 1: Research and reverse engineering

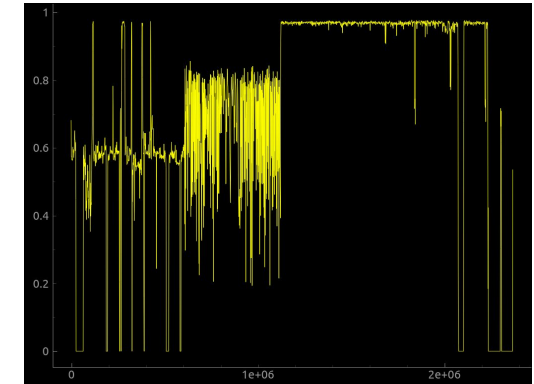
- Product manuals
- Schematics, FW
- Repair tutorials
- Patent filings
- FCC filings
- Datasheets
- (Un)Labelled ports



https://cdn.kthlaw.com/media/public/content/images/fcc_id.png



https://www.autodesk.com/products/fusion-360/blog/wp-content/uploads/2021/08/cyber-2377718_1280-1024x686.jpg



<https://sergioprado.blog/images/20200206-binwalk-entropy.png>



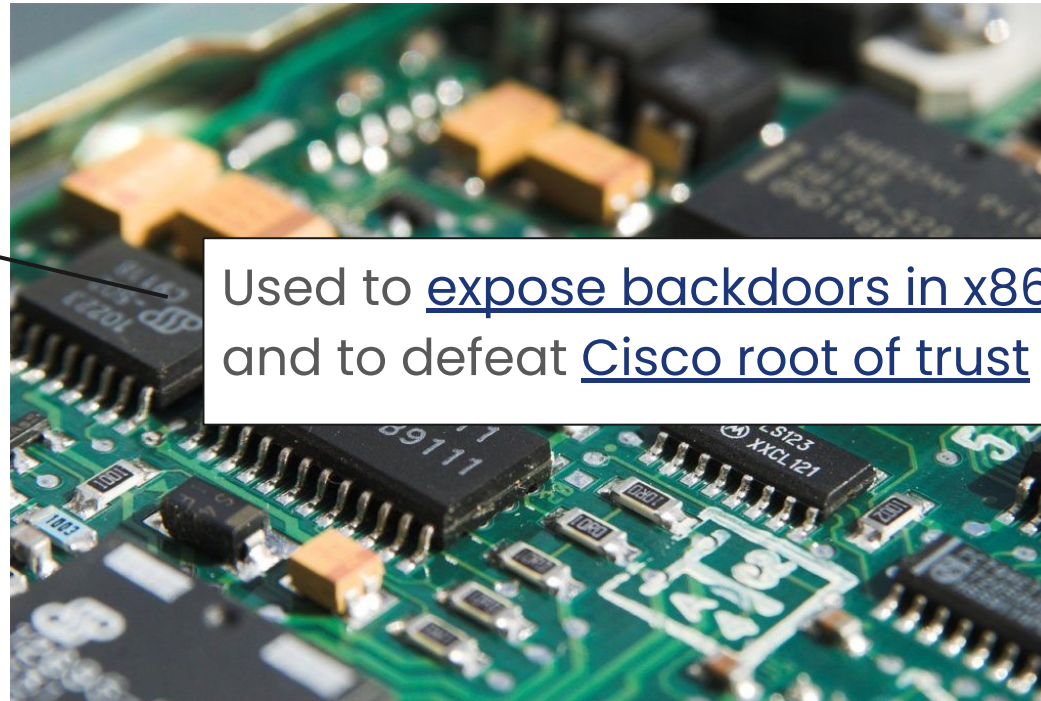
Threat capabilities

<https://tinyurl.com/3kw99dfc>



Step 1: Research and reverse engineering

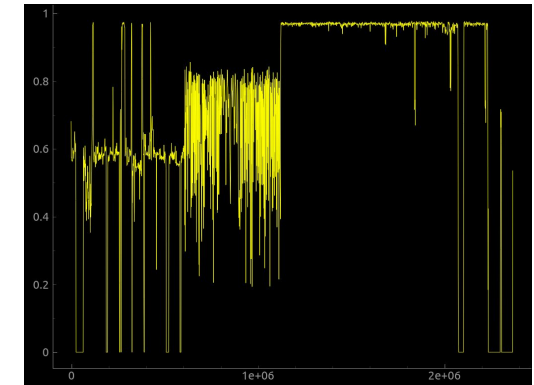
- Product manuals
- Schematics, FW
- Repair tutorials
- **Patent filings**
- FCC filings
- Datasheets
- (Un)Labelled ports



Used to expose backdoors in x86 CPUs (2018)
and to defeat Cisco root of trust (2020)!!



https://cdn.kthlaw.com/media/public/content/images/fcc_id.png



radio.blog/images/walk-entropy.png

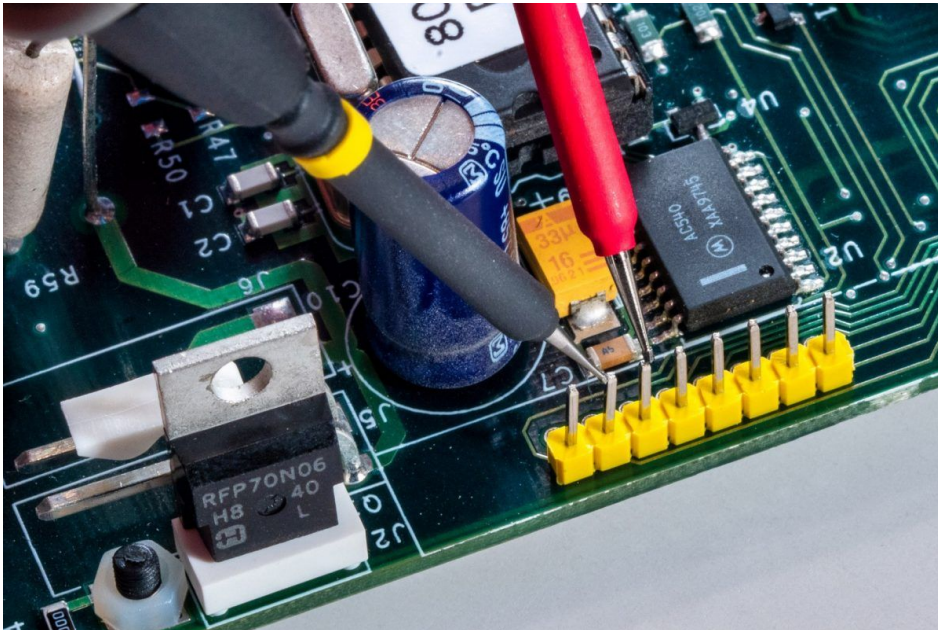


https://www.autodesk.com/products/fusion-360/blog/wp-content/uploads/2021/08/cyber-2377718_1280-1024x686.jpg

Threat capabilities

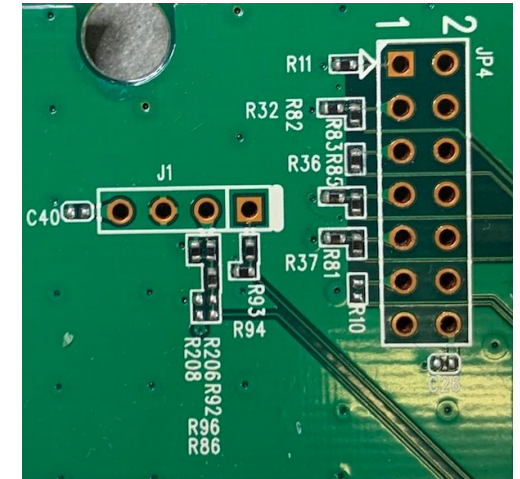
Step 2: Snooping

- JTAG
- UART
- SPI
- Peripherals



<https://www.tag-connect.com/wp-content/uploads/2018/03/no-slip-probes-2-1200x800.jpg>
www.embeddedsummit.com

<https://tinyurl.com/3kw99dfc>



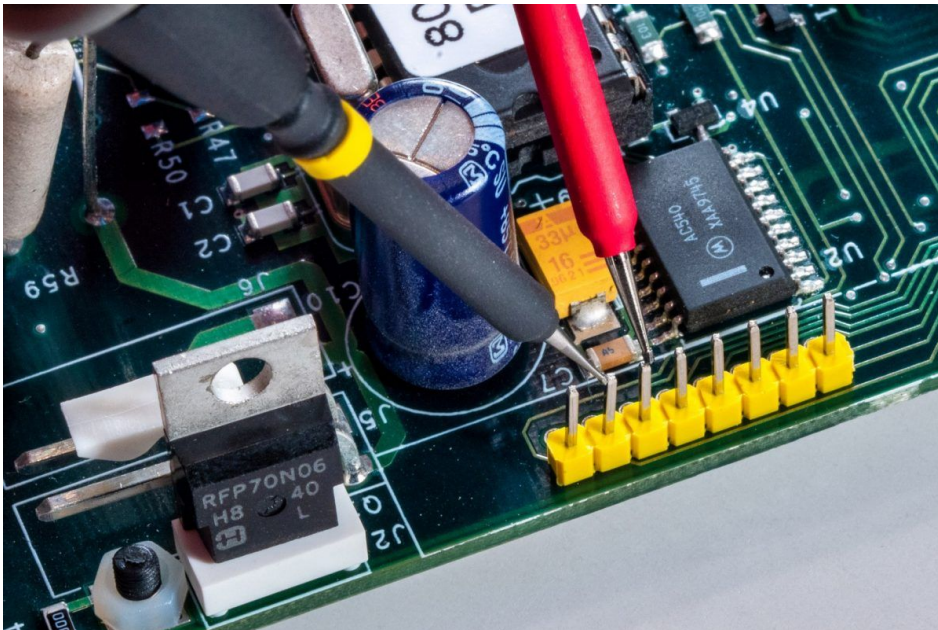
<https://riverloopsecurity.com/img/blog/hw-101-jtag/header.jpg>



Threat capabilities

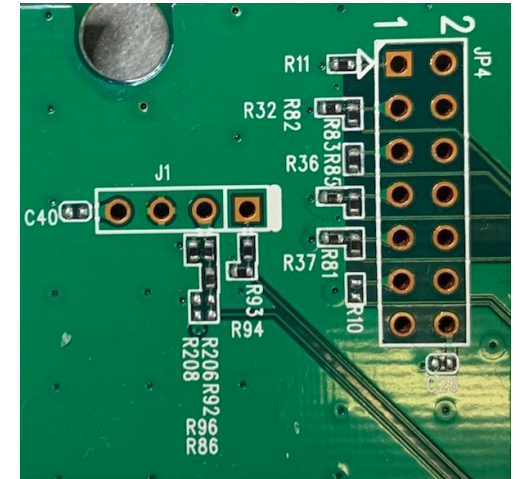
Step 2: Snooping

- JTAG ← Unlocked? Susceptible to fault injection?
- UART
- SPI
- Peripherals



<https://www.tag-connect.com/wp-content/uploads/2018/03/no-slip-probes-2-1200x800.jpg>
www.embeddedsummit.com

<https://tinyurl.com/3kw99dfc>



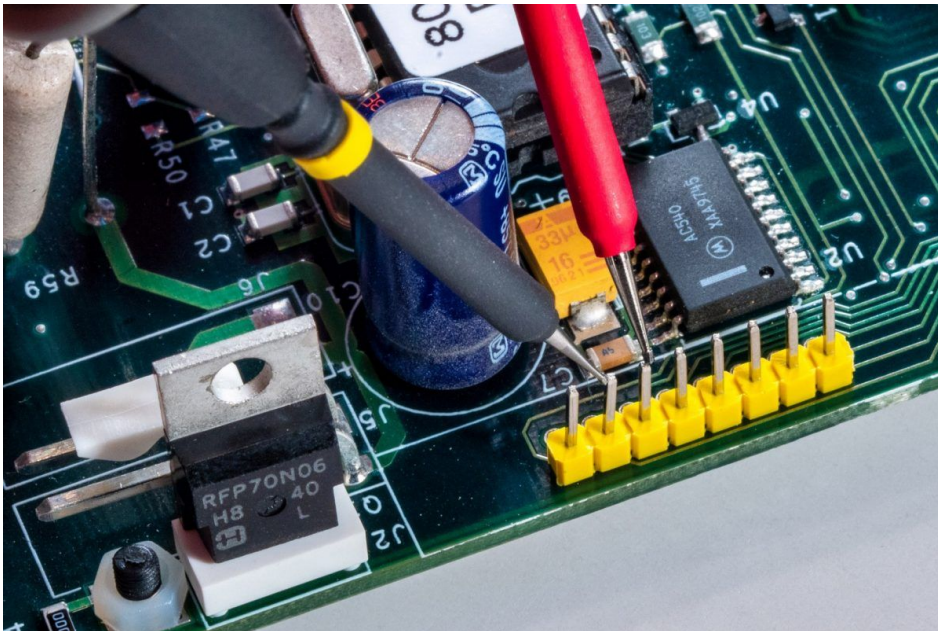
<https://riverloopsecurity.com/img/blog/hw-101-jtag/header.jpg>



Threat capabilities

Step 2: Snooping

- JTAG ← Unlocked? Susceptible to fault injection?
- UART ← Does execution time leak the correct pw?
- SPI
- Peripherals

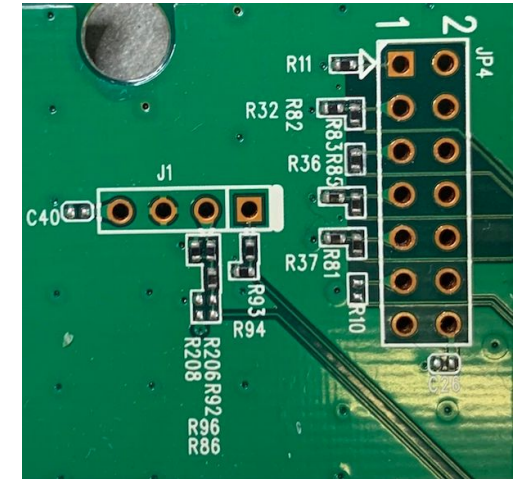


<https://www.tag-connect.com/wp-content/uploads/2018/03/no-slip-probes-2-1200x800.jpg>
www.embeddedsummit.com

<https://tinyurl.com/3kw99dfc>



MBEDDED
SYSTEMS
SUMMIT



<https://riverloopsecurity.com/img/blog/hw-101-jtag/header.jpg>



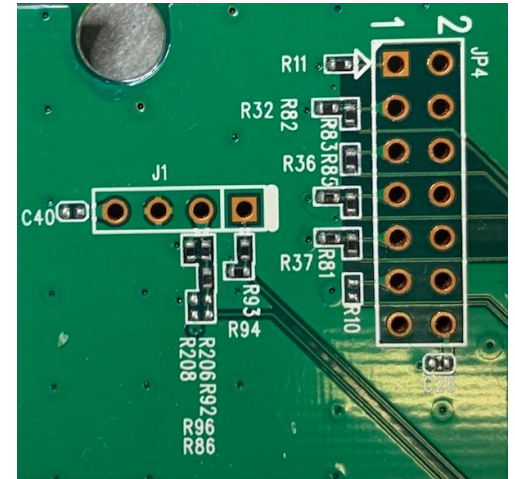
Threat capabilities

<https://tinyurl.com/3kw99dfc>

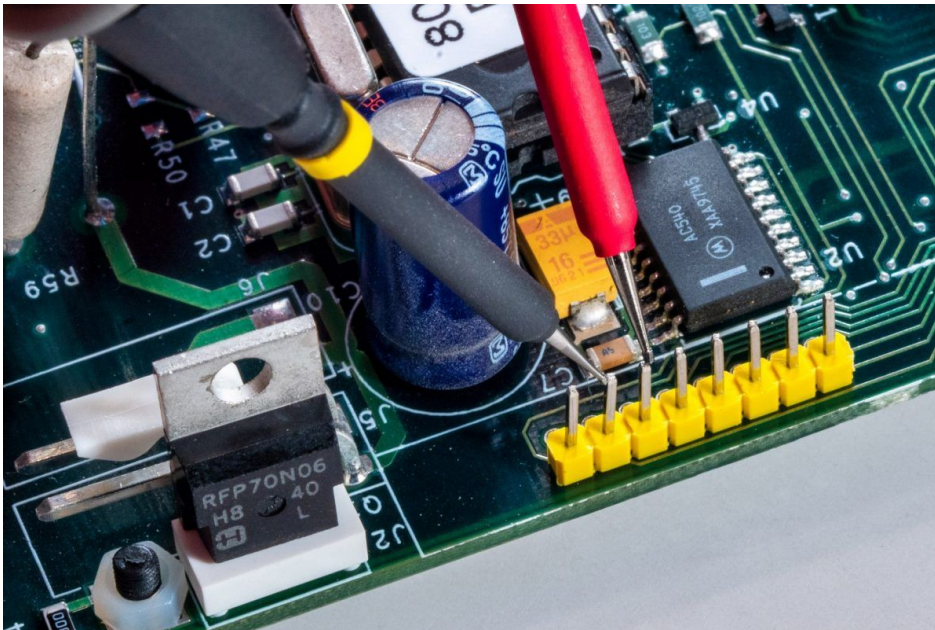


Step 2: Snooping

- JTAG ← Unlocked? Susceptible to fault injection?
- UART ← Does execution time leak the correct pw?
- SPI
- Peripherals ← Can we spoof or fault these signals for an atk?



<https://riverloopsecurity.com/img/blog/hw-101-jtag/header.jpg>



<https://www.tag-connect.com/wp-content/uploads/2018/03/no-slip-probes-2-1200x800.jpg>
www.embeddedsummit.com

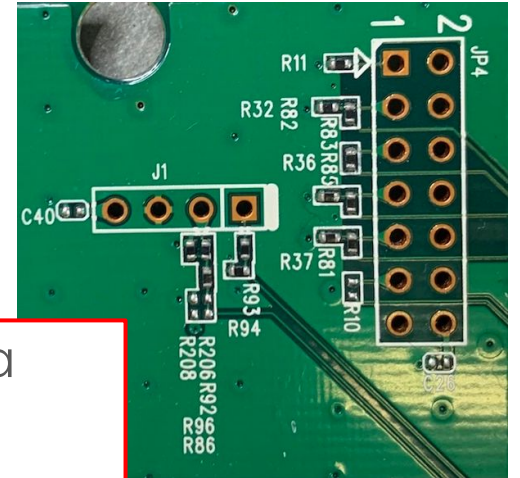


Threat capabilities

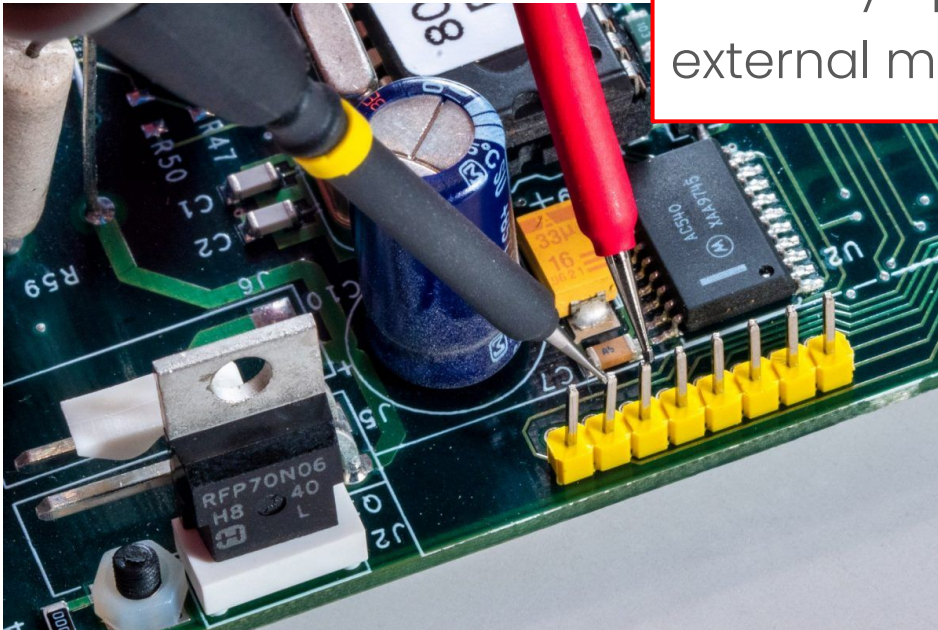
Step 2: Snooping

- JTAG ← Unlocked? Susceptible to fault injection?
- UART ← Does execution time leak the correct pw?
- SPI ← Playstation 3 was hacked in 2010 by glitching a
- Peripherals ← memory operation between the MCU and the

Playstation 3 was hacked in 2010 by glitching a memory operation between the MCU and external memory that resulted in a memory leak!



hw-101-jtag/header.jpg

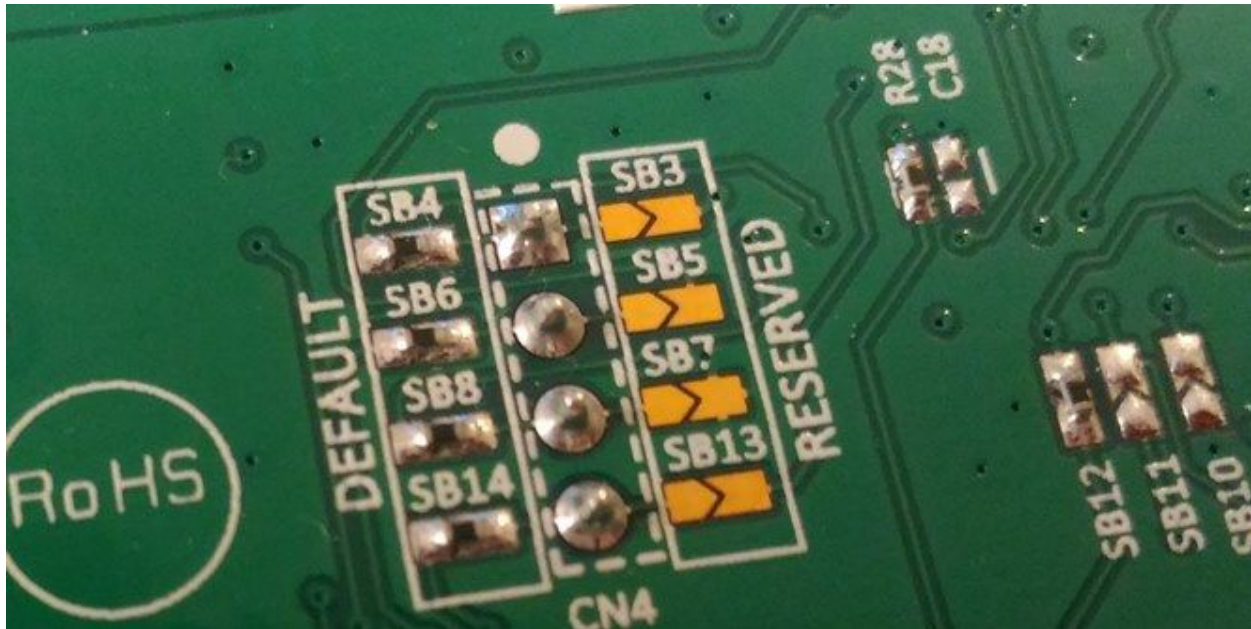


<https://www.tag-connect.com/wp-content/uploads/2018/03/no-slip-probes-2-1200x800.jpg>
www.embeddedsummit.com

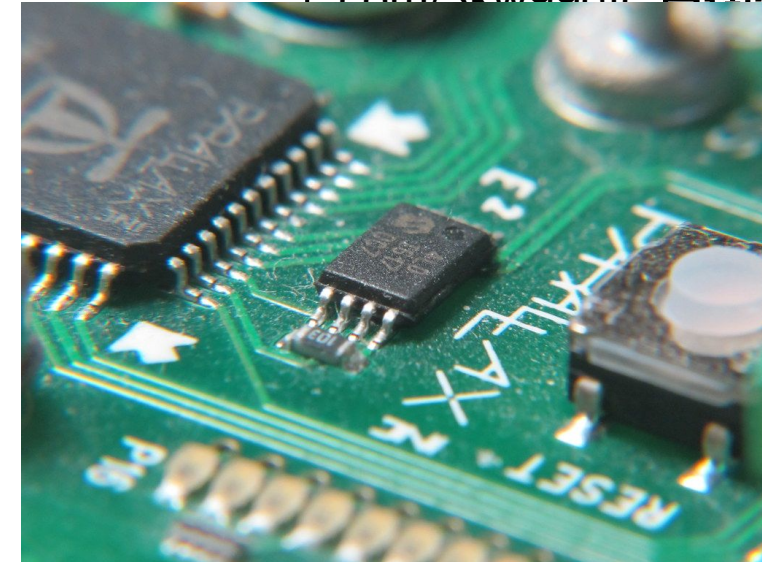
Threat capabilities

Step 3: Alter the PCB

- Configuration resistors/jumpers
- Read from/write to external memory



<https://i.sstatic.net/iis7l.jpg>



https://www.totalphase.com/media/blog/2020/06/4653066327_e29062b4b7_b.jpg



Threat capabilities

Step 4: Side-channel attacks

<https://tinyurl.com/3kw99dfc>

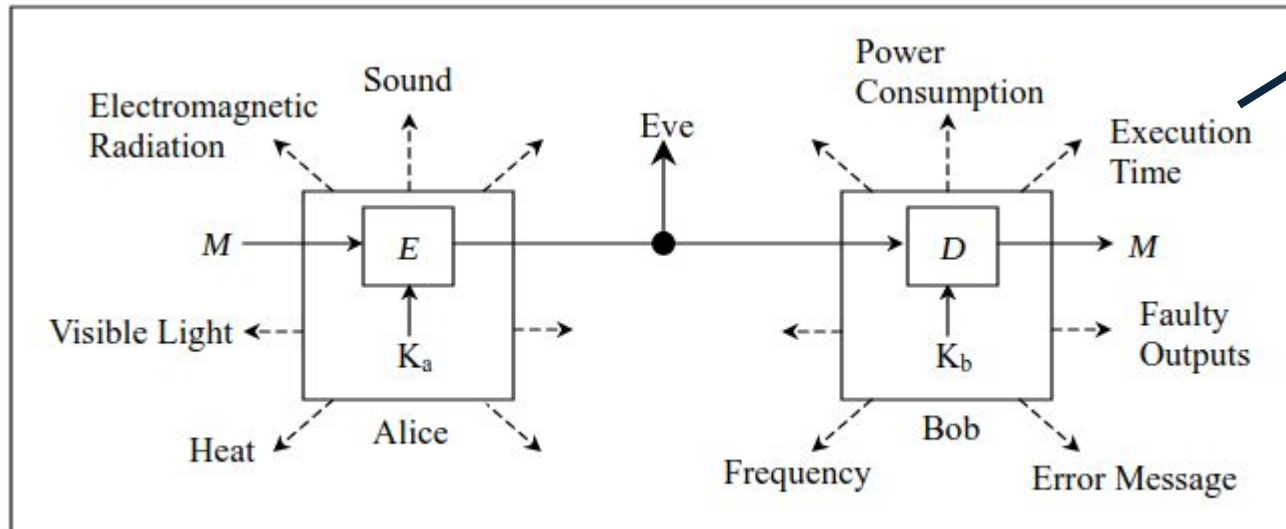
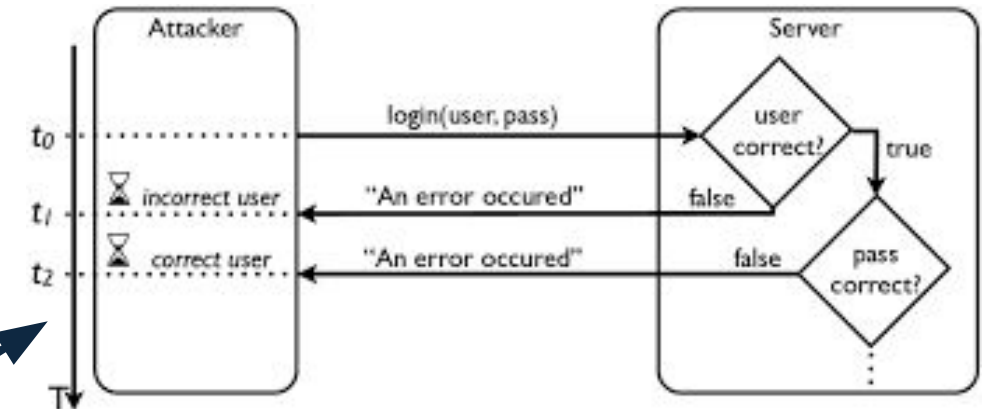


Figure 2: The cryptographic model including side-channel

Zhou, Yongbin & Feng, DengGuo. (2005). Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing.. IACR Cryptology ePrint Archive. 2005. 388. <https://eprint.iacr.org/2005/388.pdf>



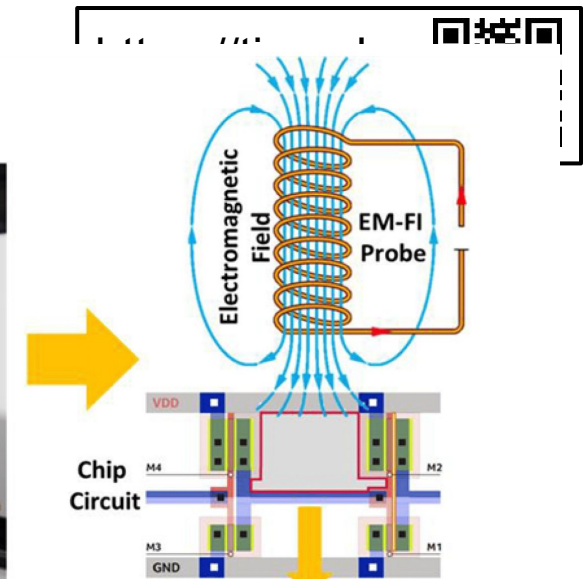
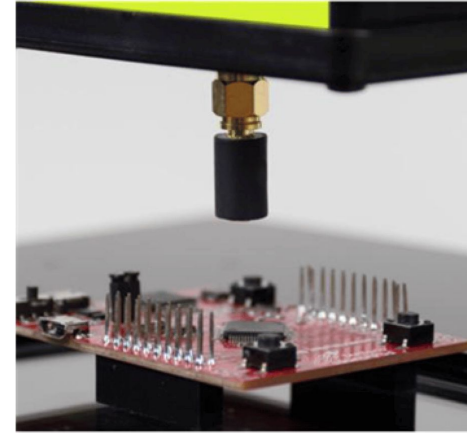
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTq3YSO_cL3ENPKtwCp76ygZUaLMUKaDqirw&s



Threat capabilities

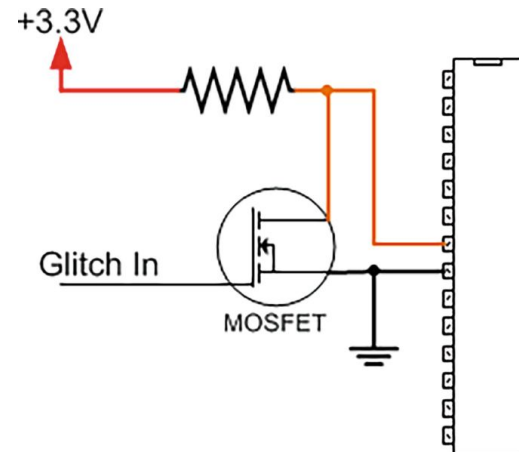
Step 5: Fault injection attacks

Fault Injection Attacks



Eddy Current → Transistor Error

https://pub.mdpi-res.com/applsci/applsci-10-03849/article_deploy/html/images/applsci-10-03849-g001.png?1592831188



<https://tinyurl.com/5cy3pt7x>



<https://www.intel.com/content/dam/www/central-libraries/us/en/images/2025-07/security-daniel-blog-figure2.png>

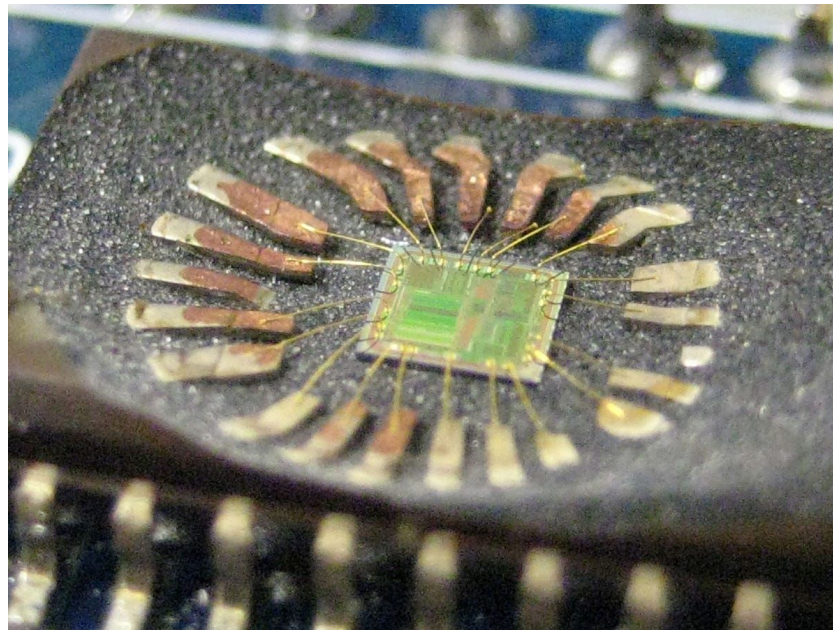
Threat capabilities

<https://tinyurl.com/3kw99dfc>

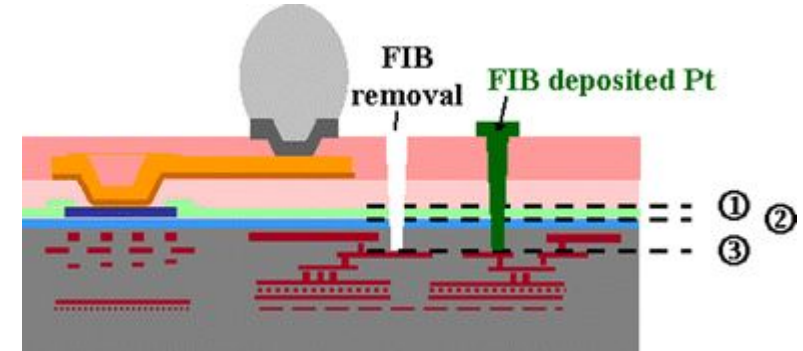


Step 6: Decapping (followed by more RE, SCAs, FIs)

- Imaging (microscope, SEM)
- Microprobing
- Optical emission analysis
- Optical FI
- FIB editing



https://miro.medium.com/v2/resize:fit:1200/1*RU_BdQQ5xEUrWMaqUTbEzA.jpeg



<https://tinyurl.com/2fvm7cyr>



All's not lost!

<https://tinyurl.com/3kw99dfc>



Determine *what* secrets you're trying to protect, *who* could attack them, and *how*. Then employ countermeasures against those *specific* attacks.



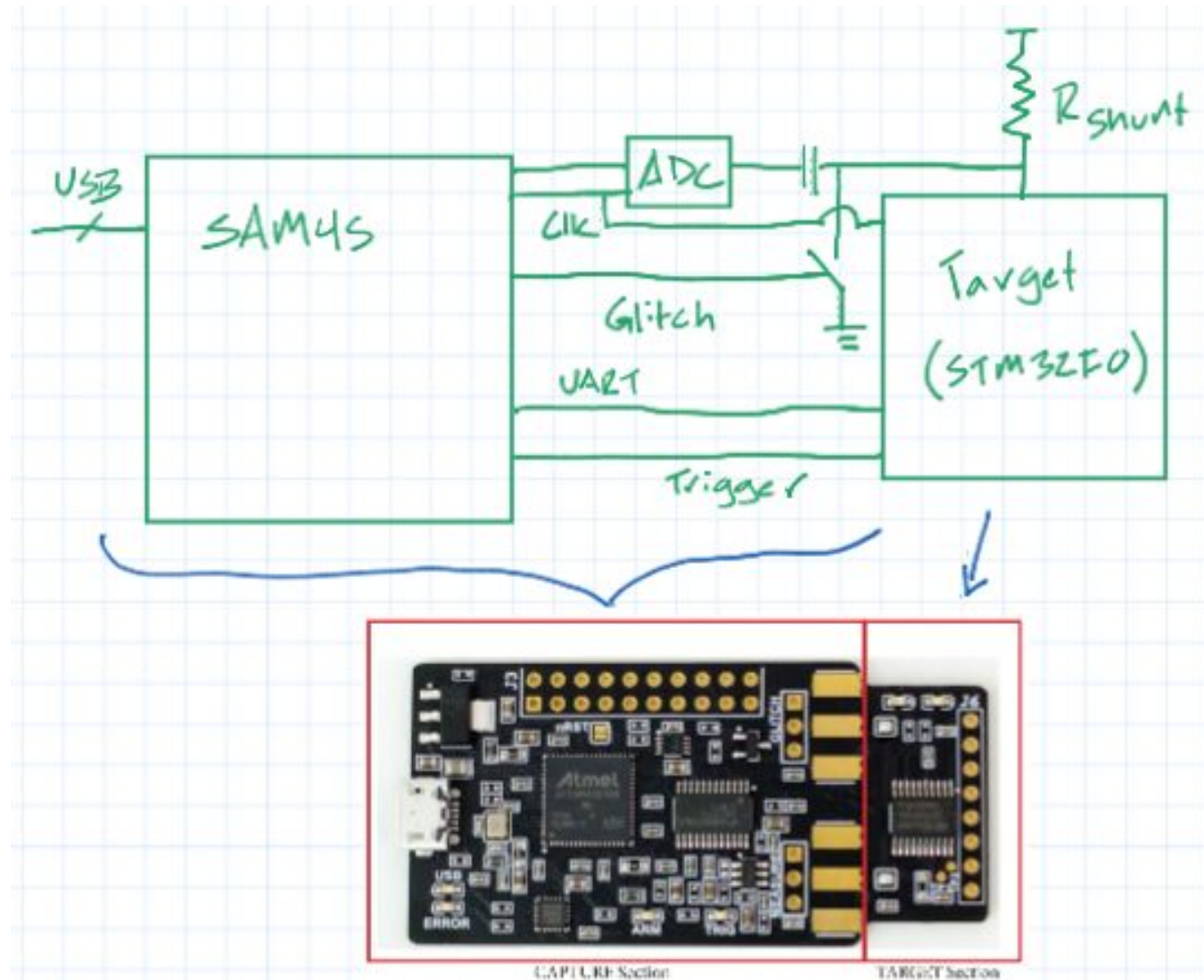


3

Introduction to the CW-Nano

CW-Nano

<https://tinyurl.com/3kw99dfc>



Cheatsheet

<https://tinyurl.com/3kw99dfc>



```
int main(void)
{
    simpleserial_addcmd('c', 0, myCmd)
    while(1) simpleserial_get();
}

uint8_t myCmd(uint8_t* pt, uint8_t len)
{
    trigger_high();
    // Do stuff
    return 0x00;
}
```

```
# Define variables
SCOPE_TYPE = 'CWNANO'
PLATFORM = 'CWNANO'
CRYPTO_TARGET='TINYAES128C'
SS_VER = 'SS_VER_1_1'
#or SS_VER = 'SS_VER_2_1'

# Import CW lib, create scope/target objects
%run "../../Setup_Scripts/Setup_Generic.ipynb"

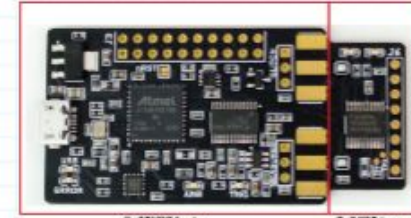
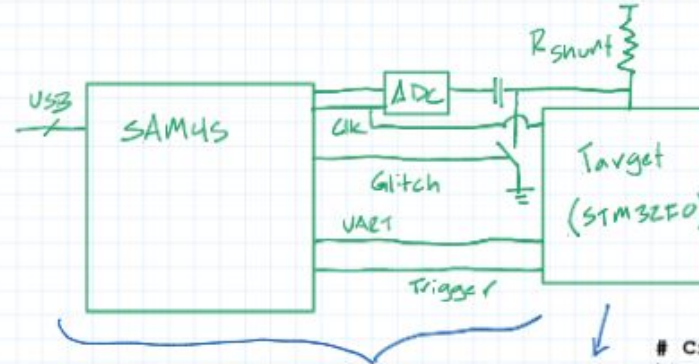
# Build and flash a project for the STM32F0
%%bash -s "$PLATFORM" "$CRYPTO_TARGET" "$SS_VER"
cd ../../../../firmware/mcu/{PROJECT}
make PLATFORM=$1 CRYPTO_TARGET=$2 SS_VER=$3 -j

cw.program_target(scope, prog, "/path/to/project-{}.hex".format(PLATFORM))

# Get power traces
scope.arm()
scope.capture() # Check for (non-zero) error
# Trigger operation of interest on target
trace = scope.get_last_trace()

# Glitch
scope.glitch.repeat = (length of glitch)
scope.glitch.ext_offset = (delay)
scope.arm()
# Trigger operation of interest on target
scope.capture() # Check for (non-zero) error

# Disconnect from HW
scope.dis()
target.dis()
```



```
# Create random AES key and 16-byte "text"
ktp = cw.ktp.Basic()
key, text = ktp.next()

# Set AES key
target.set_key(key)

# Send data to target
target.simpleserial_write(cmd, payload)
Ex:
# Get SimpleSerial version number
target.simpleserial_write('v', bytearray([]))
# Send plaintext for encryption
target.simpleserial_write('p', text)

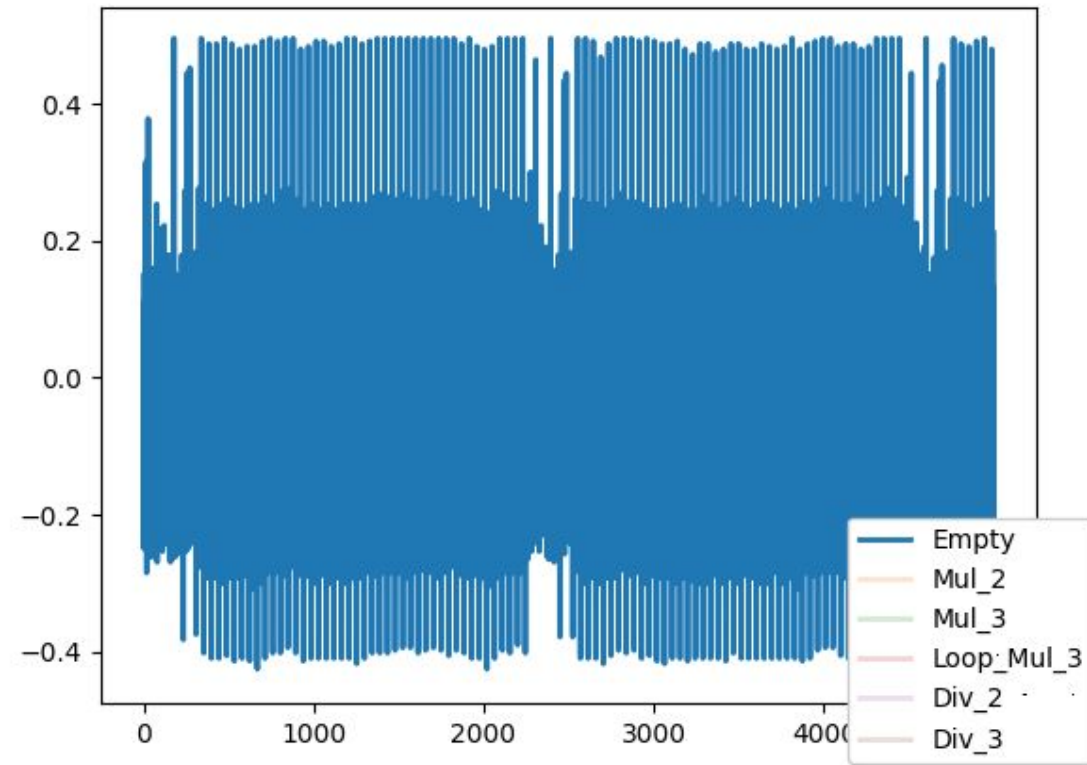
# Receive data from target
# Any message
target.simpleserial_read()
# 'r' message (expecting 16 bytes)
target.simpleserial_read('r', 16)
```

Your turn!

<https://tinyurl.com/3kw99dfc>



Follow SCA101/Lab 2_1A to get a single power trace.





4

Attacking a target

Side-channel analyses

MCUs leak information in many different ways

<https://tinyurl.com/3kw99dfc>

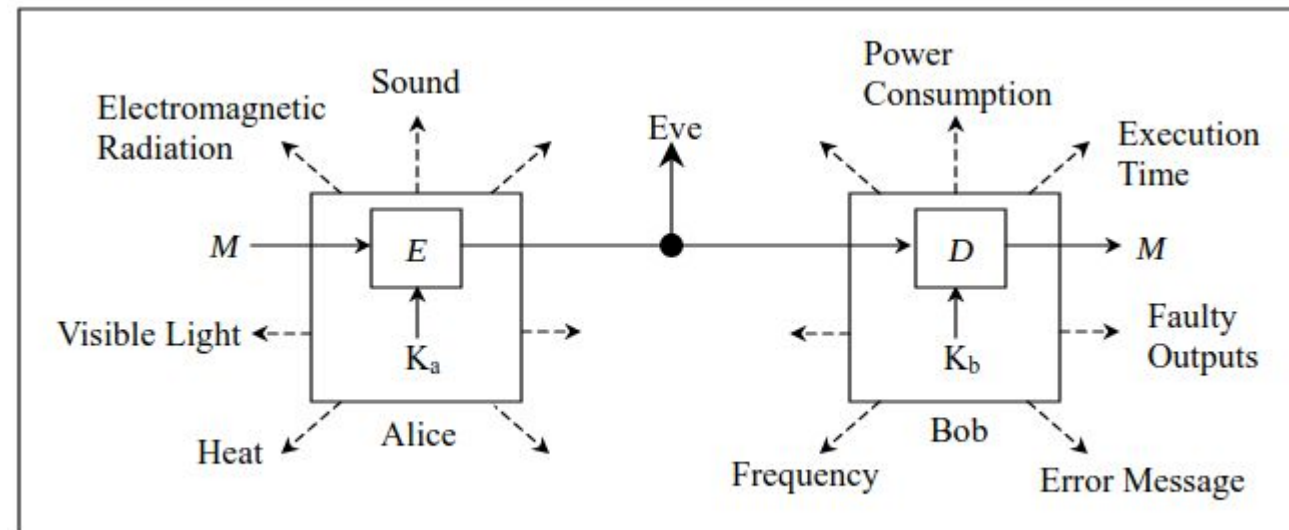
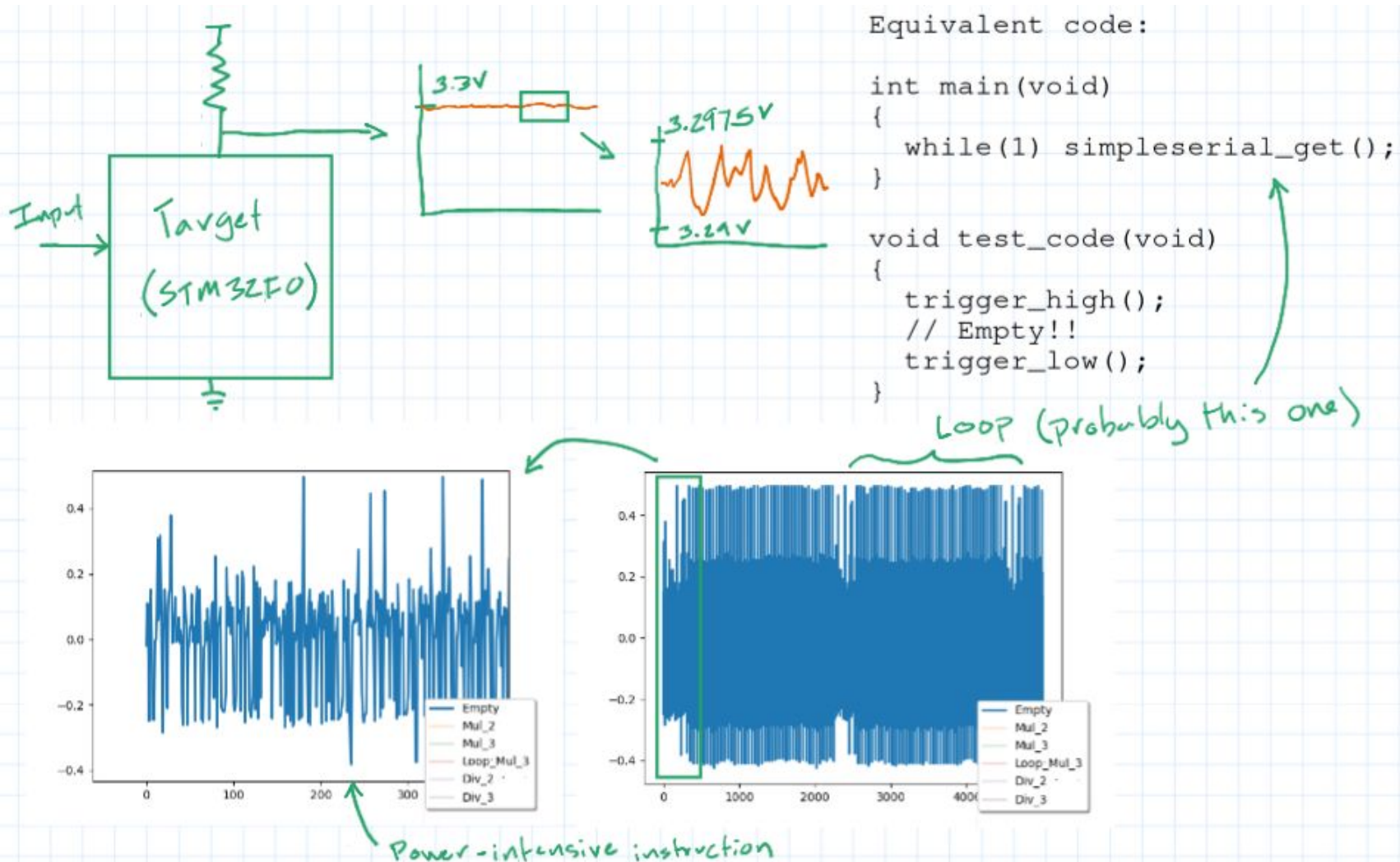


Figure 2: The cryptographic model including side-channel

Zhou, Yongbin & Feng, DengGuo. (2005). Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing.. IACR Cryptology ePrint Archive. 2005. 388. <https://eprint.iacr.org/2005/388.pdf>

Power traces leak information about what an MCU is doing (SCA101/Lab 2_1A)

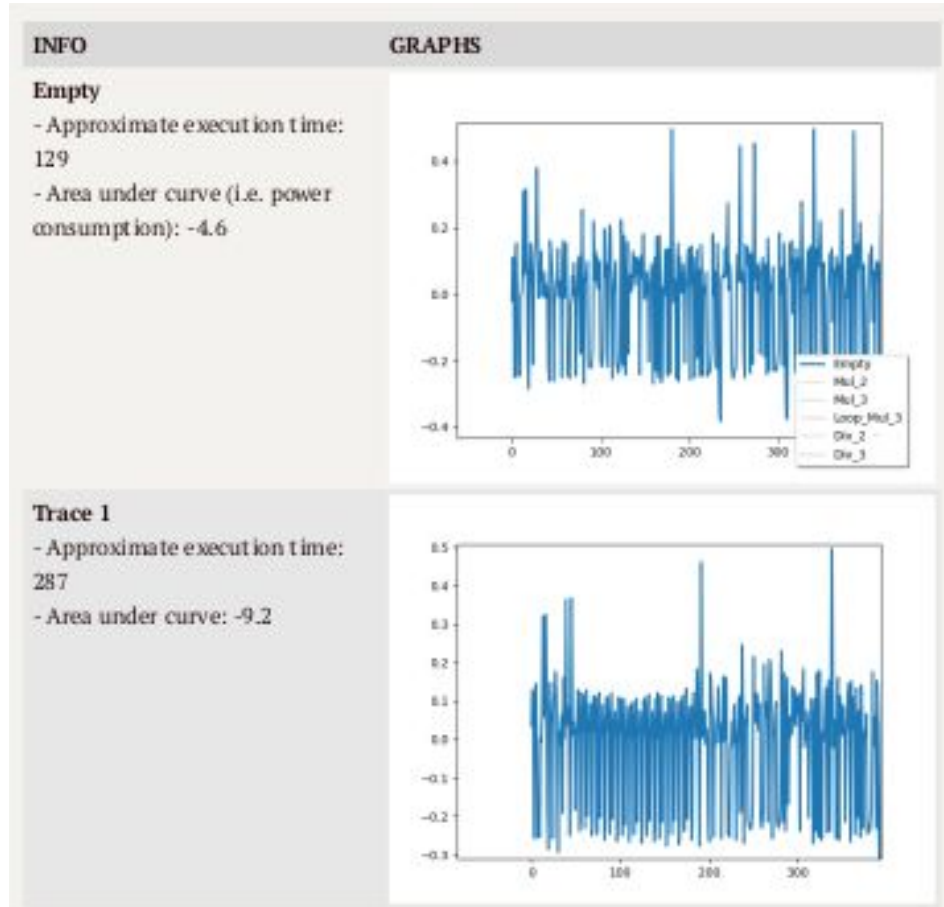


Your turn!

<https://tinyurl.com/3kw99dfc>



Match each trace to its code snippet!



?

Multiplication by 2

```
void test_code(void)
{
    trigger_high();
    volatile long int A = 0xE6A9;
    A *= 2;
    // 19 more times
}
```

Multiplication by 3 (loop)

```
void test_code(void)
{
    trigger_high();
    volatile long int A = 0xE6A9;
    for(volatile int i = 0; i < 20; i++) A *= 3;
}
```

Division by 2

```
void test_code(void)
{
    trigger_high();
    volatile long int A = 0xE6A9;
    A /= 2;
    // 19 more times
}
```

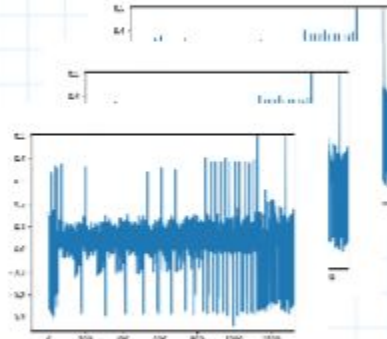
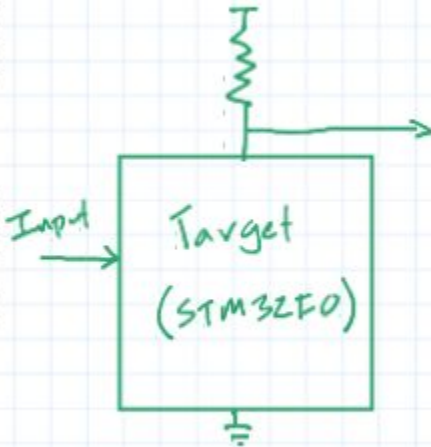
Power traces even leak information about how many bits are 1

<https://tinyurl.com/3kw99dfc>

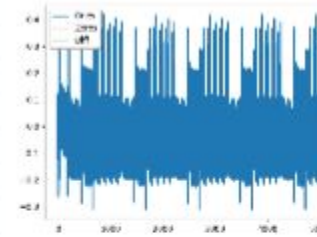


[0x34, 0xA0, ...]
[0x7D, 0x06, ...]
[0x0C, 0xF1, ...]
[0xB5, 0x82, ...]

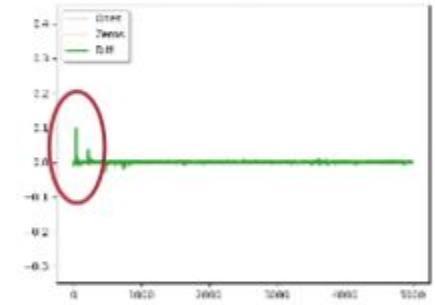
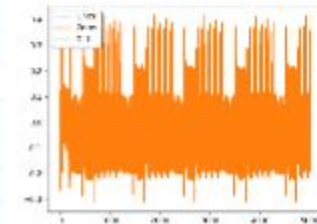
[0x00, 0xA0, ...]
[0xFF, 0x06, ...]
[0x00, 0xF1, ...]
[0xFF, 0x82, ...]



[0xFF, 0x06, ...]
[0xFF, 0x82, ...]

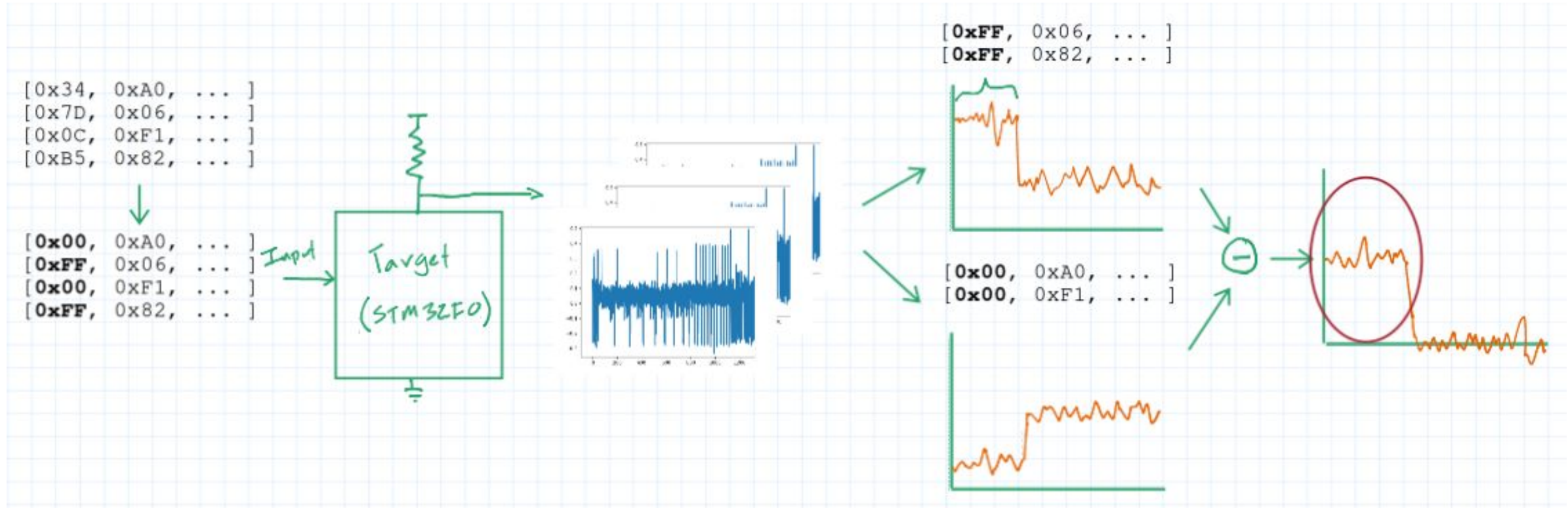


[0x00, 0xA0, ...]
[0x00, 0xF1, ...]



Power traces even leak information about how many bits are 1

<https://tinyurl.com/3kw99dfc>



Your turn!

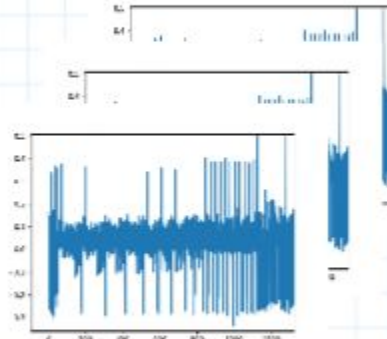
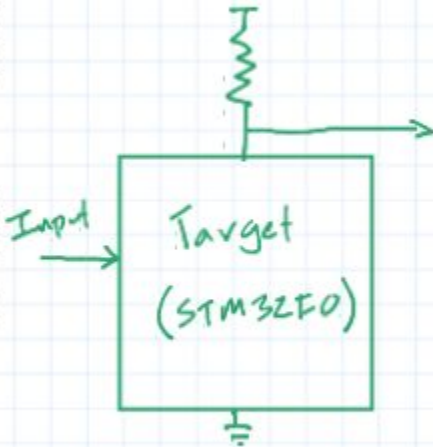
Complete SCA101/Lab 3_1

<https://tinyurl.com/3kw99dfc>

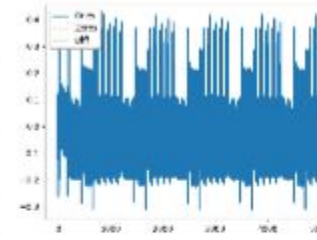


[0x34, 0xA0, ...]
[0x7D, 0x06, ...]
[0x0C, 0xF1, ...]
[0xB5, 0x82, ...]

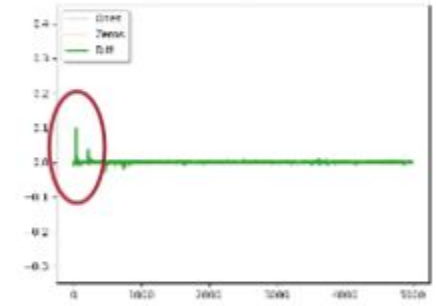
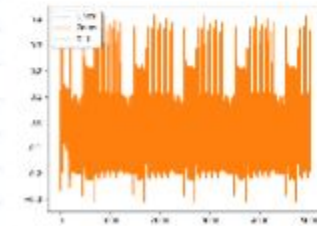
[0x00, 0xA0, ...]
[0xFF, 0x06, ...]
[0x00, 0xF1, ...]
[0xFF, 0x82, ...]



[0xFF, 0x06, ...]
[0xFF, 0x82, ...]

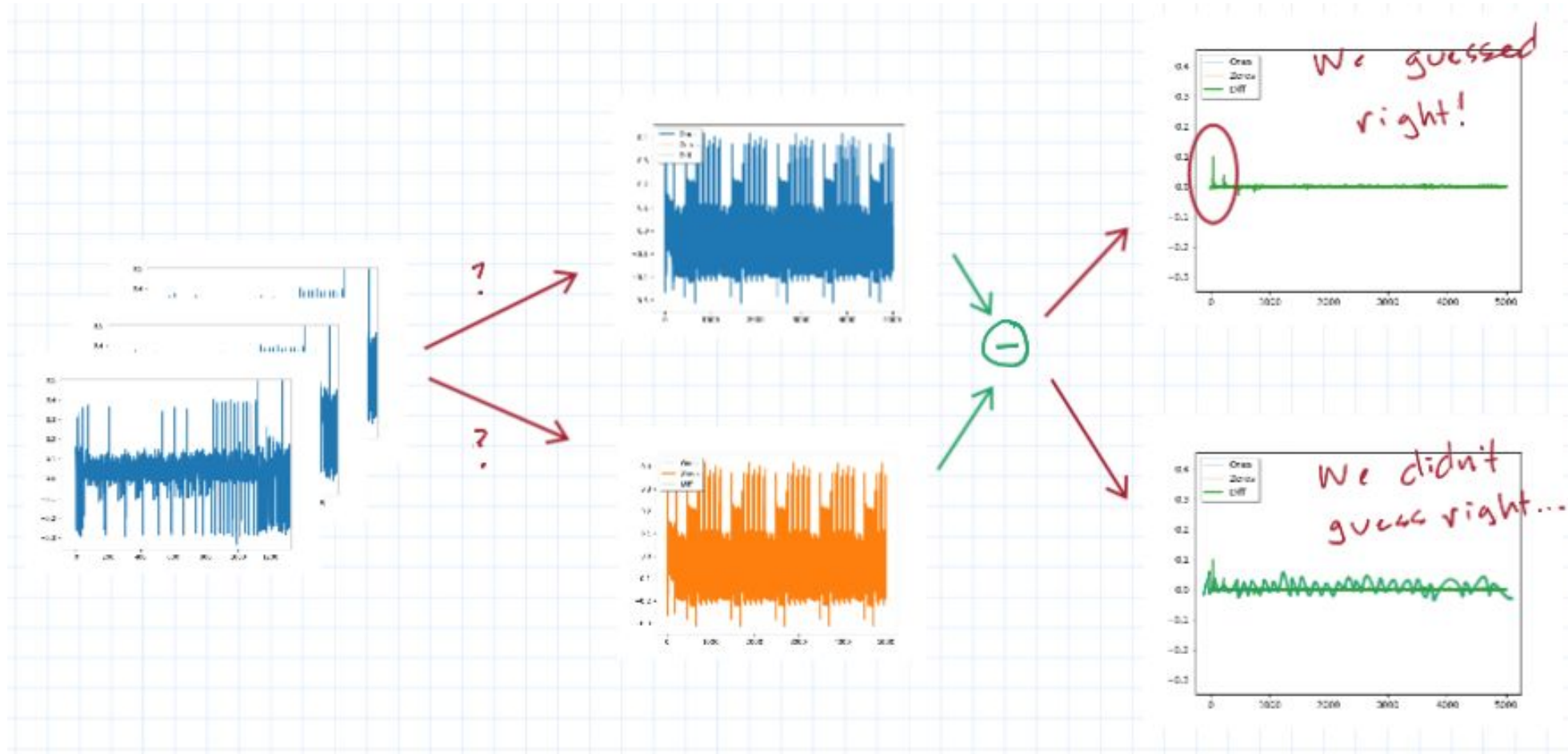


[0x00, 0xA0, ...]
[0x00, 0xF1, ...]



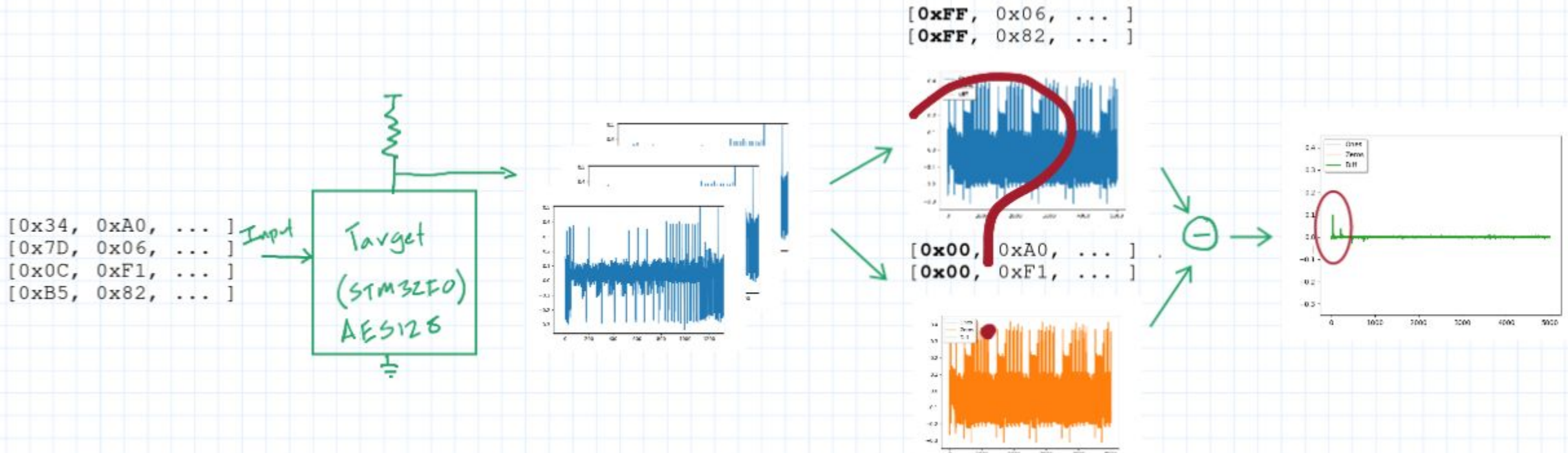
What if we didn't know which trace was which?

<https://tinyurl.com/3kw99dfc>



How can we use this to break AES128?

<https://tinyurl.com/3kw99dfc>



AES encryption

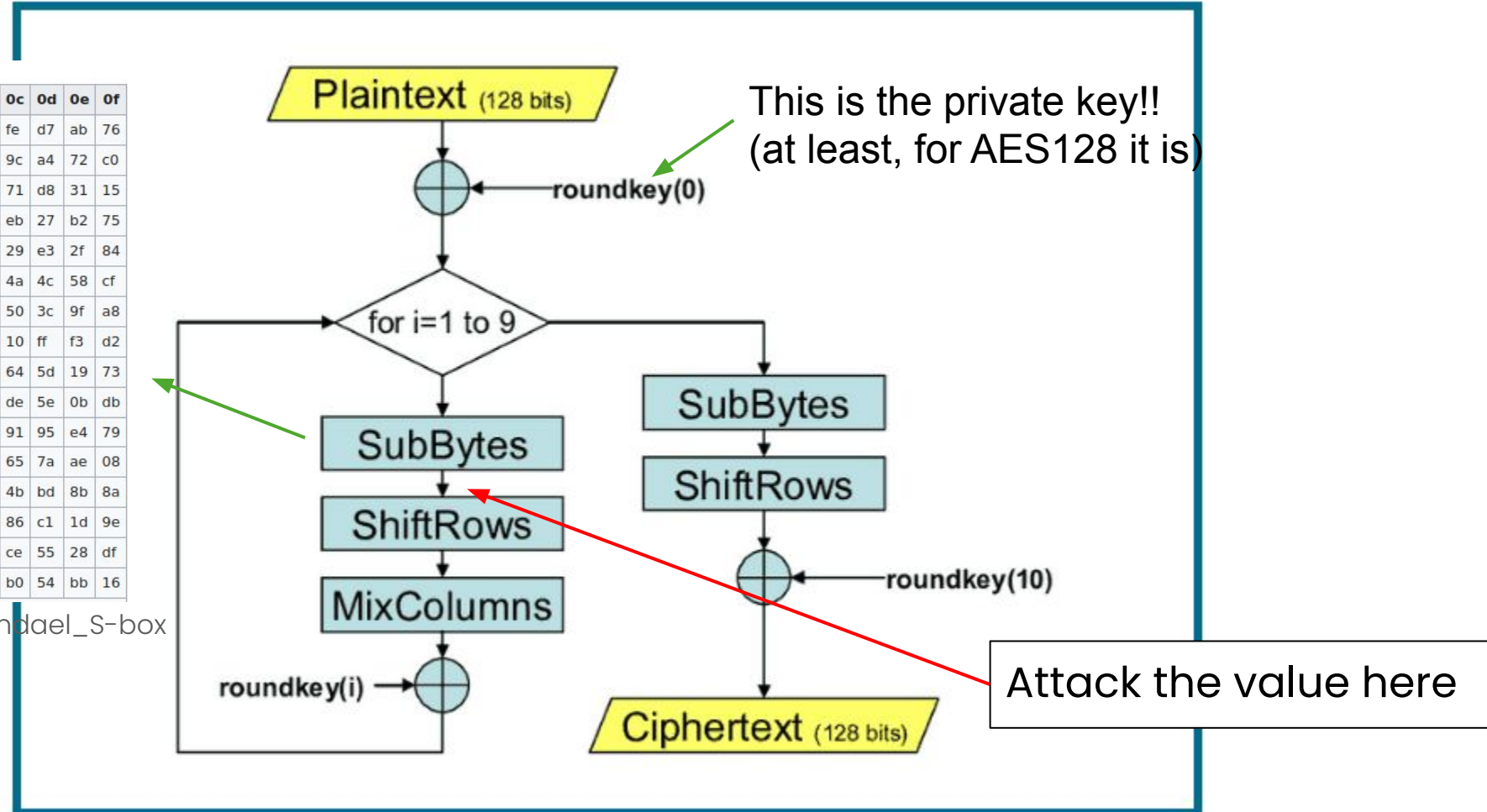
<https://tinyurl.com/3kw99dfc>



AES S-box

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

https://en.wikipedia.org/wiki/Rijndael_S-box



<https://privacycanada.net/app/uploads/2019/01/aes-algorithm-diagram.png>

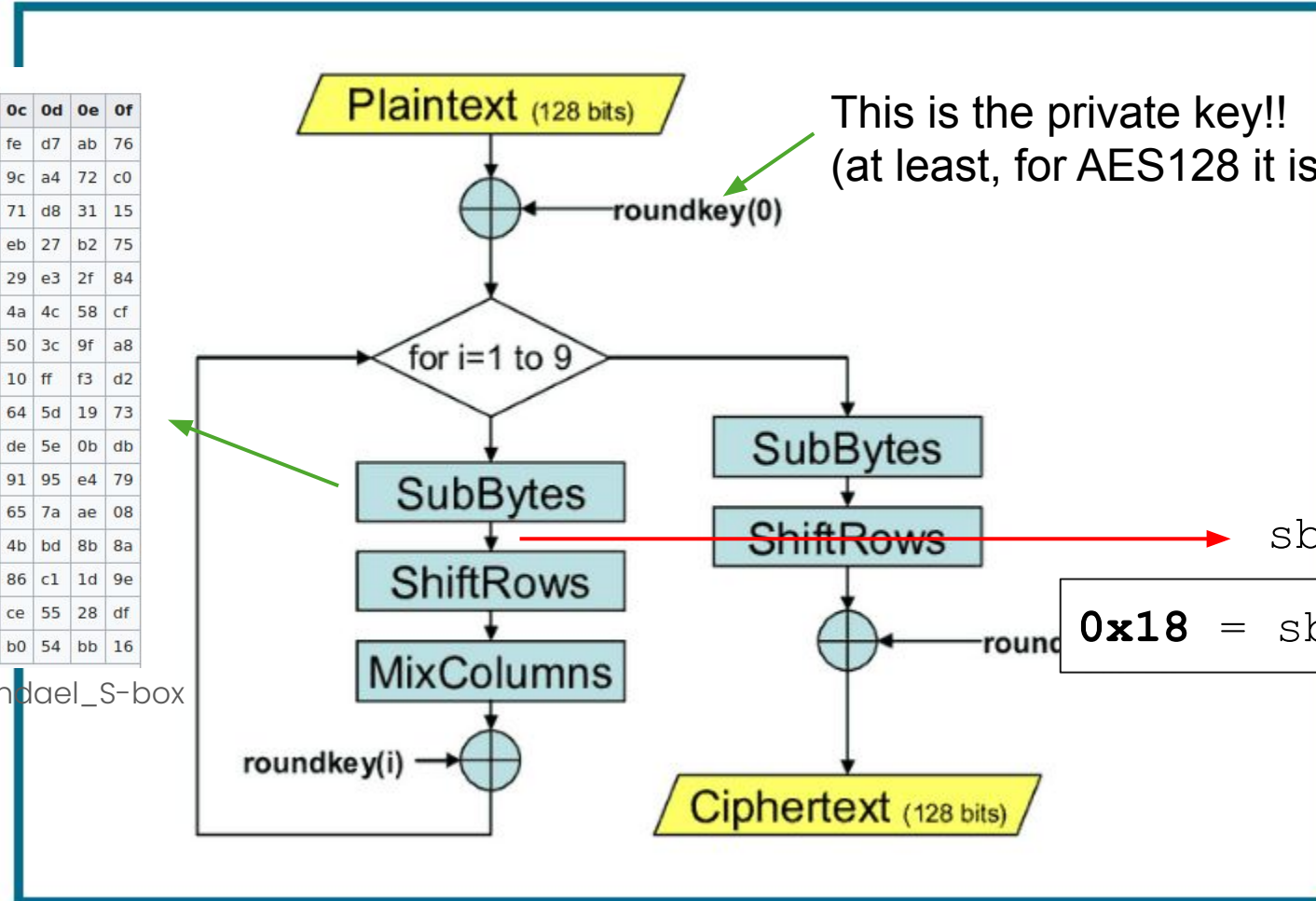
AES encryption

<https://tinyurl.com/3kw99dfc>



	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

https://en.wikipedia.org/wiki/Rijndael_S-box



<https://privacycanada.net/app/uploads/2019/01/aes-algorithm-diagram.png>

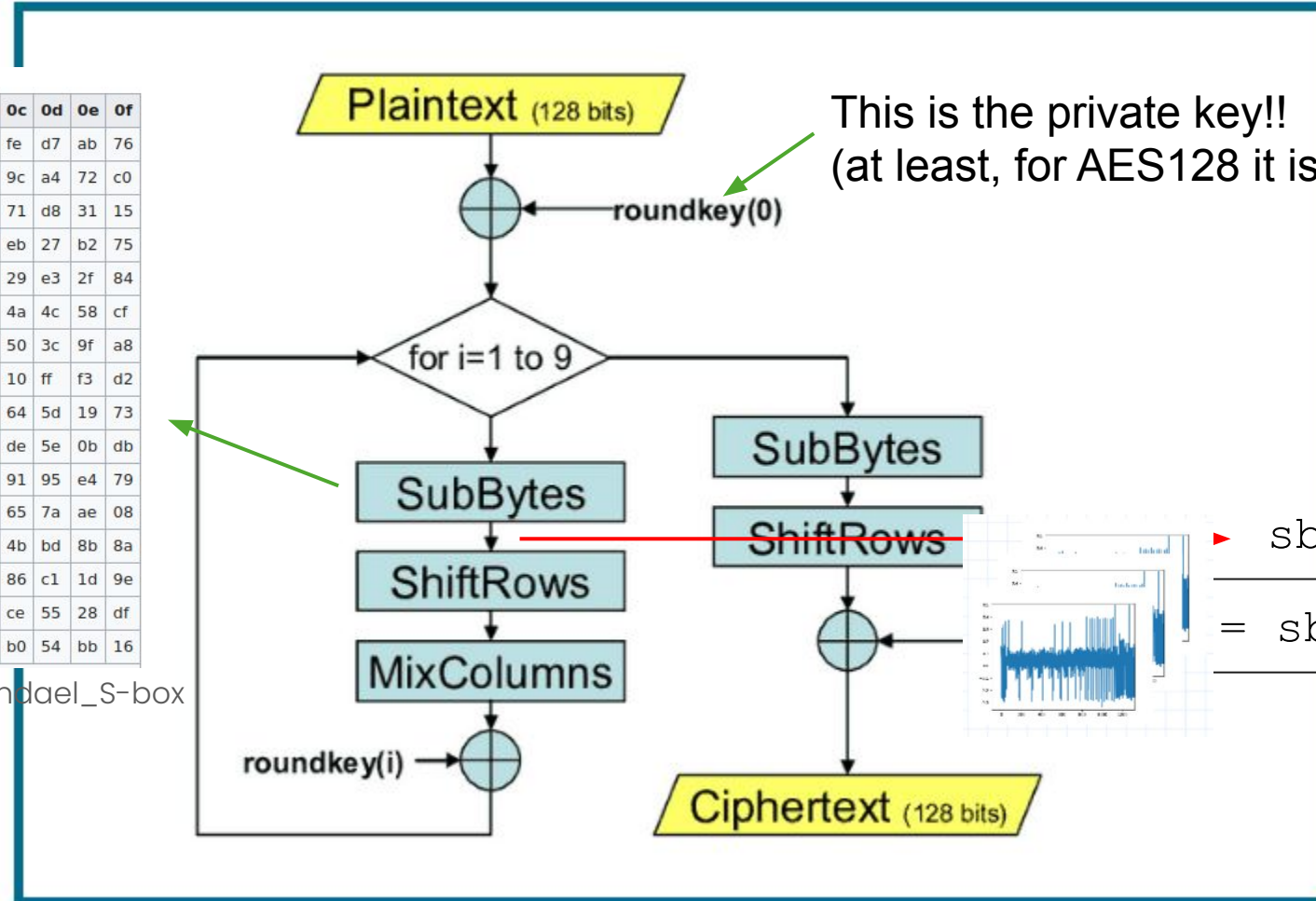
AES encryption

<https://tinyurl.com/3kw99dfc>



	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

https://en.wikipedia.org/wiki/Rijndael_S-box



This is the private key!!
(at least, for AES128 it is)

$$\text{sbox}[\text{pt}[n] \wedge \text{key}[n]] \\ = \text{sbox}[\mathbf{0x34} \wedge \text{key}[0]]$$

???

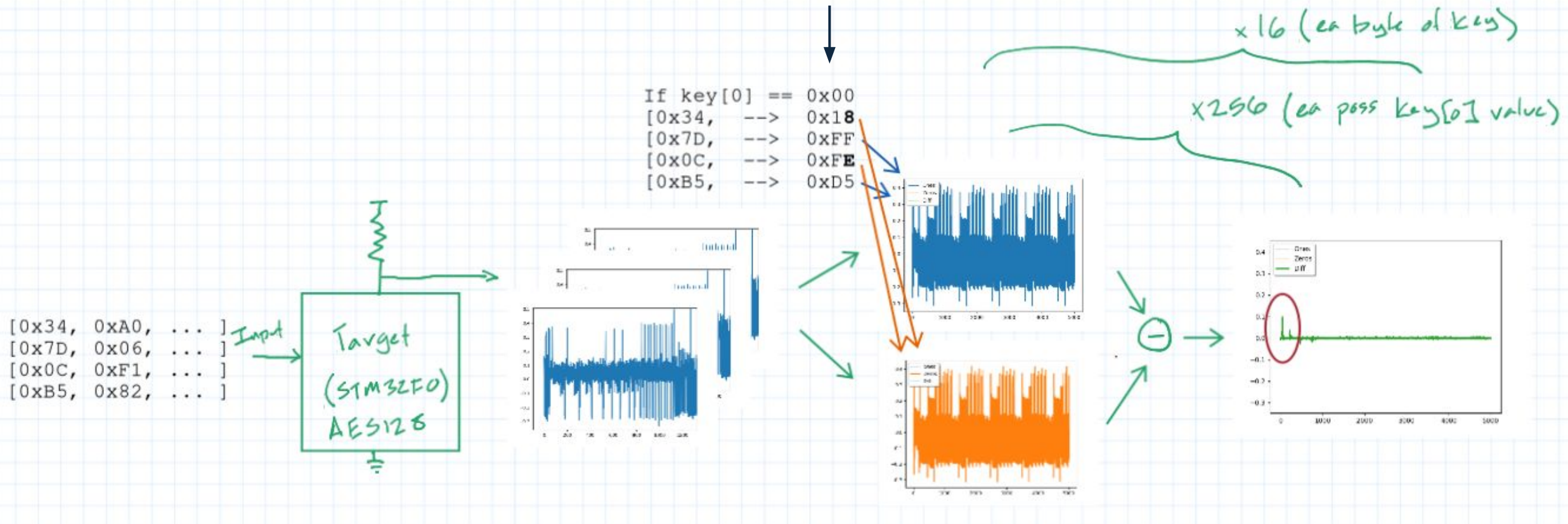
<https://privacycanada.net/app/uploads/2019/01/aes-algorithm-diagram.png>

How can we use this to break AES128?

<https://tinyurl.com/3kw99dfc>



$\text{sbox}[\text{pt}[0] \wedge \text{key}[0]]$



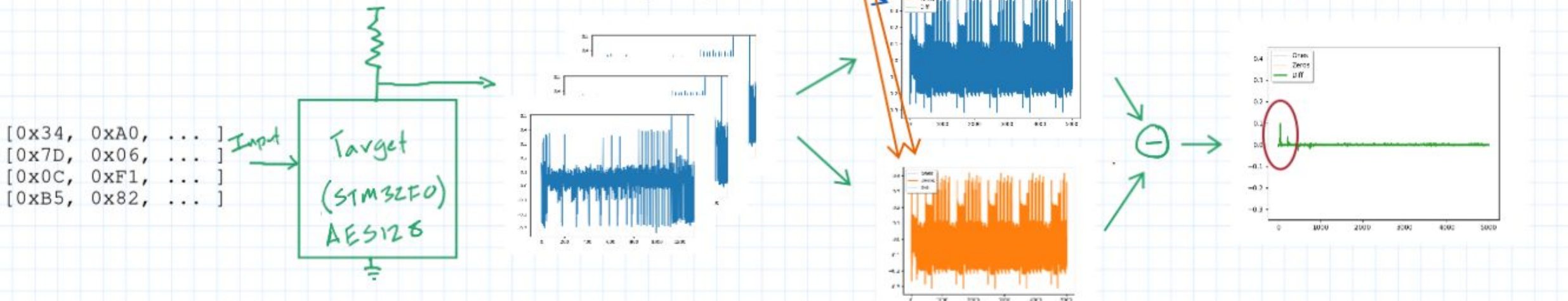
Your turn!

Complete SCA101/Lab 3_3

<https://tinyurl.com/3kw99dfc>



`sbox[pt[0]^key[0]]`





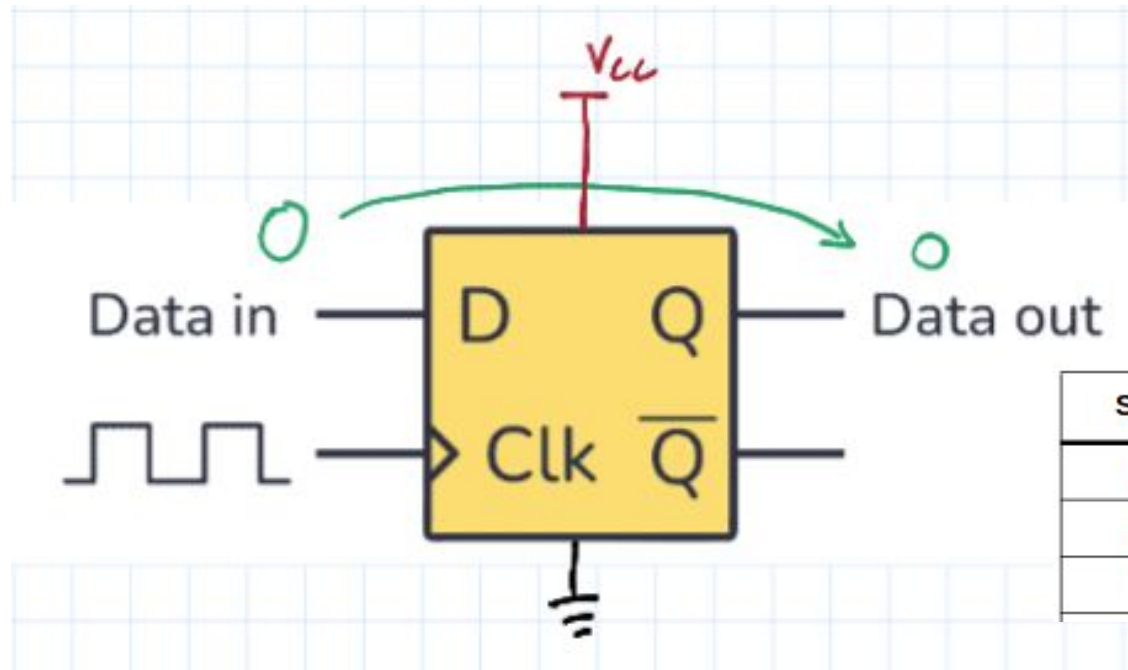
4

Attacking a target

Fault injections

Required operating conditions

<https://tinyurl.com/3kw99dfc>



For the STM32F042 on our CW-Nano:

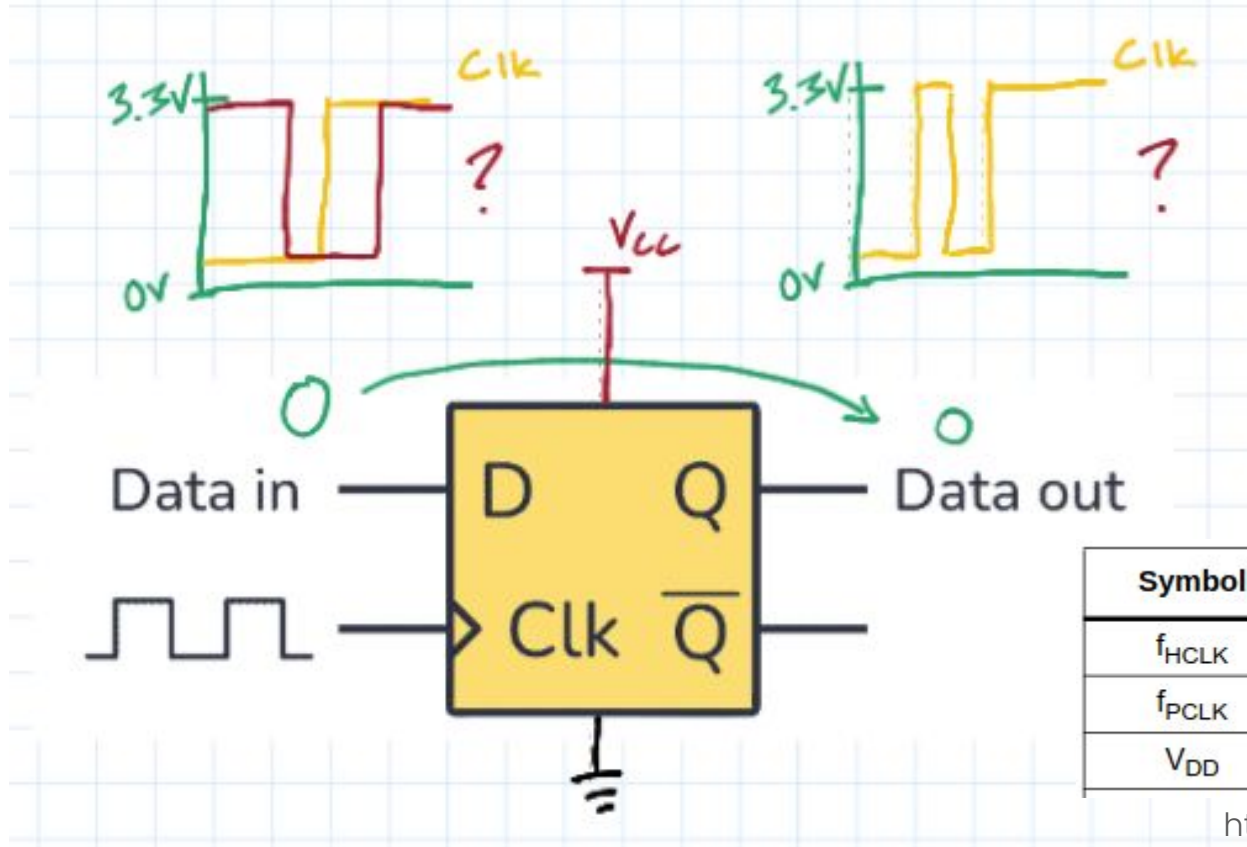
Symbol	Parameter	Min	Max	Unit
f_{HCLK}	Internal AHB clock frequency	0	48	MHz
f_{PCLK}	Internal APB clock frequency	0	48	
V_{DD}	Standard operating voltage	2.0	3.6	V

<https://www.st.com/resource/en/datasheet/stm32f042f4.pdf>

<https://www.build-electronic-circuits.com/wp-content/uploads/2022/11/D-flip-flop-basics.png>

But what happens if we don't?

<https://tinyurl.com/3kw99dfc>



For the STM32F042 on our CW-Nano:

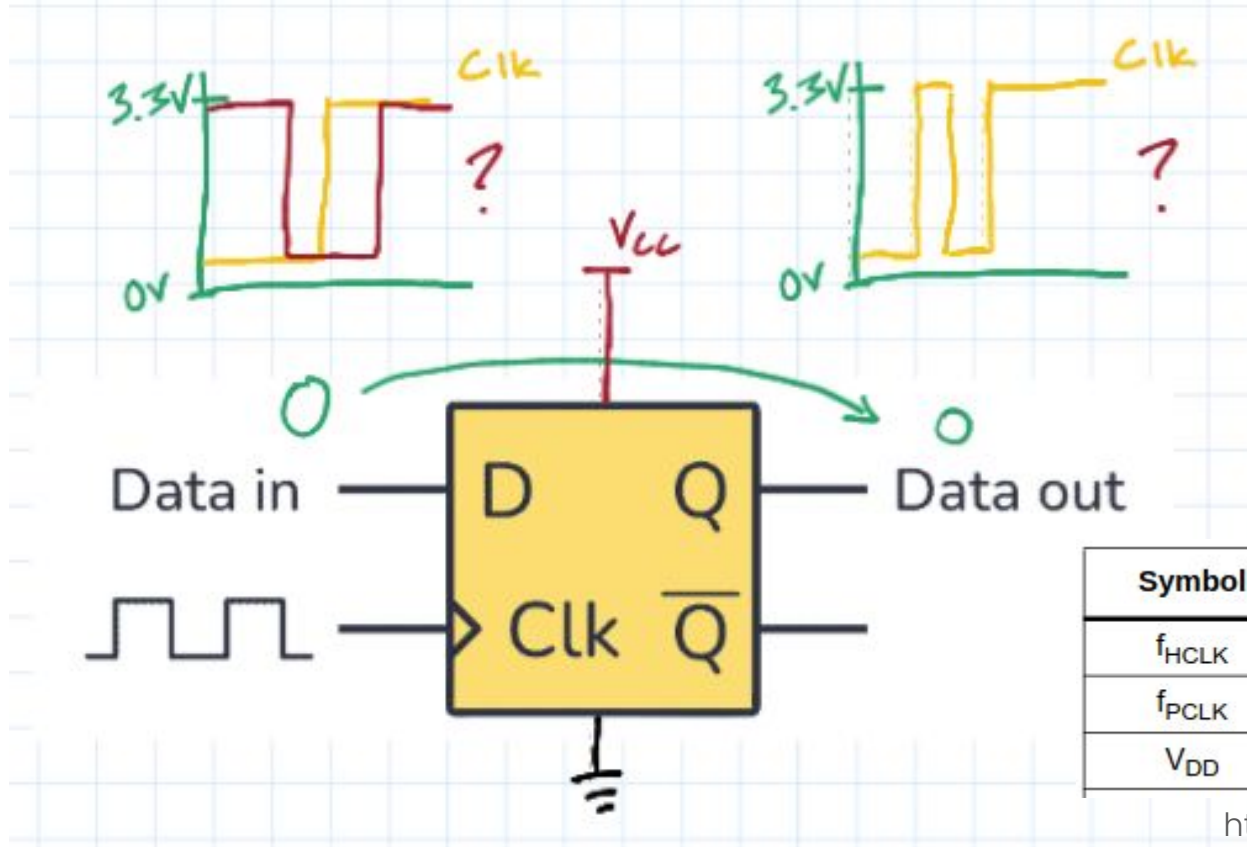
Symbol	Parameter	Min	Max	Unit
f_{HCLK}	Internal AHB clock frequency	0	48	MHz
f_{PCLK}	Internal APB clock frequency	0	48	
V_{DD}	Standard operating voltage	2.0	3.6	V

<https://www.st.com/resource/en/datasheet/stm32f042f4.pdf>

<https://www.build-electronic-circuits.com/wp-content/uploads/2022/11/D-flip-flop-basics.png>

But what happens if we don't?

<https://tinyurl.com/3kw99dfc>



We don't know! Most likely reset, but occasionally:

- Skipped instructions
- Altered/corrupted instructions
- Altered/corrupted memory

For the STM32F042 on our CW-Nano:

Symbol	Parameter	Min	Max	Unit
f_{HCLK}	Internal AHB clock frequency	0	48	MHz
f_{PCLK}	Internal APB clock frequency	0	48	
V_{DD}	Standard operating voltage	2.0	3.6	V

<https://www.st.com/resource/en/datasheet/stm32f042f4.pdf>

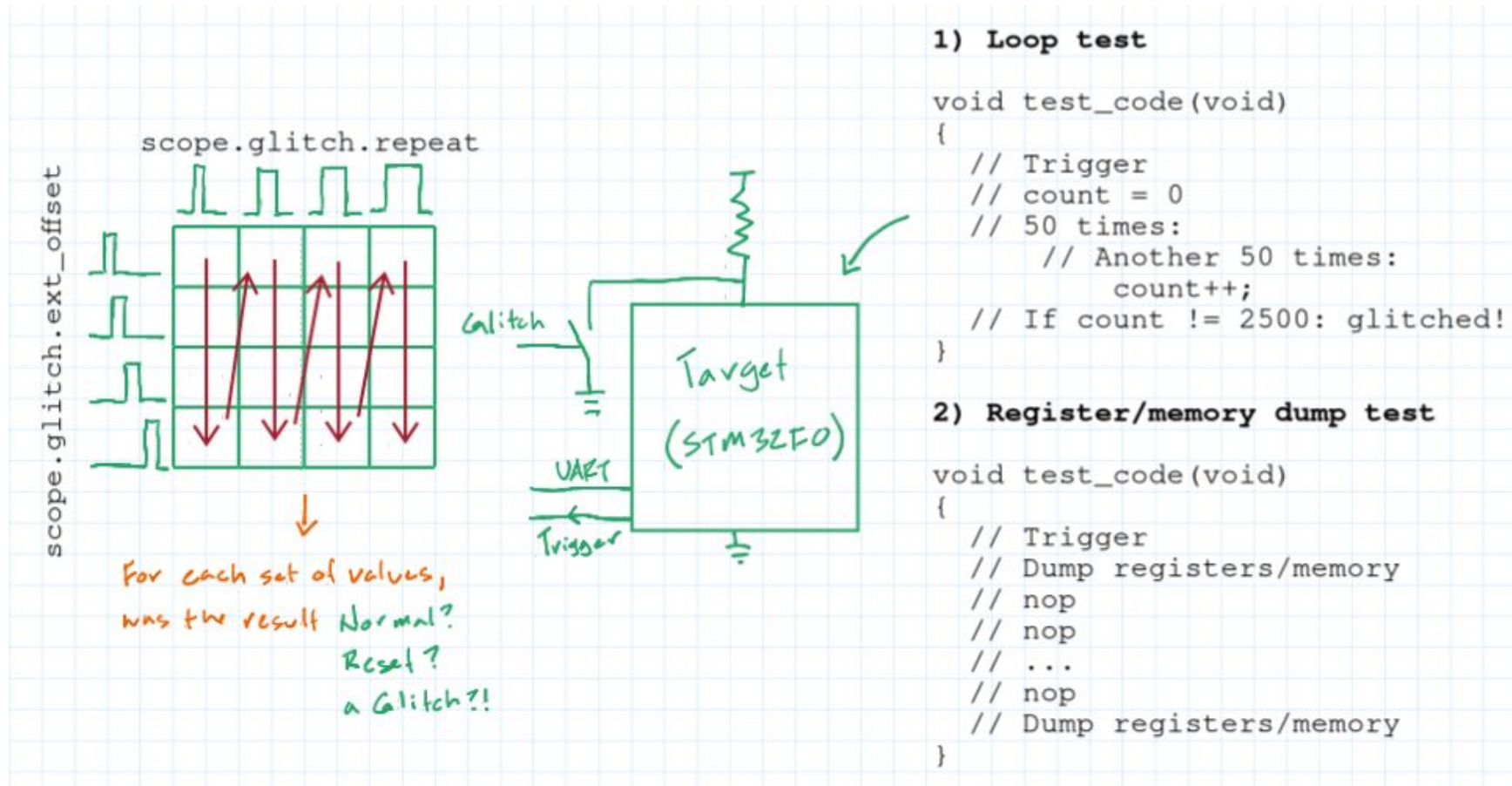
<https://www.build-electronic-circuits.com/wp-content/uploads/2022/11/D-flip-flop-basics.png>

Glitch characterization

<https://tinyurl.com/3kw99dfc>



Fault101/SOLN_Fault 2_1B

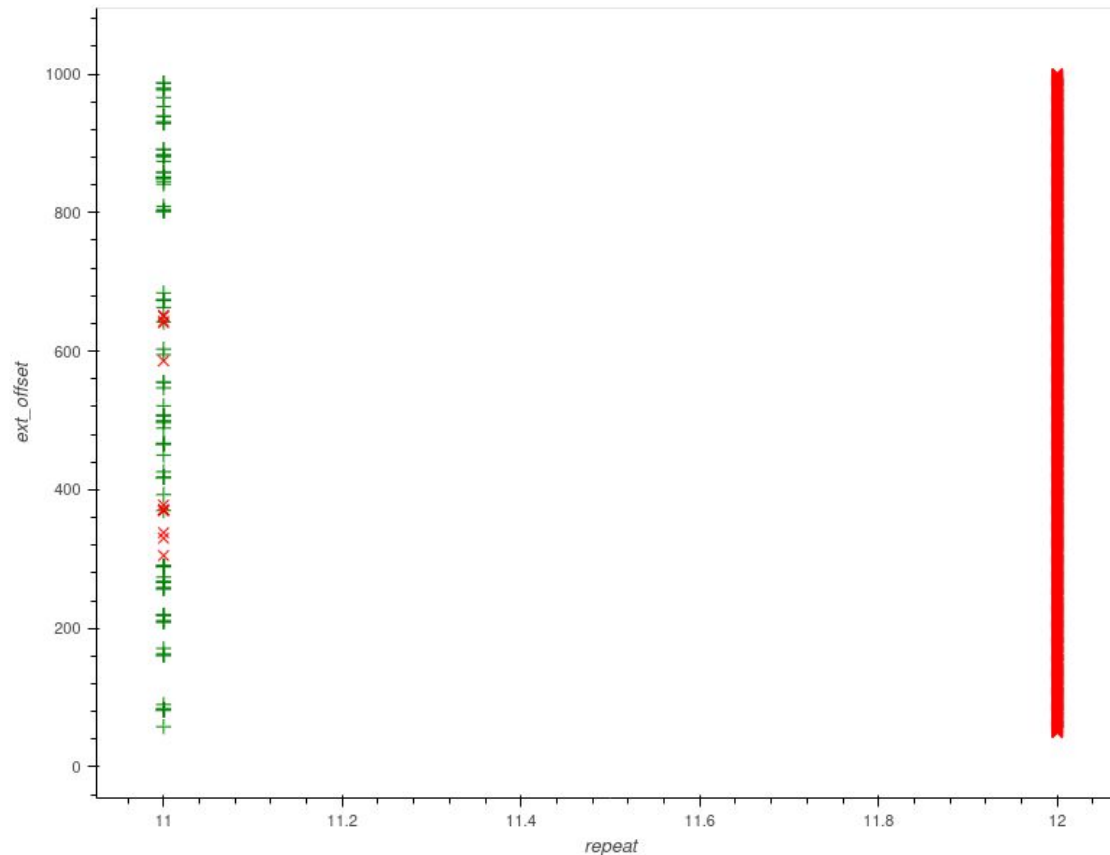


Your turn!

<https://tinyurl.com/3kw99dfc>



Fault101/SOLN_Fault 2_1B



```
cw.set_all_log_levels(cw.logging.CRITICAL)
```

```
2501
successes = 2, resets = 0, repeat = 11, ext_offset = 674
2501
successes = 1, resets = 0, repeat = 11, ext_offset = 684
2502
2502
successes = 3, resets = 0, repeat = 11, ext_offset = 802
2502
successes = 1, resets = 0, repeat = 11, ext_offset = 804
2499
successes = 1, resets = 0, repeat = 11, ext_offset = 809
```

Preparing to glitch: Identifying a target

<https://tinyurl.com/3kw99dfc>



```
#if SS_VER == SS_VER_2_1
uint8_t password(uint8_t cmd, uint8_t scmd, uint8_t len, uint8_t* pw)
#else
uint8_t password(uint8_t* pw, uint8_t len)
#endif
{
    char passwd[] = "touch";
    char passok = 1;
    int cnt;

    trigger_high();

    //Simple test - doesn't check for too-long password!
    for(cnt = 0; cnt < 5; cnt++){
        if (pw[cnt] != passwd[cnt]){
            passok = 0;
        }
    }

    trigger_low();

    if(passok) access = true;

    simpleserial_put('r', 1, (uint8_t*)&passok);
    return 0x00;
}
```

Skip passok = 0;

```
96:simpleserial-glitch.c ****          passok = 0;
385          .loc 1 96 13 is_stmt 1 view .LVU83
386          .loc 1 96 20 is_stmt 0 view .LVU84
387 0056 0B23      movs    r3, #11
388 0058 FB18      adds    r3, r7, r3
                        movs    r2, #0
                        strb    r2, [r3]
.L17:
94:simpleserial-glitch.c ****          if (pw[cnt] != passwd[cnt]){
392          .loc 1 94 27 is_stmt 1 discriminator 2 view .LVU85
94:simpleserial-glitch.c ****          if (pw[cnt] != passwd[cnt]){
393          .loc 1 94 30 is_stmt 0 discriminator 2 view .LVU86
394 005e 7B69      ldr     r3, [r7, #20]
395 0060 0133      adds    r3, r3, #1
396 0062 7B61      str     r3, [r7, #20]
397          .LVL29:
398          .L16:
94:simpleserial-glitch.c ****          if (pw[cnt] != passwd[cnt]){
399          .loc 1 94 18 is_stmt 1 discriminator 1 view .LVU87
94:simpleserial-glitch.c ****          if (pw[cnt] != passwd[cnt]){
400          .loc 1 94 5 is_stmt 0 discriminator 1 view .LVU88
401 0064 7B69      ldr     r3, [r7, #20]
                        cmp     r3, #4
                        ble     .L18
```

Skip backwards jump

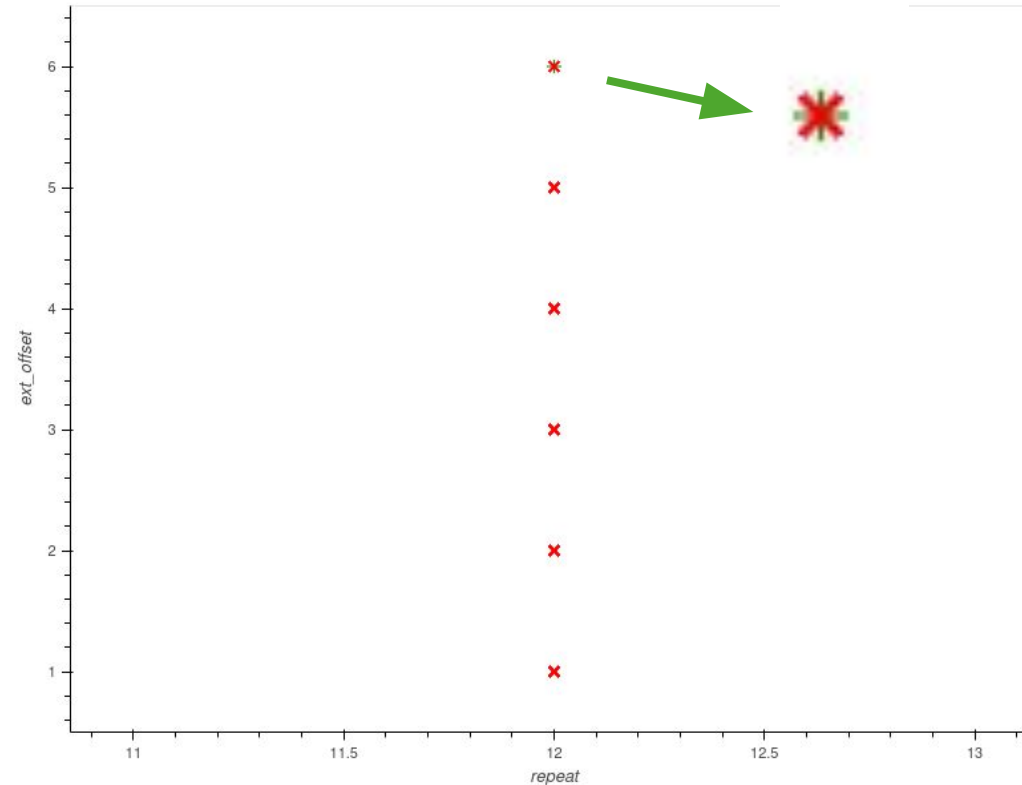
or afterward, corrupt passok to be non-0

Your turn!

<https://tinyurl.com/3kw99dfc>



Fault101/SOLN_Fault 2_2B



`{'valid': True, 'payload': CWbtearray(b'01'), 'full_response': CWbtearray(b'00 72 01 01 d4 00'), 'rv': bytearray(b'\x00')}`
`CWbtearray(b'01')`
`12 6`





5

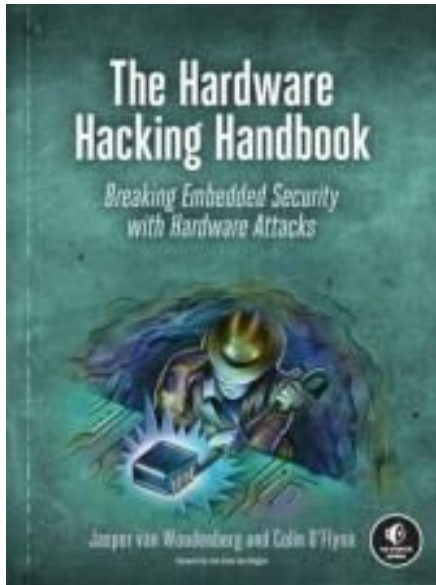
Closing thoughts

Closing thoughts

<https://tinyurl.com/3kw99dfc>



- Countermeasures *do* exist for these attacks
- For more fun:



ChipSHOUTER® Kit

SKU: NAE-CW520-KIT

\$4,605.00

Quantity *

— 1 +

Add to Cart

Features

- The ChipSHOUTER CW520 main unit is the EMFI fault injection platform itself.



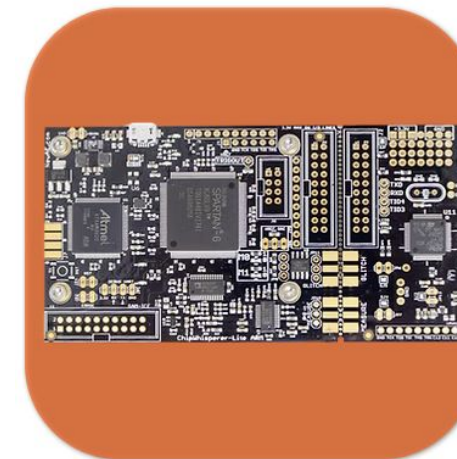
Husky

Our Flagship Whisperer

- High-speed logic analyzers (to visualize glitches)
- Real-time data streaming for attacking asymmetric algorithms
- Support for JTAG/SWD programming with an FTDI-compatible mode
- Additional I/O expansion pins

\$599 USD

Buy Now



Lite

Hacking on a Budget

The Lite offers power analysis measurement, device programming, glitching, serial communications, and an example target that can be loaded with cryptographic algorithms all into a single board.

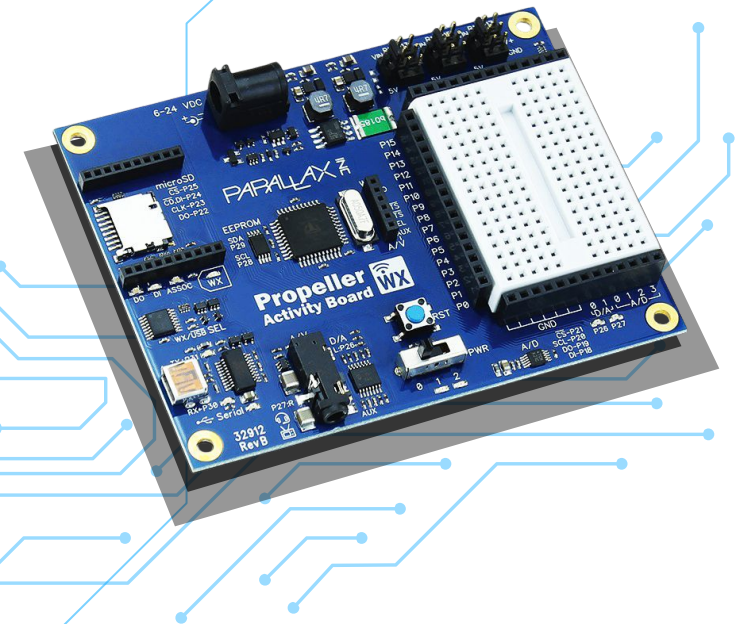
\$340 USD

Buy Now



John Malloy

“Security is not a one-time event. It’s an ongoing process.”



MITRE eCTF

<https://tinyurl.com/3kw99dfc>



Design Phase

Teams design and implement systems that meet security and functionality requirements



Handoff

Organizers test each design for functionality

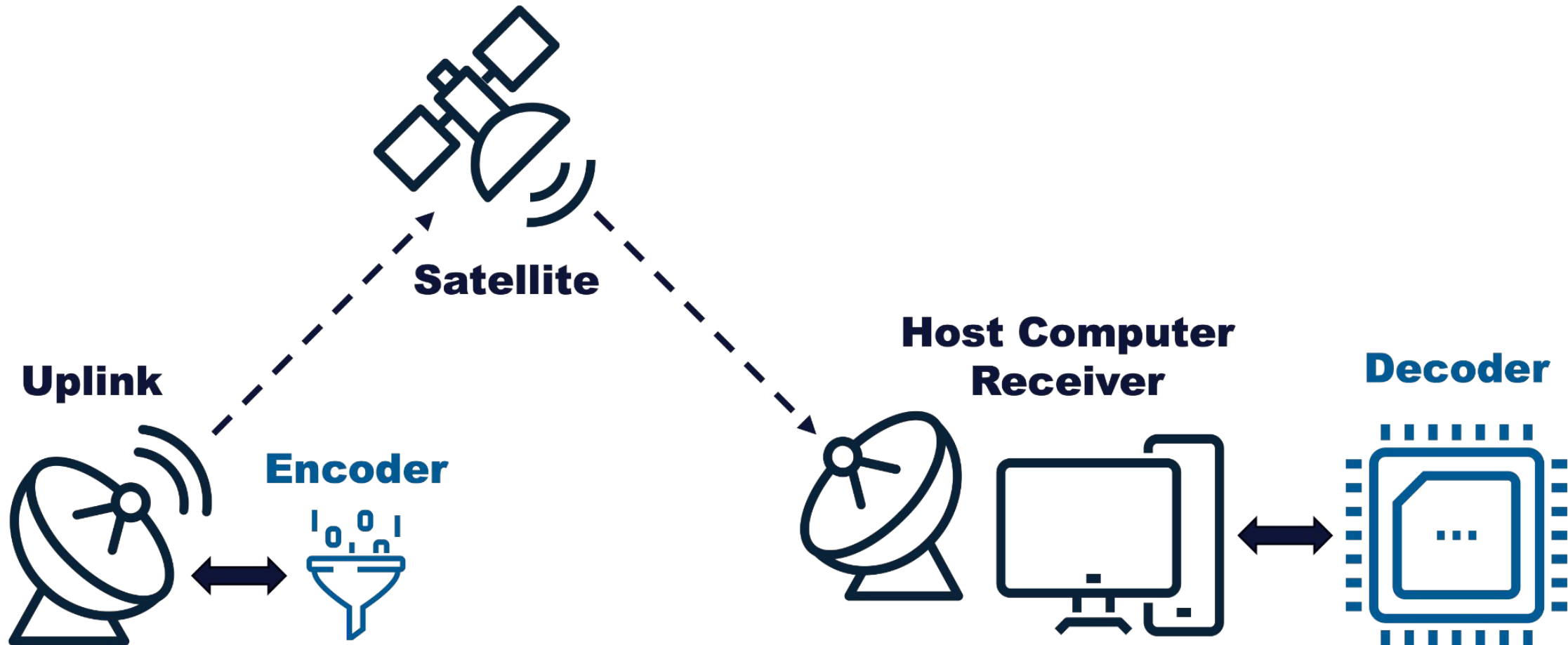


Attack Phase

Teams analyze and attack each other's designs for points

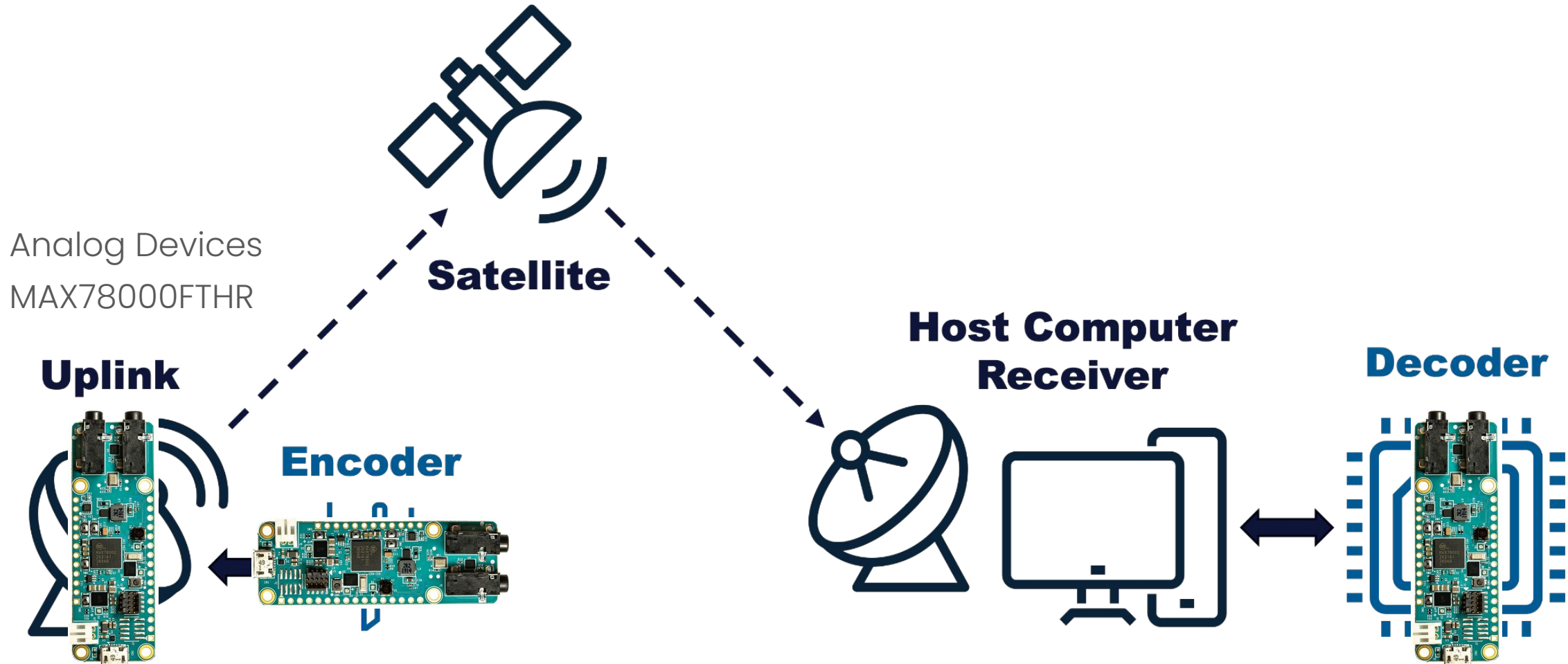
2025 eCTF: Secure Satellite TV

<https://tinyurl.com/3kw99dfc>



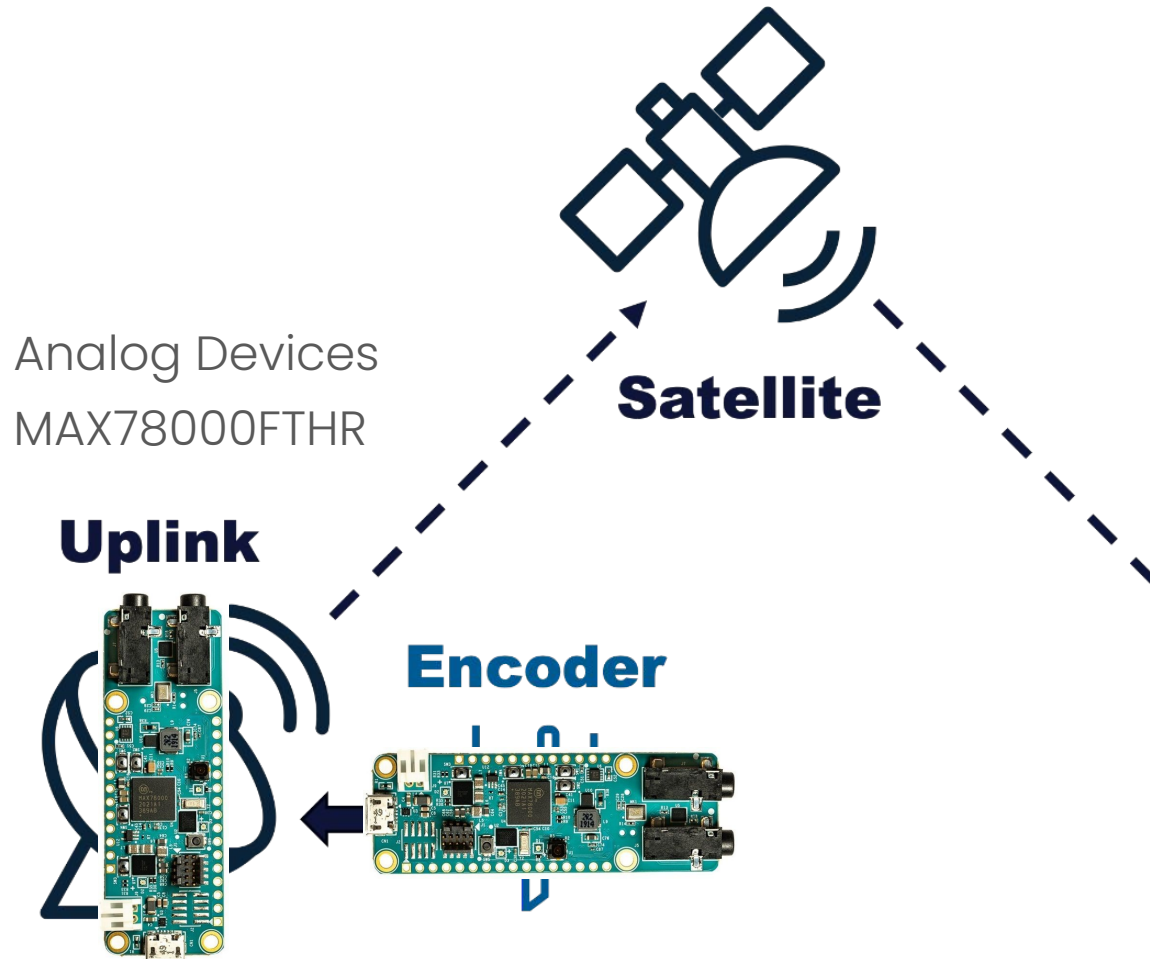
2025 eCTF: Secure Satellite TV

<https://tinyurl.com/3kw99dfc>



2025 eCTF: Secure Satellite TV

<https://tinyurl.com/3kw99dfc>

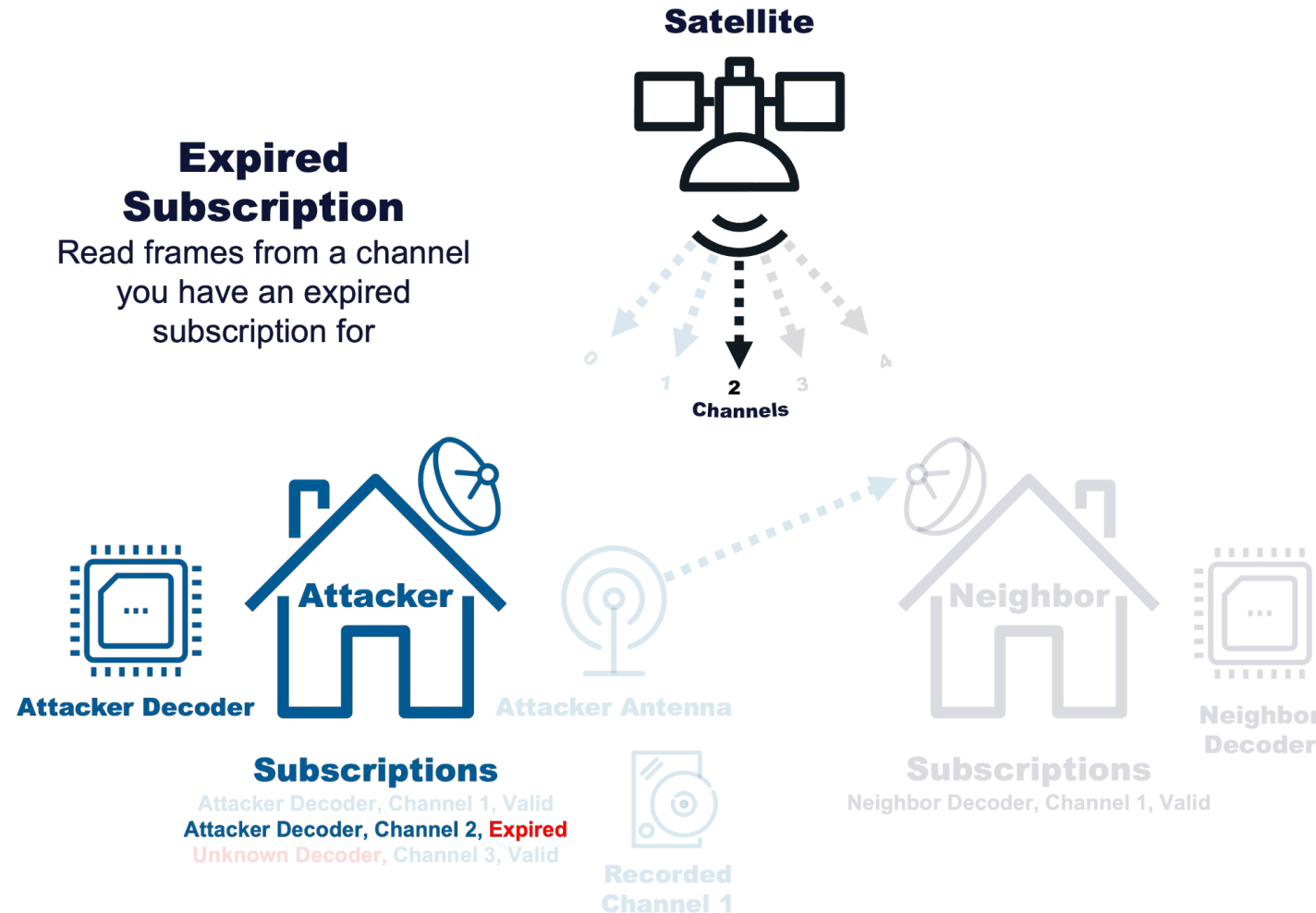


Security requirements:

1. An attacker should not be able to decode TV frames without a Decoder that has a valid, active subscription to that channel.
2. The Decoder should only decode valid TV frames generated by the Satellite System the Decoder was provisioned for.
3. The Decoder should only decode frames with strictly monotonically increasing timestamps.

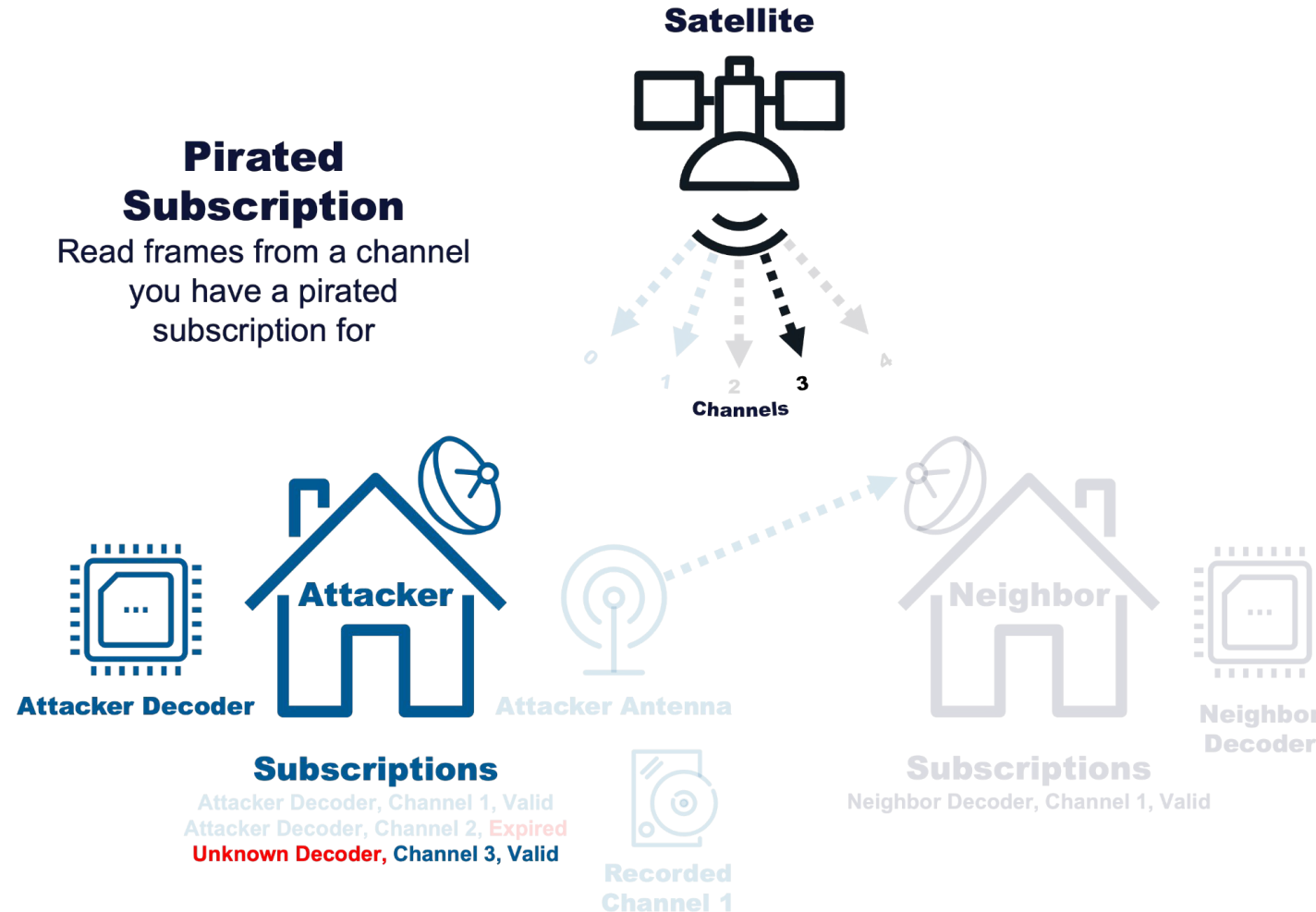
2025 eCTF: Secure Satellite TV

<https://tinyurl.com/3kw99dfc>



2025 eCTF: Secure Satellite TV

<https://tinyurl.com/3kw99dfc>



Past competitors

<https://tinyurl.com/3kw99dfc>



Air Force Institute of Technology • Amherst College • Amrita Vishwa Vidyapeetham University • Arizona State University • Asia Pacific University of Technology and Innovation • Baldwin Wallace University • BASIS Chandler • Binghamton University • Boston University • Brigham Young University • California State University, Fullerton • California State University, Los Angeles • Carnegie Mellon University • Center I (Albemarle County Public Schools) • City College of San Francisco • Clemson University • Colombe Academy of Technology • Colorado State University • Columbia University • Cornell University • CyberAegis • Dakota State University • Delaware Area Career Center • Duke University • East Tennessee State University • Ecole Royale de l'Air • Embry-Riddle Aeronautical University, Prescott • Essex North Shore Agricultural and Technical School • Faculte des sciences Ain Chock • Fairport High School • Flinders University • Florida A&M University • Florida Atlantic University • Florida International University • Florida State University • George Mason University • Georgia Institute of Technology • Guru Gobind Singh Indraprastha University • Hampton University • Harmony Science Academy • Ho Chi Minh City University of Technology • HTL Pinkafeld • Indian Institute of Technology Dharwad • Indian Institute of Technology Indore • Indian Institute of Technology Madras • Indiana Institute of Technology • Institut Bisnis

Past competitors

<https://tinyurl.com/3kw99dfc>

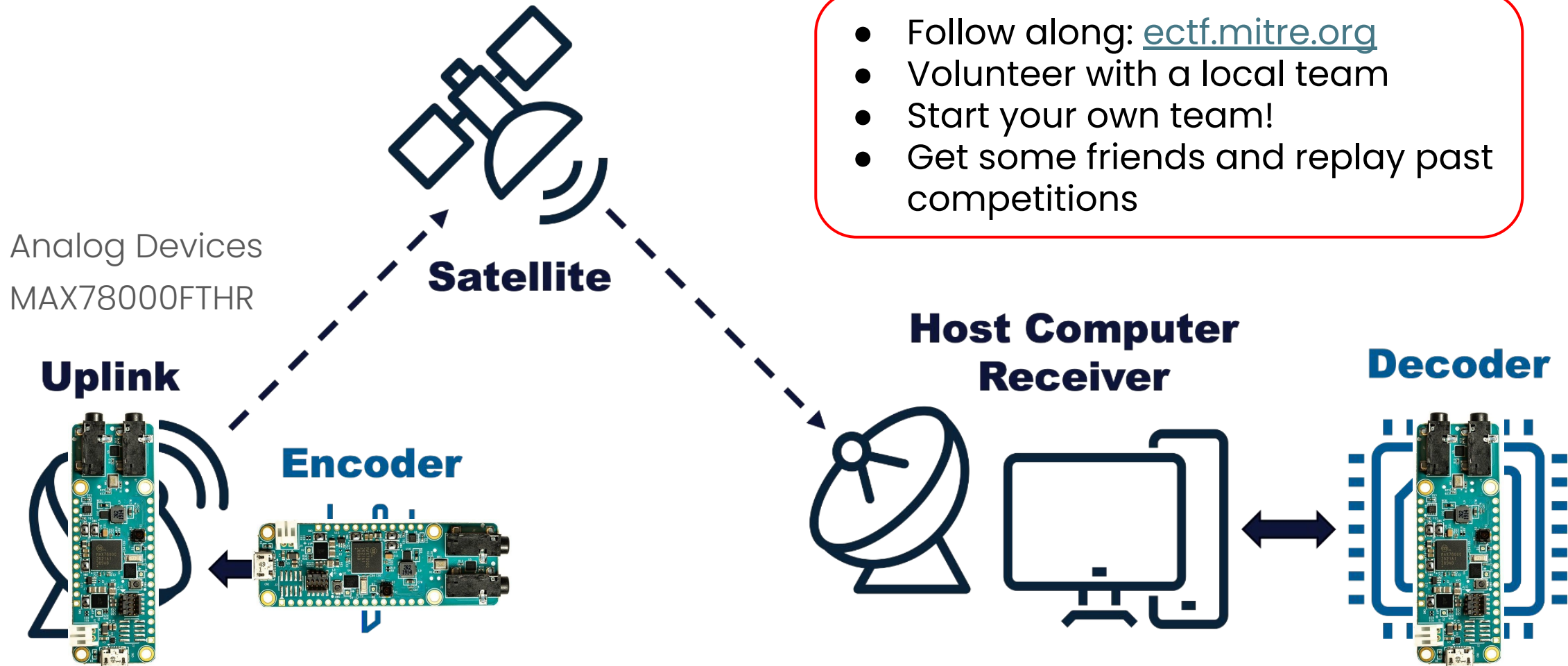


Air Force Institute of Technology • Amherst College • Amrita Vishwa Vidyapeetham University • Arizona State University • Asia Pacific University of Technology and Innovation • Baldwin Wallace University • BASIS Chandler • Binghamton University • Boston University • Brigham Young University • California State University, Fullerton • California State University, Los Angeles • Carnegie Mellon University • Center I (Albemarle County Public Schools) • City College of San Francisco • Clemson University • Colombe Academy of Technology • Colorado State University • Columbia University • Cornell University • CyberAegis • Dakota State University • Delaware Area Career Center • Duke University • Eastern Michigan University • Ecole Royale de l'Air • Embry-Riddle Aeronautical University, Prescott • Essex Community College • Georgia Institute of Technology • Harbin Engineering University • Fairport High School • Flinders University • Florida A&M University • Florida Atlantic University • Florida International University • Florida State University • George Mason University • Georgia Institute of Technology • Guru Gobind Singh Indraprastha University • Hampton University • Harmony Science Academy • Ho Chi Minh City University of Technology • HTL Pinkafeld • Indian Institute of Technology Dharwad • Indian Institute of Technology Indore • Indian Institute of Technology Madras • Indiana Institute of Technology • Institut Bisnis

+9 others!

For more hardware hacking fun

<https://tinyurl.com/3kw99dfc>





Thank you

www.embeddedsummit.com