TEMPEST Attacks Against AES

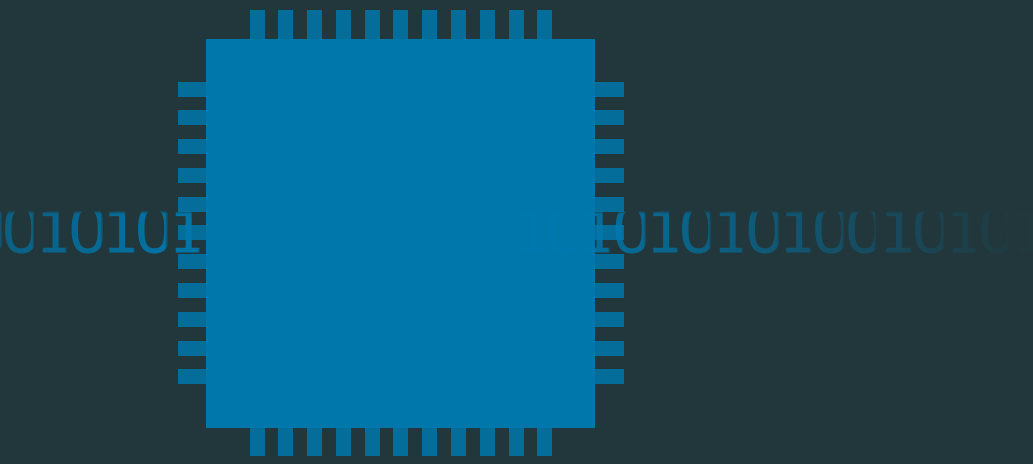
Covertly stealing keys for $200

Craig Ramsay & Jasper Lohuis

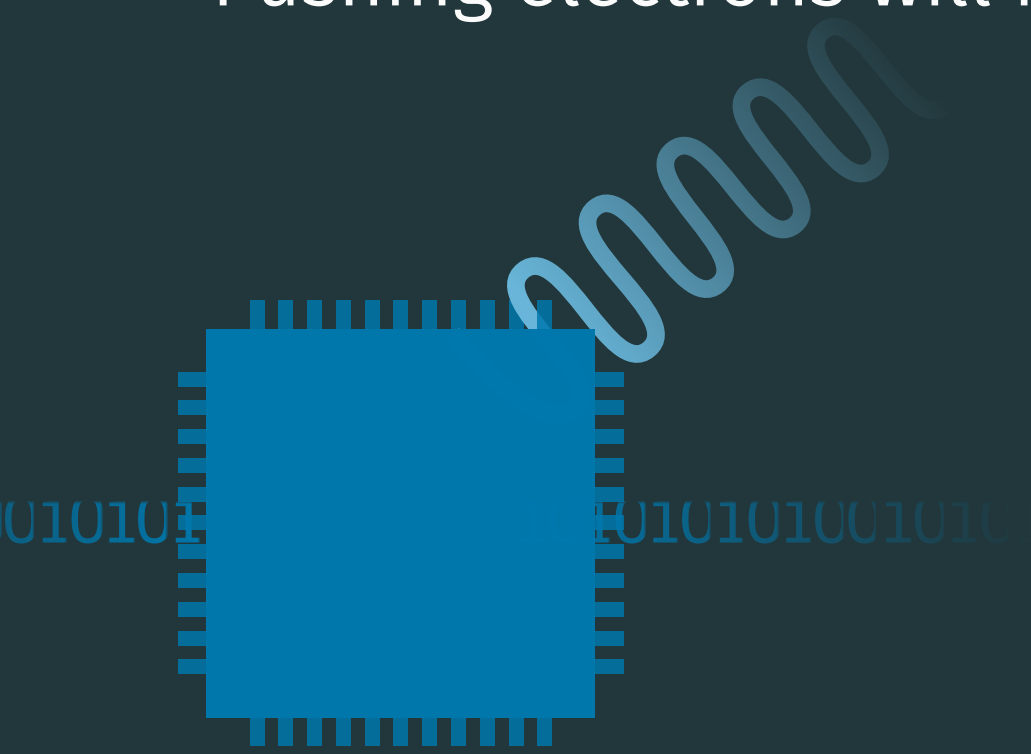
September 22, 2017

Introduction

Your code just pushes electrons around.

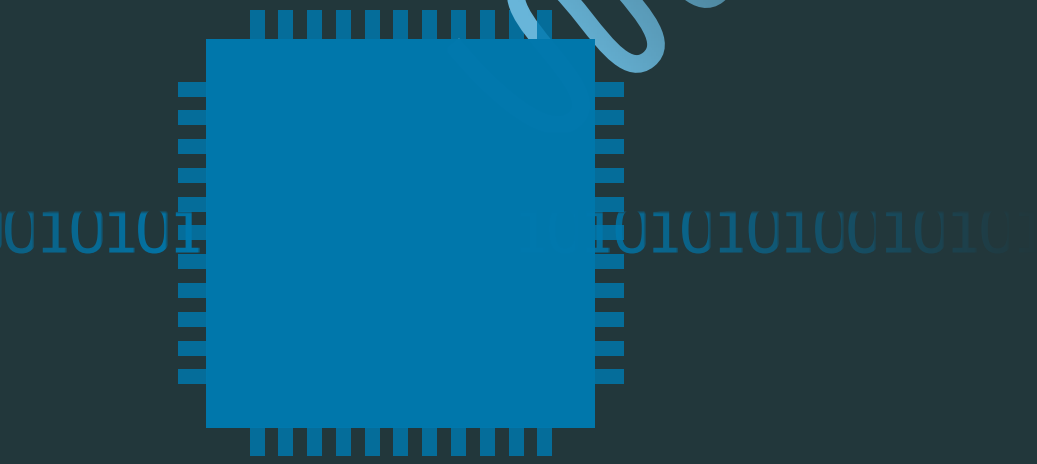


Pushing electrons will make magnetic fields.




TEMPEST attacks measure this from a distance.

TEMPEST attacks measure this from a distance.



Project people

Duncan Lew
First intern. Close-by FPGA attacks

Freek van Tienen
We'll see!

Craig Ramsay
Radio-based workflow & attacking ARM

Jasper Lohuis
Cheap shielding, SDRs & antennas

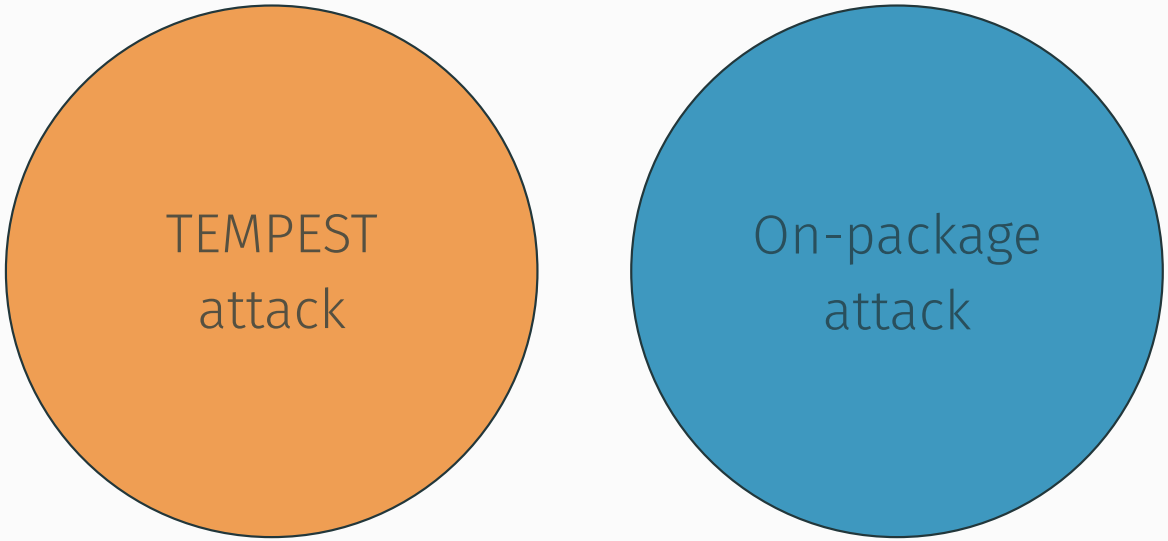
Thanks for feeding us, folks

FOX IT

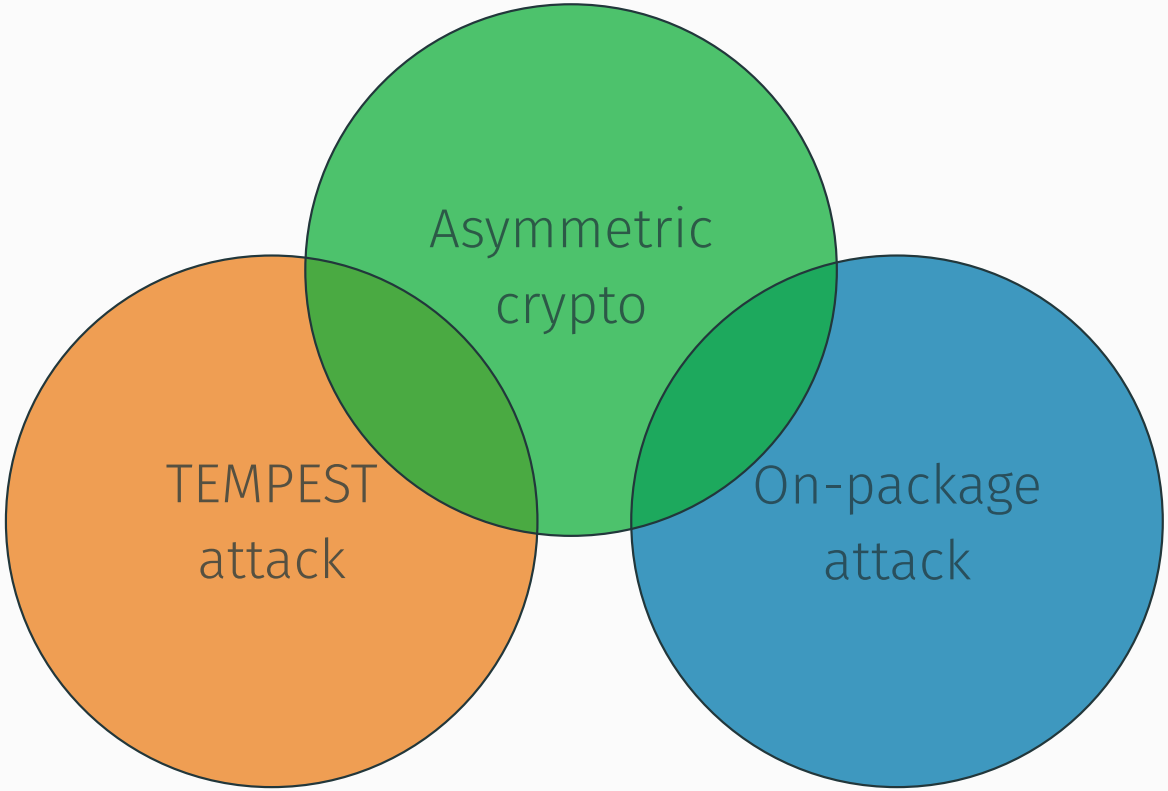
FOR A MORE SECURE SOCIETY

riscure

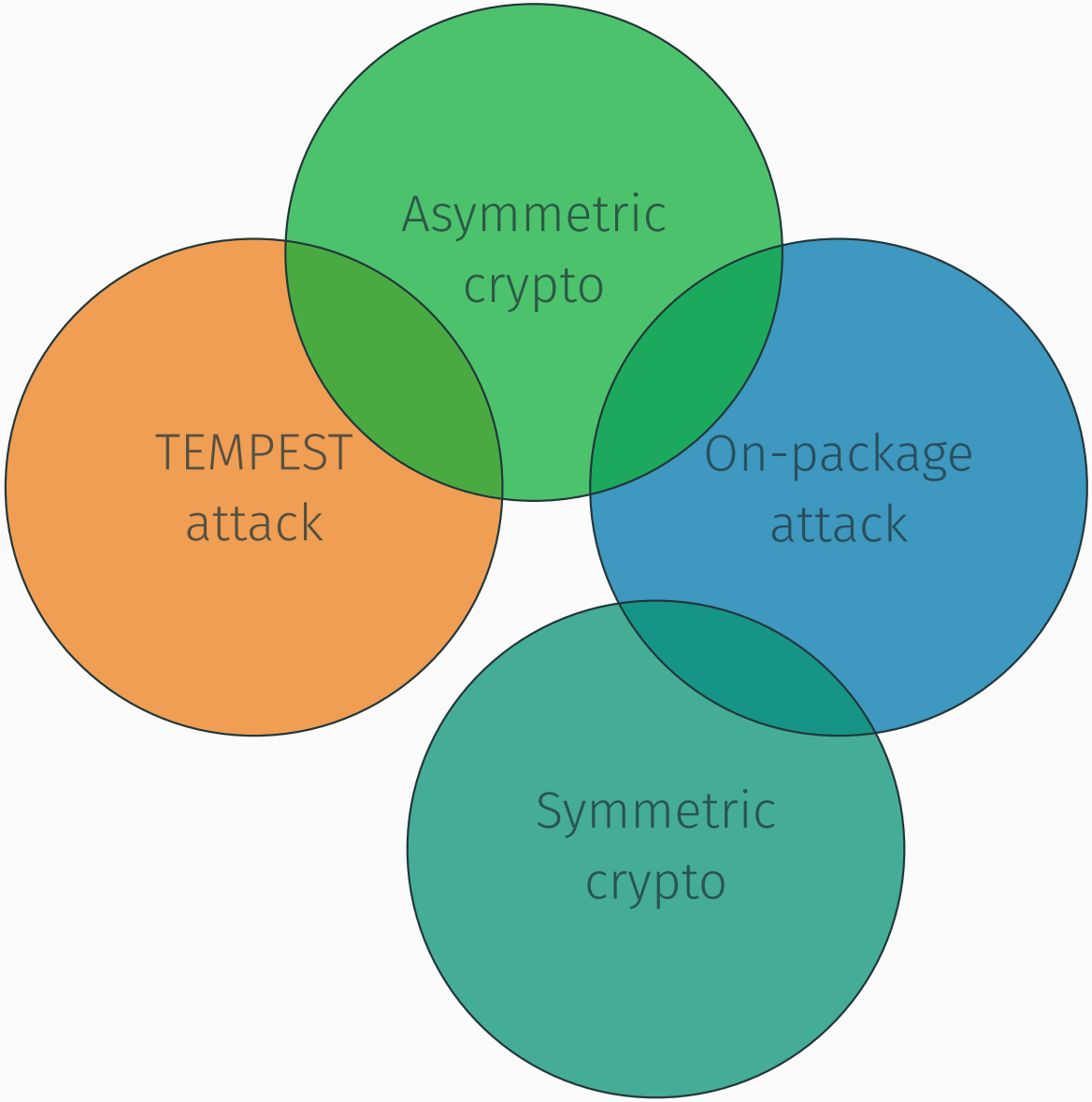
Existing Work



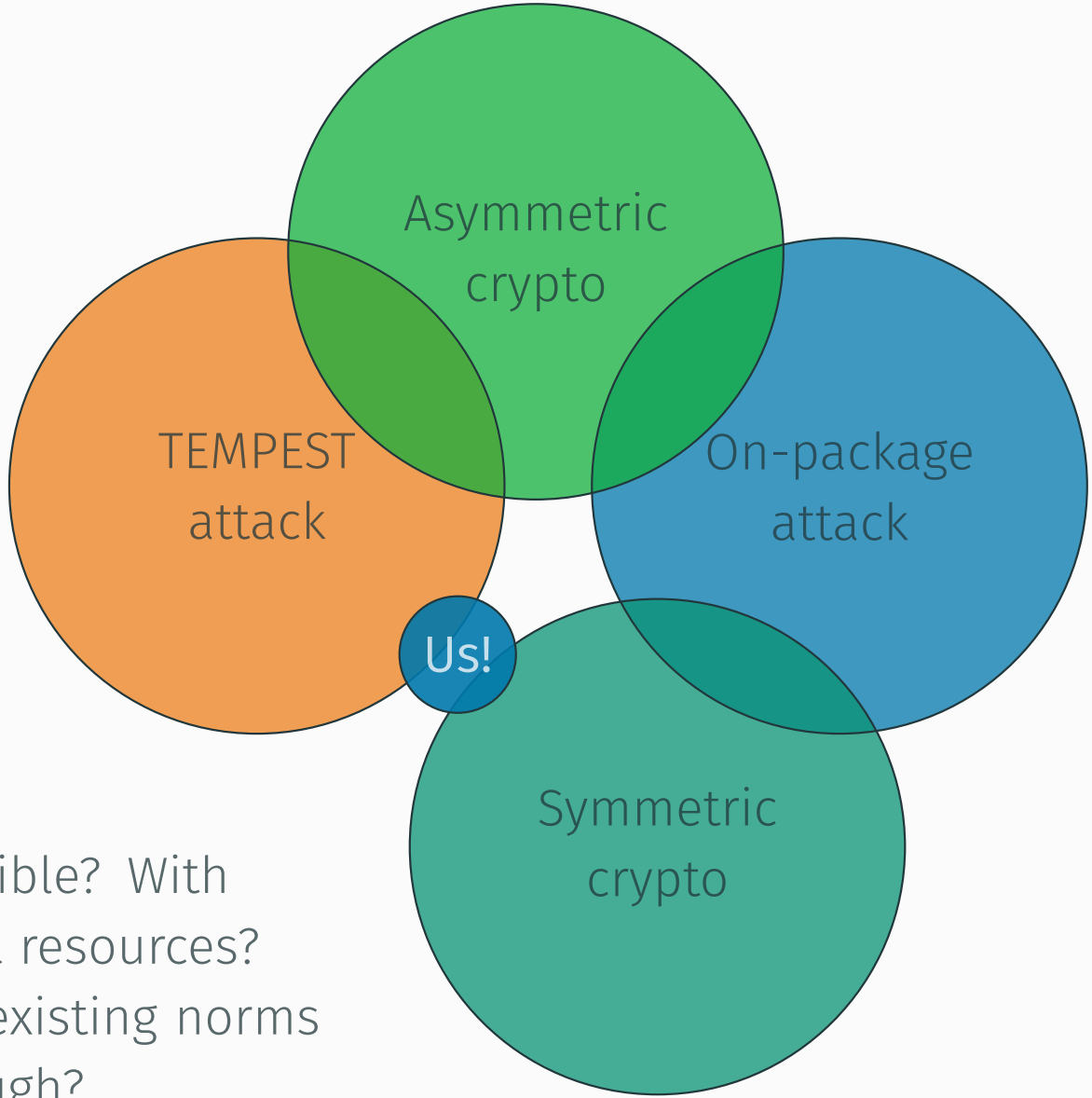
Existing Work



Existing Work



Existing Work



Possible? With what resources? Are existing norms enough?

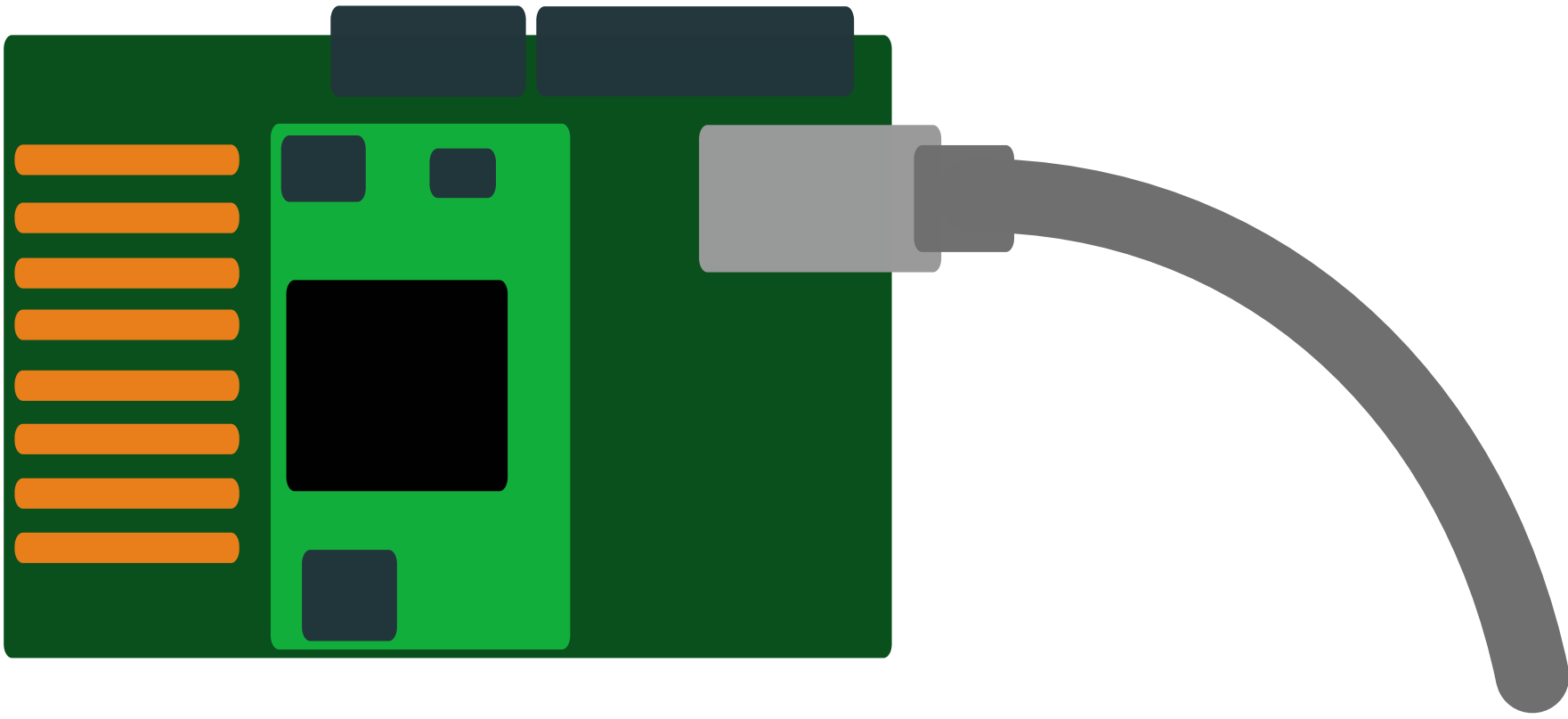
Replicating On-Package Attack

Overview


Target Device
Analogue Measurement
Radio Recording
Preprocessing
Analysis

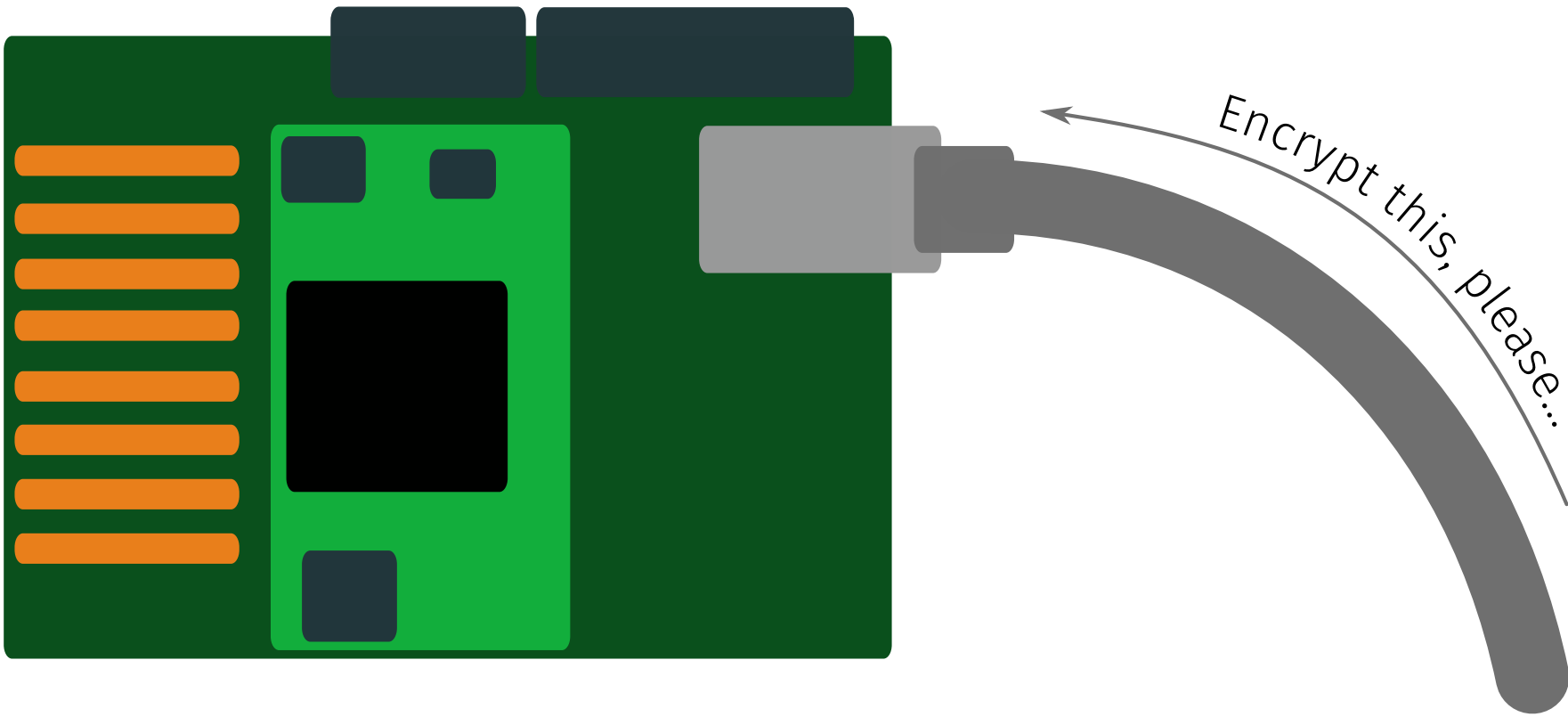
Overview


Target Device
Analogue Measurement
Radio Recording
Preprocessing
Analysis



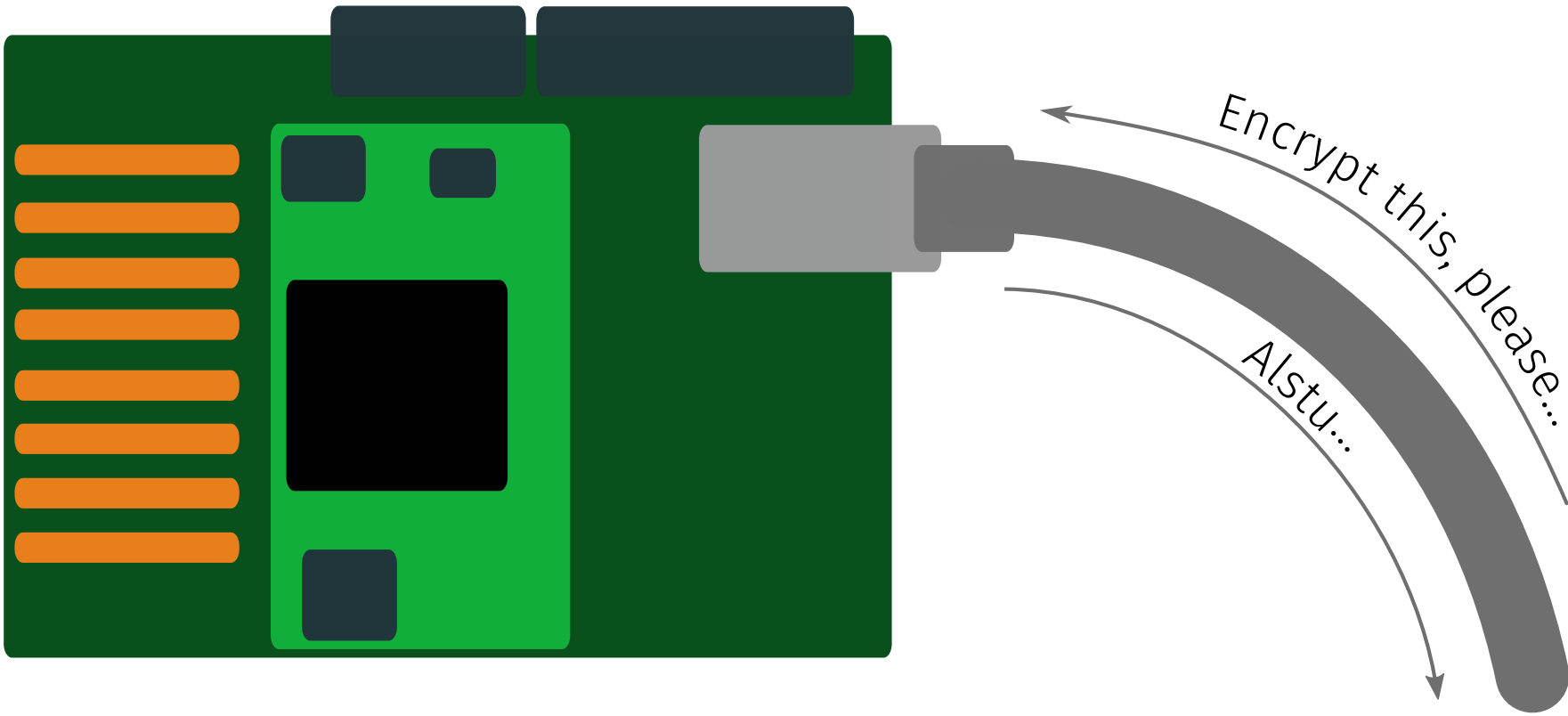
Overview


Target Device
Analogue Measurement
Radio Recording
Preprocessing
Analysis


Encrypt this, please...

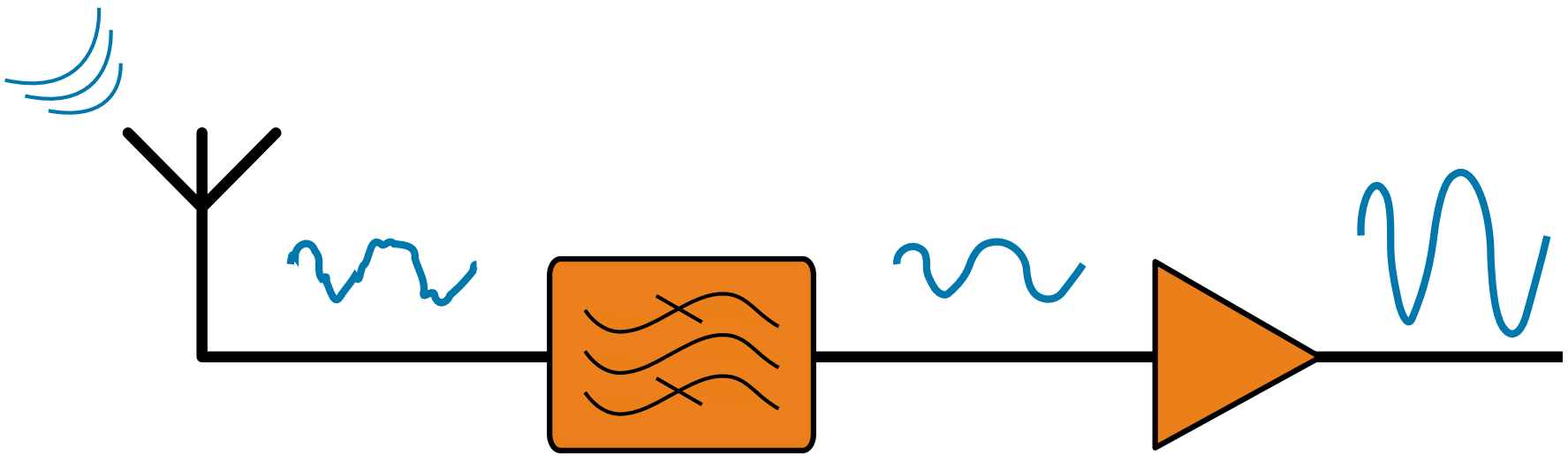
Overview


Target Device
Analogue Measurement
Radio Recording
Preprocessing
Analysis


Encrypt this, please...
Alstu...

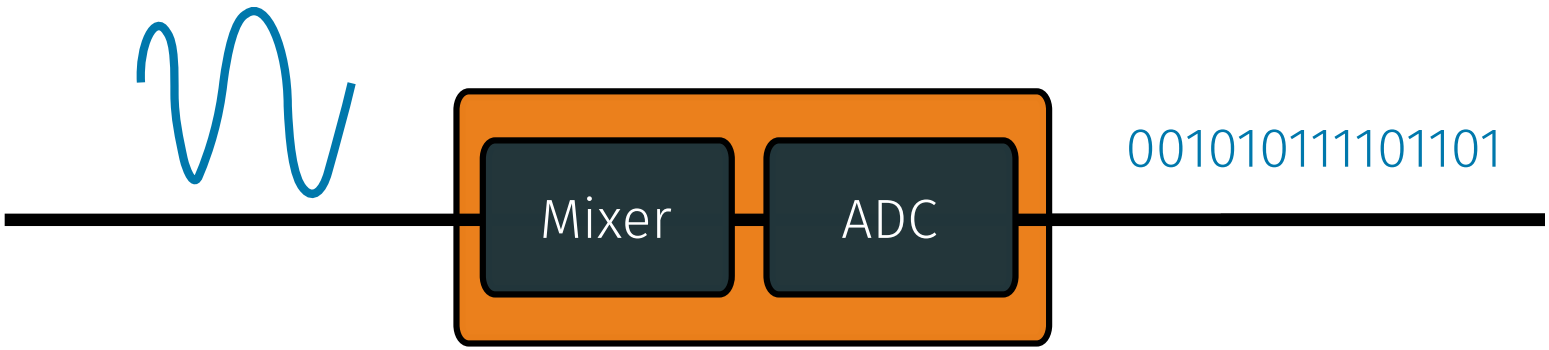
Overview


Target Device
Analogue Measurement
Radio Recording
Preprocessing
Analysis



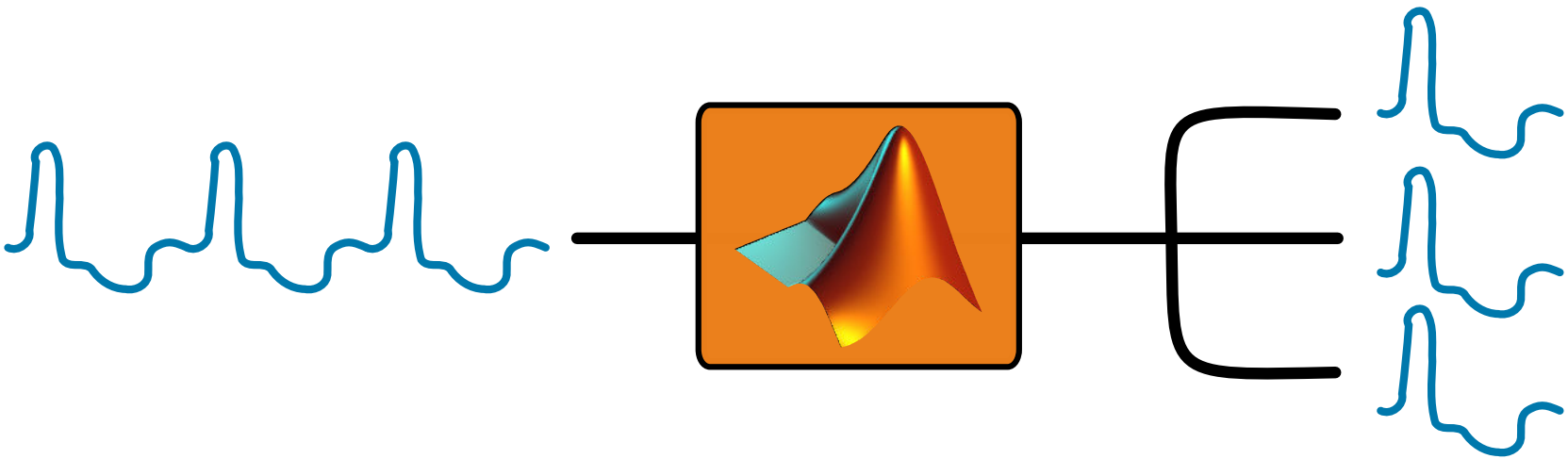
Overview


Target Device
Analogue Measurement
Radio Recording
Preprocessing
Analysis


Mixer
ADC
001010111101101

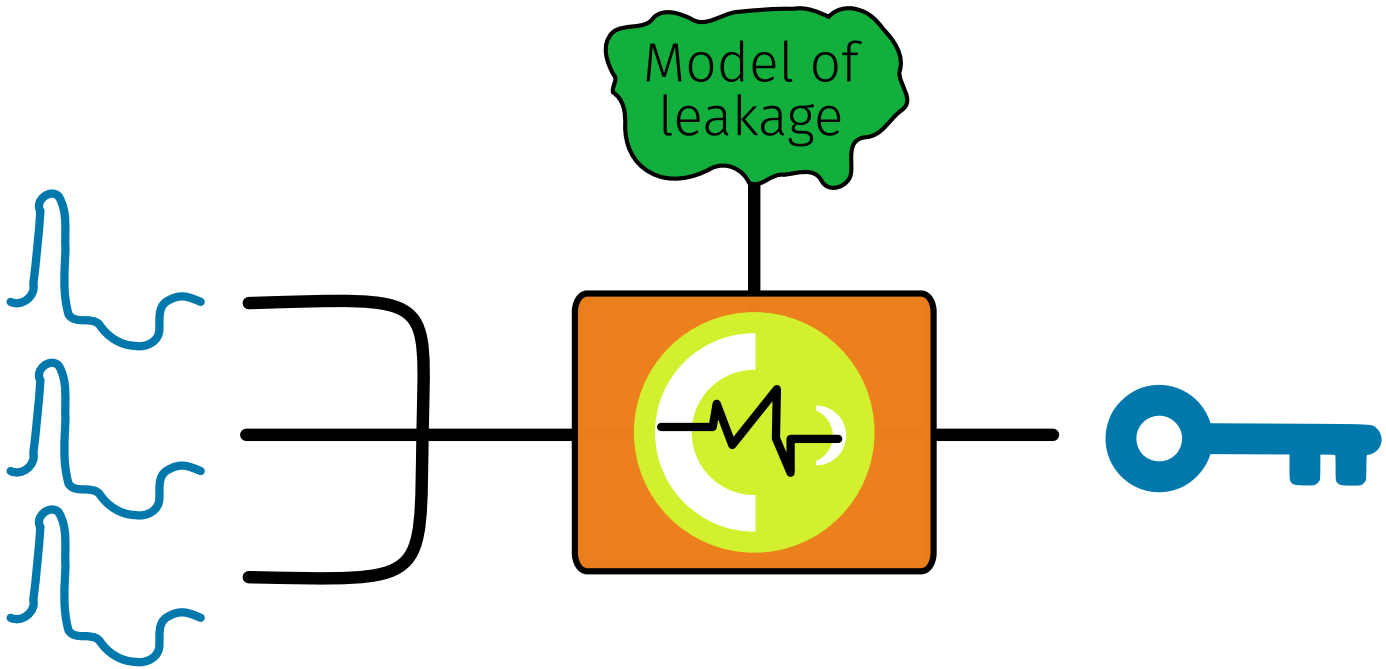
Overview


Target Device
Analogue Measurement
Radio Recording
Preprocessing
Analysis



Overview


Target Device
Analogue Measurement
Radio Recording
Preprocessing
Analysis

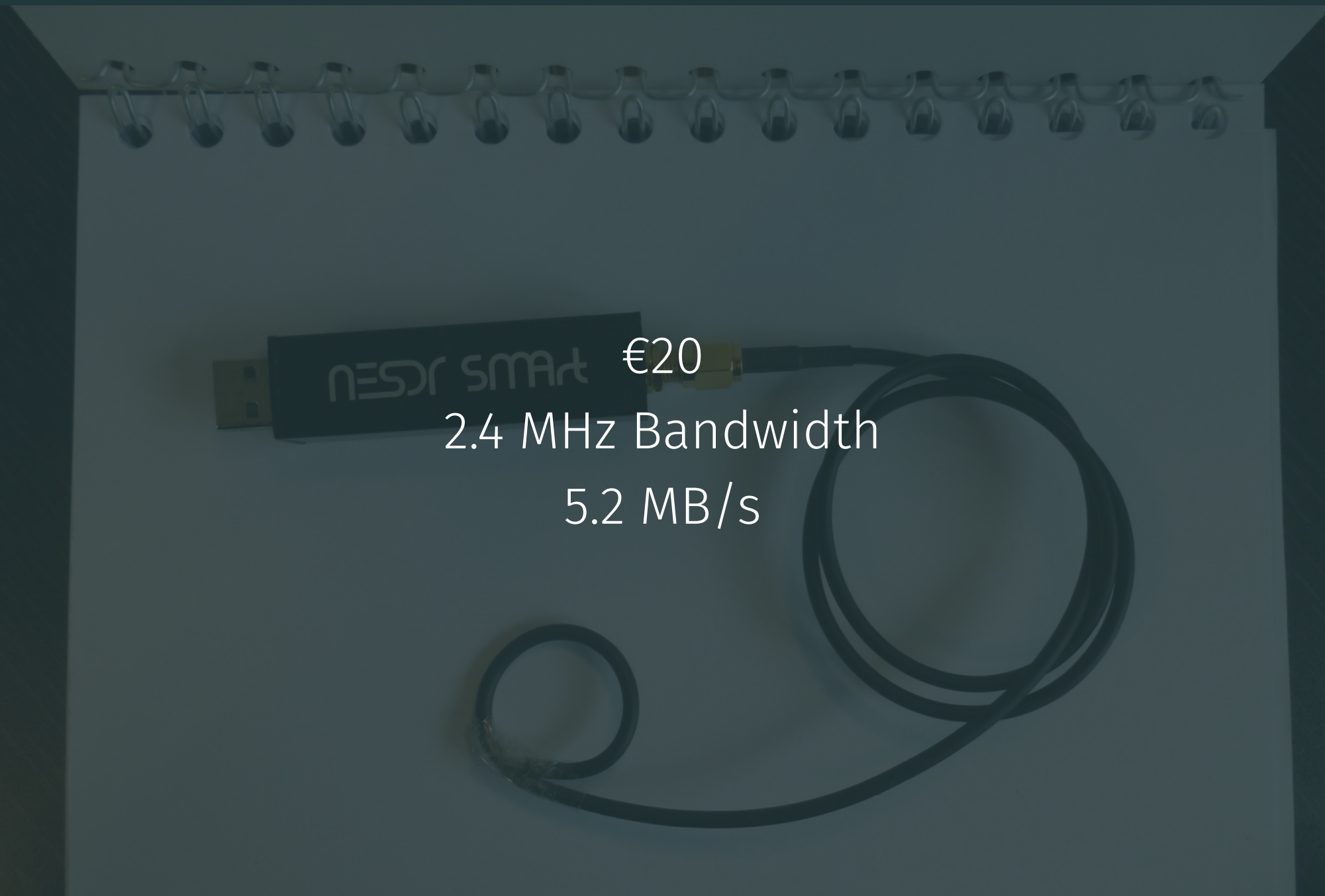

Model of leakage

Measuring the field

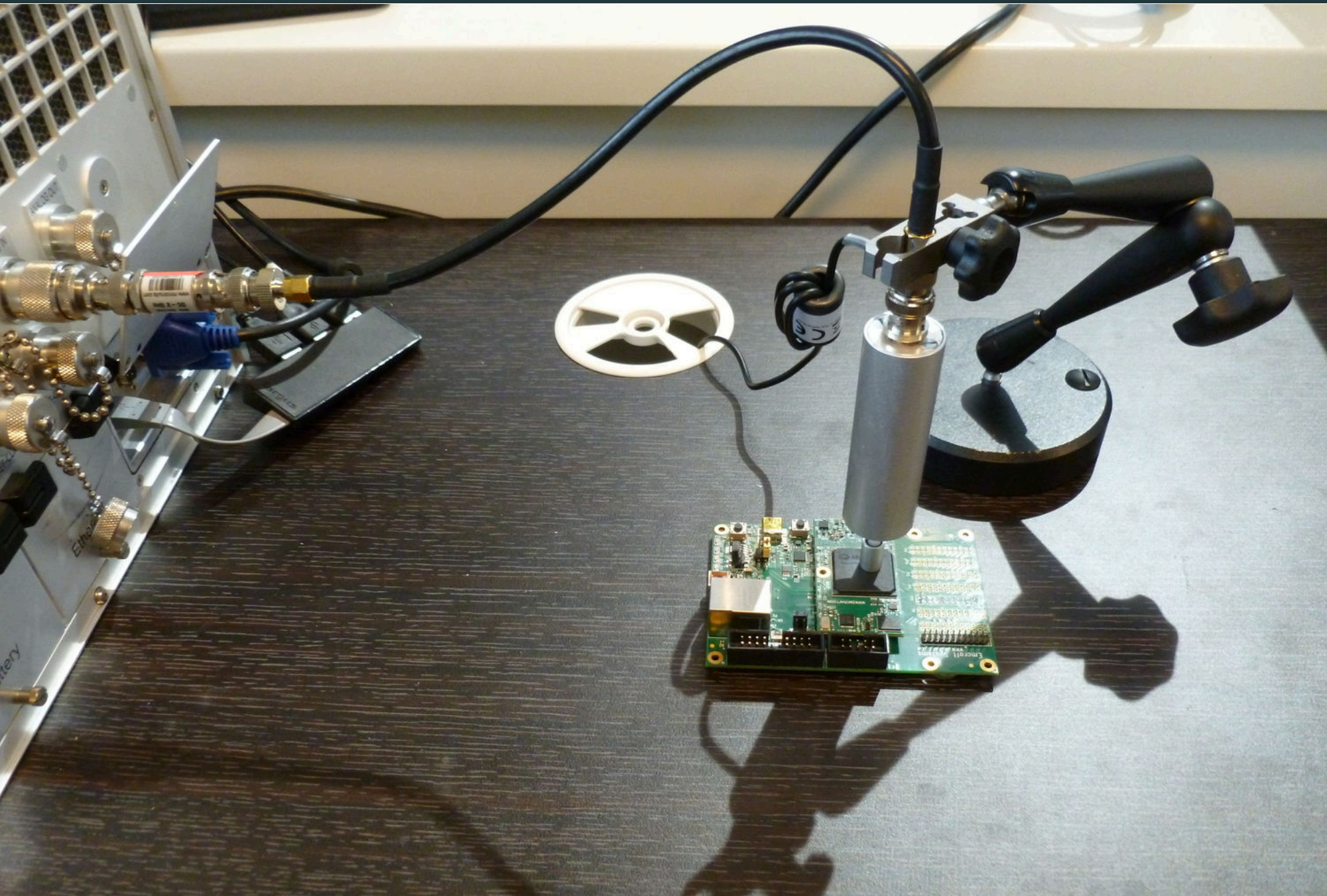
Measuring the field

Recording —Low-end

Recording —Low-end

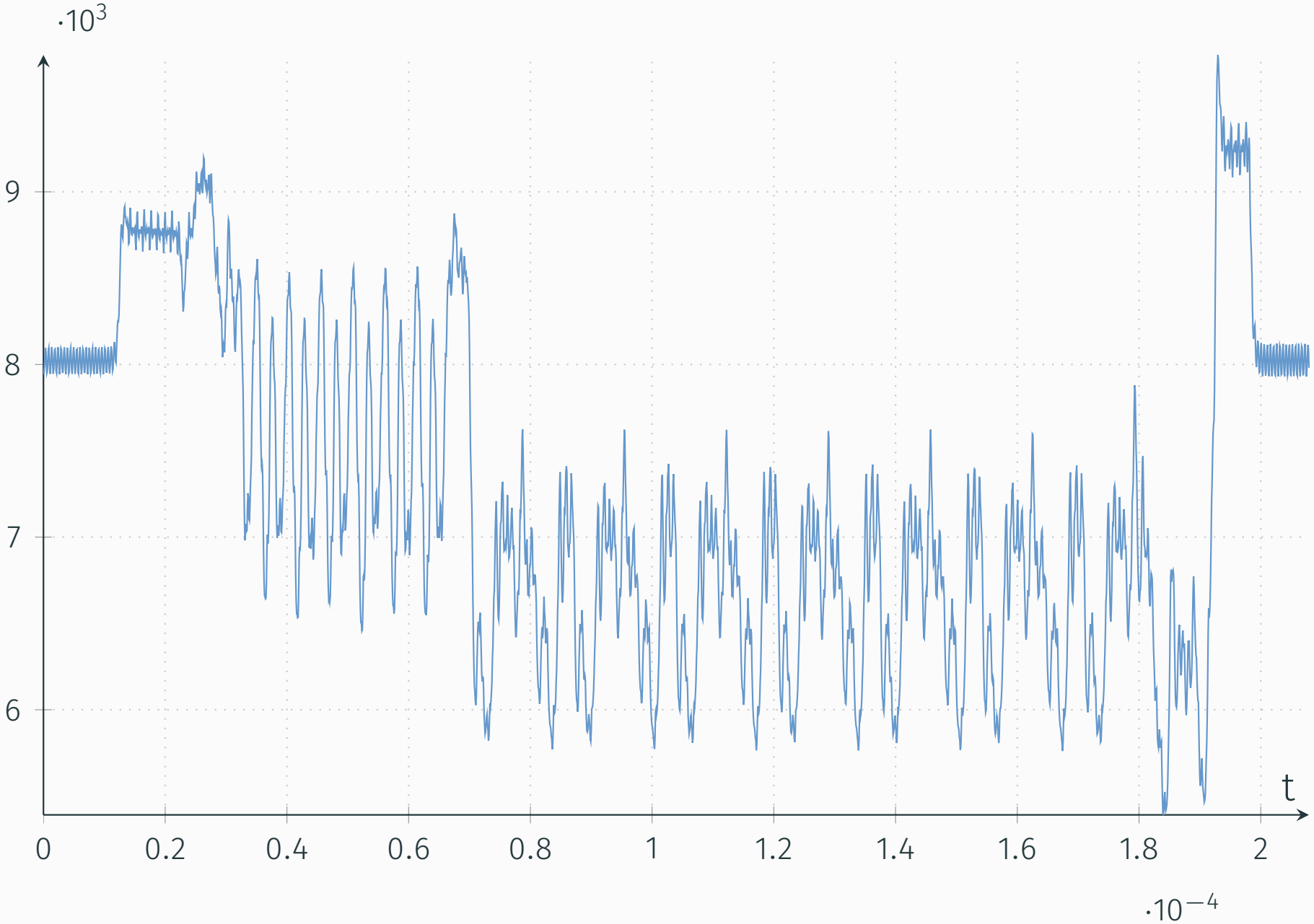


Positioning

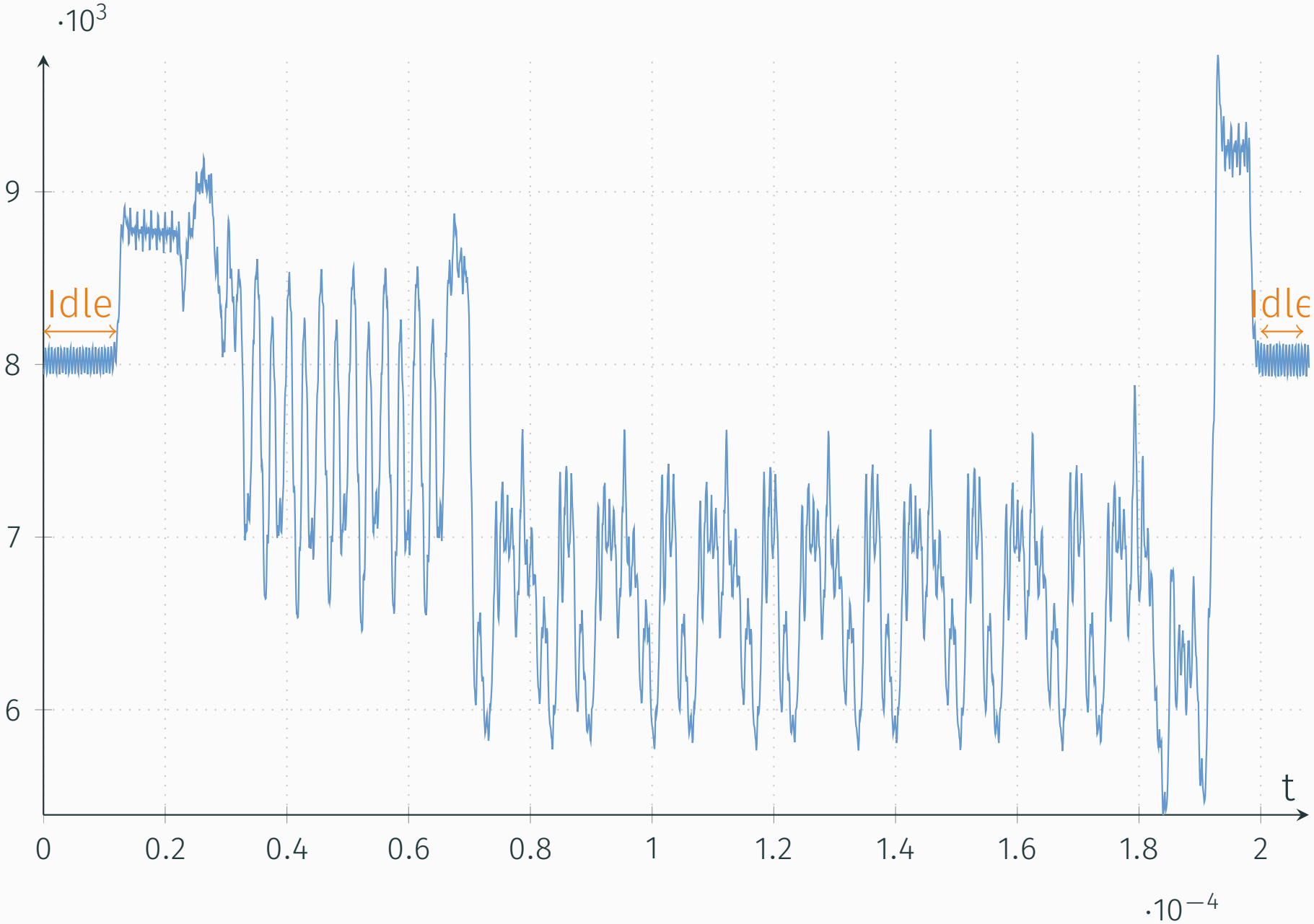


Positioning

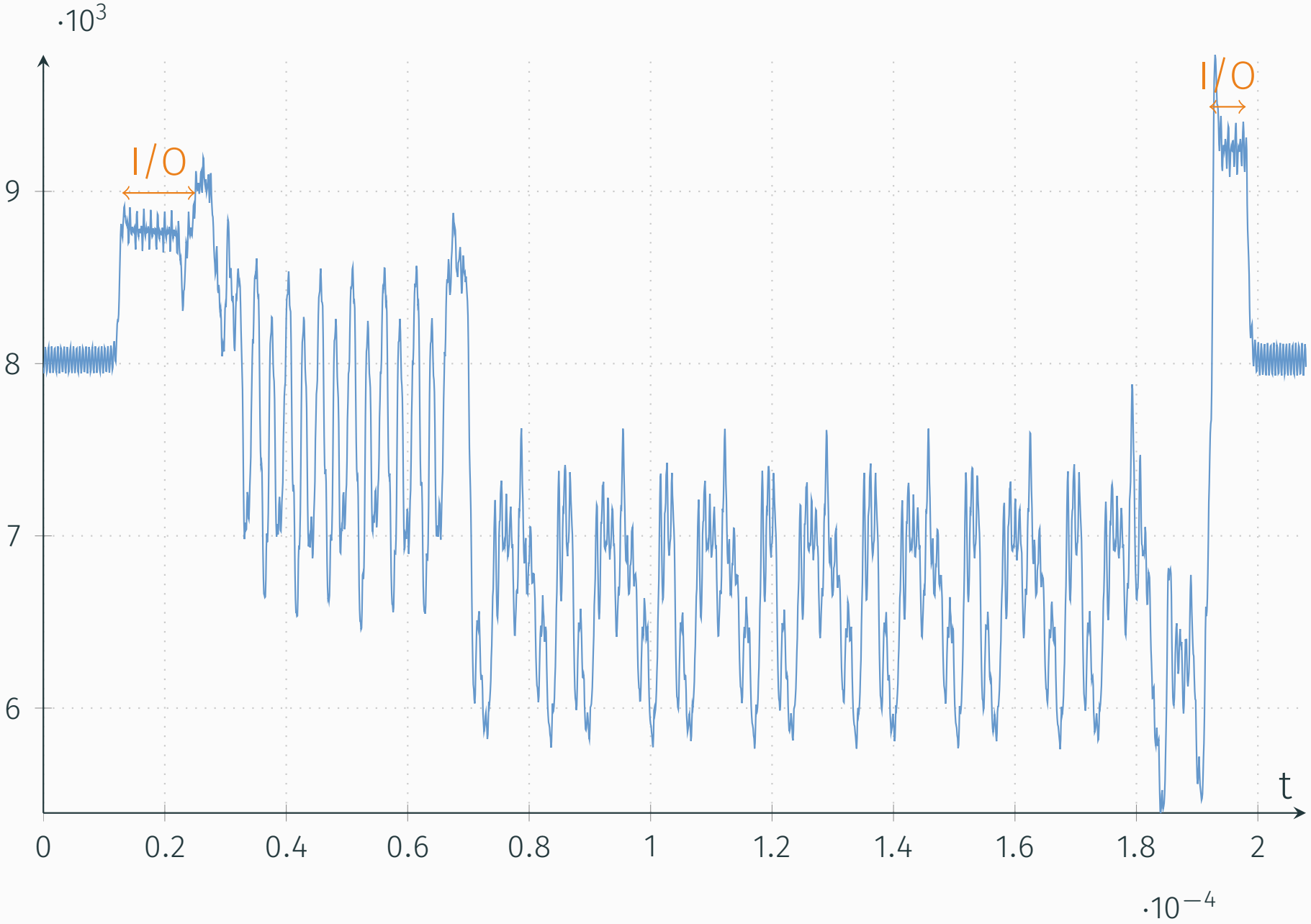
ARM Software Trace



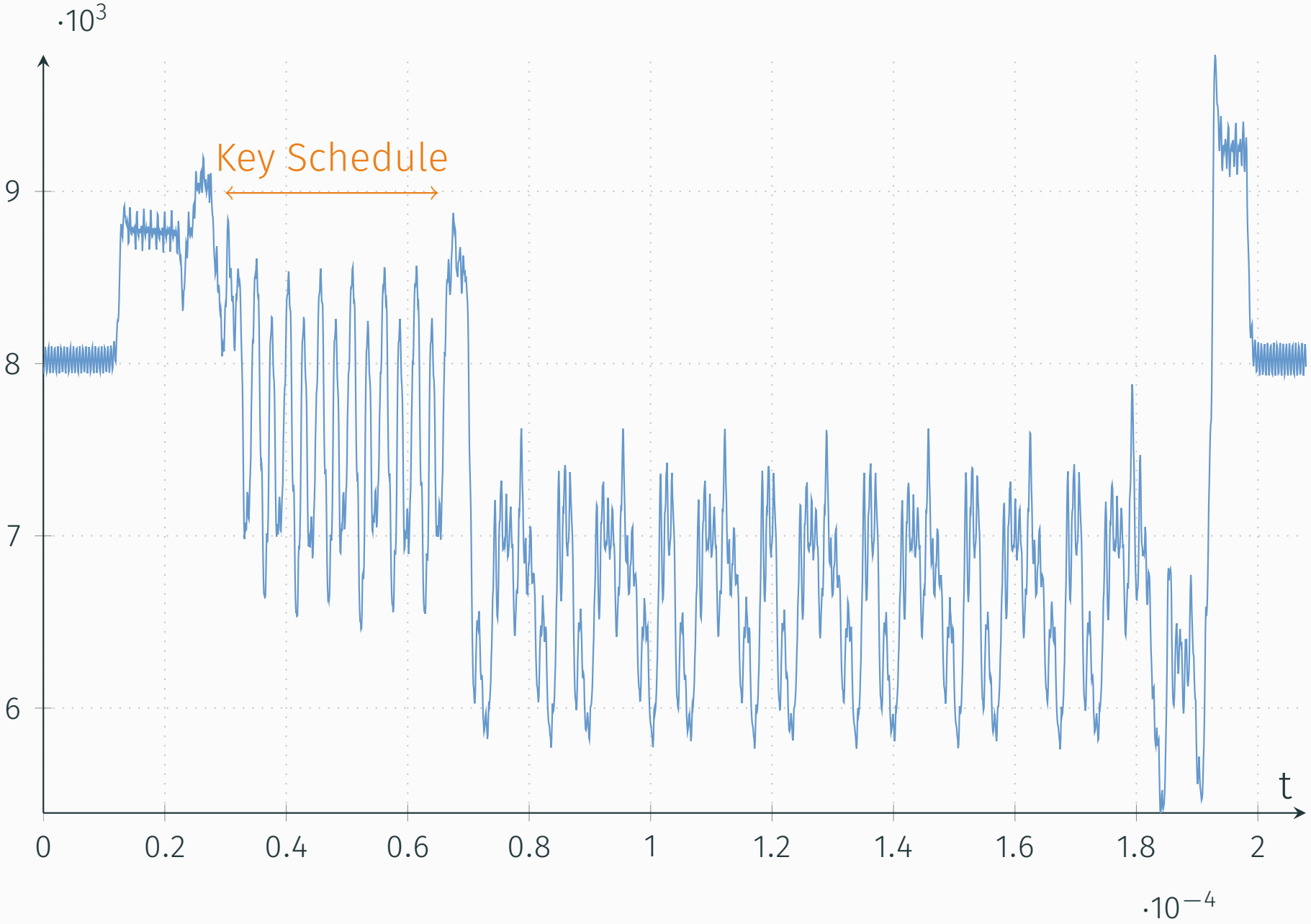
ARM Software Trace



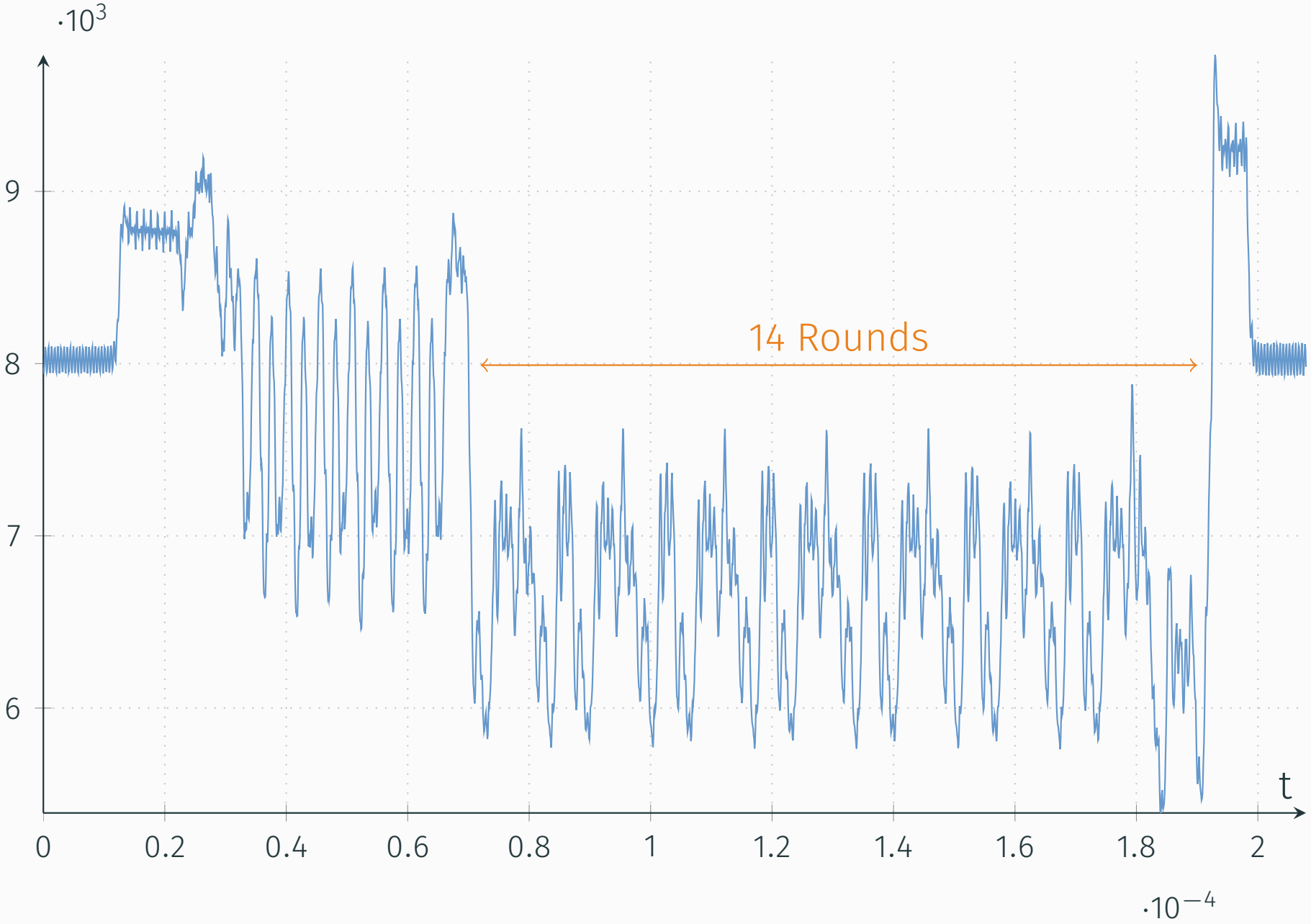
ARM Software Trace



ARM Software Trace



ARM Software Trace



Nice... but still, how do you get a key?

(This part is just existing SCA techniques)

Our trace is related to “power consumption”.

$$P \approx P_{static} + P_{noise} + P_{data} + P_{operation}$$

$$P \approx P_{static} + P_{noise} + P_{data} + P_{operation}$$

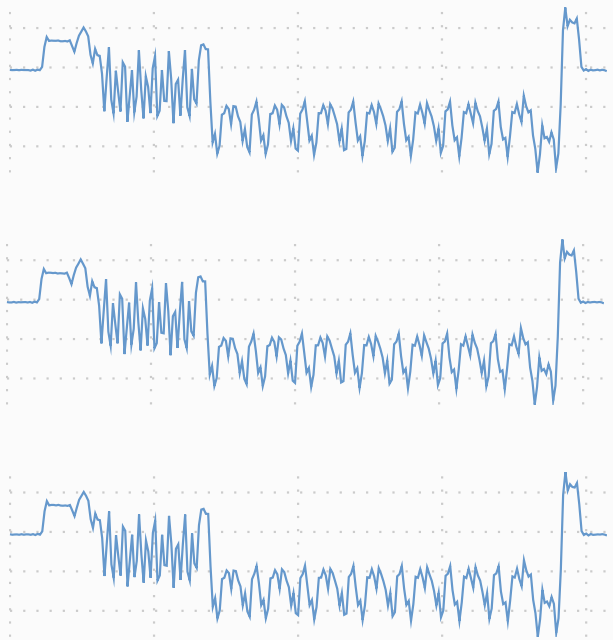
Correlation Intro

Input Byte

0011 0101

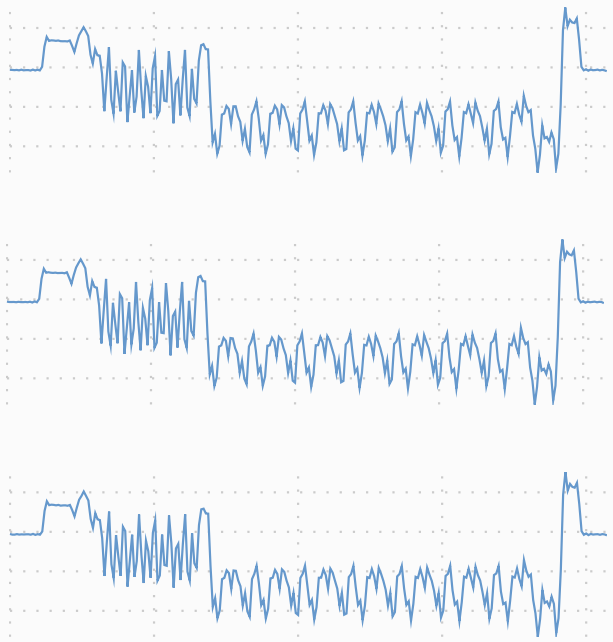
0100 0000

1101 1110



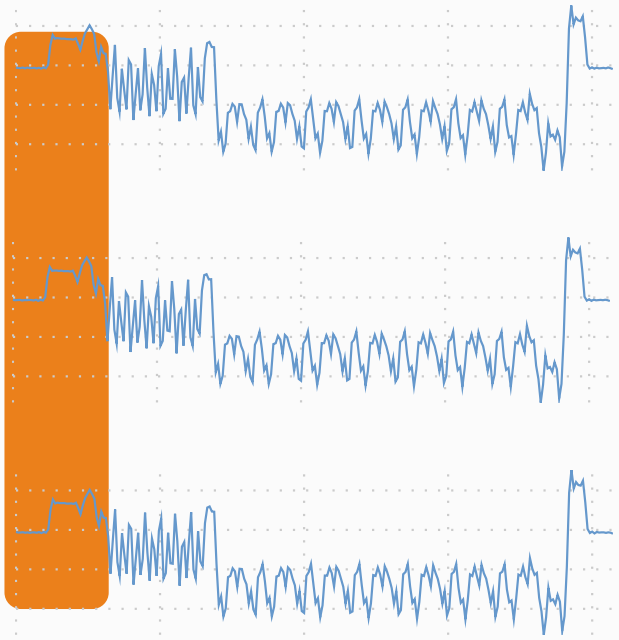
Correlation Intro

Input Byte	(# '1' bits) Power estimate
`0011 0101`	4
`0100 0000`	1
`1101 1110`	6



Correlation Intro

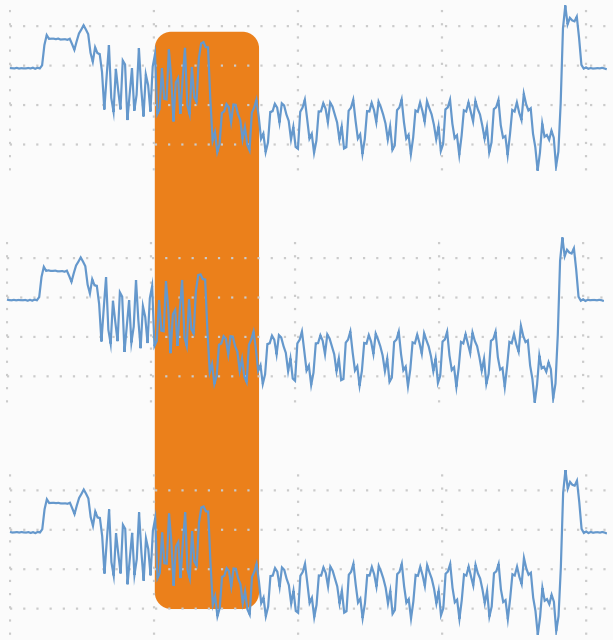
Input Byte	(# '1' bits) Power estimate
`0011 0101`	4
`0100 0000`	1
`1101 1110`	6



Correlate!

Correlation Intro

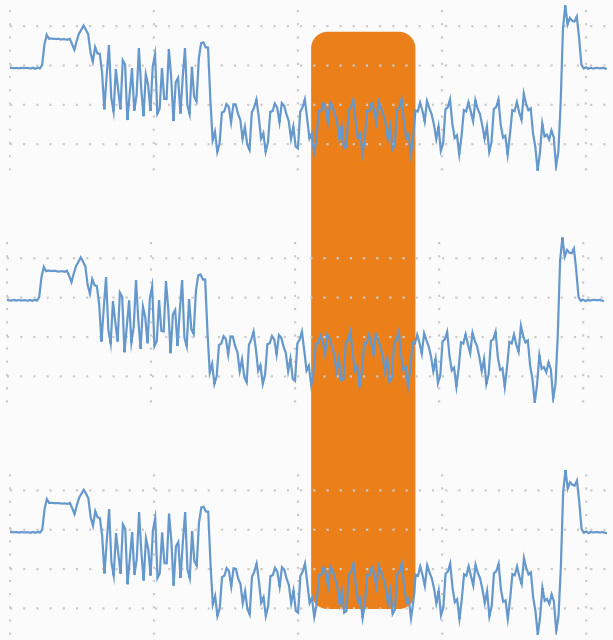
Input Byte	(# '1' bits) Power estimate
`0011 0101`	4
`0100 0000`	1
`1101 1110`	6



Correlate!

Correlation Intro

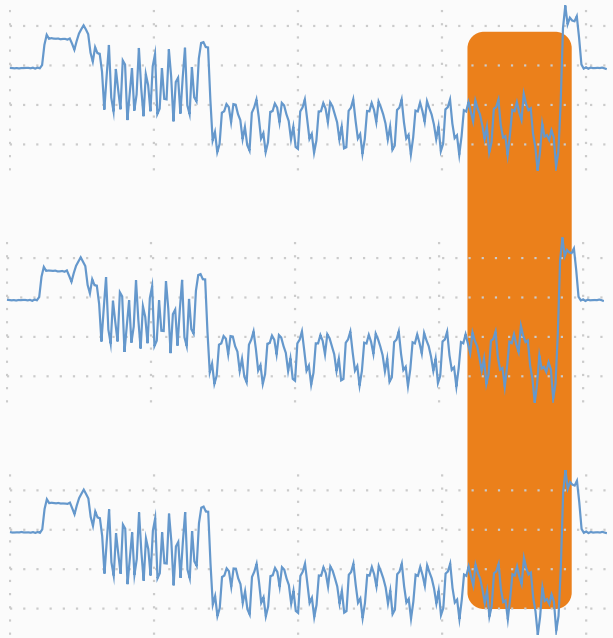
Input Byte	(# '1' bits) Power estimate
0011 0101	4
0100 0000	1
1101 1110	6



Correlate!

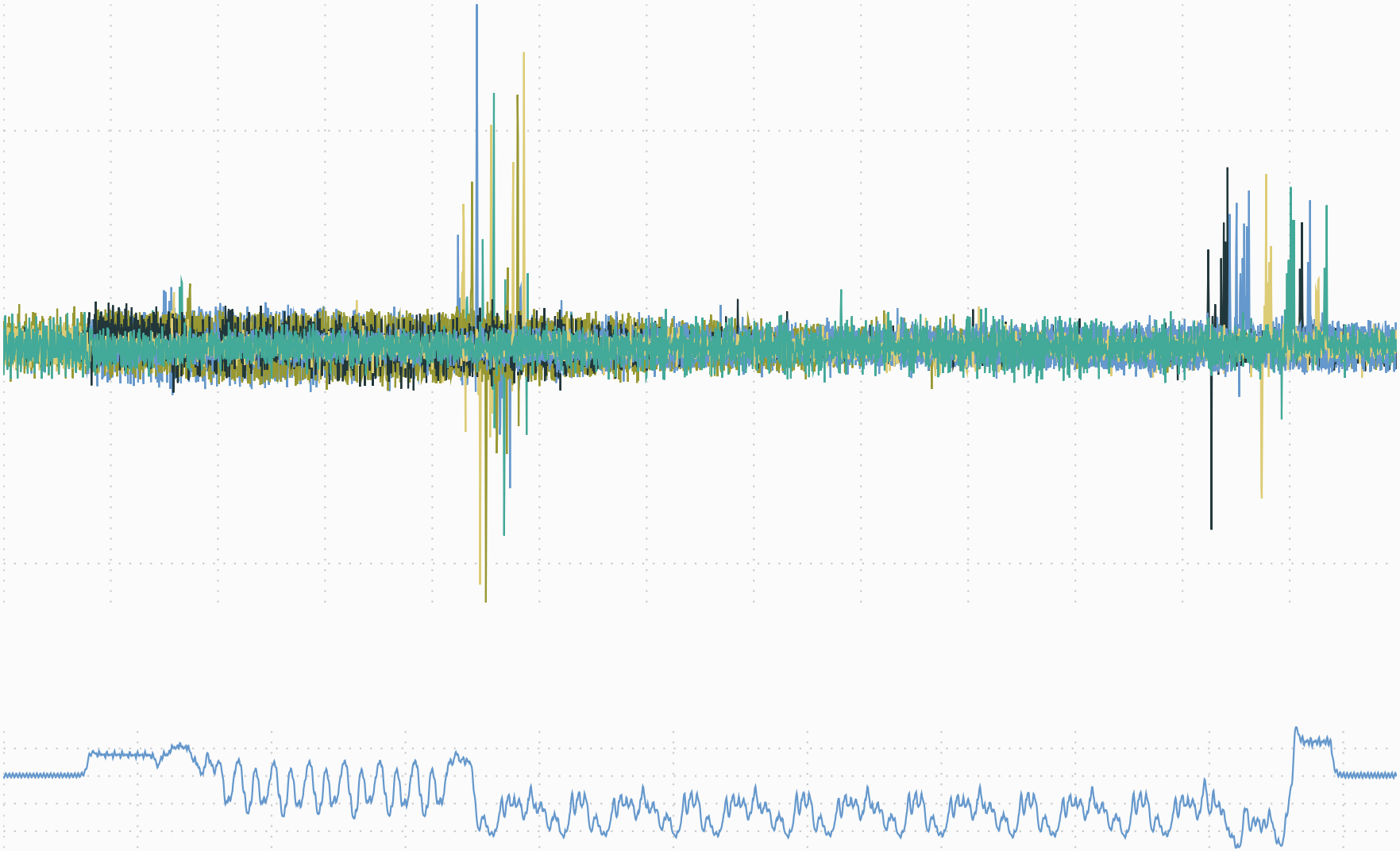
Correlation Intro

Input Byte	(# '1' bits) Power estimate
`0011 0101`	4
`0100 0000`	1
`1101 1110`	6



Correlate!

I/O Correlation



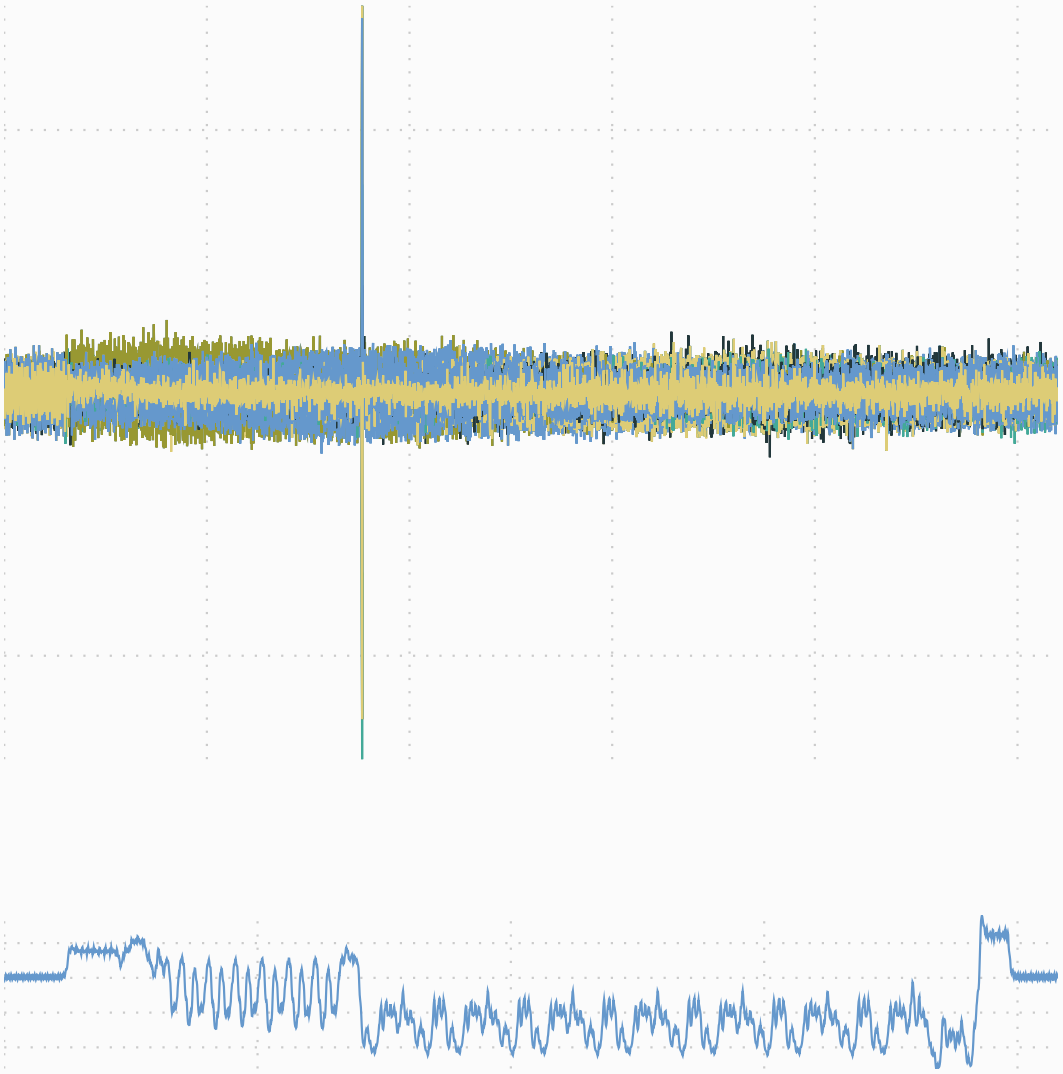
We can detect data!

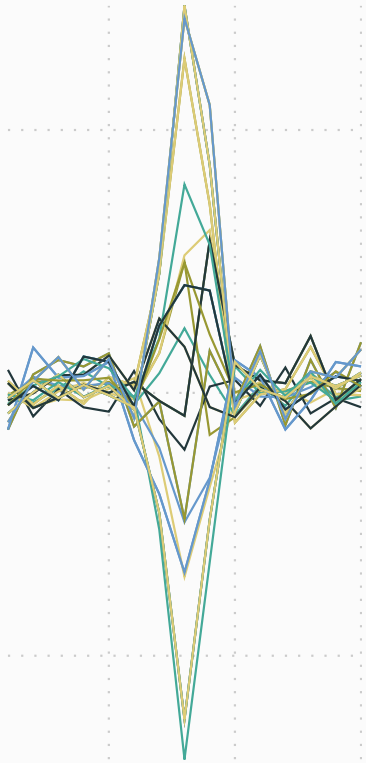
Let’s find a value using 1 key byte
and correlate for all 256 possibilities

32×2^8 guesses
(instead of 2^{256})

8192 guesses
(instead of 10^{77})

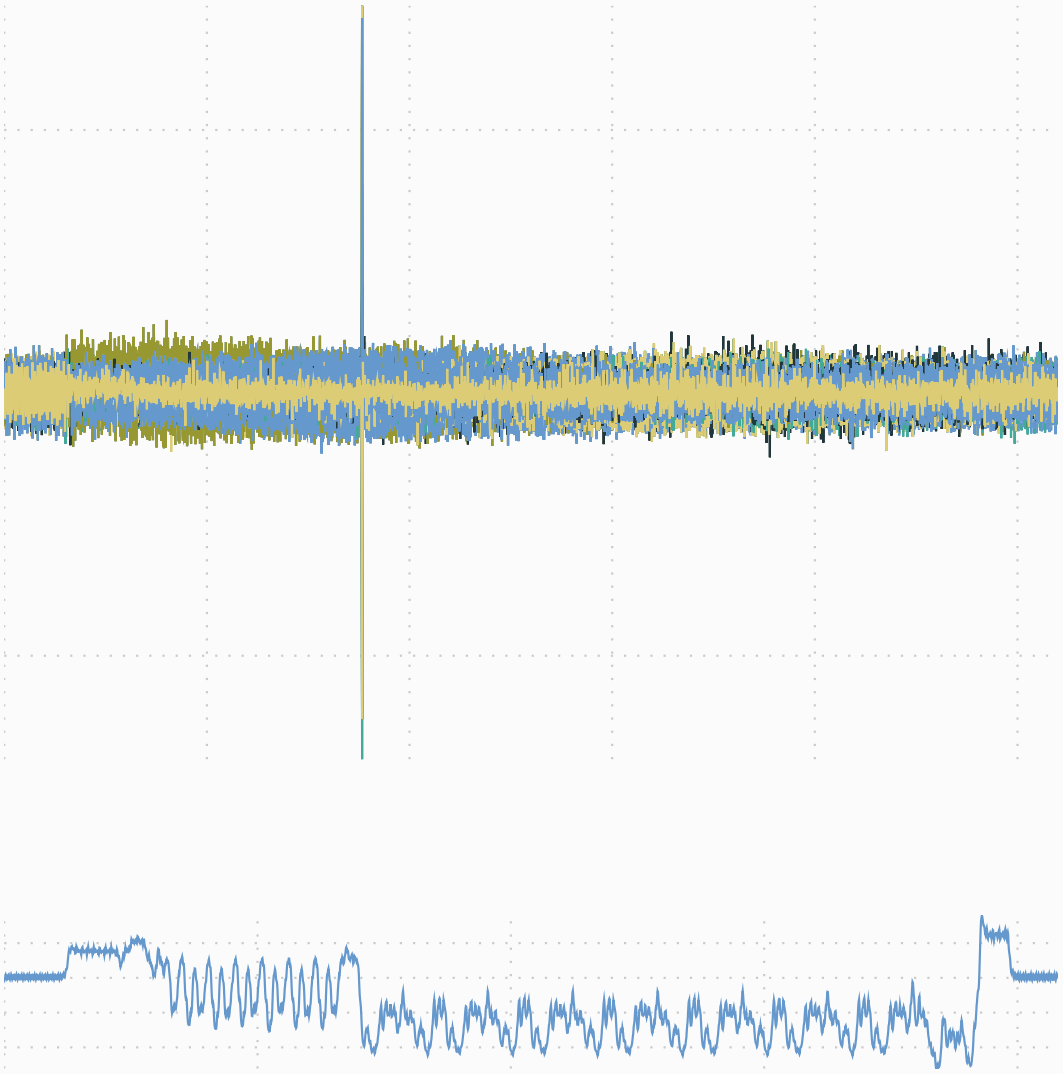
T Table Correlation

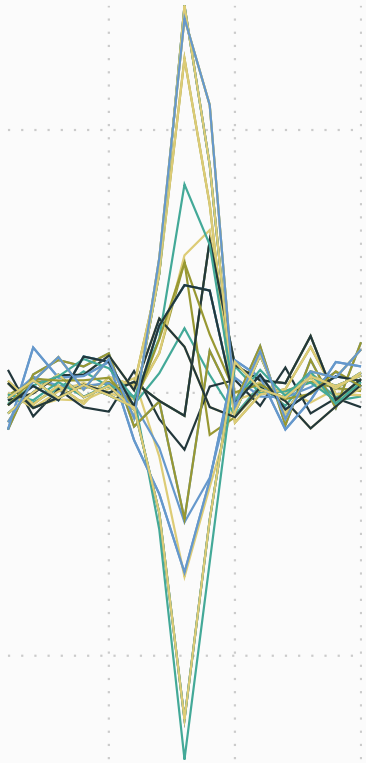




Known-key bitwise on T Table lookup

T Table Correlation





Easy & works
(but could do better)

Known-key bitwise on T Table lookup

“You know can addresses leak too, right?”

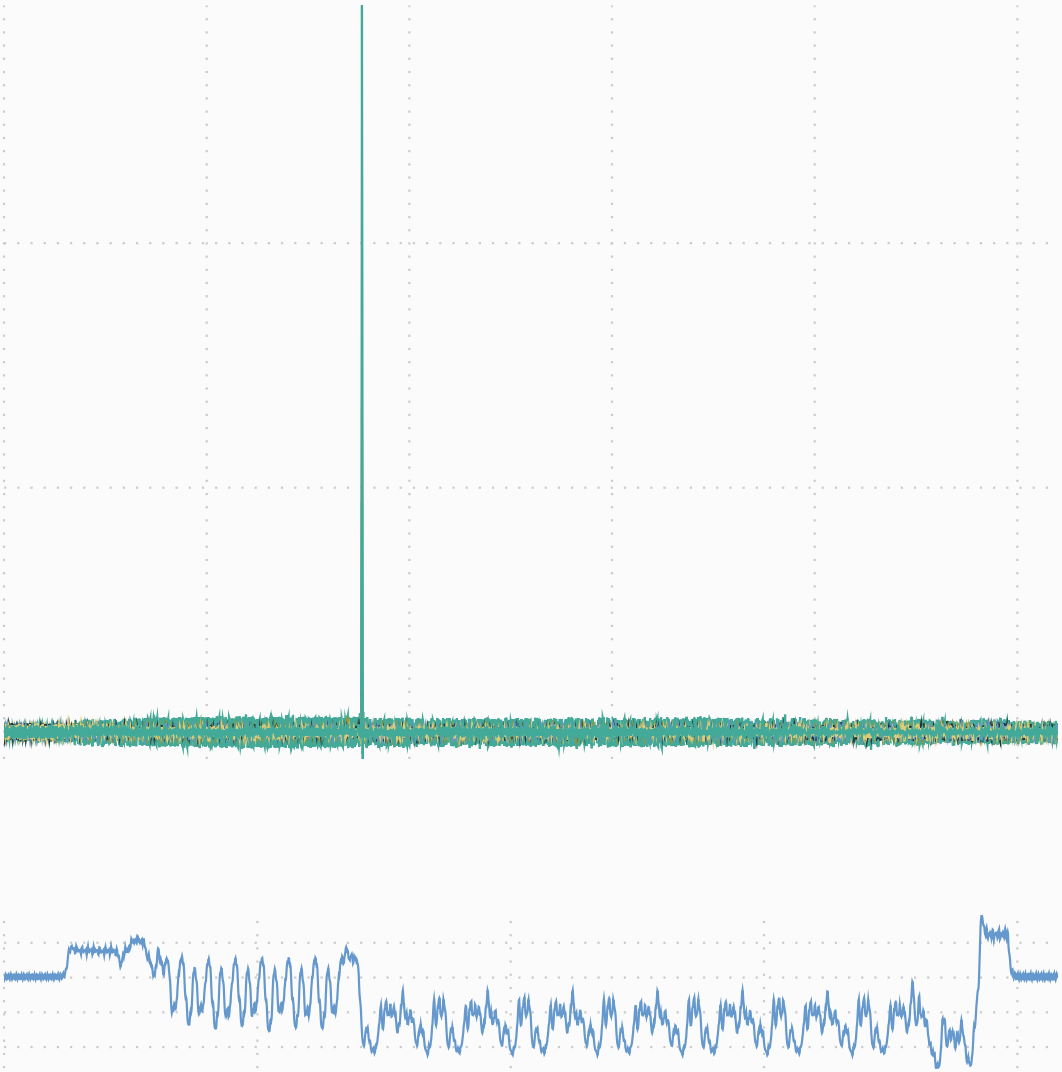
—Riscure, 2017

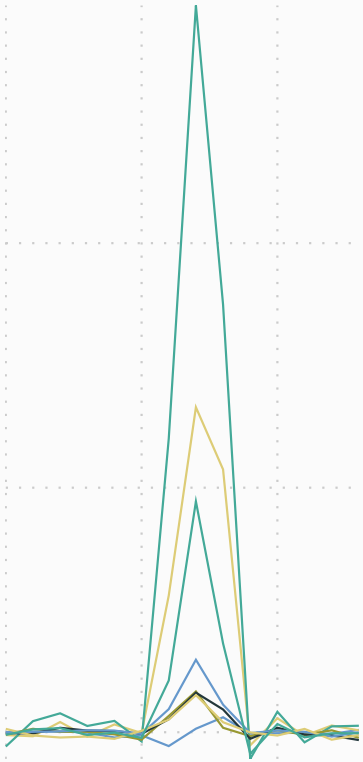
“*Oh… thanks.*” — Me, 2017

ARM T Table Addresses

```
1508: 4b4a          ldr   r3, [pc, #296]
150a: 681b          ldr   r3, [r3, #0]
150c: 0e1b          lsrs  r3, r3, #24
150e: 4a4d          ldr   r2, [pc, \#308]
1510: f852 2023     ldr.w r2, [r2, r3, lsl #2]
1514: 4b48          ldr   r3, [pc, #288]
1516: 681b          ldr   r3, [r3, #0]
1518: 0c1b          lsrs  r3, r3, #16
151a: b2db          uxtb  r3, r3
151c: 494a          ldr   r1, [pc, #296]
```

T Table Address Correlation

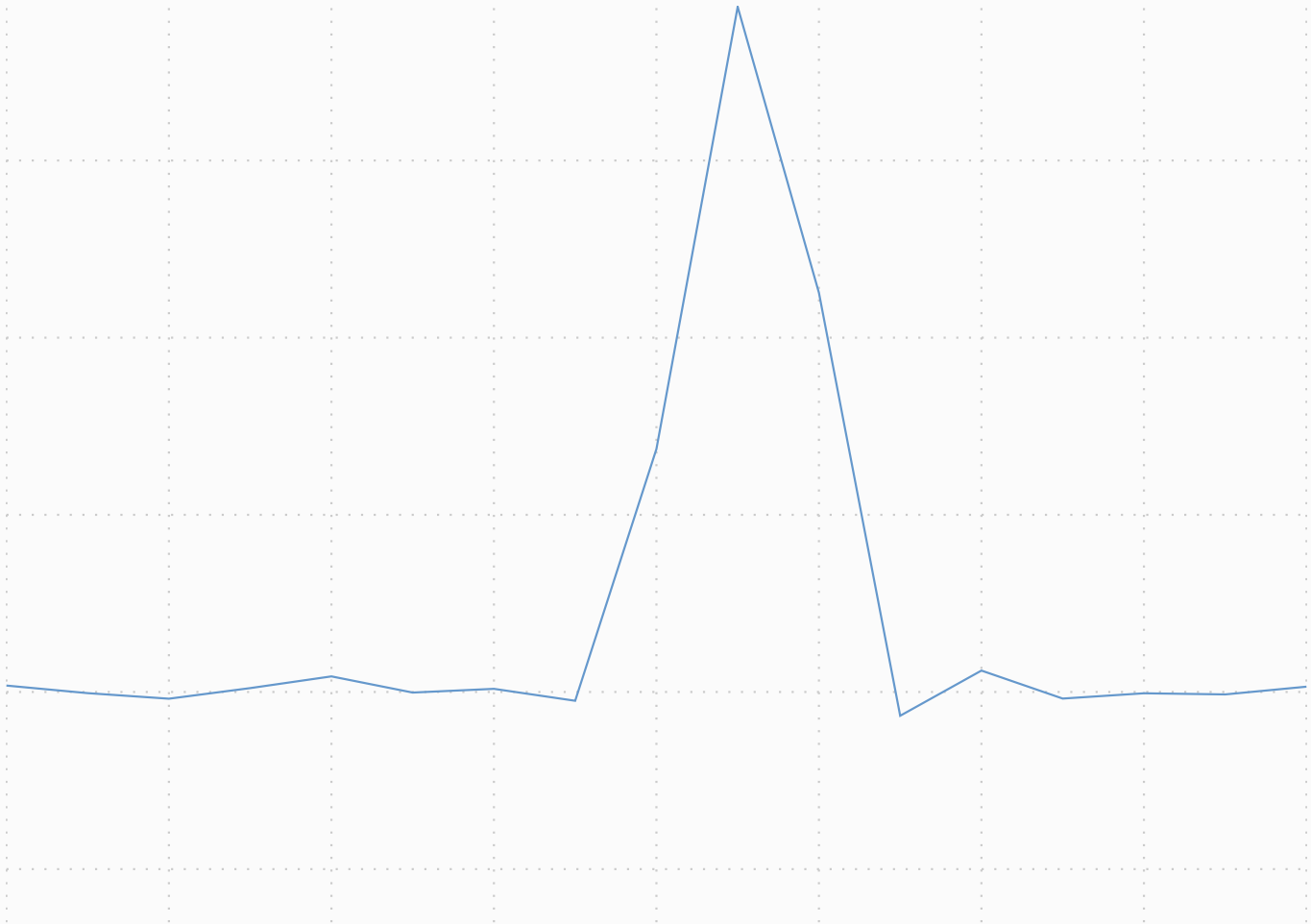




Known-key bitwise on T Table lookup address $\oplus$ previous address

If the correlation for the *correct* key byte is biggest, we have an attack.

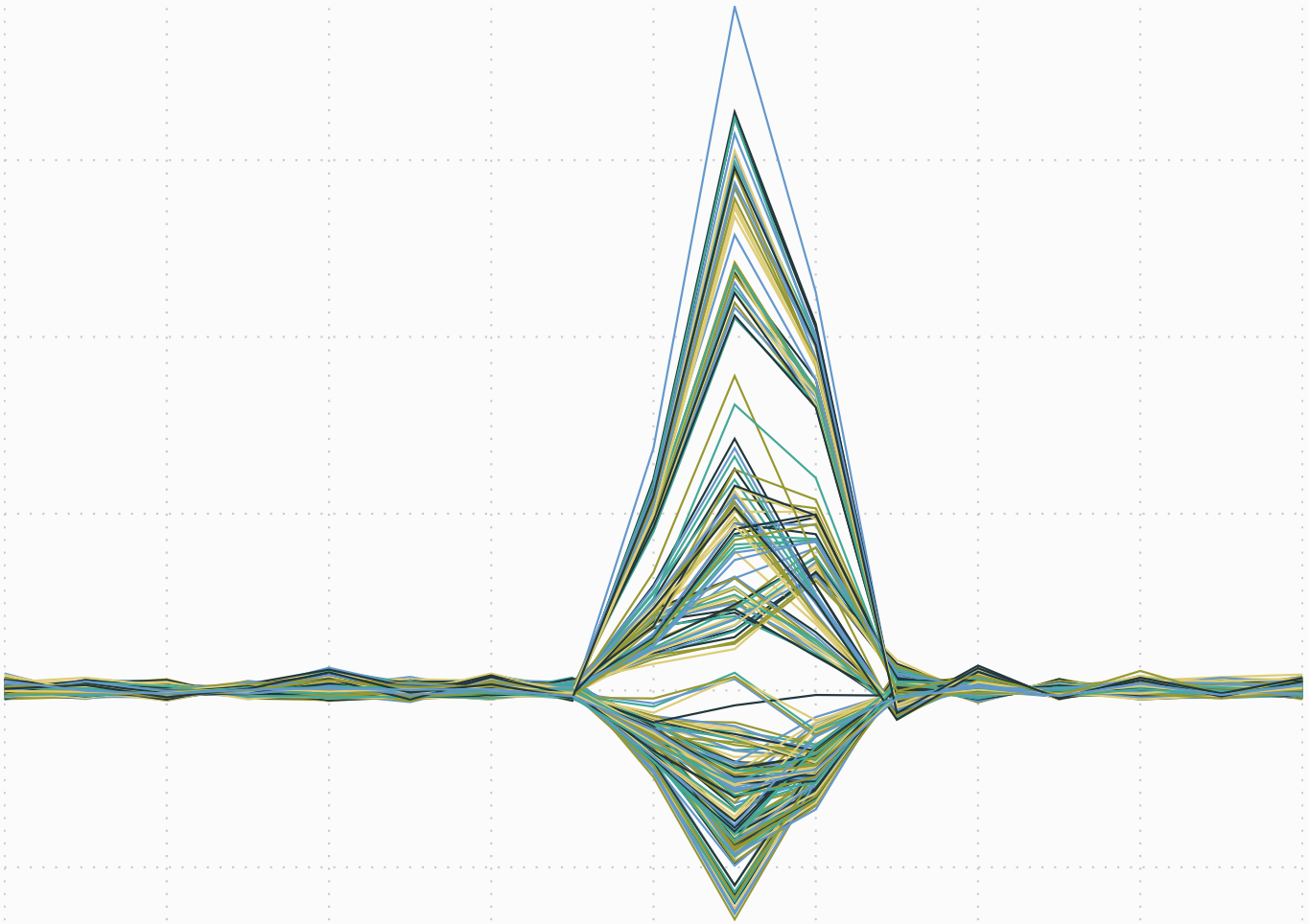
T Table Attack



Correct key byte...

HD on T Table lookup address (real attack)

T Table Attack



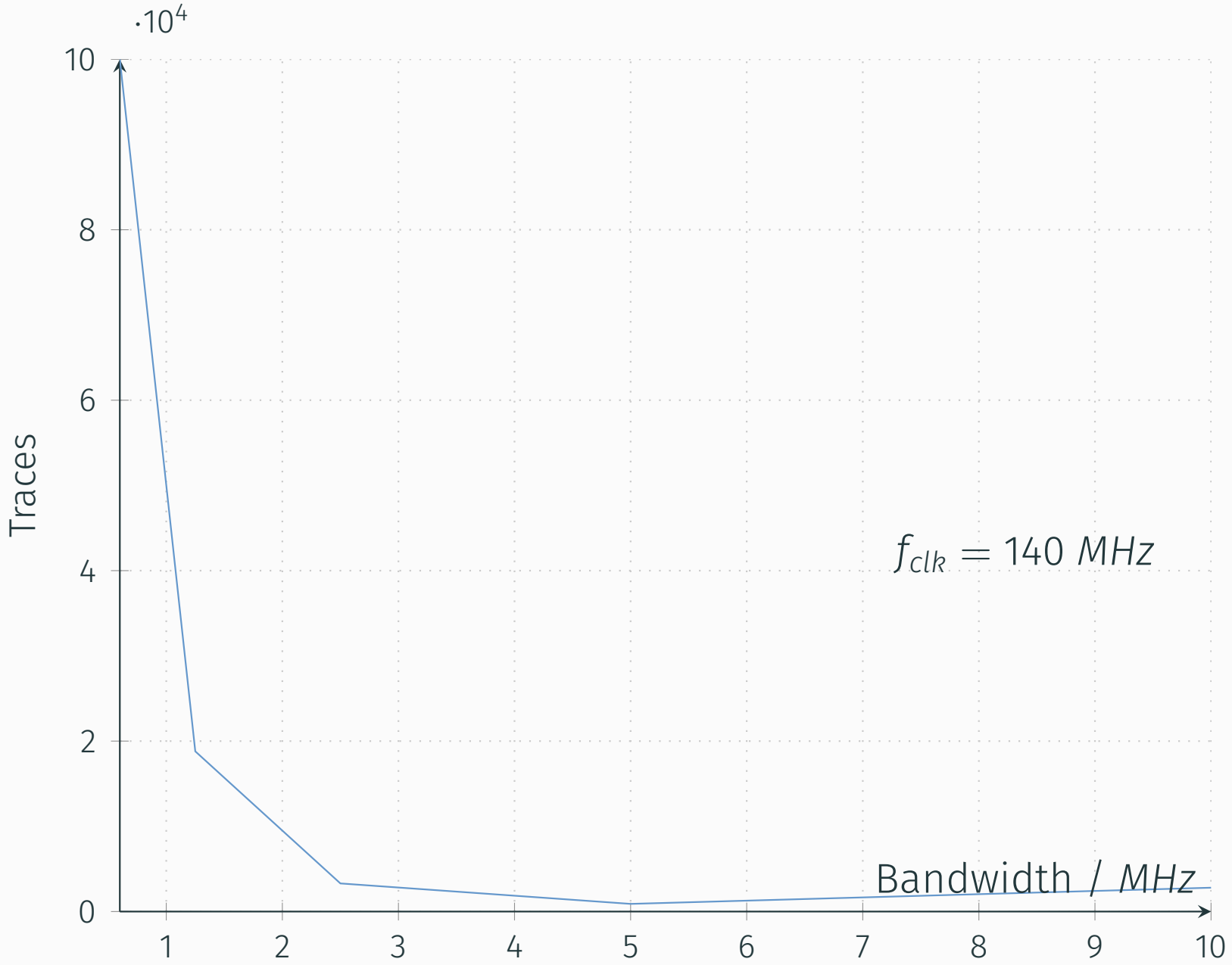
All key byte guesses. We win!

HD on T Table lookup address (real attack)

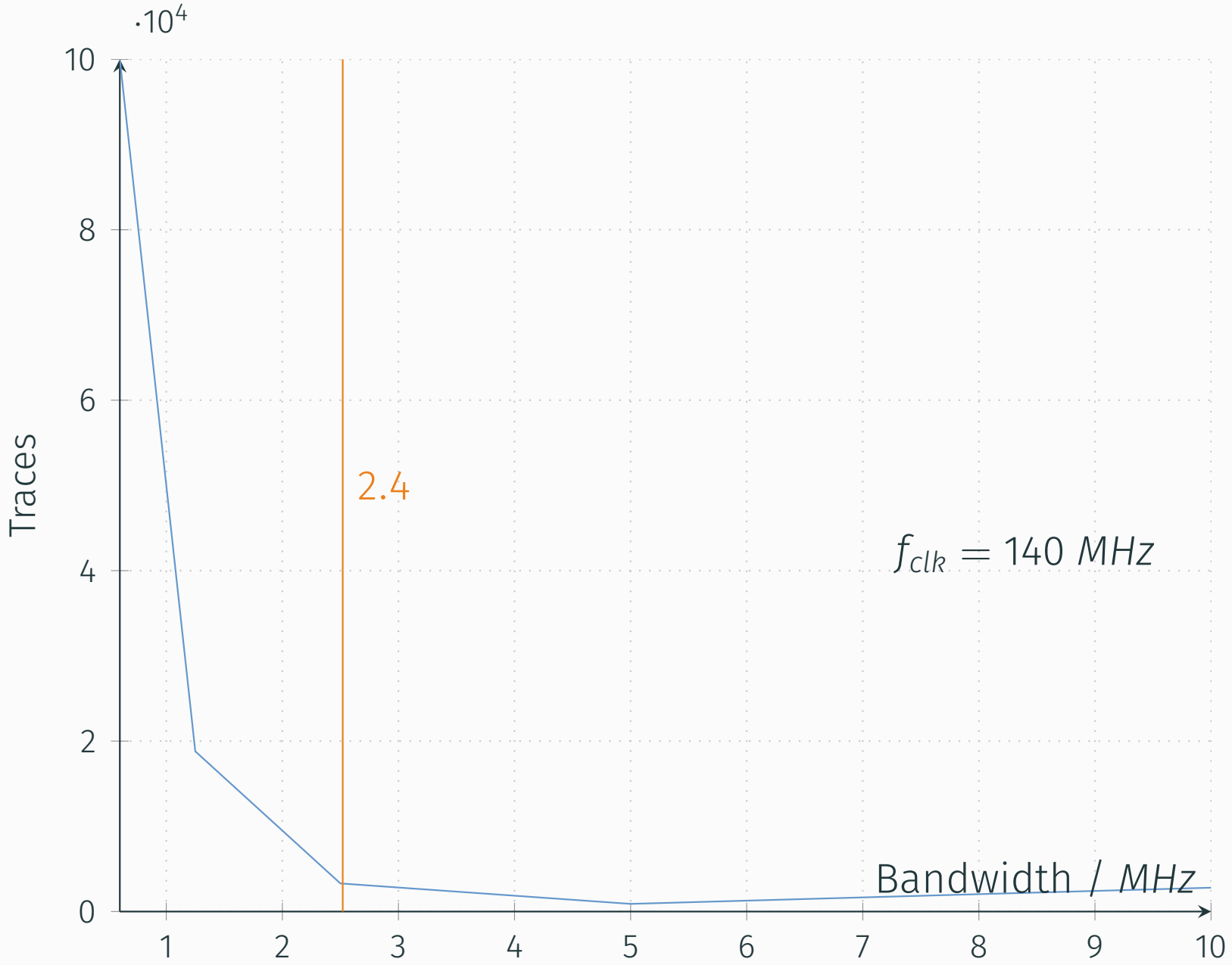
Repeat this for all 32 key bytes and we have the full key

On-package attack results

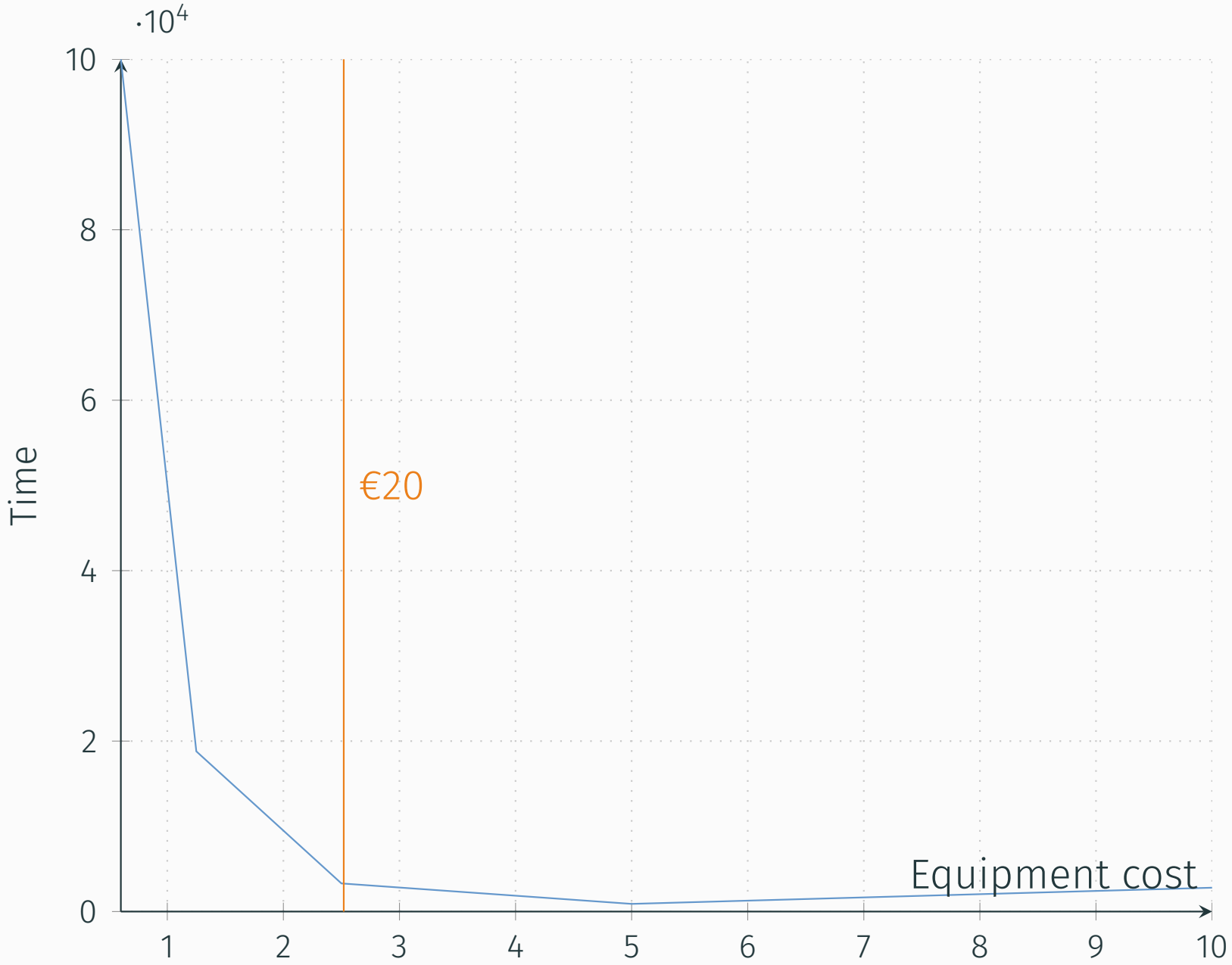
Bandwidth vs # traces



Bandwidth vs # traces



Bandwidth vs # traces



Getting some distance

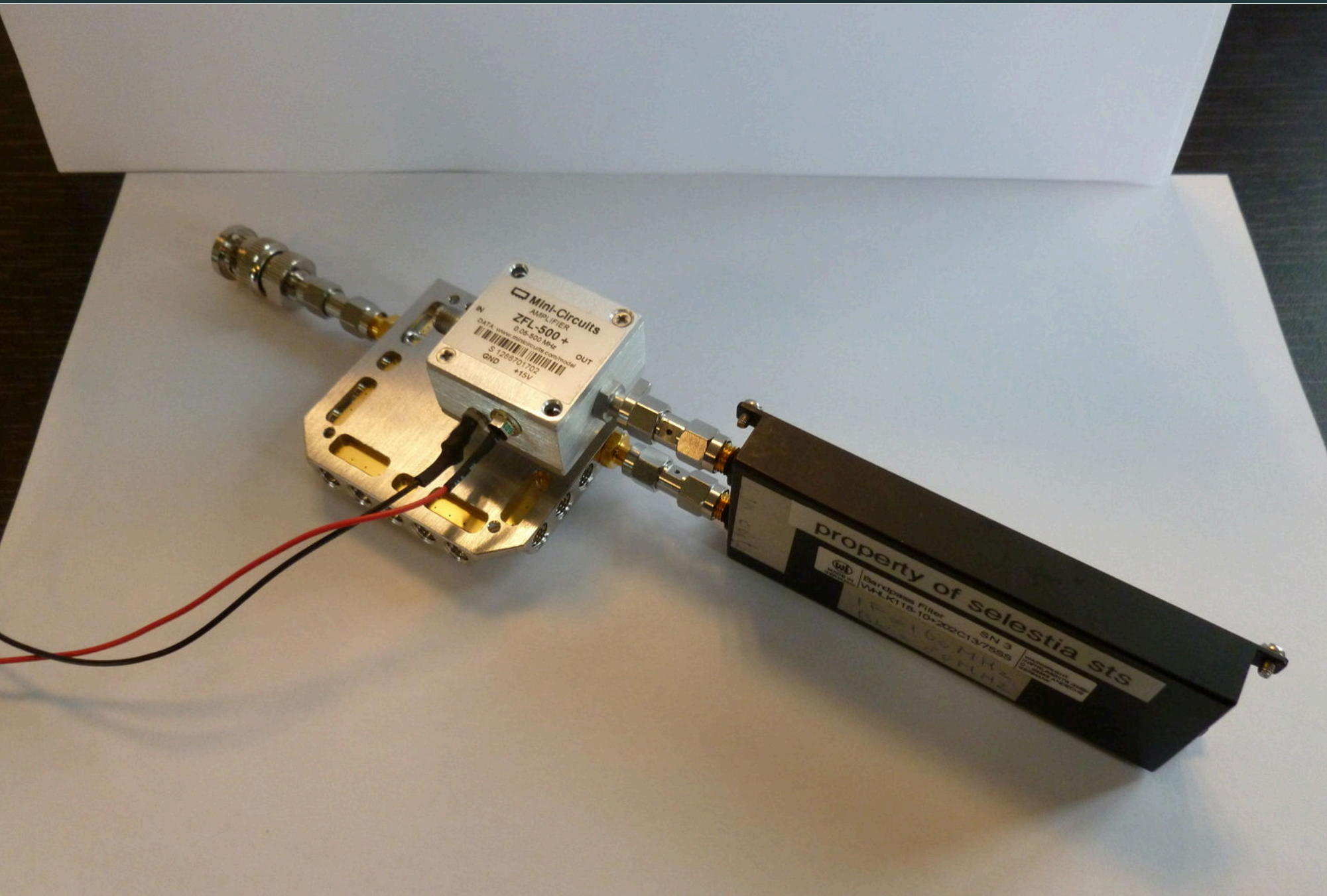
Only need to improve analogue side.

Analysis is the same.

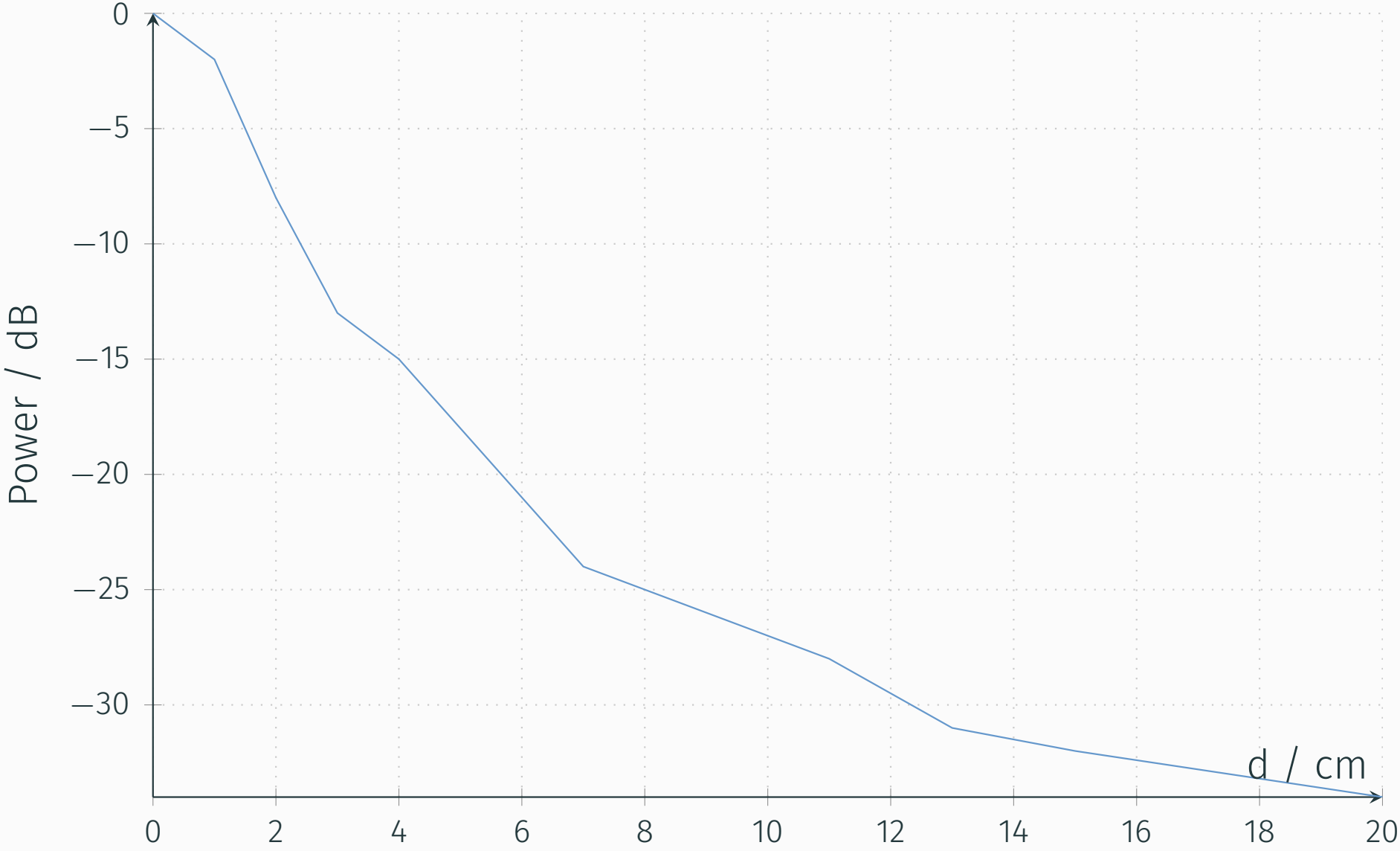
Loop size

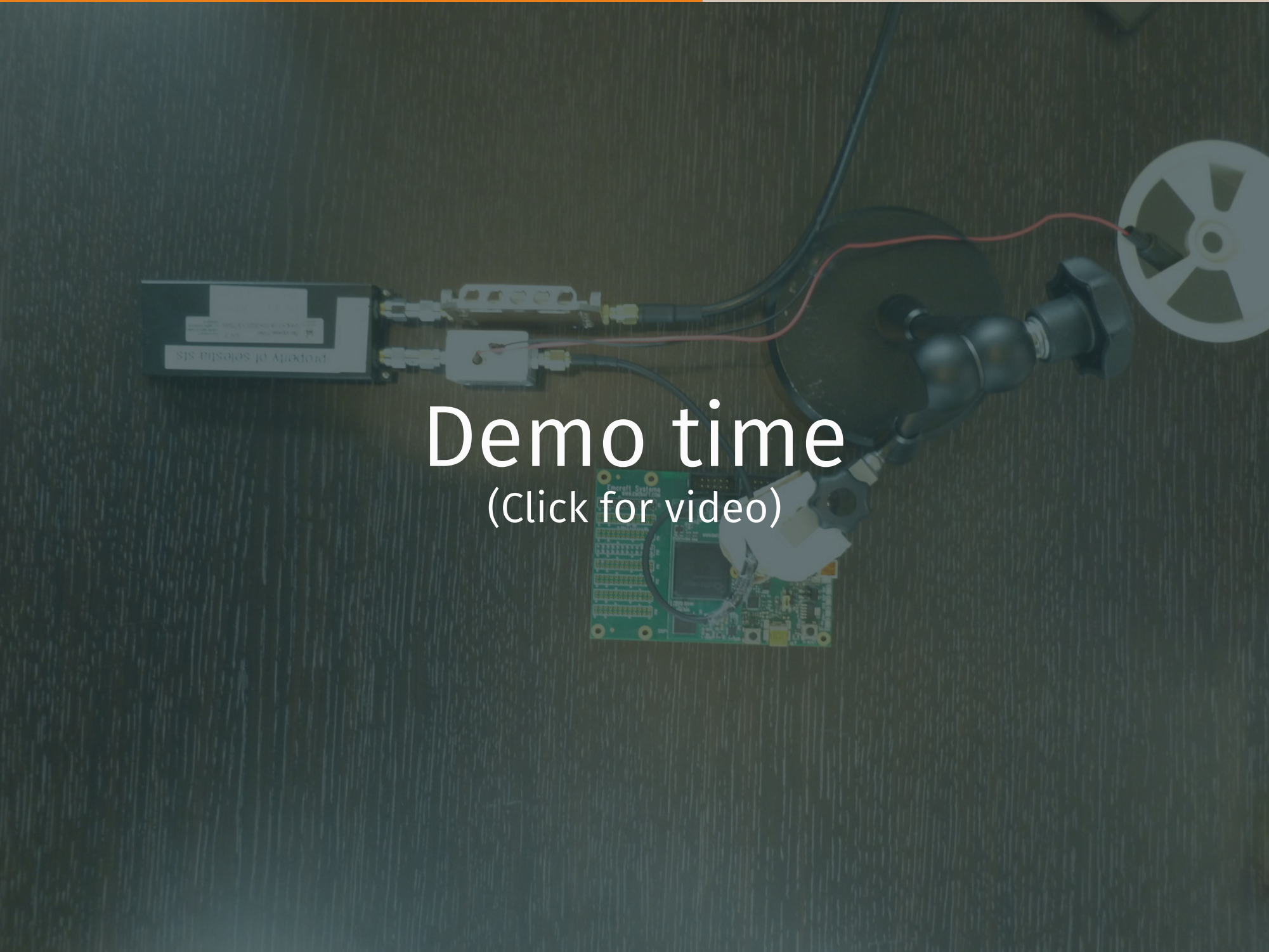
Loop size

Amplification and filtering



Small loop distance



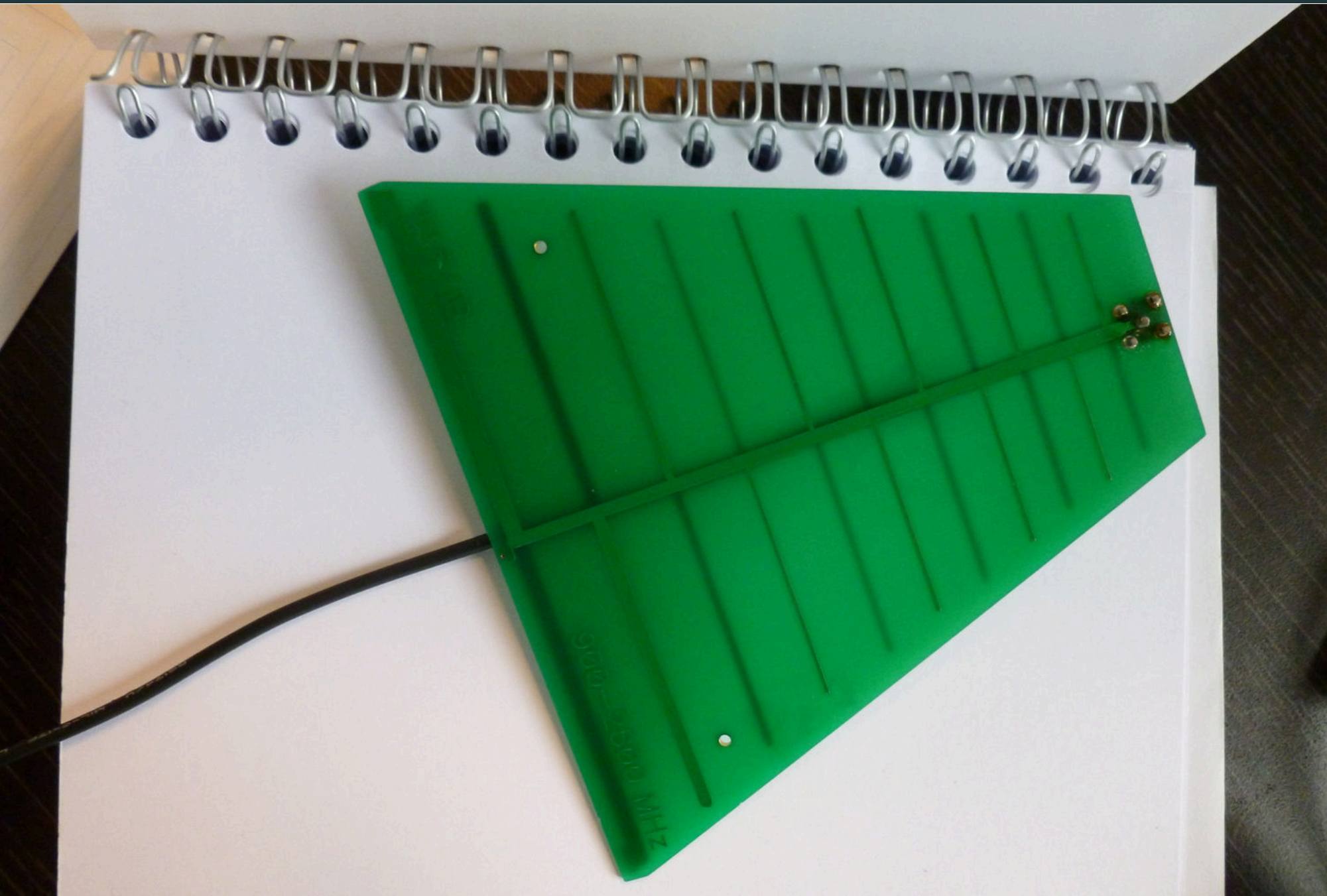


Demo time

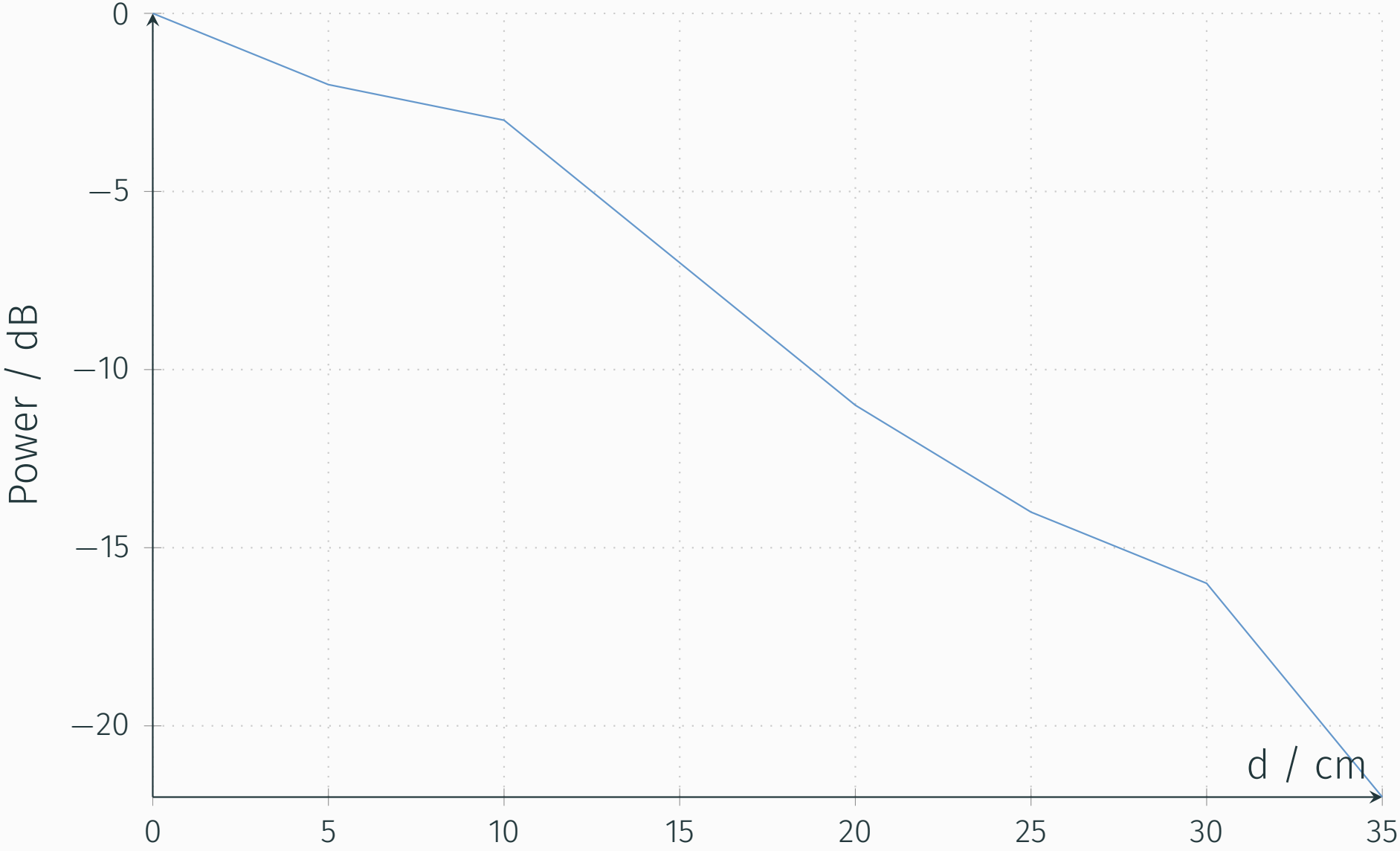
(Click for video)

Small loops are amazing for under $\approx 5\ cm$.
Won't get us to $1\ m$ though.

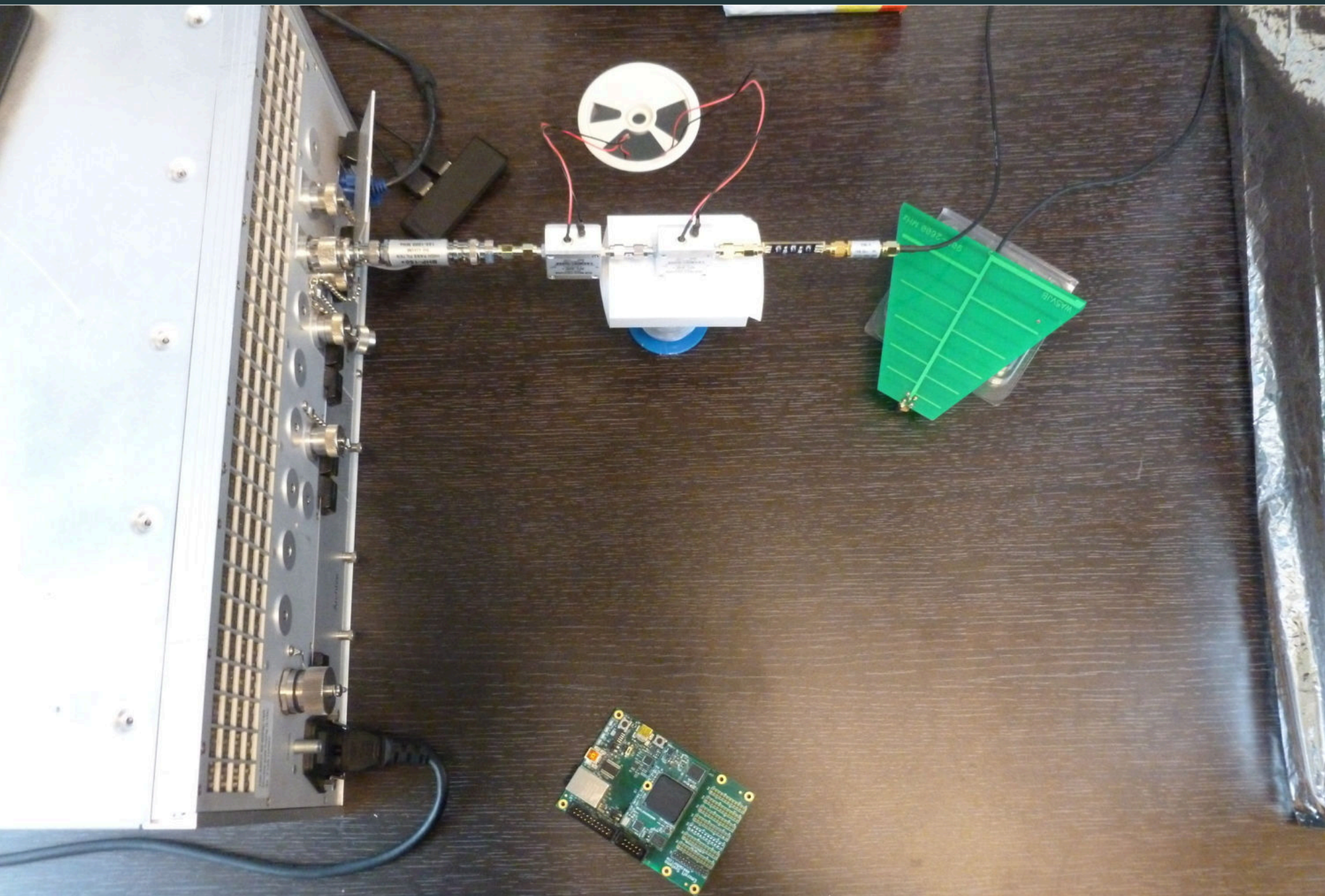
Log-periodic antenna



Log-periodic distance

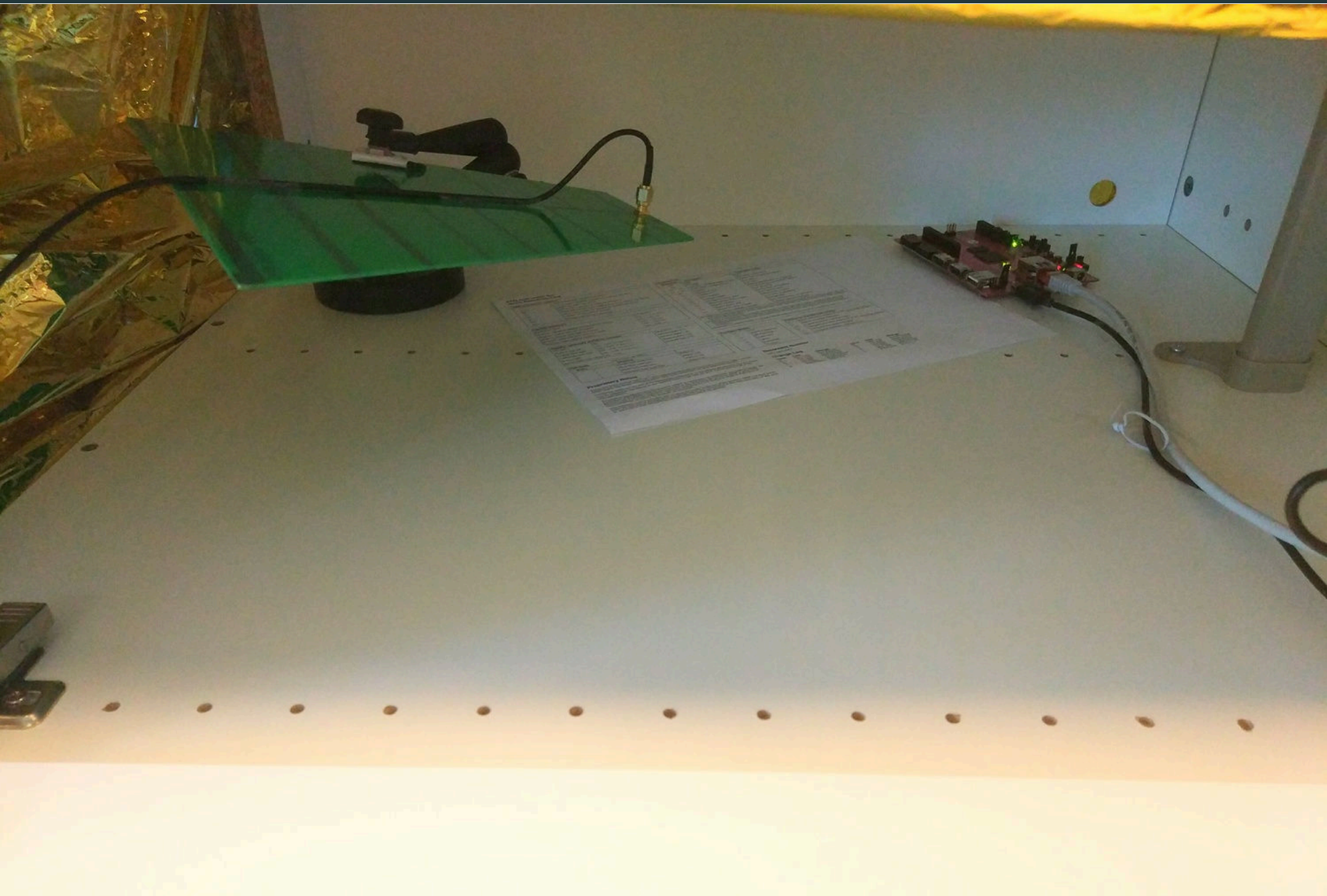


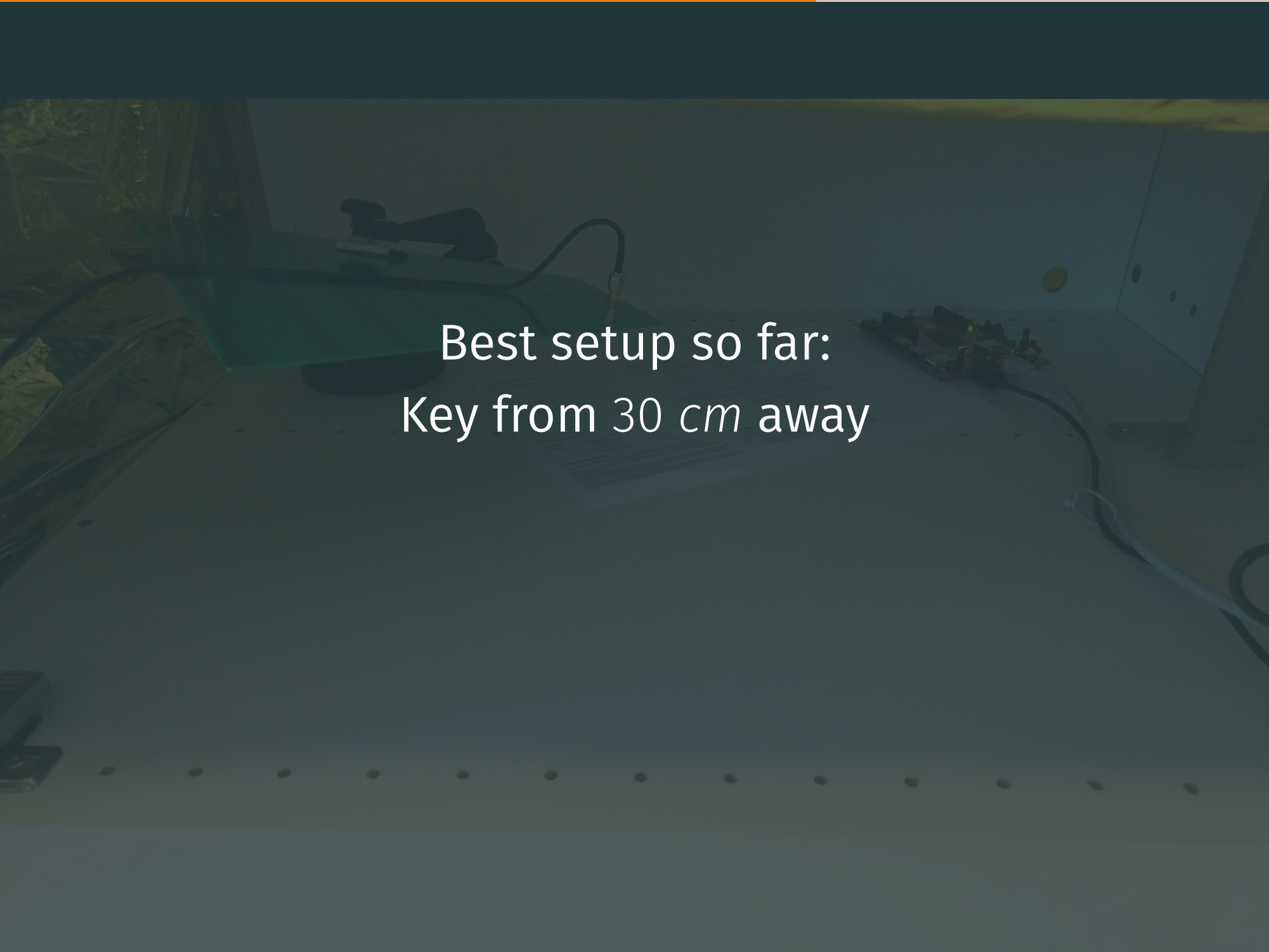
Example setup



DIY shielding

Real setup

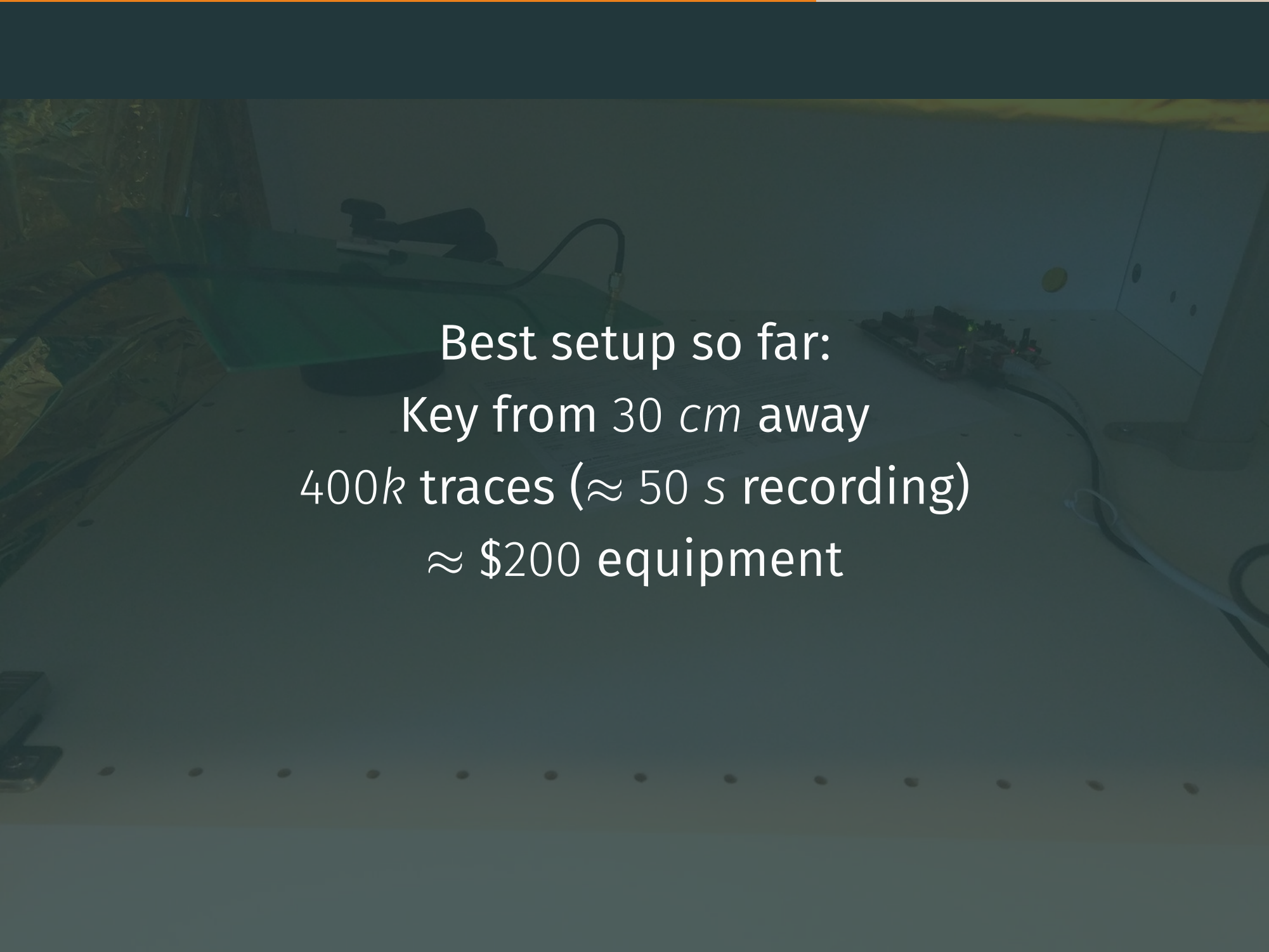



Best setup so far:
Key from 30 cm away

Best setup so far:

Key from 30 cm away

$400k$ traces ($\approx$ 50 s recording)



Best setup so far:

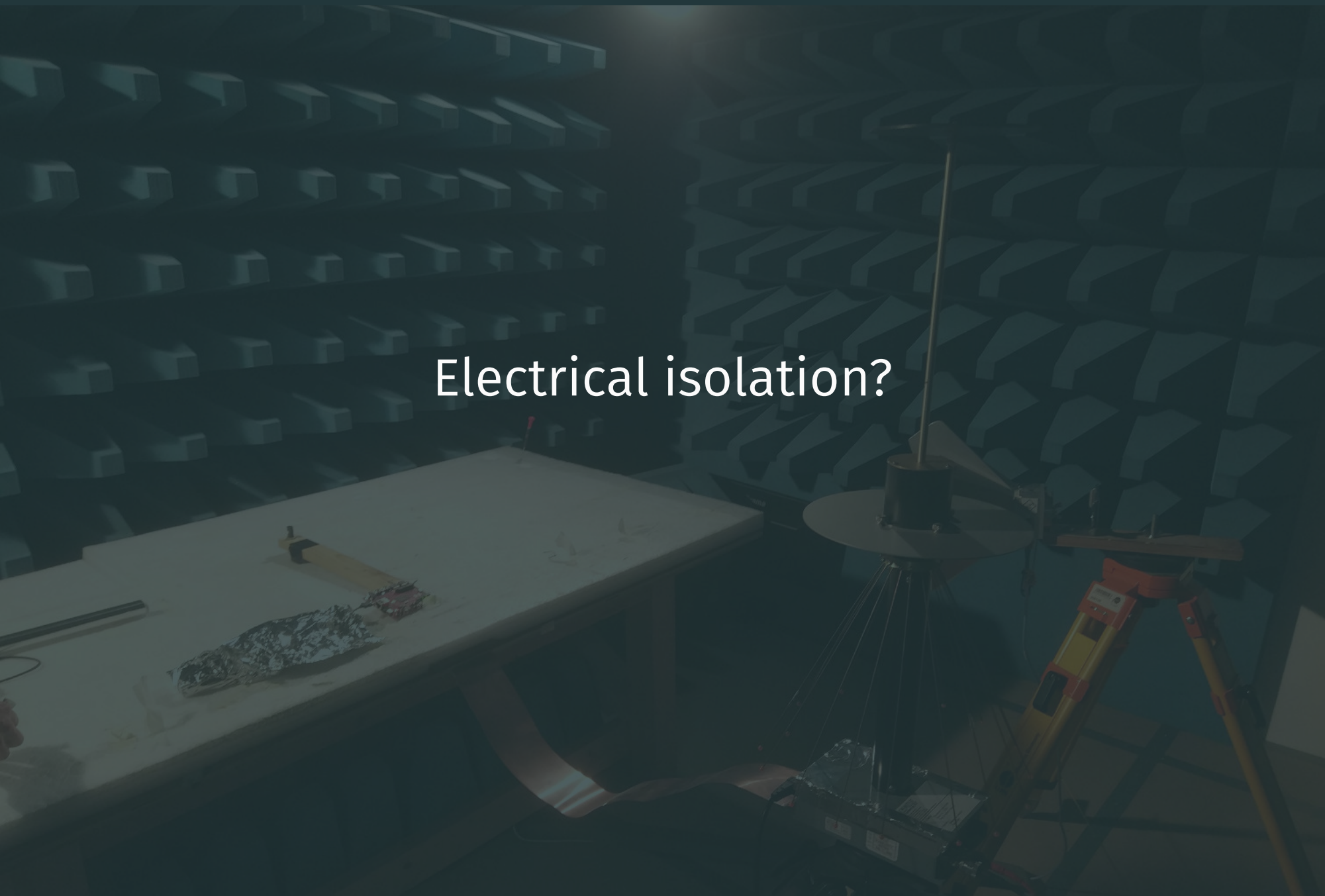
Key from $30\ cm$ away

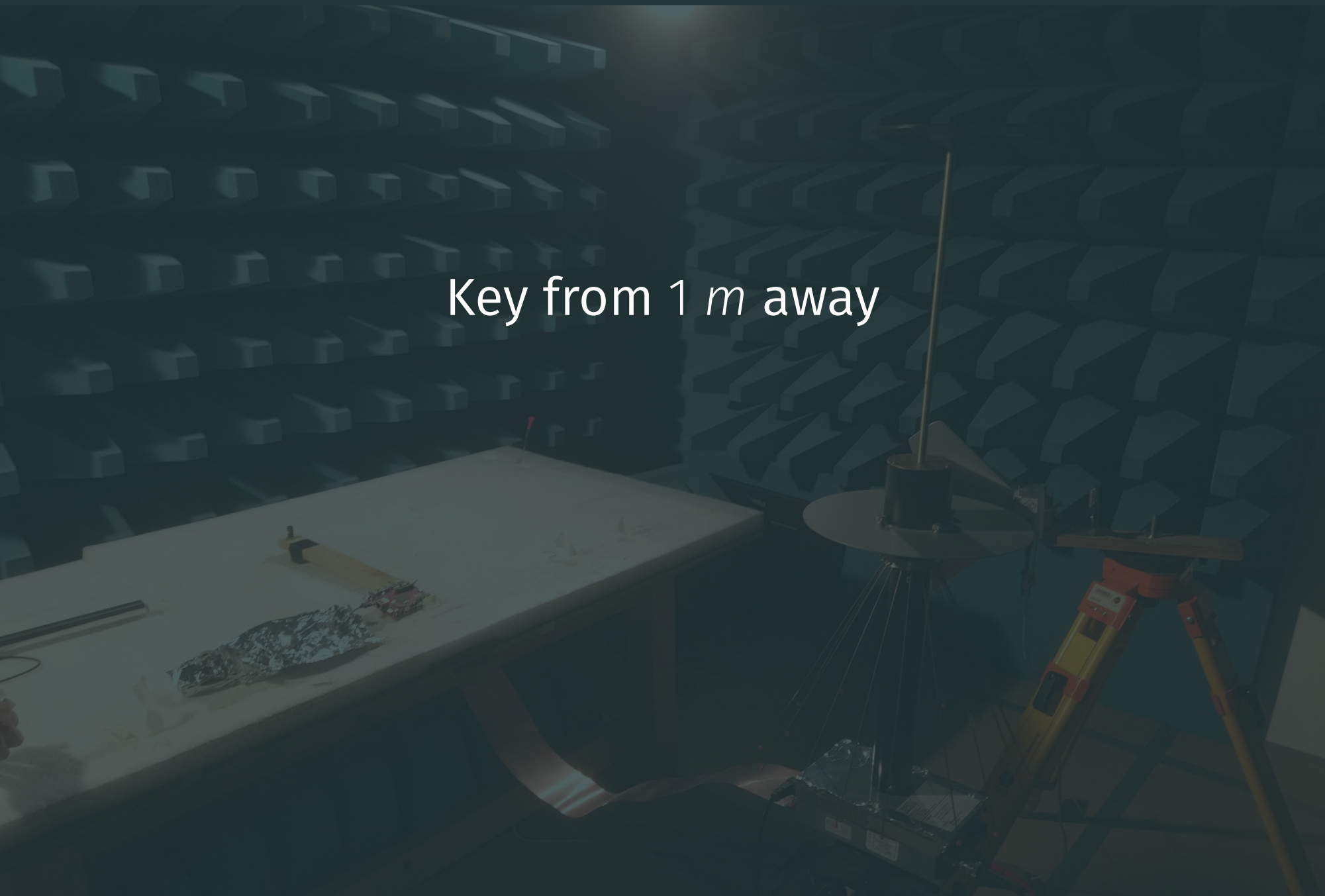
$400k$ traces ($\approx 50\ s$ recording)

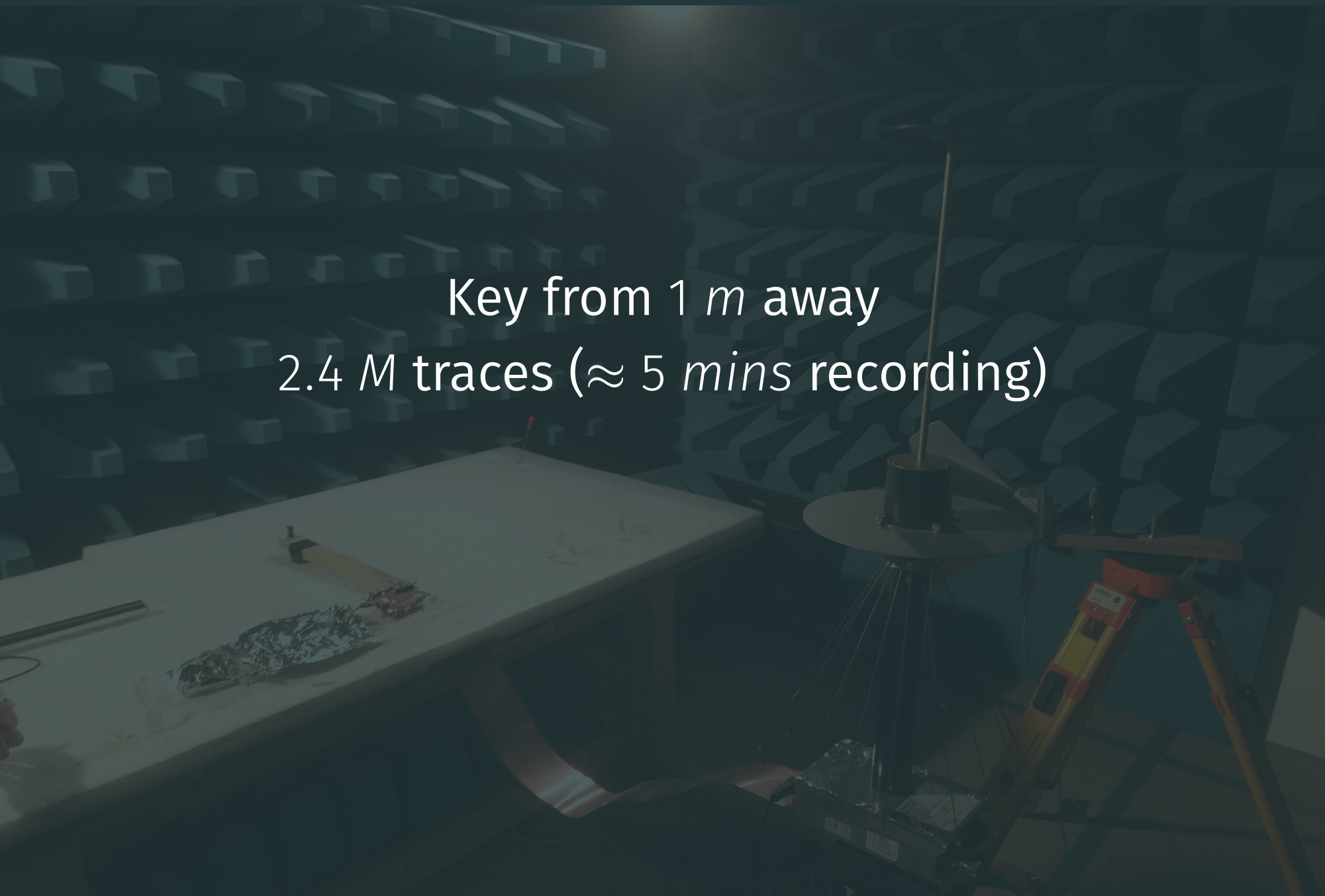
$\approx \$200$ equipment

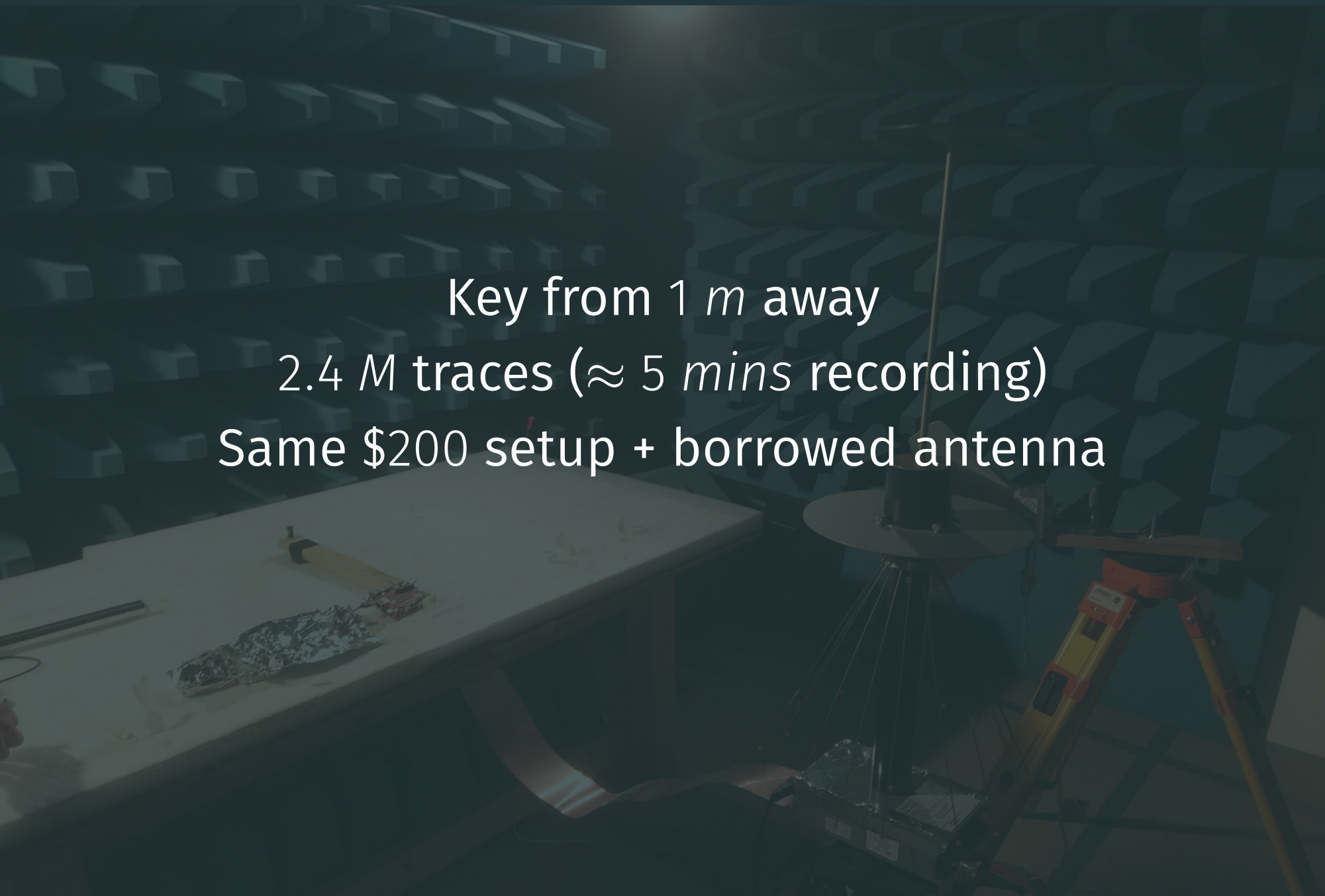
...and in ideal conditions?
(Thanks, OSPL)

Anechoic Chamber


Electrical isolation?


Key from 1 m away


Key from 1 *m* away
2.4 *M* traces (≈ 5 *mins* recording)


Key from 1 *m* away
2.4 *M* traces (≈ 5 *mins* recording)
Same $200 setup + borrowed antenna

Conclusion

Conclusion

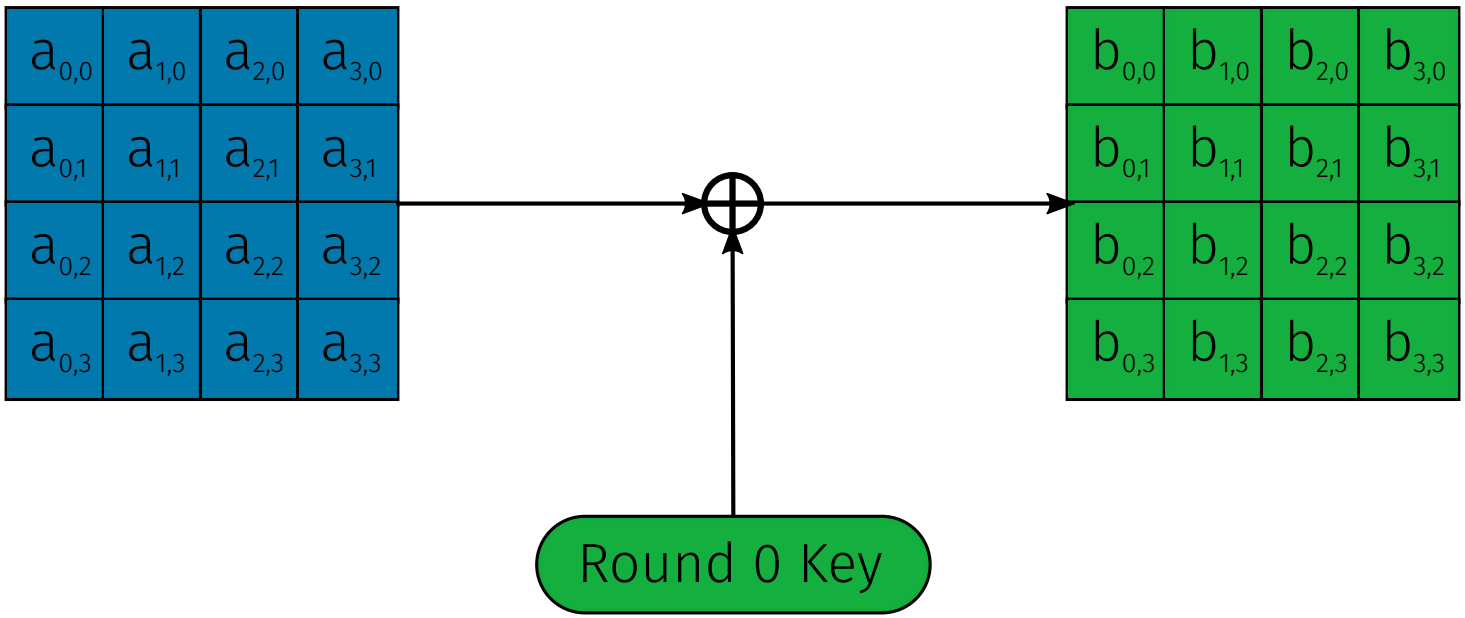
- Break OpenSSL's AES with a wire and a $20 dongle
- Radio hardware $\rightarrow$ *really* speeds up attack
- Increase attack distance with new analogue front-ends
  - First known demonstration
- 1 *m* works in 5 minutes...

Thanks! Questions?

Backup slides

Selecting an intermediate

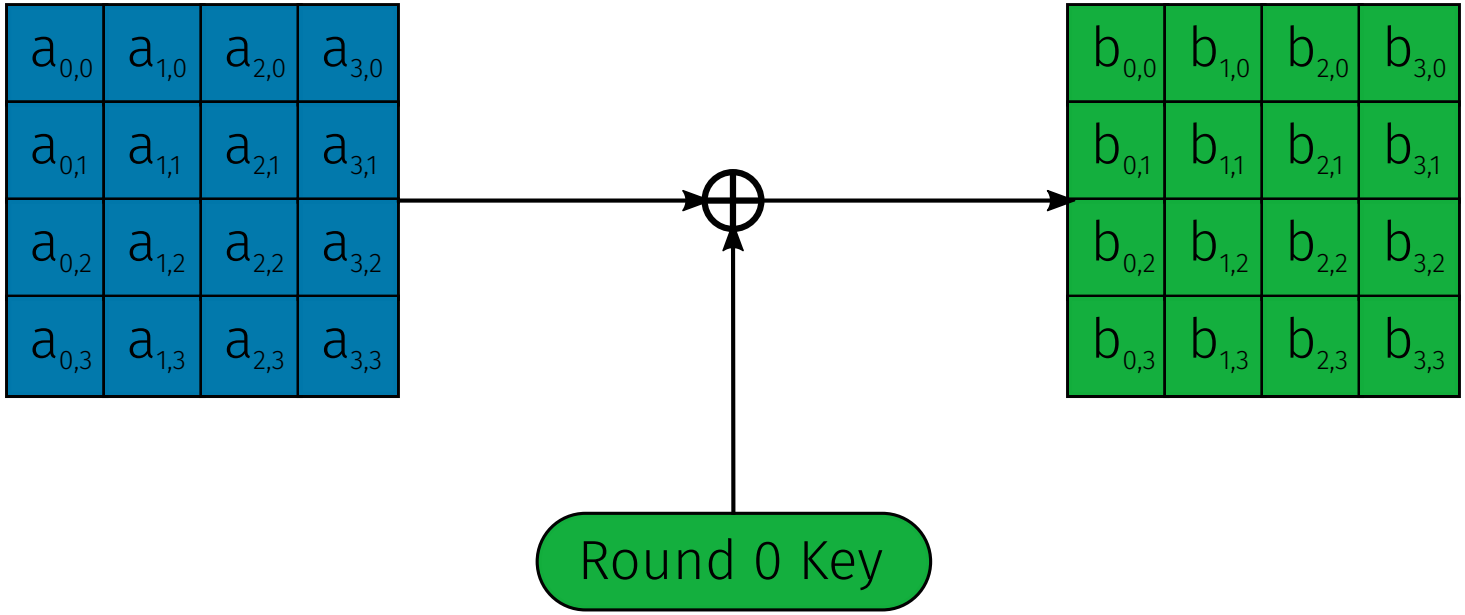
OpenSSL AES Round 0



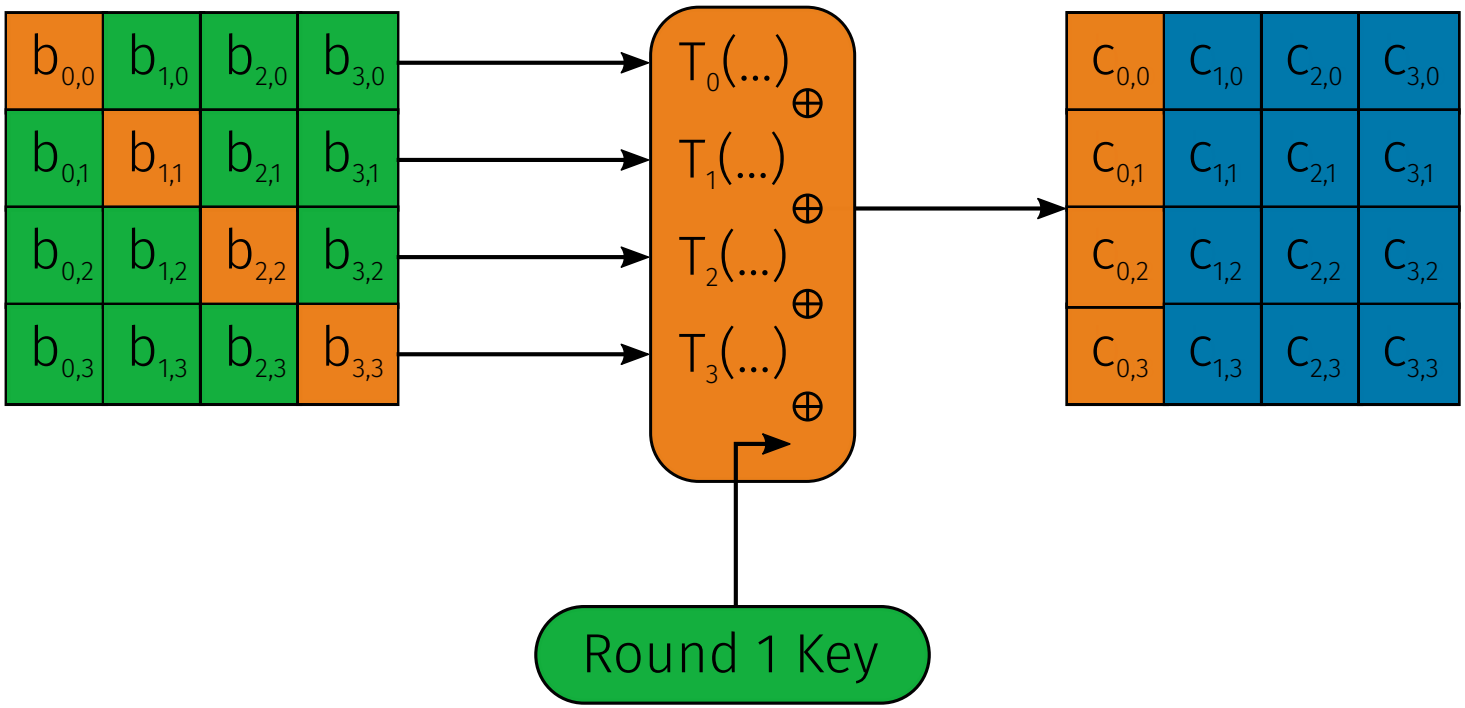
OpenSSL AES Round 0

Attack XOR with key?

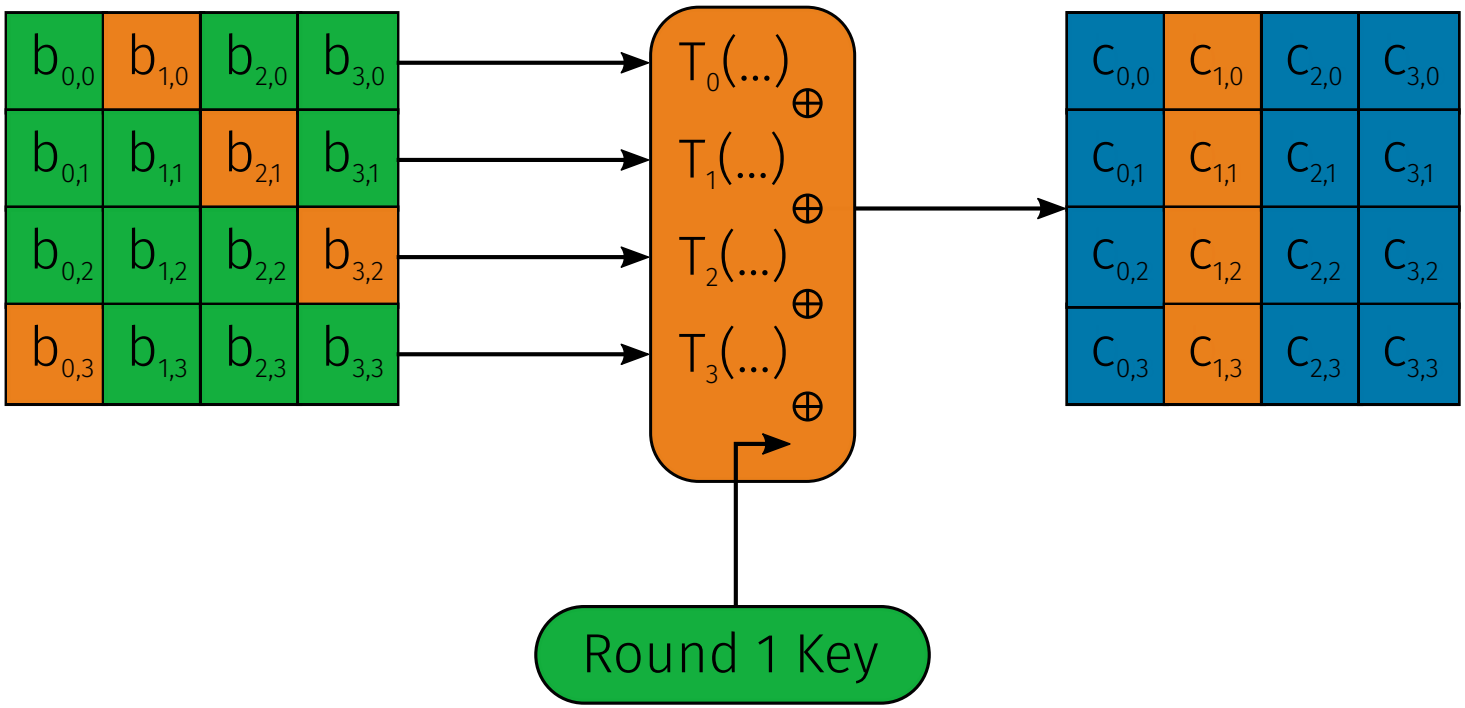
...Can do better!



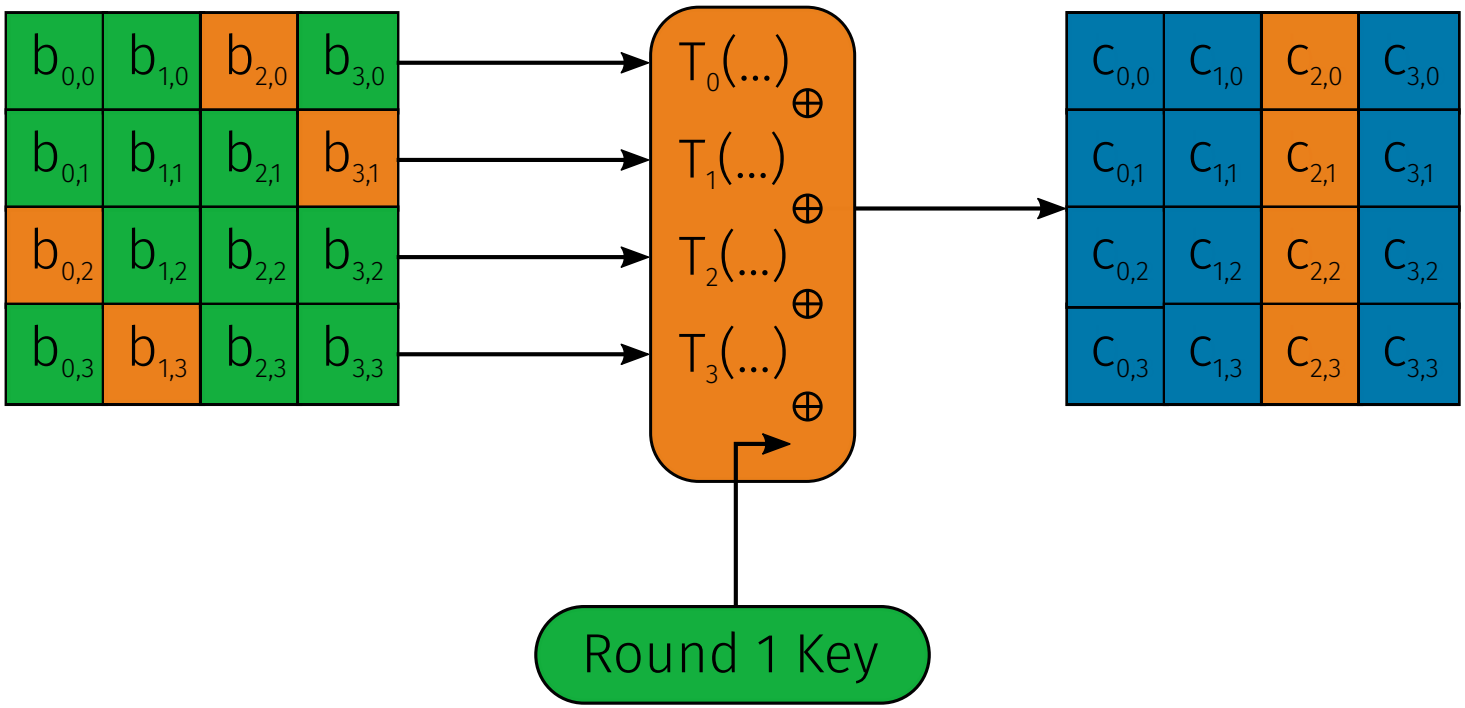
OpenSSL AES Round 1



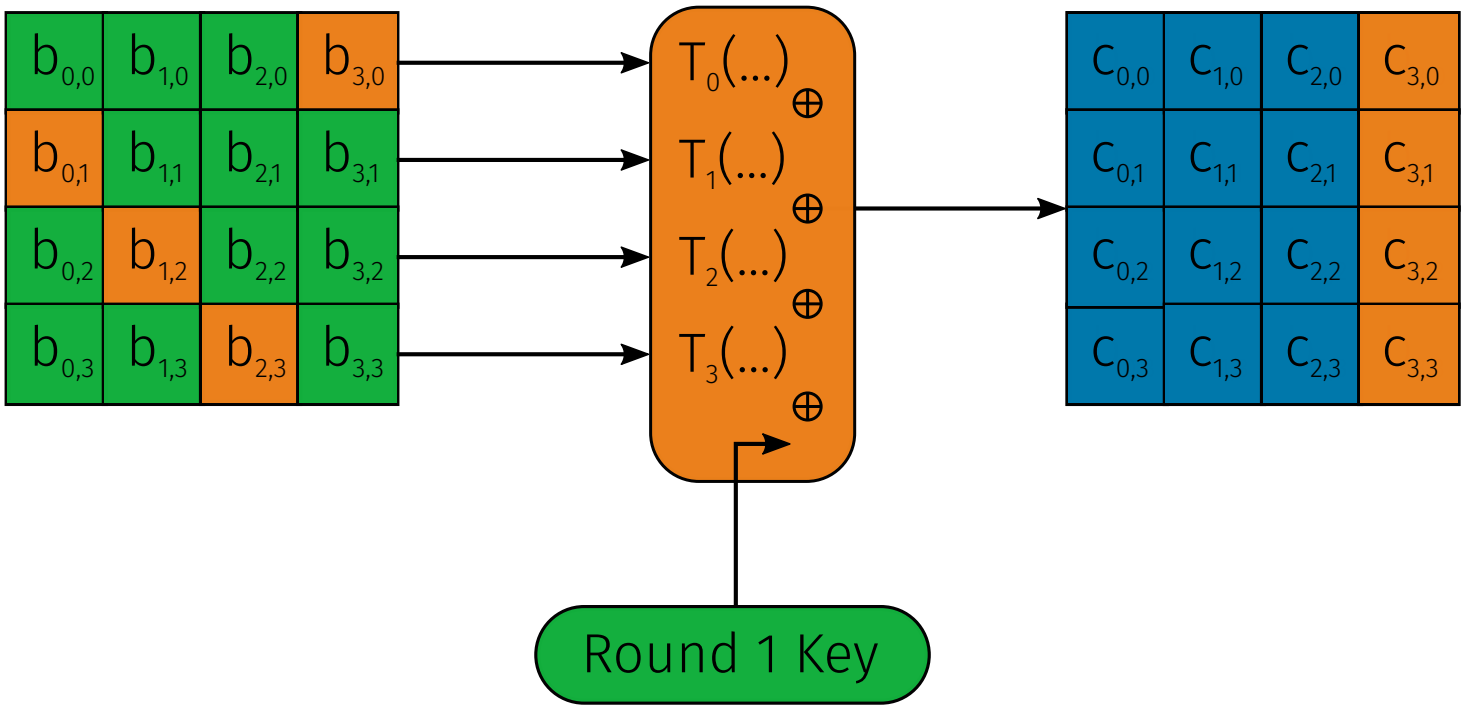
OpenSSL AES Round 1



OpenSSL AES Round 1



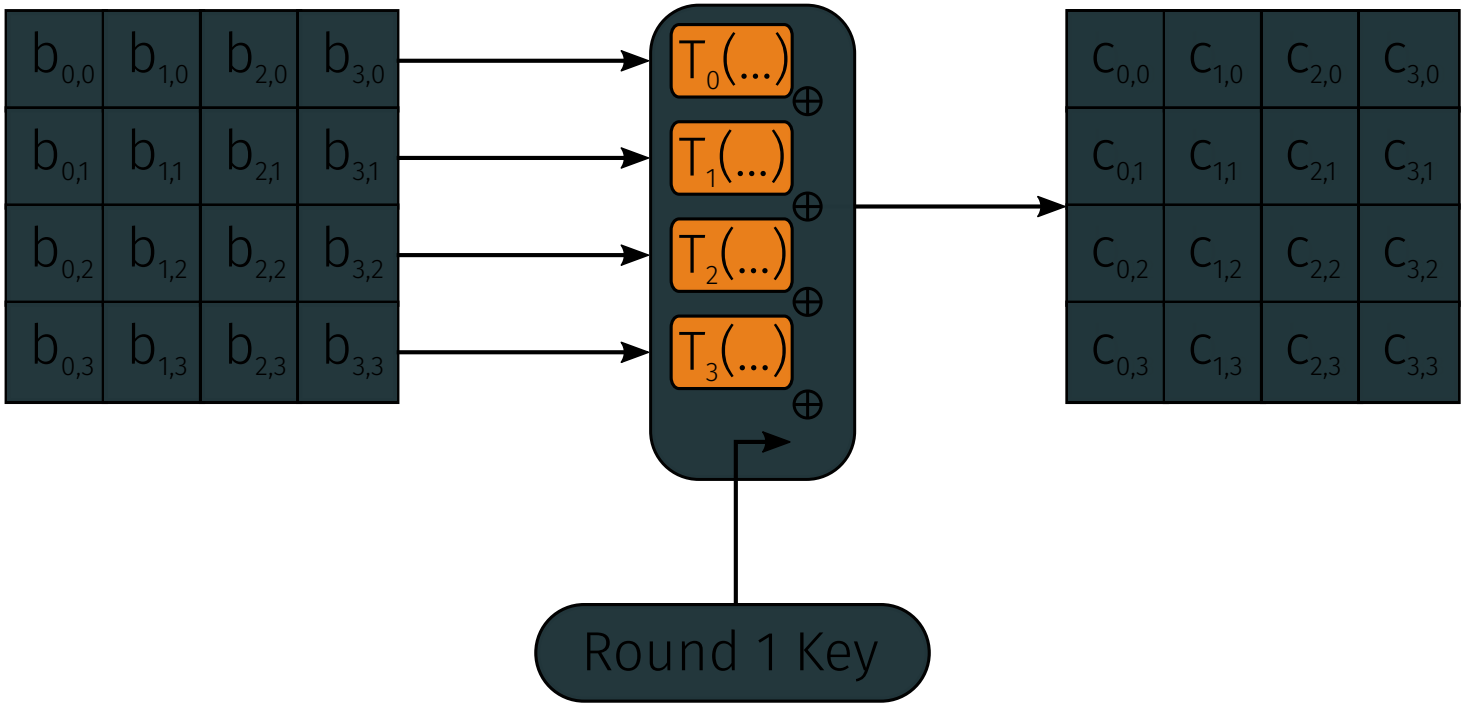
OpenSSL AES Round 1



OpenSSL AES Round 1

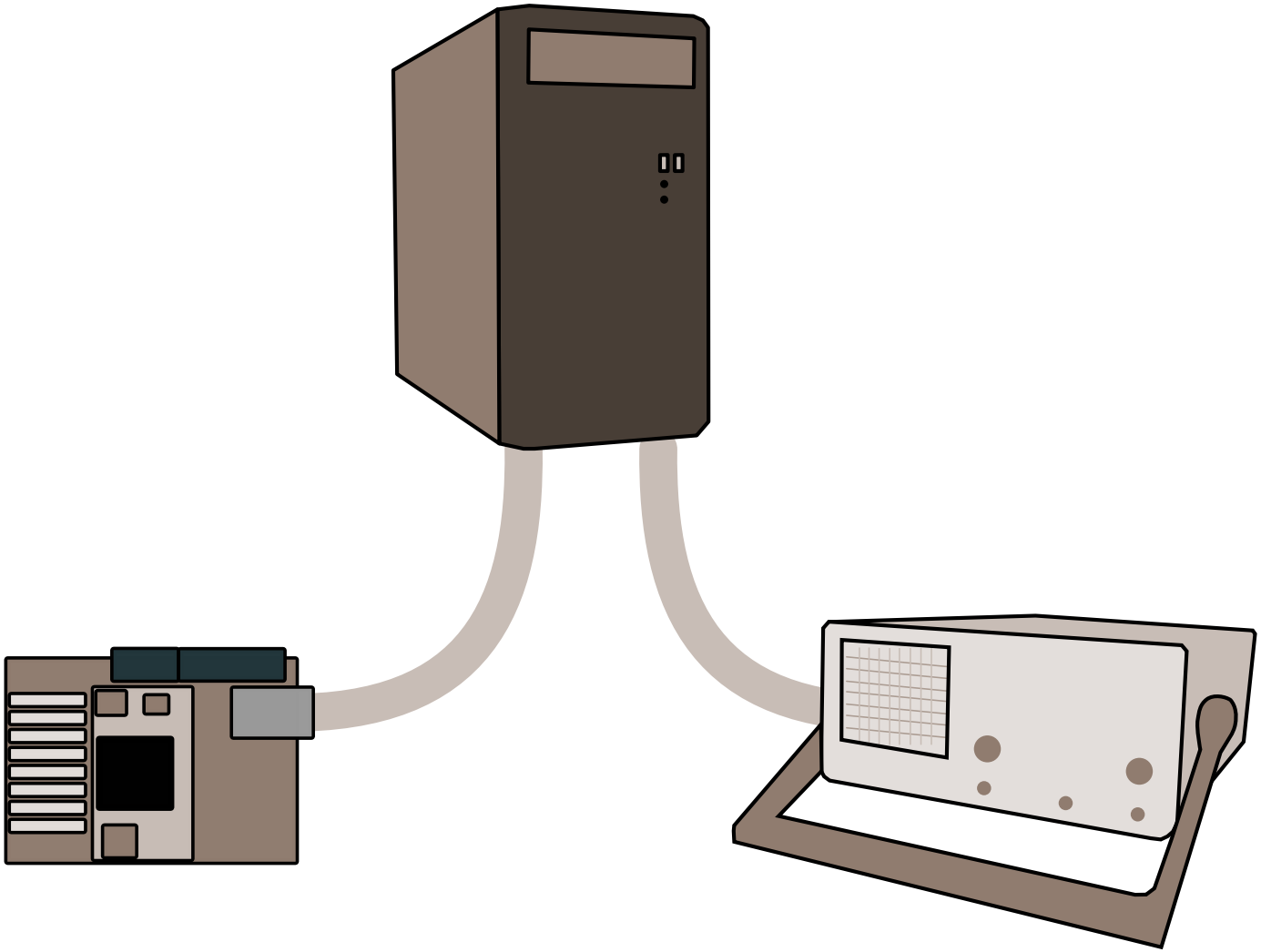
Attack these lookups.

The non-linearity is useful.

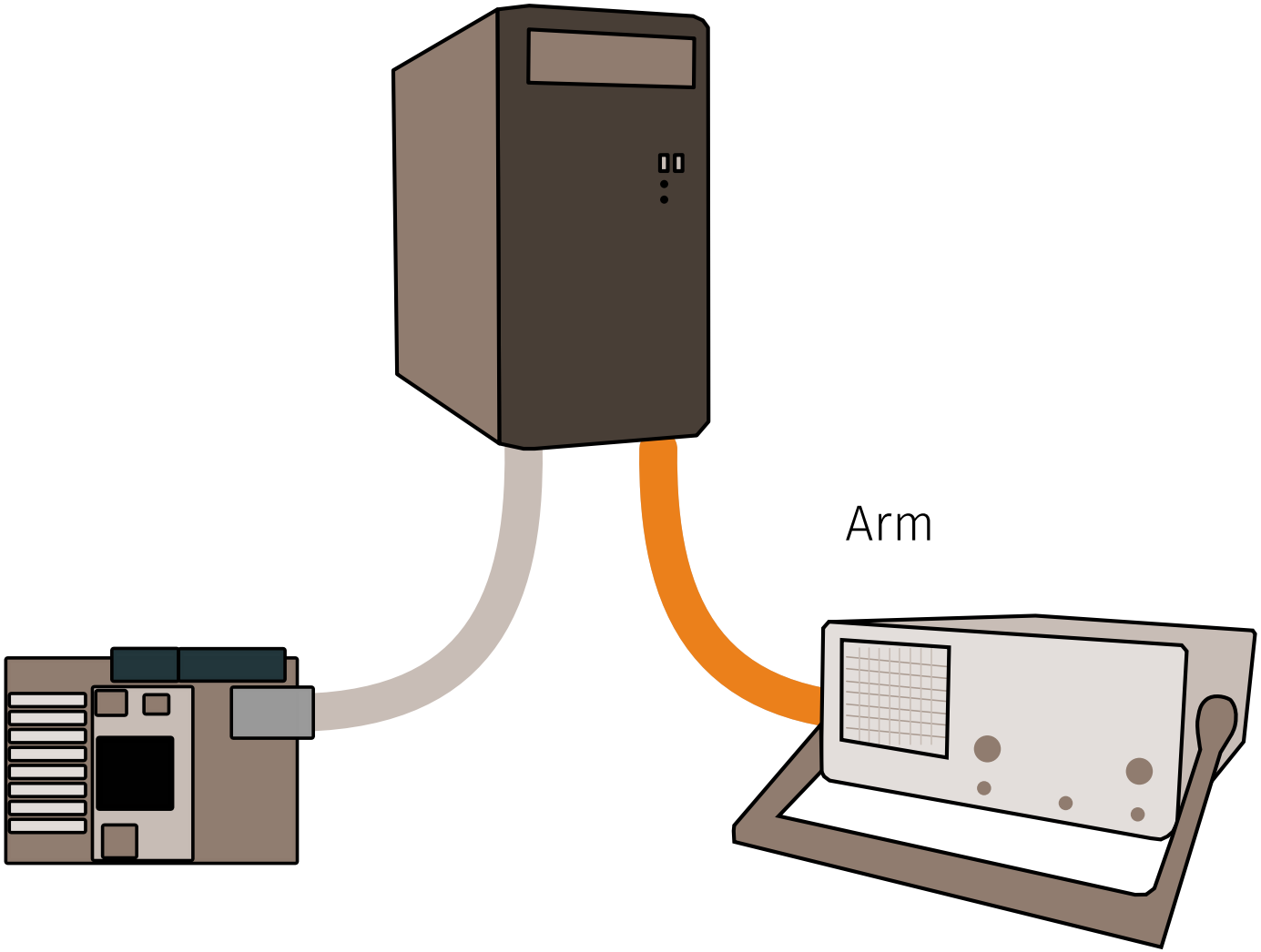


Our setup vs traditional setup

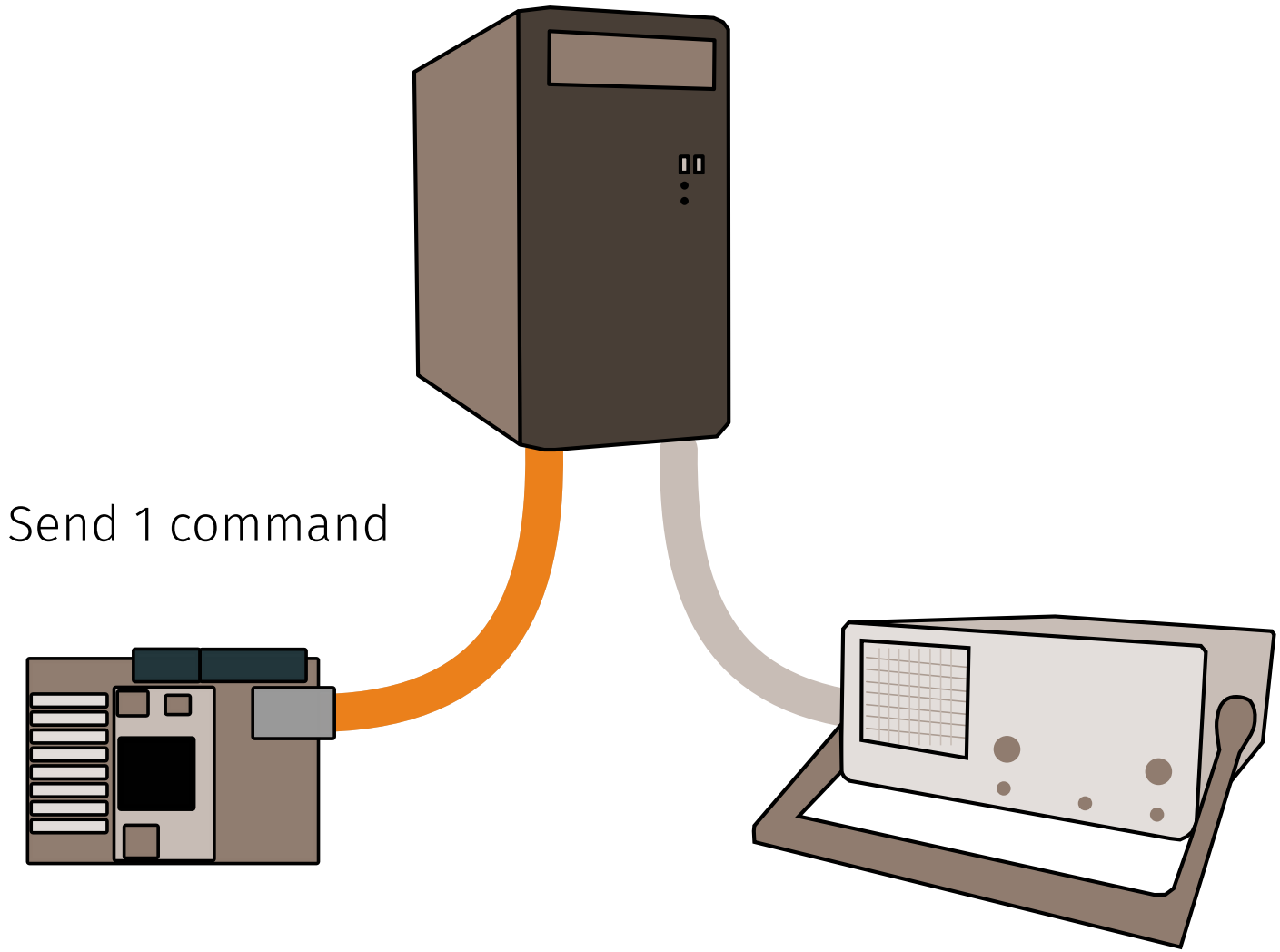
Recording comparison — ’Scope



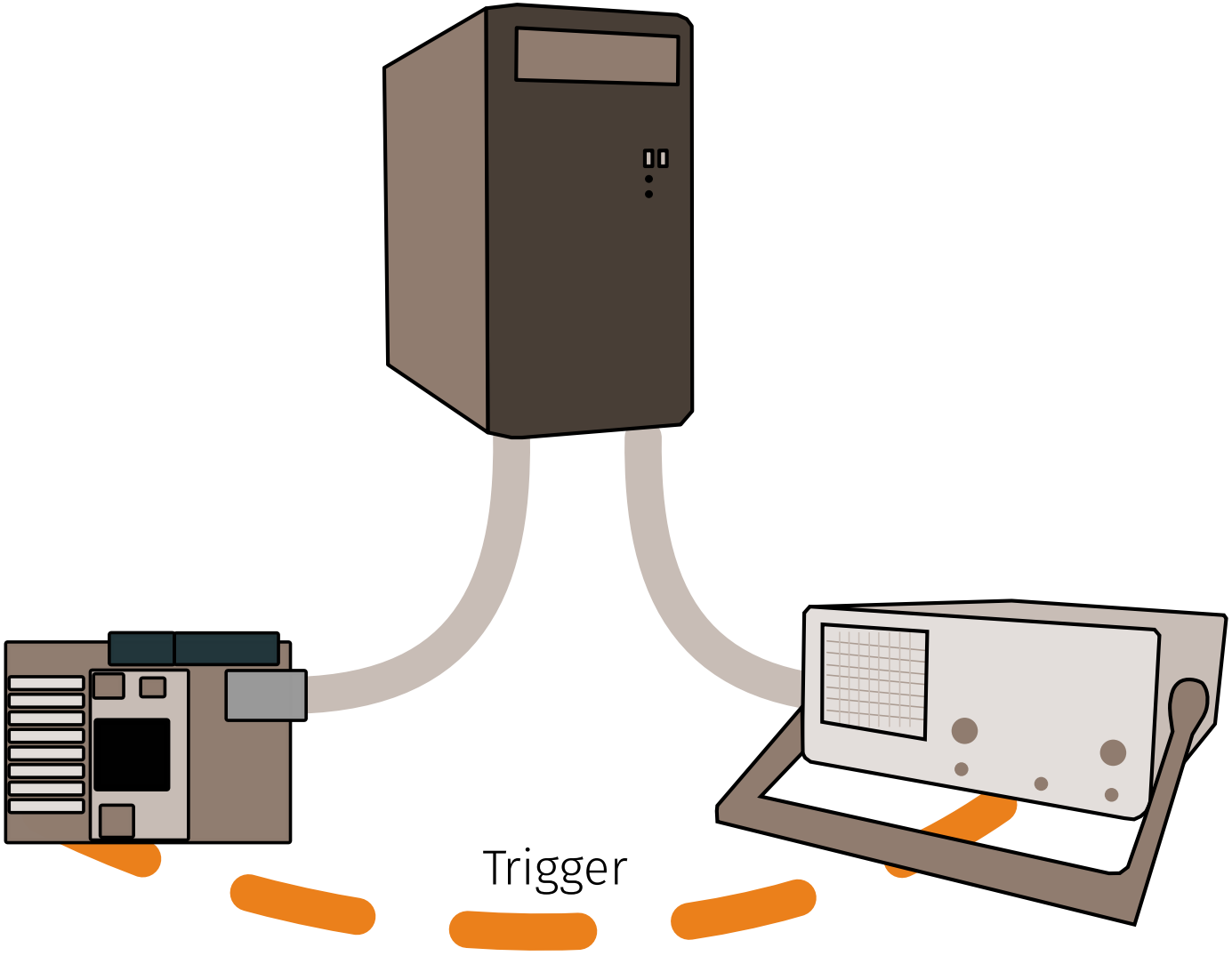
Recording comparison — ’Scope



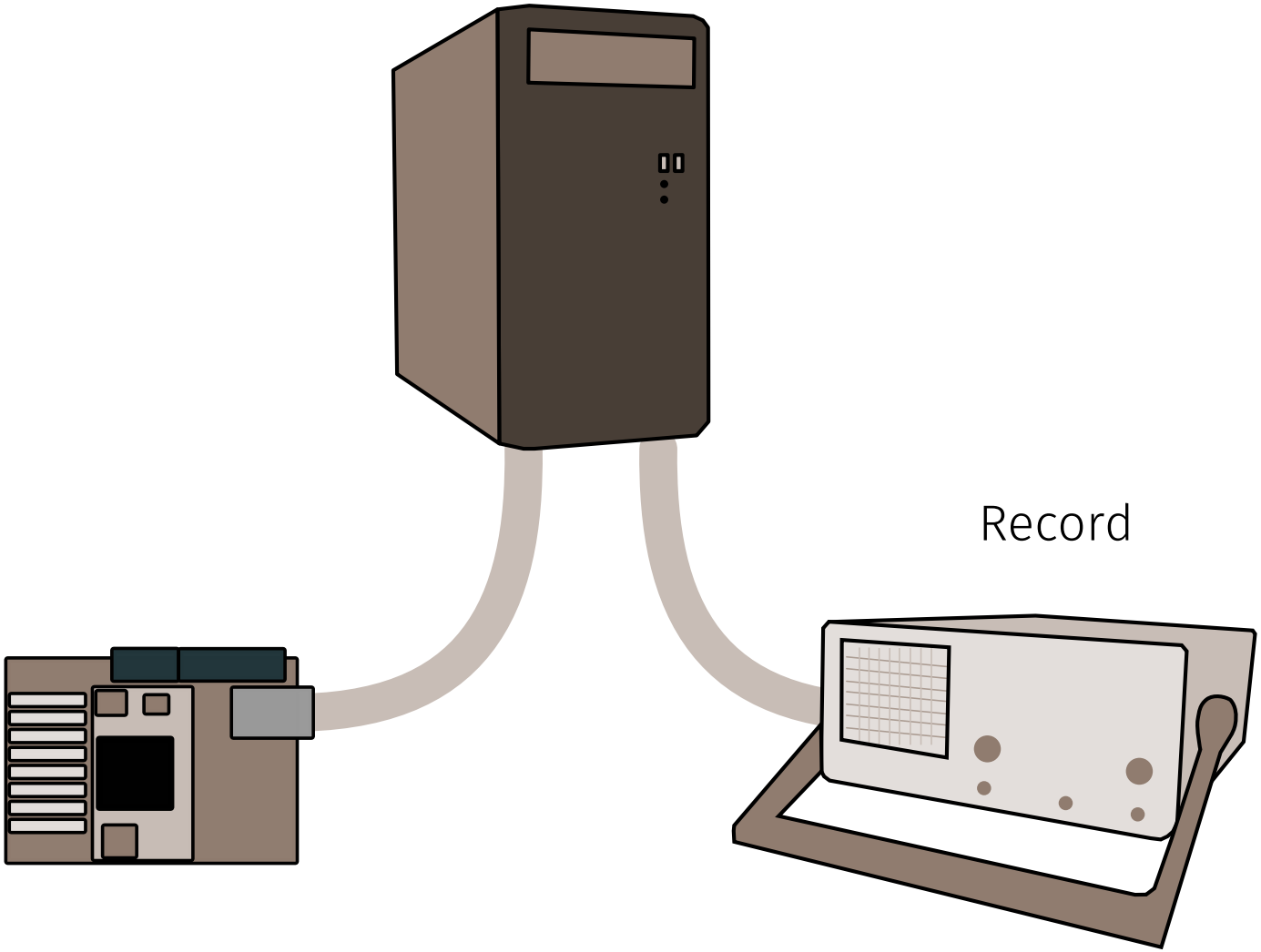
Recording comparison — ’Scope



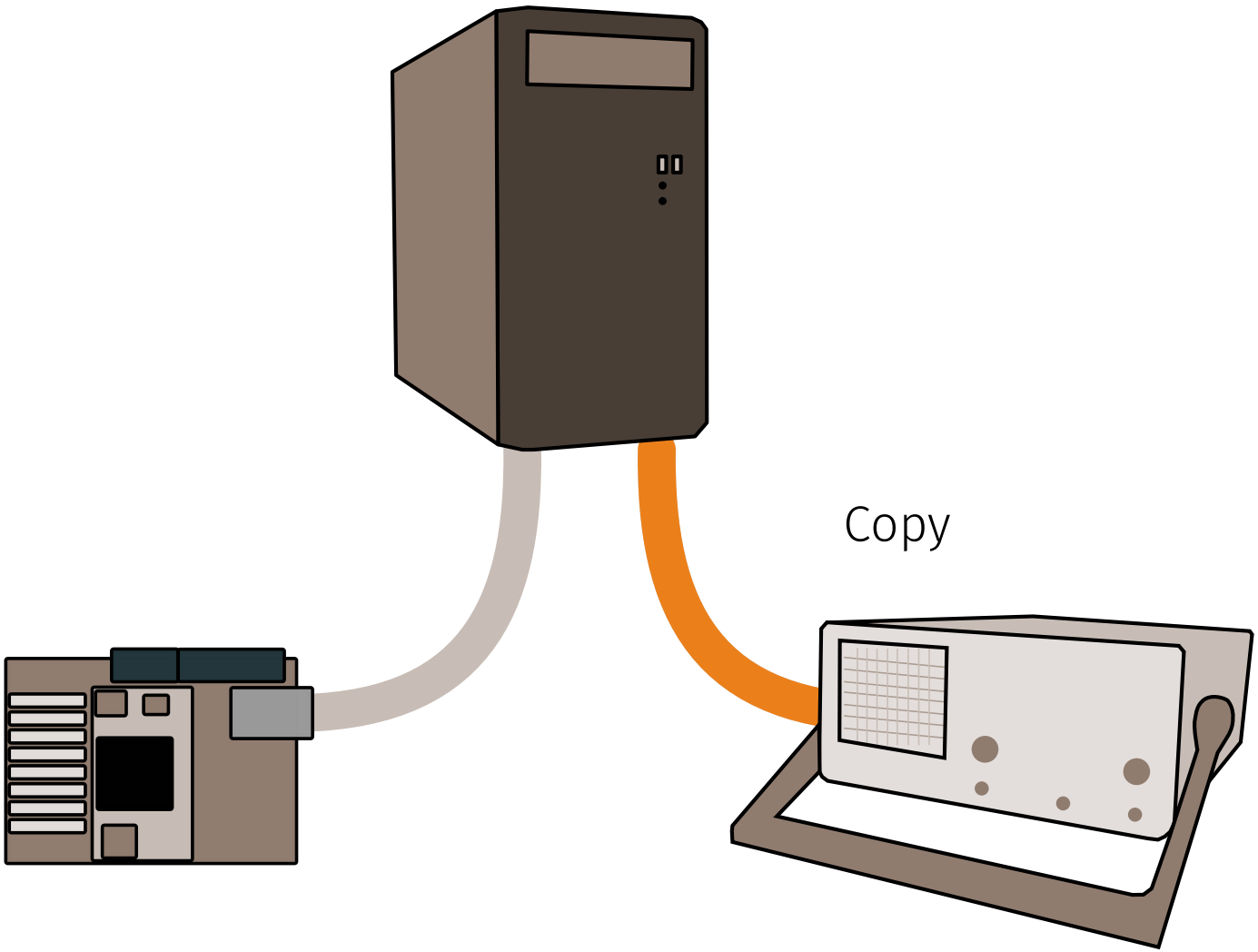
Recording comparison — ’Scope



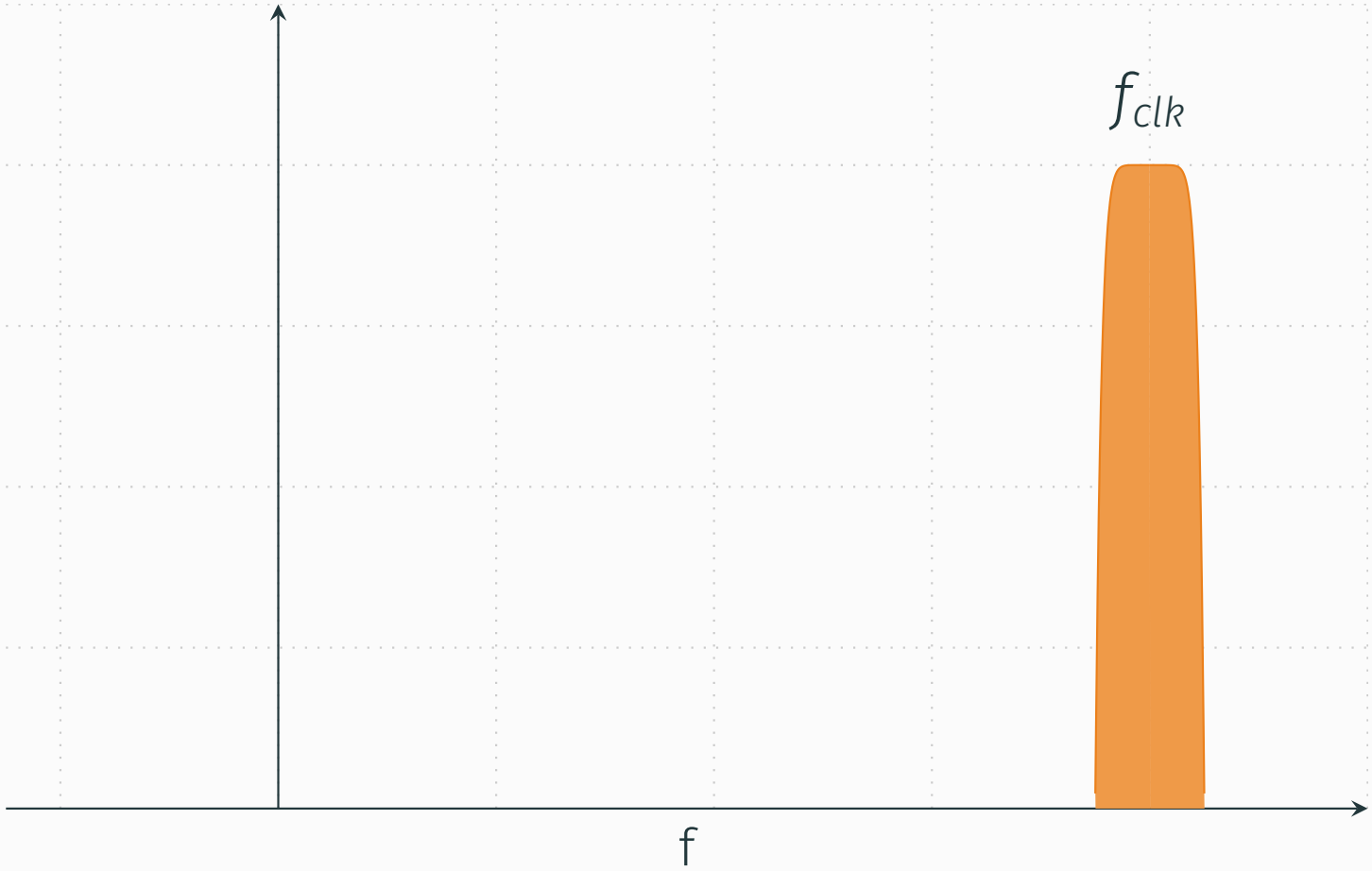
Recording comparison — ’Scope



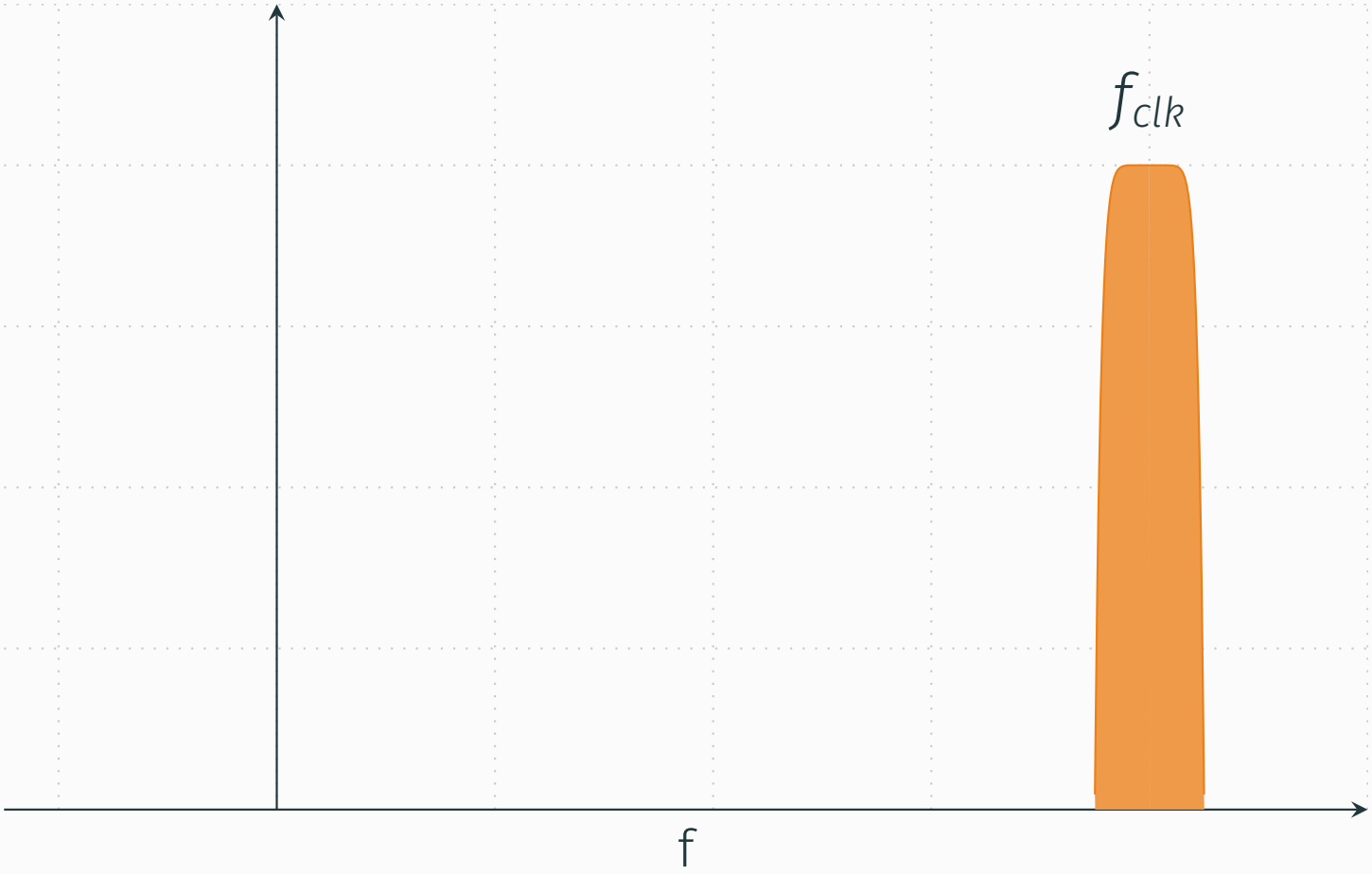
Recording comparison — ’Scope



Recording comparison — ’Scope

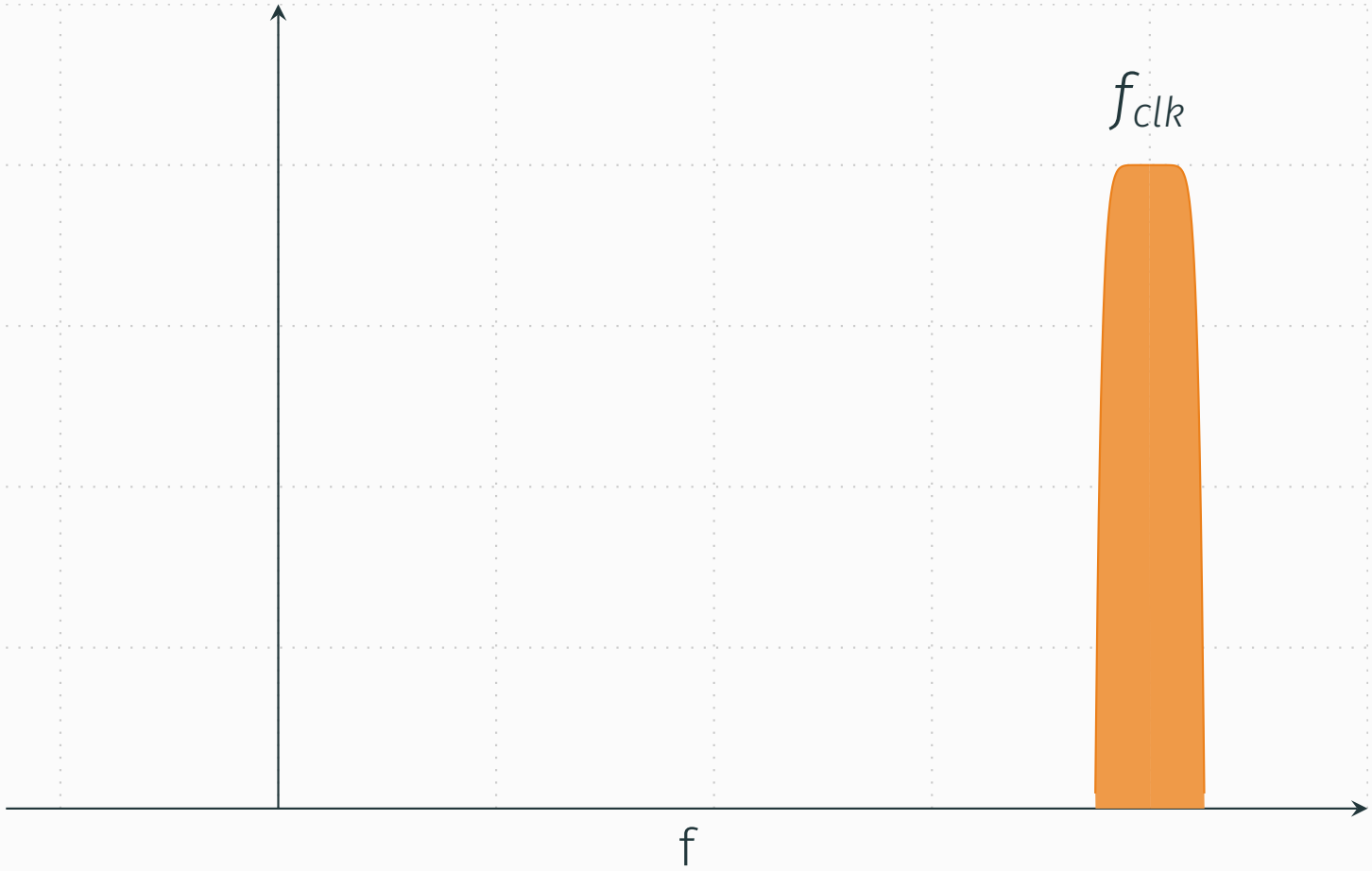


Recording comparison — 'Scope

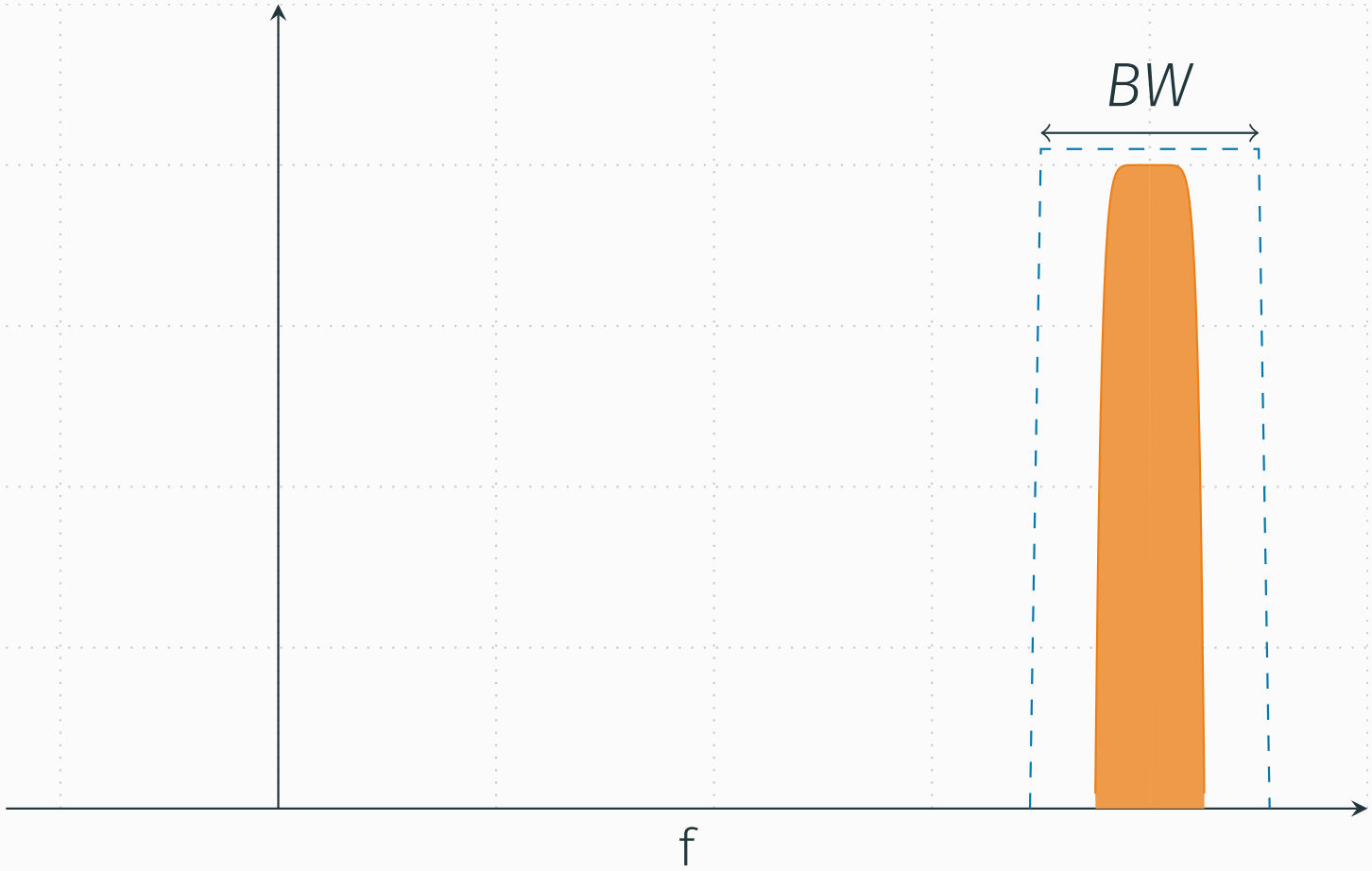


$$f_s > 2f_{clk}$$

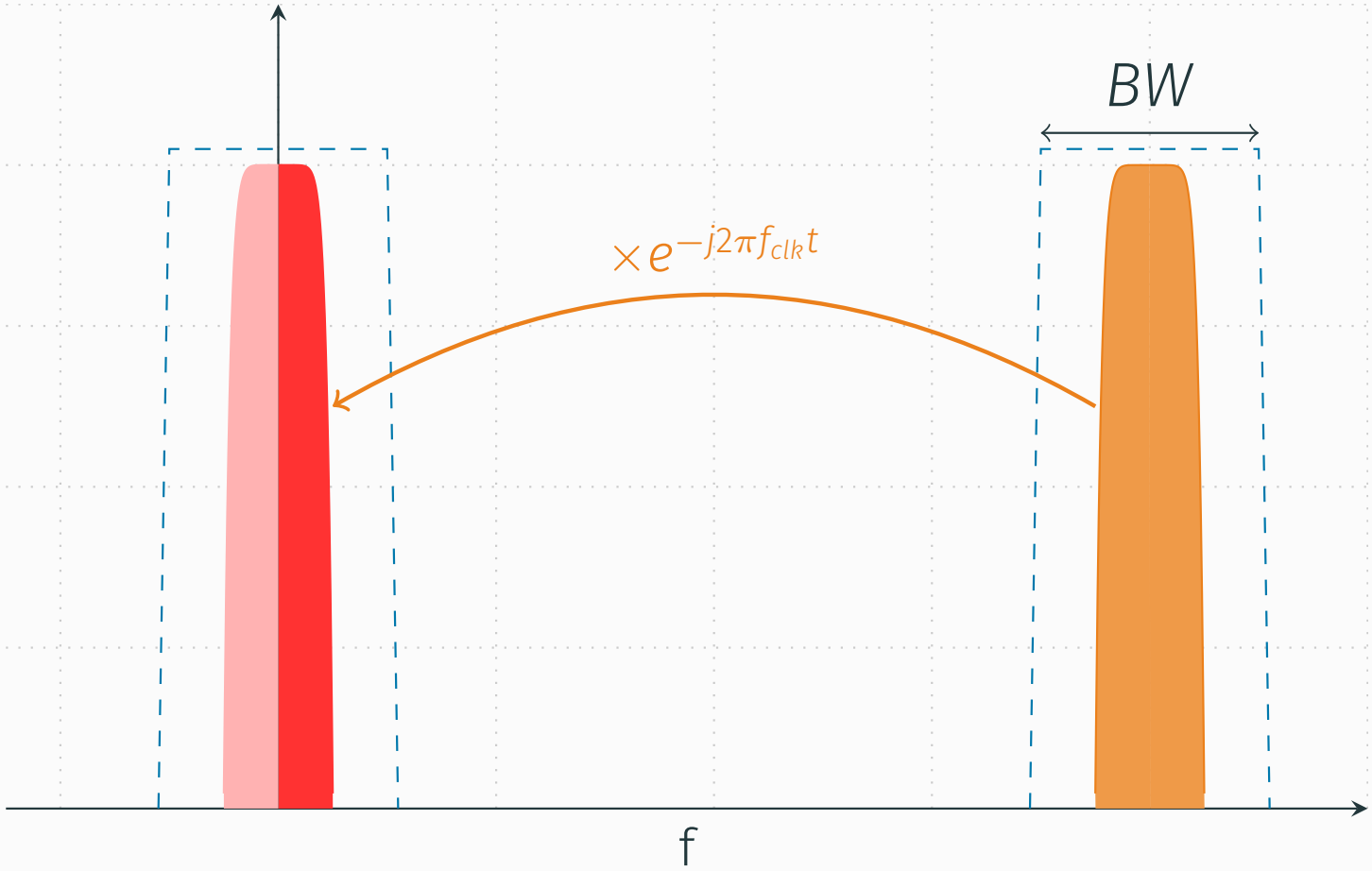
Recording comparison — Radio



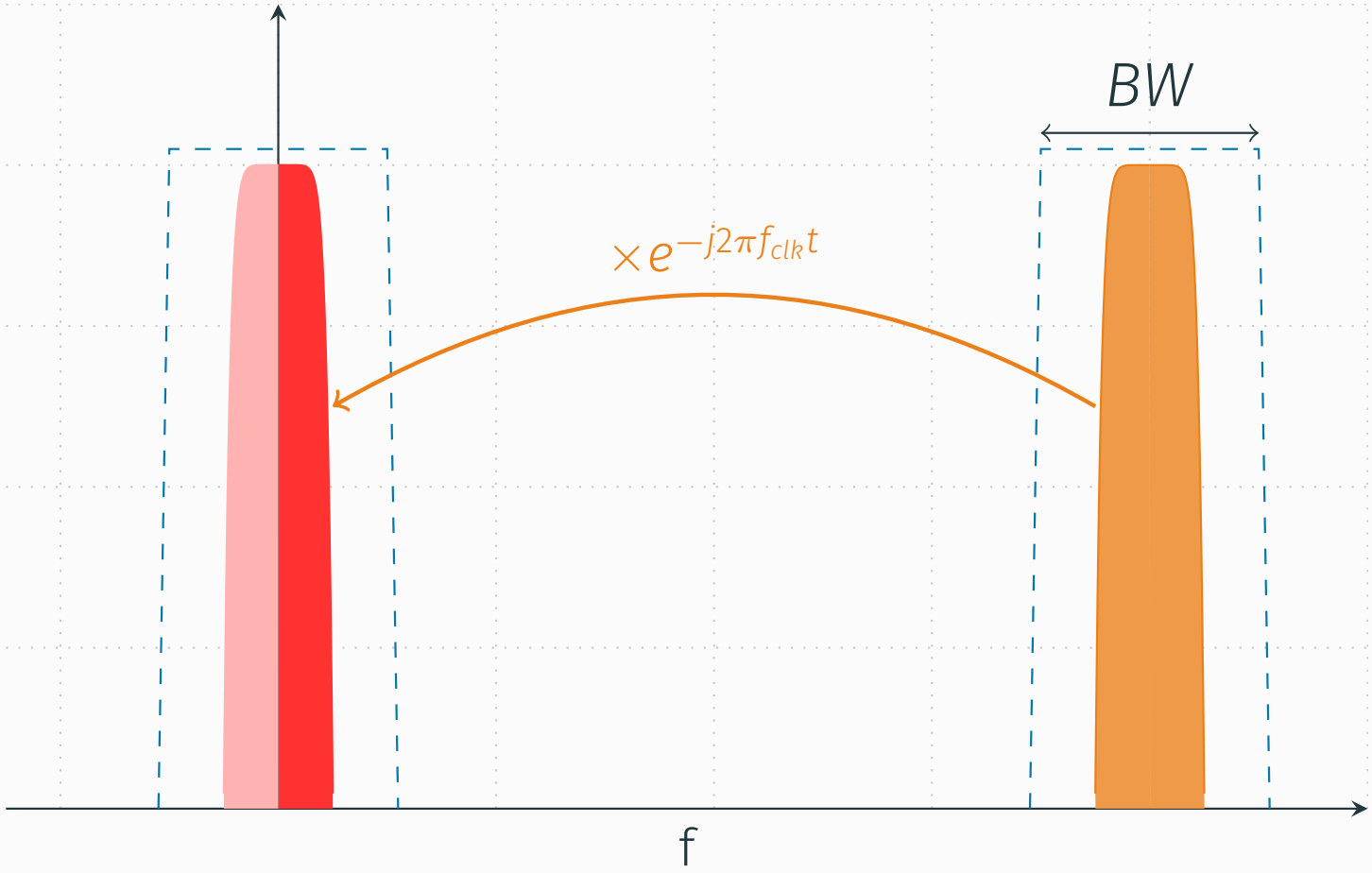
Recording comparison — Radio



Recording comparison — Radio

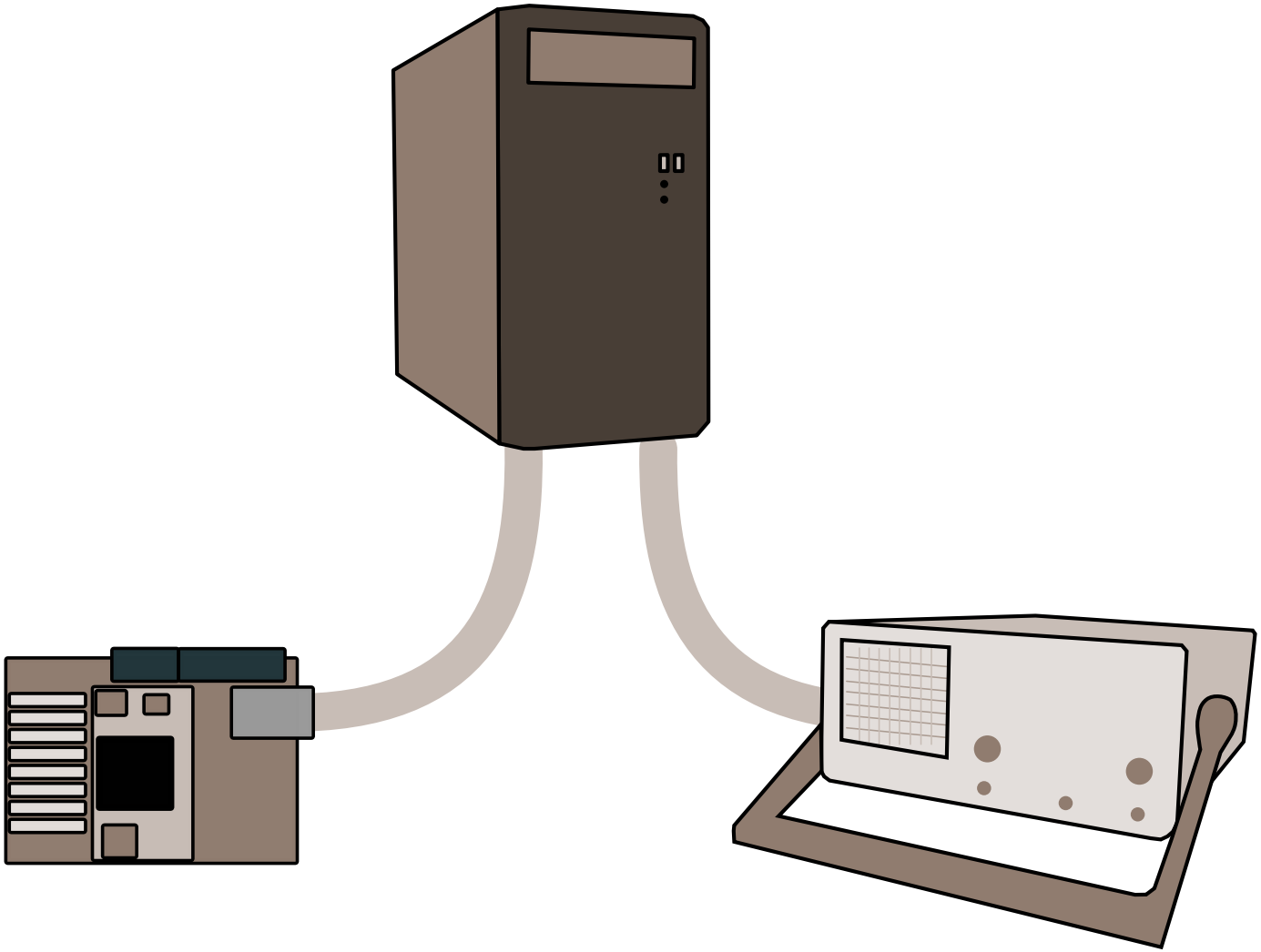


Recording comparison — Radio

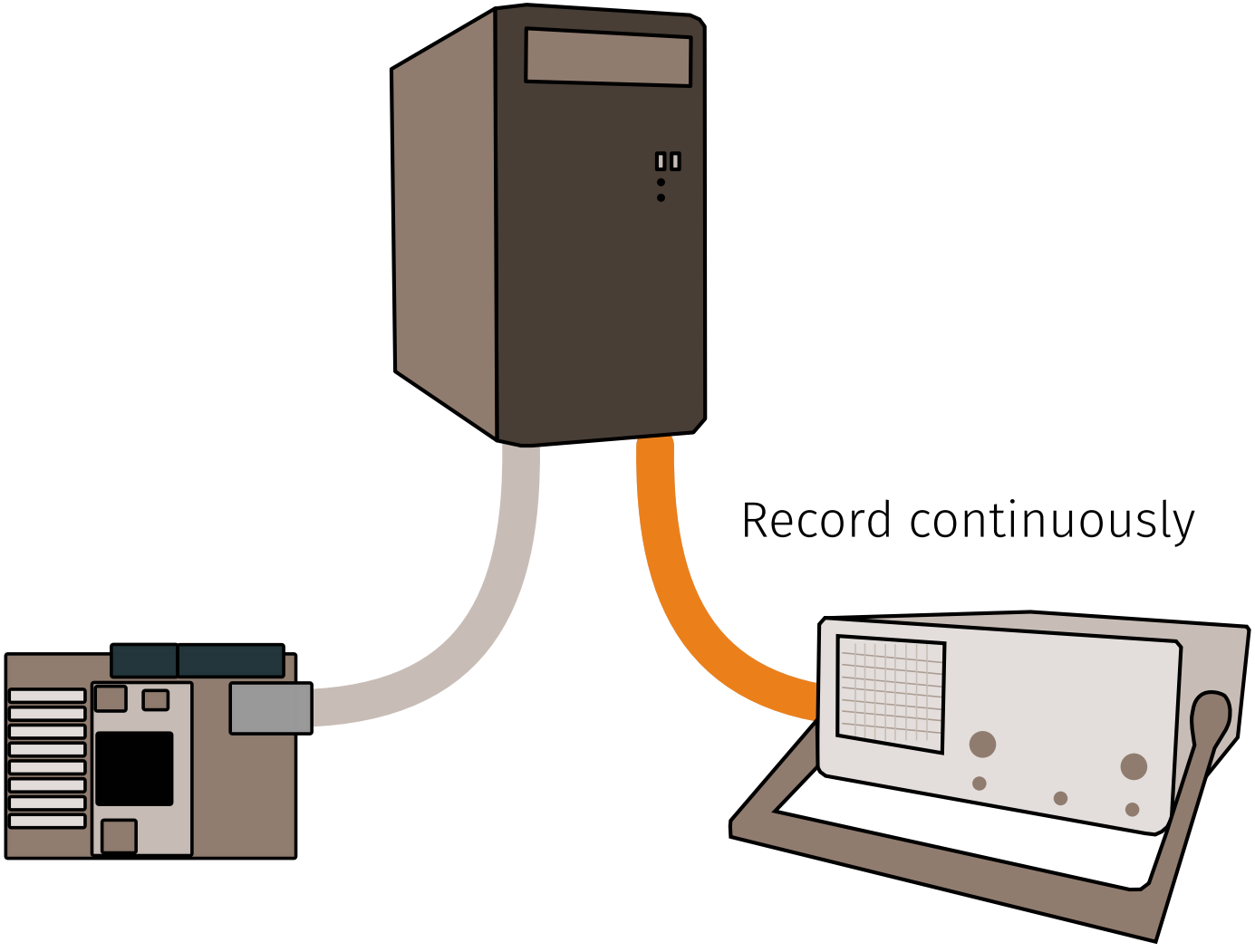


$$f_s > 2 \times BW$$

Recording comparison — Radio



Recording comparison — Radio



Recording comparison — Radio