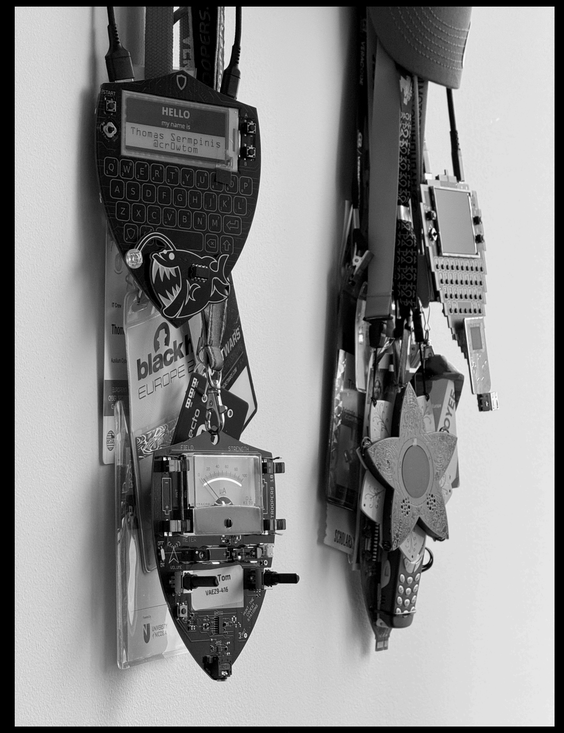
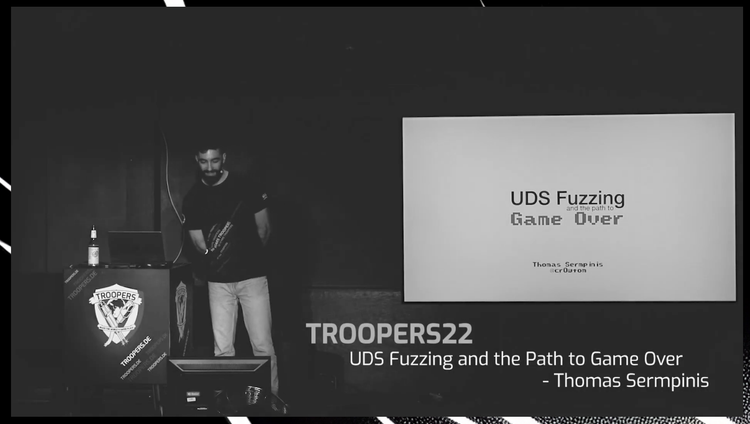

THOMAS SERMPINIS PRESENTS
BACK TO THE FUTURE
OLD VULNERABILITIES BECOMING NEW AGAIN

Whoami

- Thomas Sermpinis (a.k.a. cr0wtom)
  - Technical Director of Auxilium Pentest Labs by Day
  - Automotive Security Researcher by Night
- Hack Everything, Everywhere, All at Once (and Legally)
- Conference Speaker and part of ASRG

AUXILIUM
Pentest Labs





- For more: cr0wsplace.com

Goals of this Talk

- Analyse the state of cybersecurity in the automotive industry - from a hardware point of view
- Present unique (and hopefully interesting) use-cases, result of around 100 pentests and research projects in the industry
- Educate the new, the old and the bold
- Endorse and push more hackers to automotive
- Raise and highlight the significance of safety related hardware

챕터 #0

THE STATE

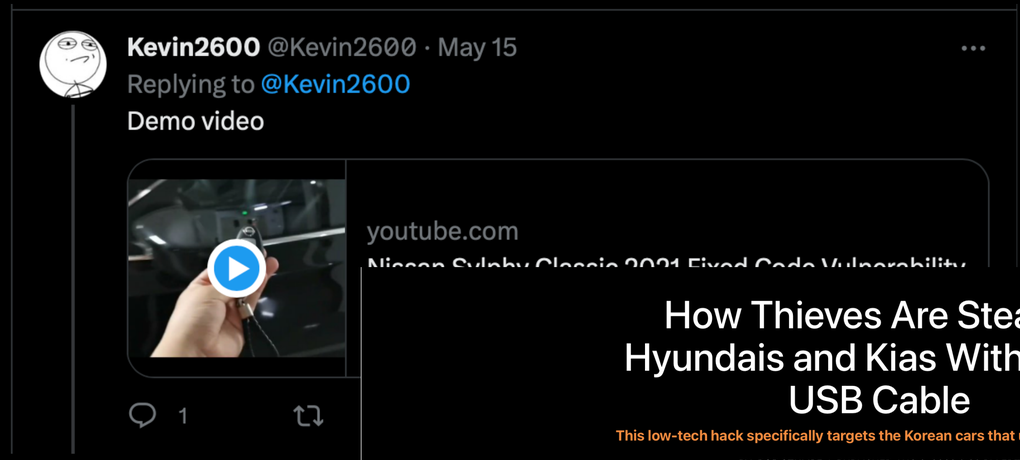
The State of Automotive Cybersecurity

Relay attacks in 2023?

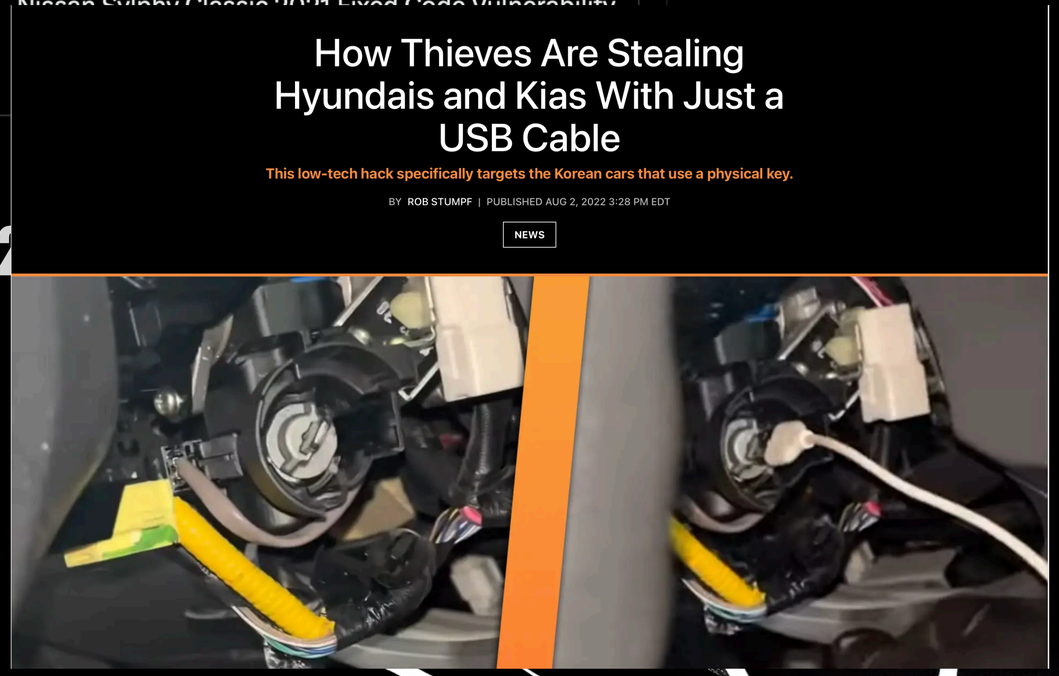
The State of Automotive Cybersecurity

Relay attacks in 2023?

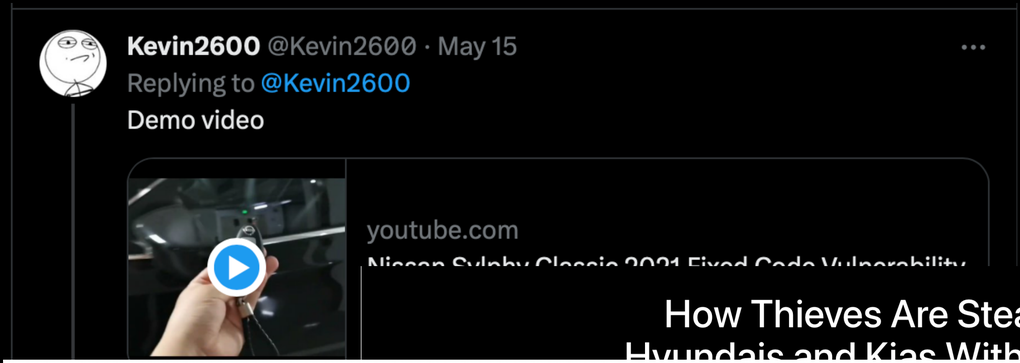
The State of Automotive Cybersecurity



Relay attacks in 202



The State of Automotive Cybersecurity



Relay

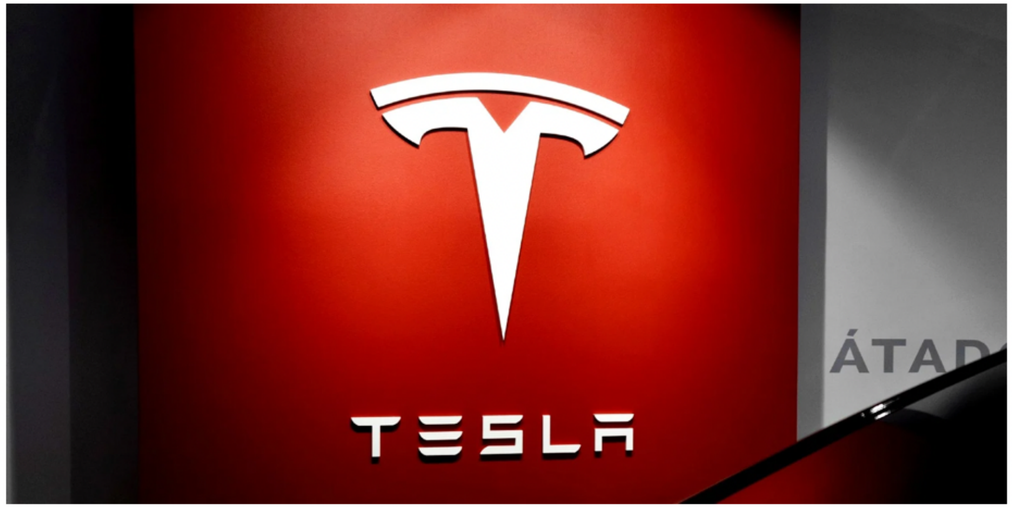
How Thieves Are Stealing Hyundais and Kias With Just a

physical key.

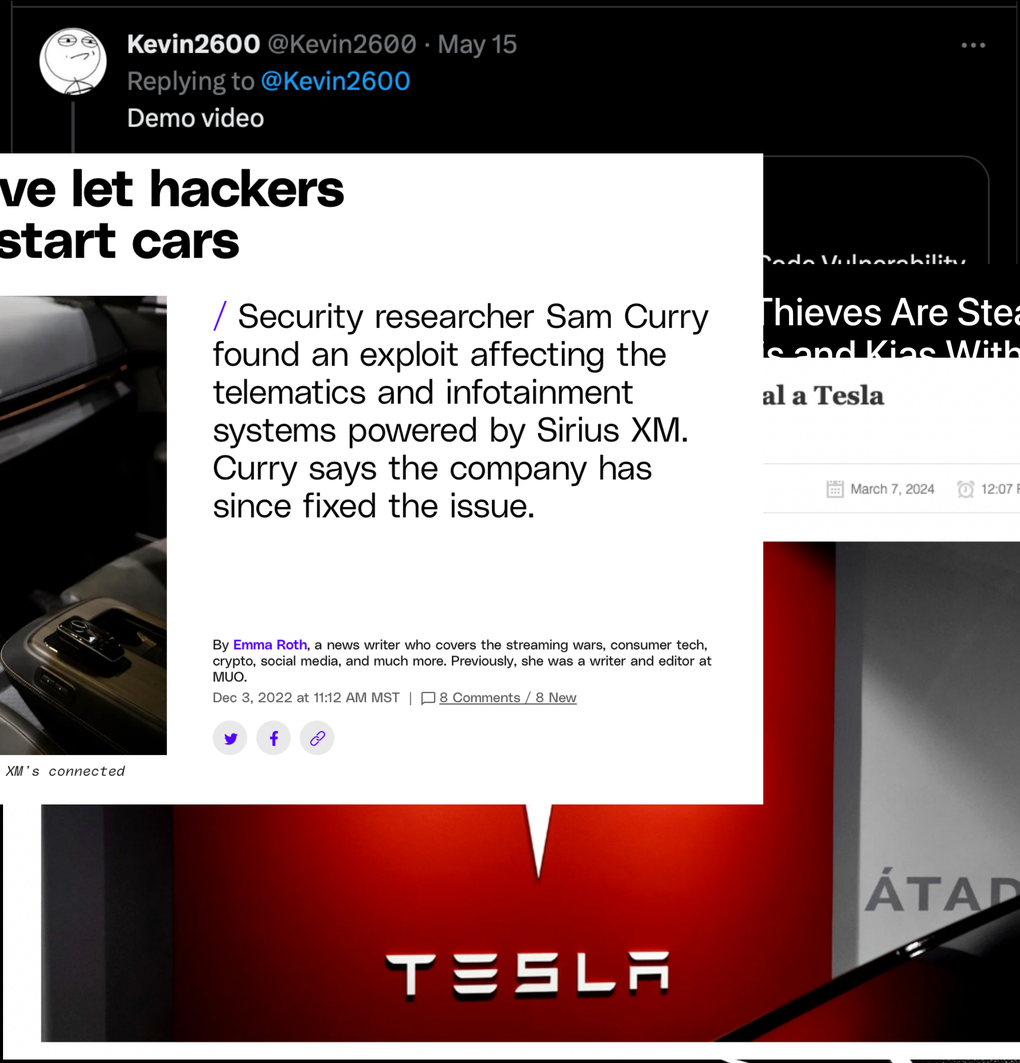
MiTM phishing attack can let attackers unlock and steal a Tesla

By Bill Toulas

March 7, 2024 12:07 PM



The State of Automotive Cybersecurity



The State of Automotive Cybersecurity

How Tech-Savvy Thieves Are Stealing Cars By Hacking Through Headlights

by **Nathan Ord** — Saturday, April 08, 2023, 02:37 PM EDT

in | 2 Comments | BECOME A PATRON

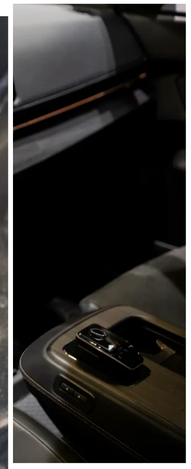
Early last year, hackers were replaying remote keyless system codes to unlock and steal Honda or Acura vehicles. This year, criminals of TikTok have been showing people how to break into certain Hyundai and Kia models with some hotwiring. However, criminals are upping their thieving game as car companies come to the rescue with patches and security solutions for vehicles. With this forced advancement come car thefts through attacks on the car's central nervous system called the Controller Area Network (CAN) bus.

Kevin2600 @Kevin2600 · May 15

Replying to @Kevin2600

Demo video

've let hackers start cars



/ Security researcher Sam Curry found an exploit affecting the telematics and infotainment systems powered by Sirius XM. Curry says the company has since fixed the issue.

By Emma Roth, a news writer who covers the streaming wars, consumer tech, crypto, social media, and much more. Previously, she was a writer and editor at MUO.

Dec 3, 2022 at 11:12 AM MST | 8 Comments / 8 New

s XM's connected

Thieves Are Stealing ...s and Kias With Just a ...al a Tesla

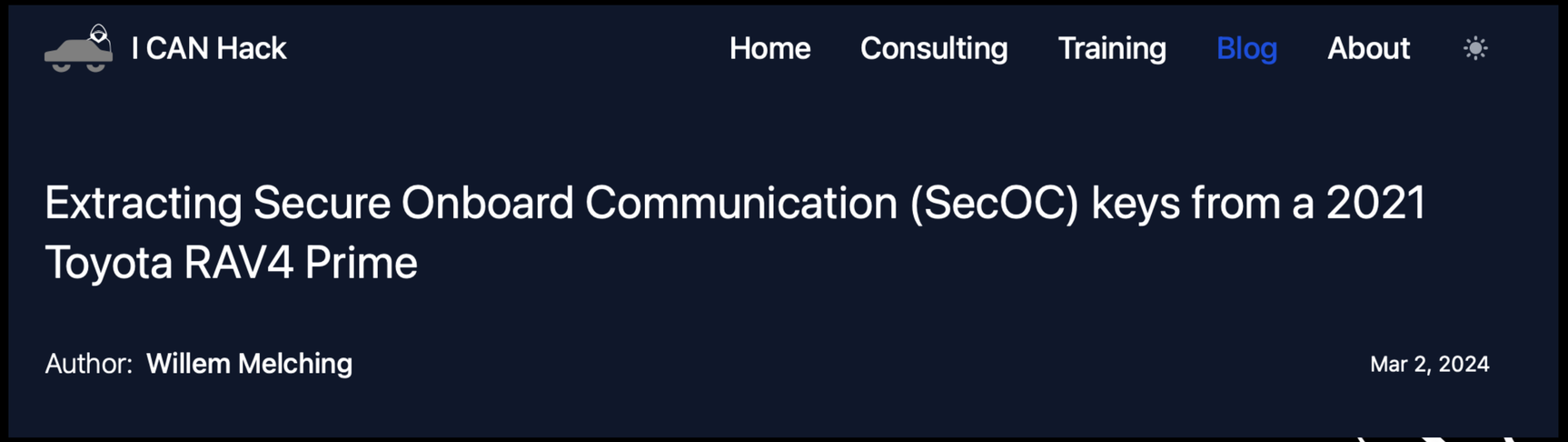
March 7, 2024 12:07 PM

...physical key.

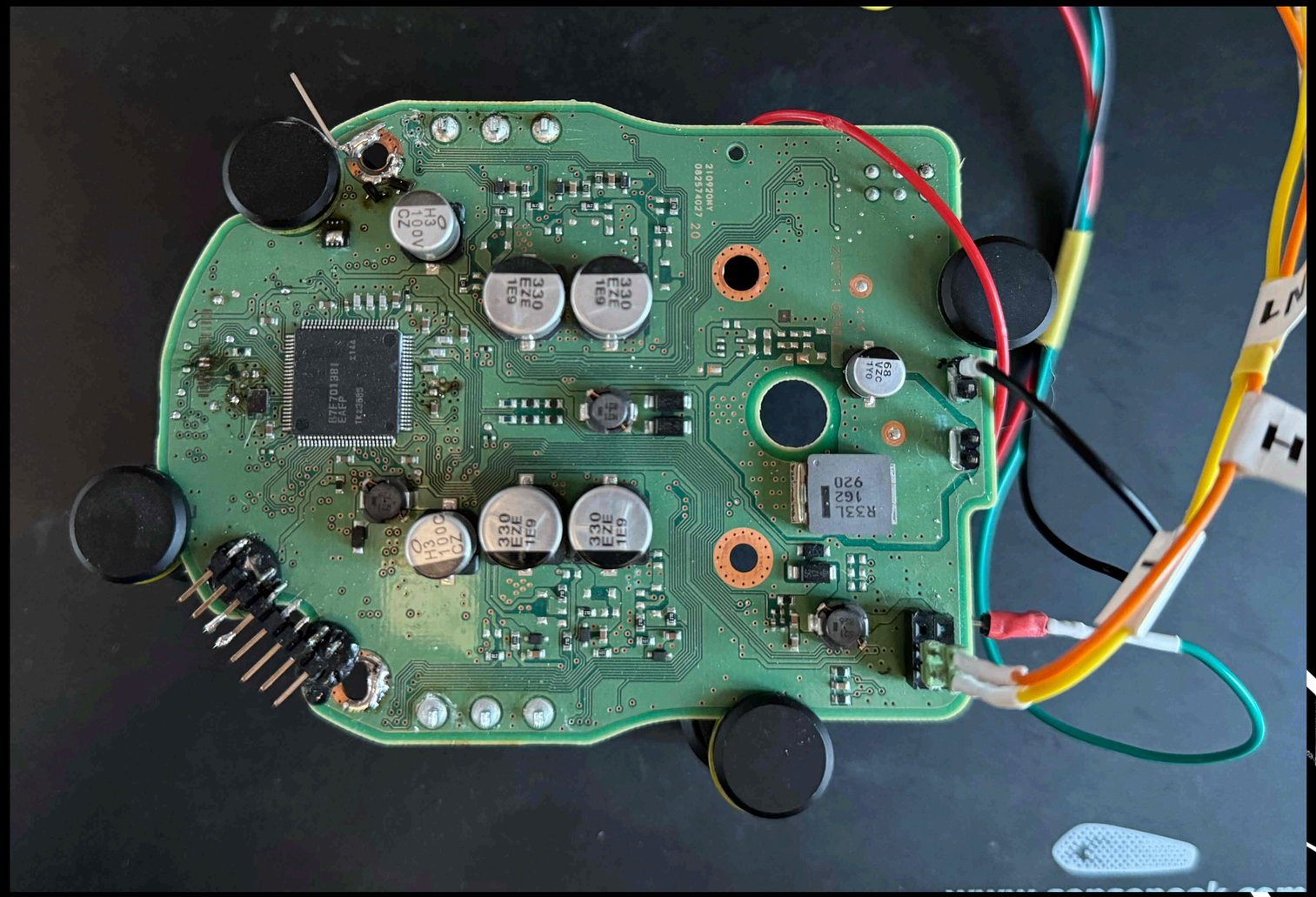
The State of Automotive Cybersecurity

Exposed debug interfaces in 2024? Is that even possible?

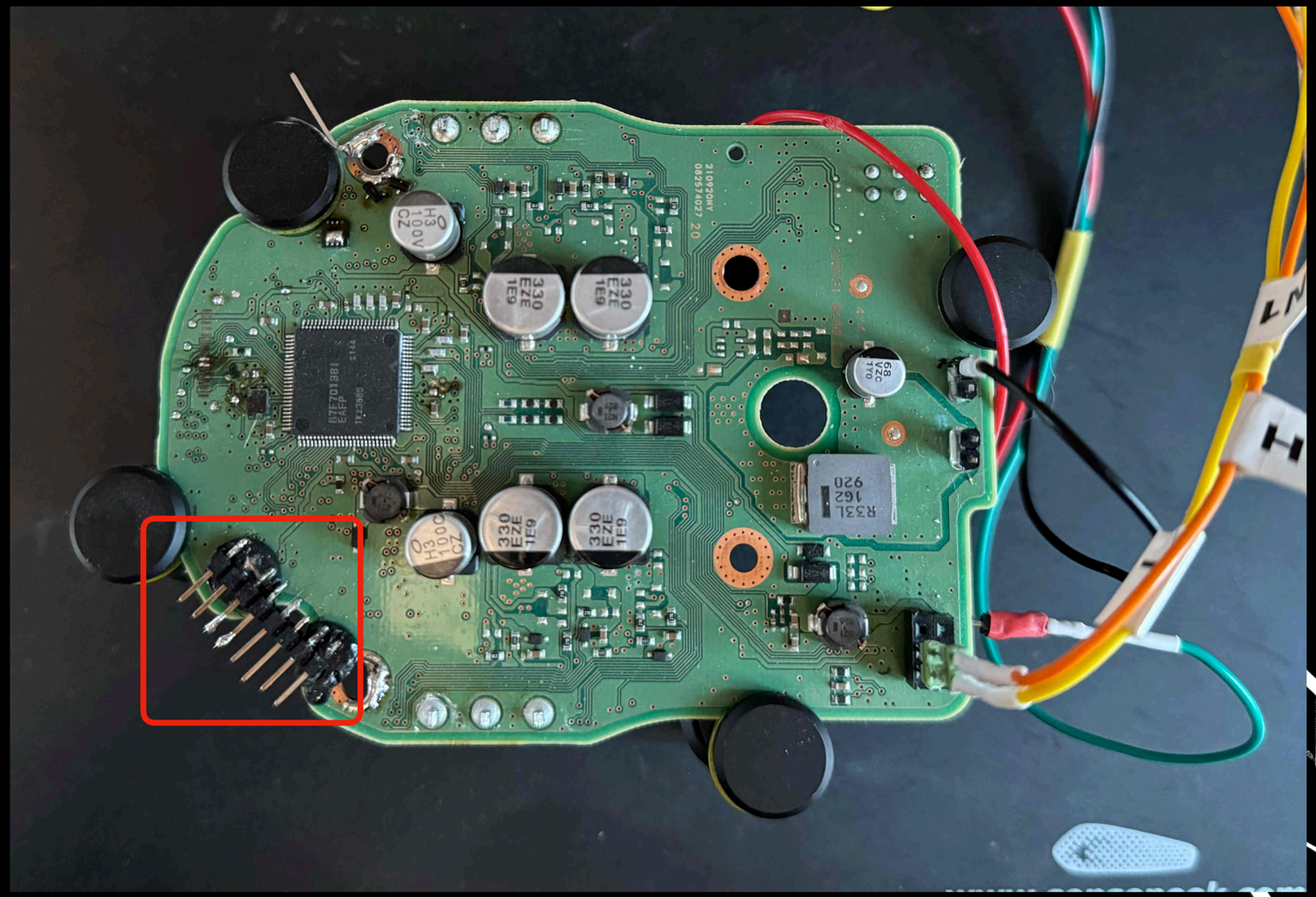
The State of Automotive Cybersecurity



The State of Automotive Cybersecurity



The State of Automotive Cybersecurity



The State of Automotive Cybersecurity

The State of Automotive Cybersecurity

- Is there a light in the end of the tunnel?
- Automotive, hardware and embedded industries are not new by any means
- Wide adoption of personal vehicles and really short life-cycles, made development shorter and targets more accessible to the "average Joe"

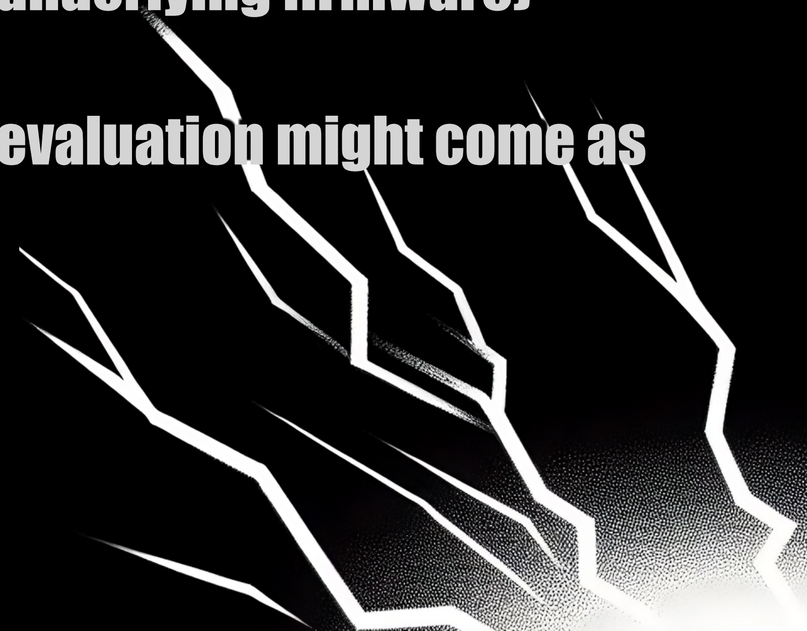
챕터 #1

THE HARDWARE

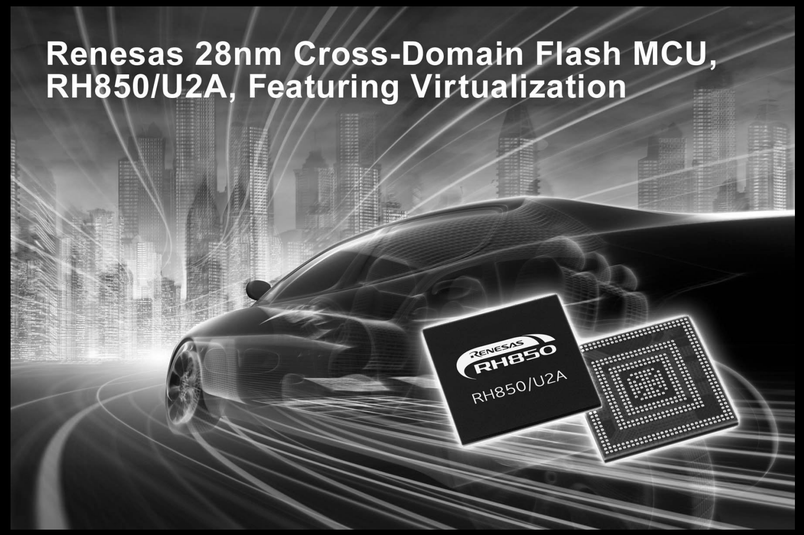
And the story of two SoCs

The Hardware

- Hardware is usually the last resort during our penetration testing projects
  - And the one where most mistakes are made
  - Higher chances of getting full access to the unit (or the underlying firmware)
  - In cases that critical information is required, hardware evaluation might come as the first step
    - e.g. missing knowledge of start up sequence



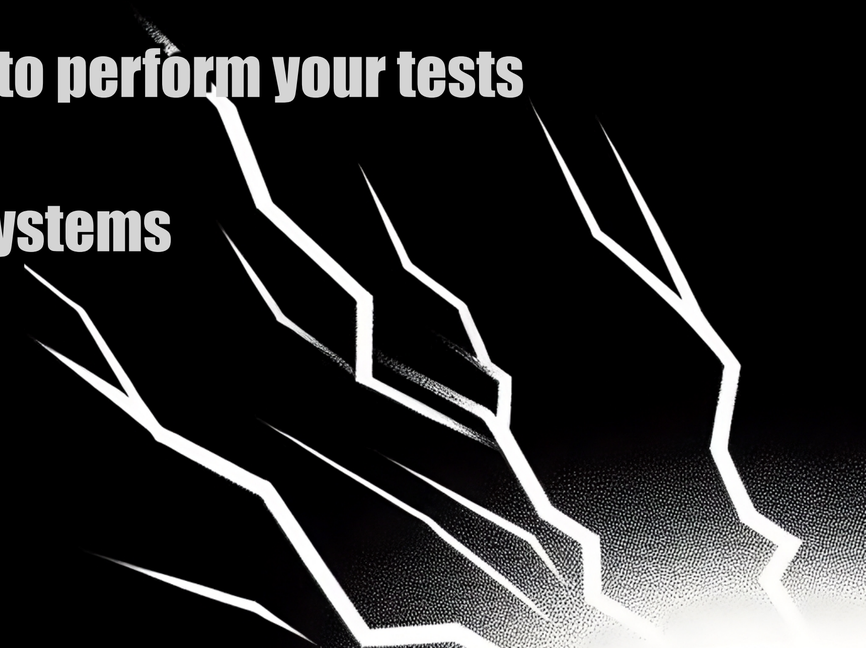
The Hardware



- Most Electronic Control Units (ECUs) use custom automotive specific architectures
  - e.g. Tricore, Renesas, etc.
  - Safety related functionality, with "real-time" responsiveness is required in automotive systems

The Hard(ware) Reality

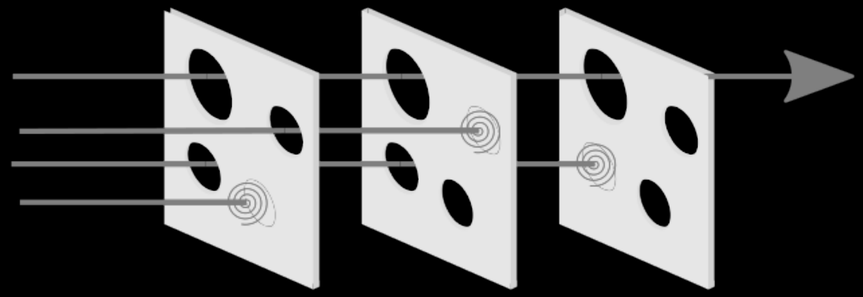
- The pace of research in those systems is extremely slower than conventional/ commercial systems (e.g. ARM, x86)
- The "buy-in" is way higher than in other industries
  - Most times you need to have access to a vehicle or ECU to perform your tests
- The product life-cycles are way longer than commercial systems
  - Both OEM and supplier products



Safety-Critical Systems

- Many ECUs are considered "Safety-critical systems"
  - Their failure may result in loss or damage to equipment, death or injury, and environmental harm
    - NOT only applicable to the Automotive Industry
  - In automotive, it mostly applies to ECUs which supply critical functionality or can directly affect safety
    - e.g. Airbag, Braking, Power Steering, and others

Safety-Critical Systems



- Hardware and software need to be implemented/developed with strict safety standards
- Often used together with the Swiss Cheese Model - "How can a threat escalate?"
- Several ways exist to minimise the risk

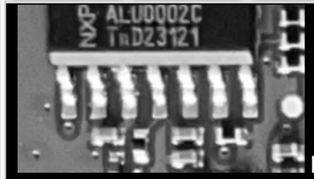
• Reason, J., Hollnagel, E., & Paries, J. (2006). Revisiting the Swiss cheese model of accidents. Journal of Clinical Engineering, 27(4), 110-115.

Hardware Testing Checklist

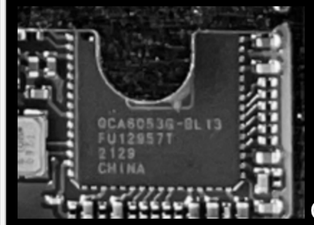
- **Analyse the target board**
- **Analyse the hardware components**
- **Find schematics and data-sheets online**
- **Discover testing points and map them to the hardware components**
- **Find the proper debugger and connect to the testing points**
- **Bypass the protections (if any) and elevate your access to the target**

- For more info, PLEASE do your self a favour and read the Hardware Hackers Handbook. ;)


dnt
QUALCOMM
MDM6200
OK
M
multicomp PRO
JBC
OK



High-Speed CAN Transceiver with Standby and Sleep Mode
https://www.nxp.com/docs/en/data-sheet/TJA1043.pdf

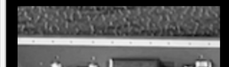


Qualcomm® APQ8053 SoCs for IoT
https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/apq8053-socs-product-brief_87-pg760-1-b.pdf

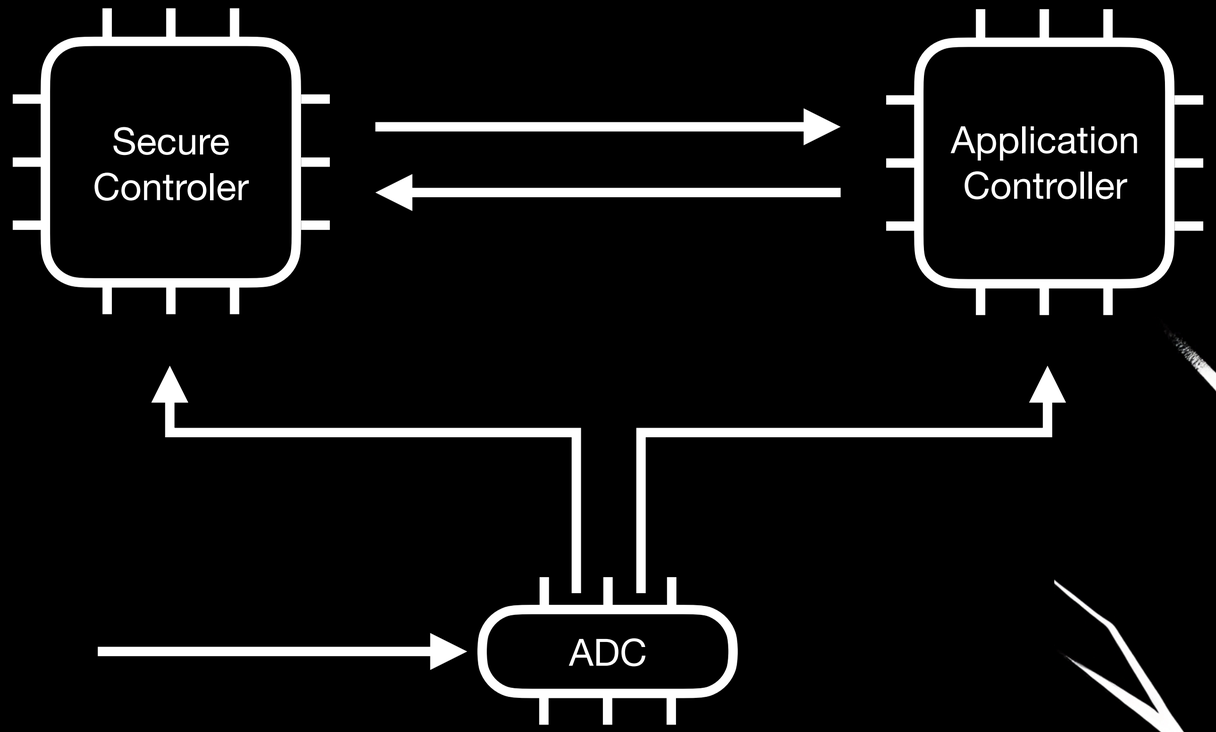
---

LTE/4G Modem Chip

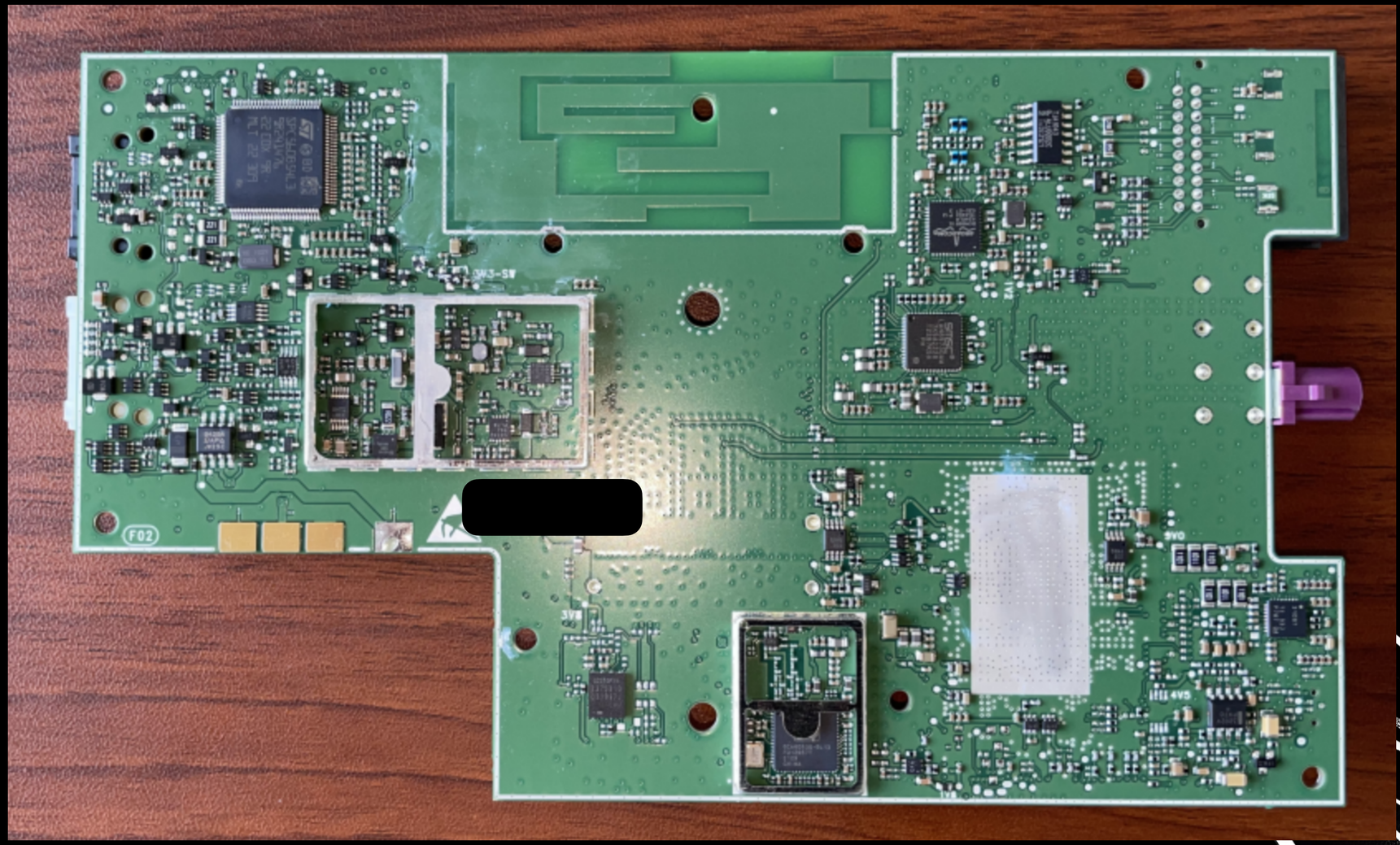
CMOS QUAD 2-INPUT NAND SCHMITT TRIGGER
https://www.ti.com/lit/ds/symlink/cd4093b-q1.pdf?ts=1694496622695



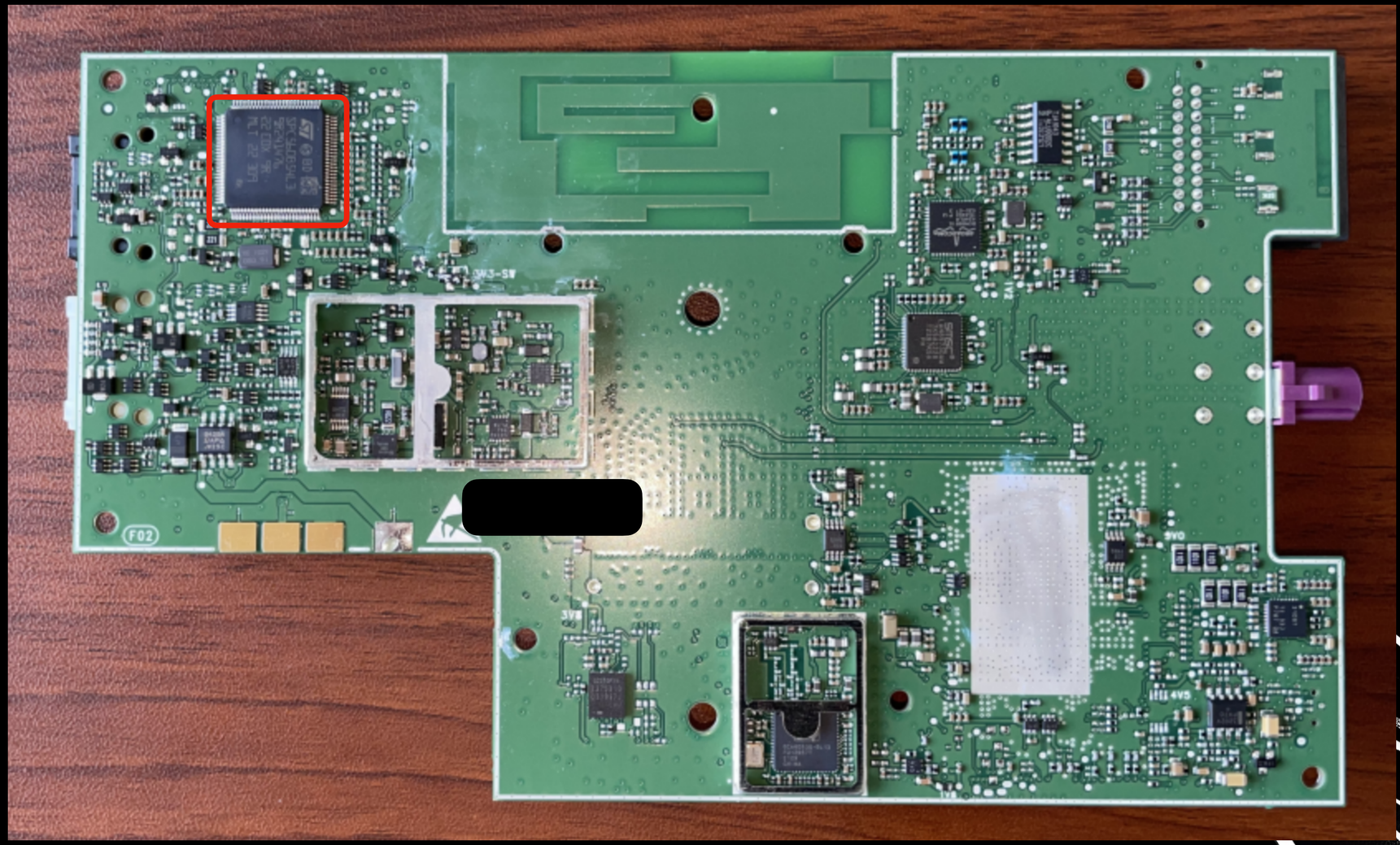
Case I - The story of two SoCs



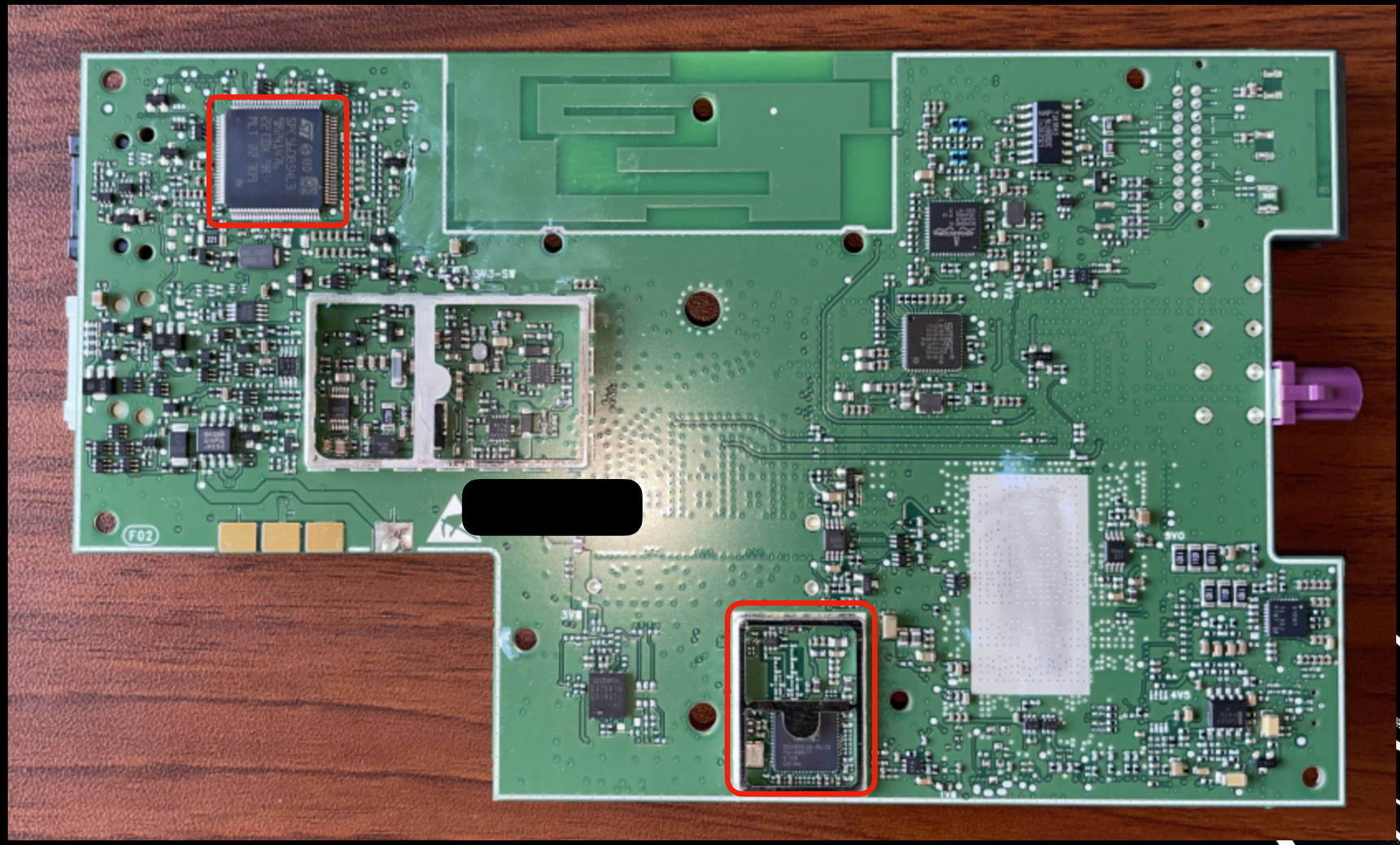
Case 1 - The story of two SoCs



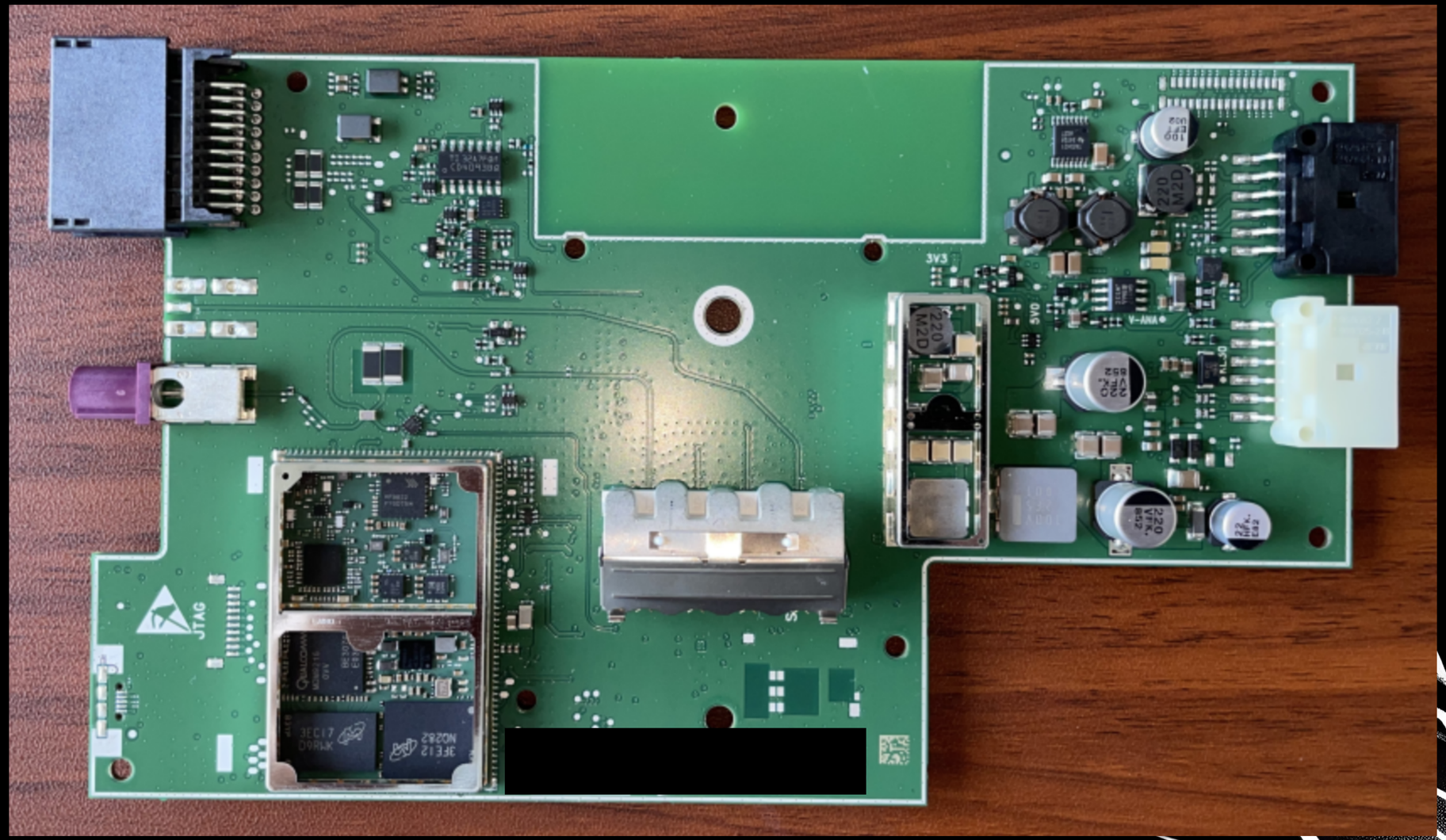
Case I - The story of two SoCs



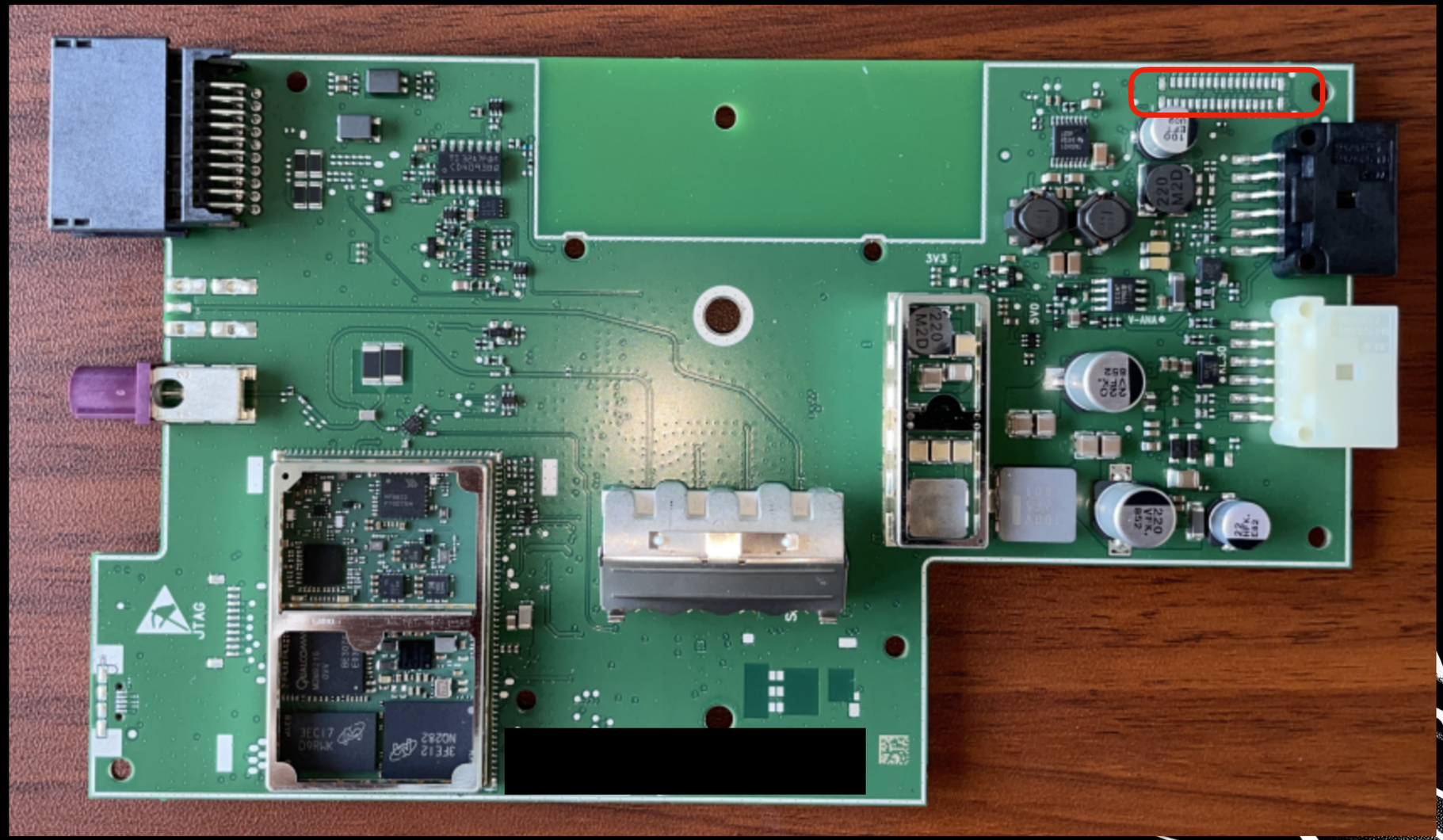
Case I - The story of two SoCs



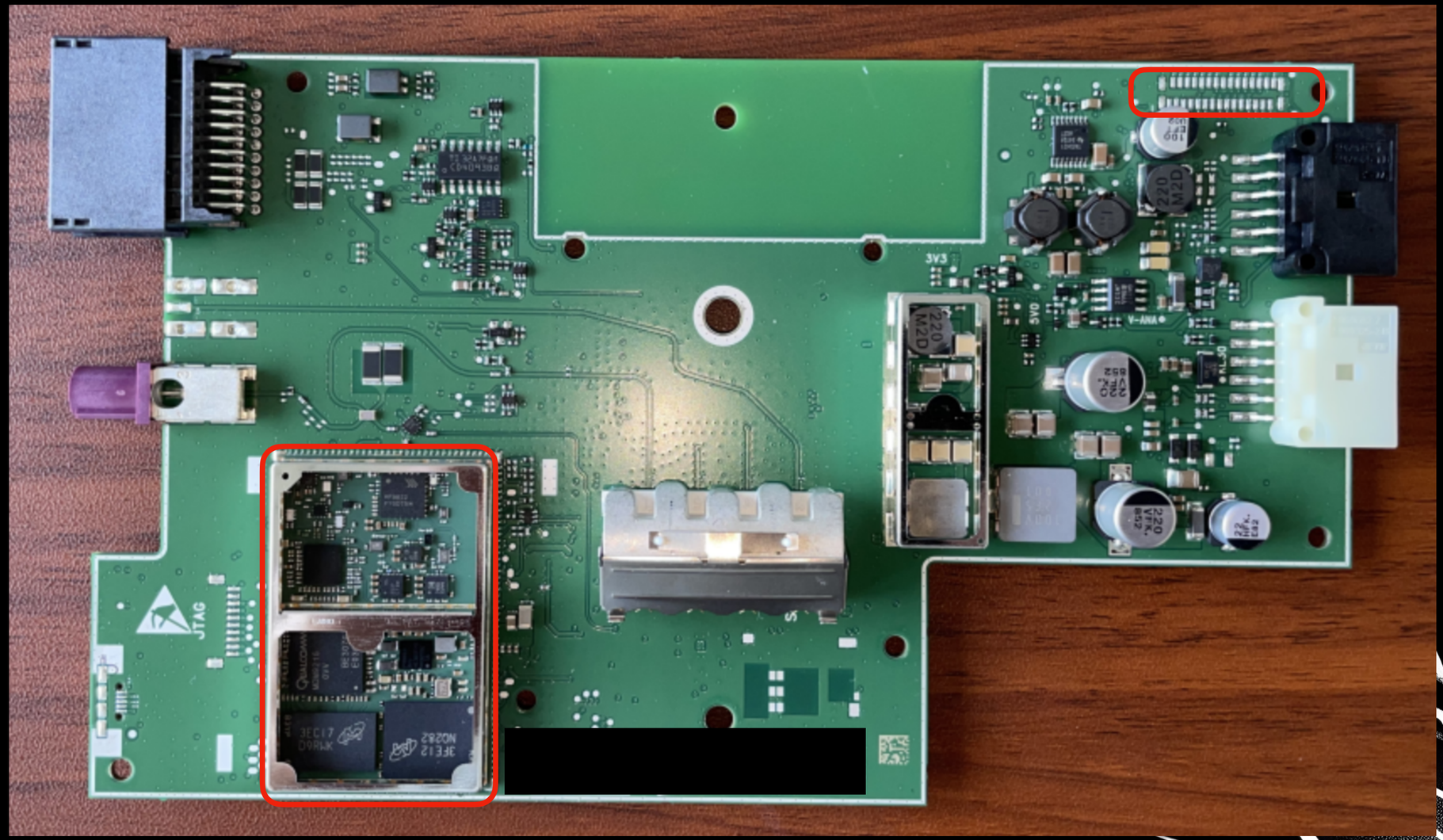
Case 1 - The story of two SoCs



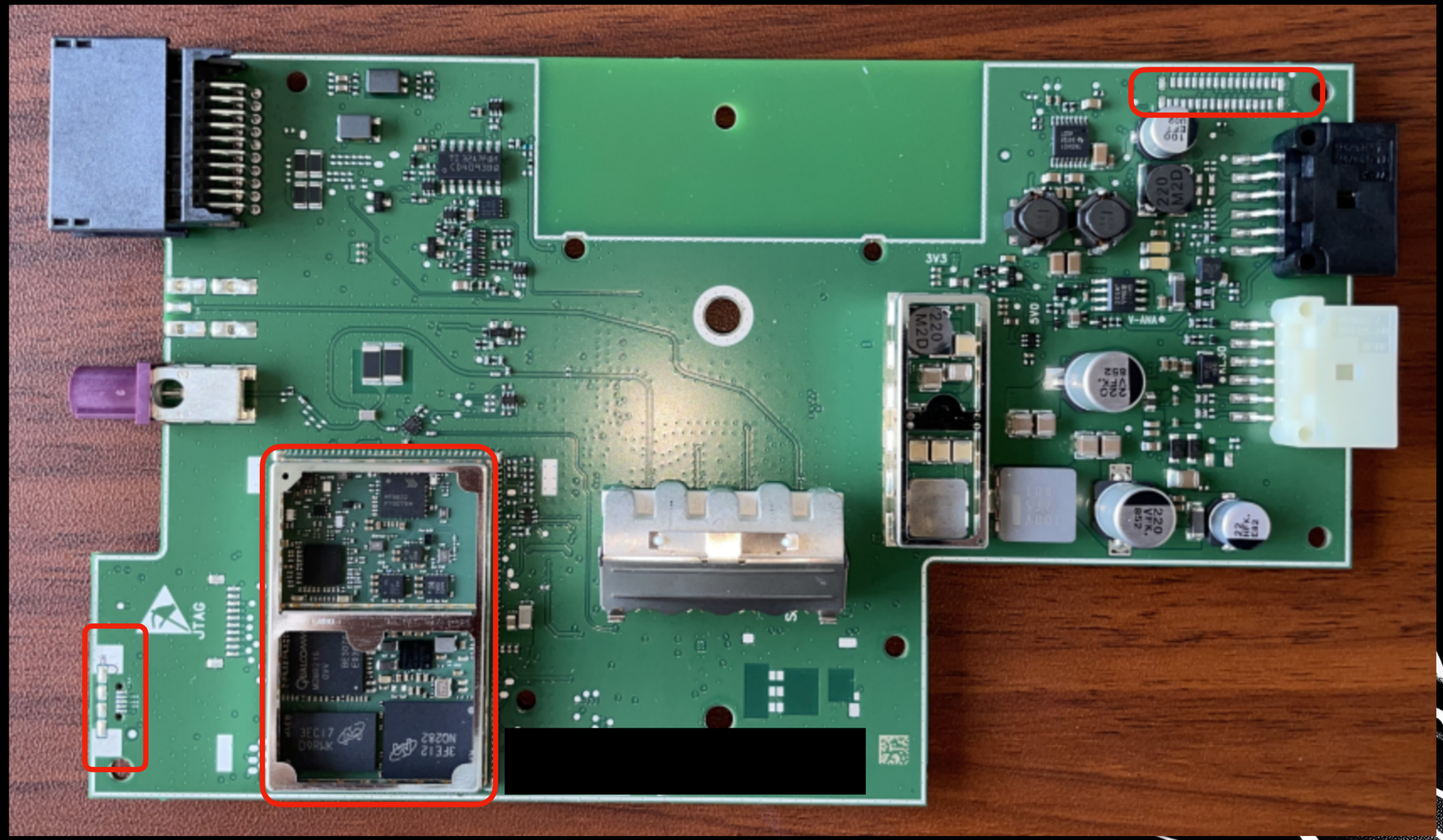
Case 1 - The story of two SoCs



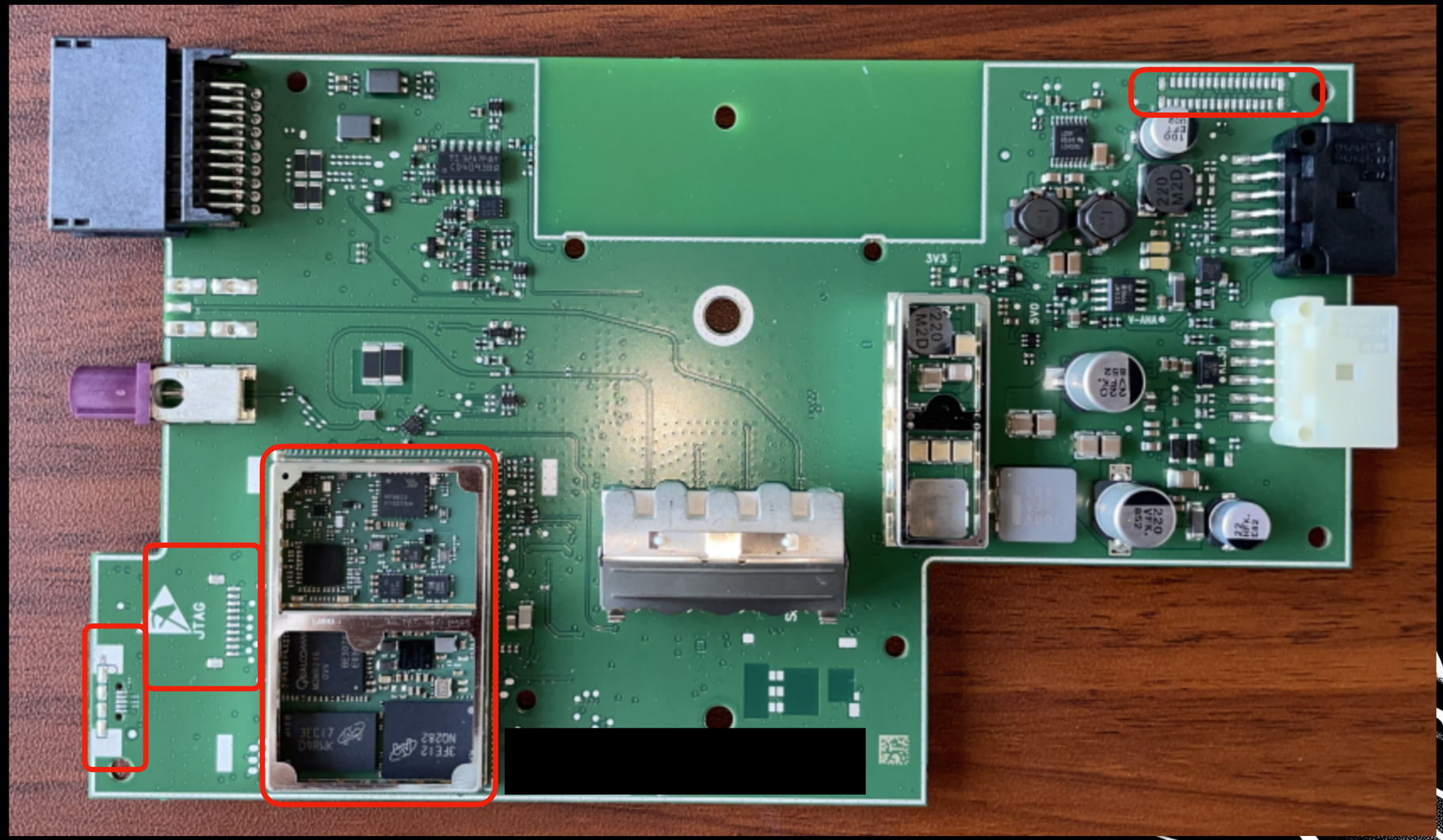
Case I - The story of two SoCs



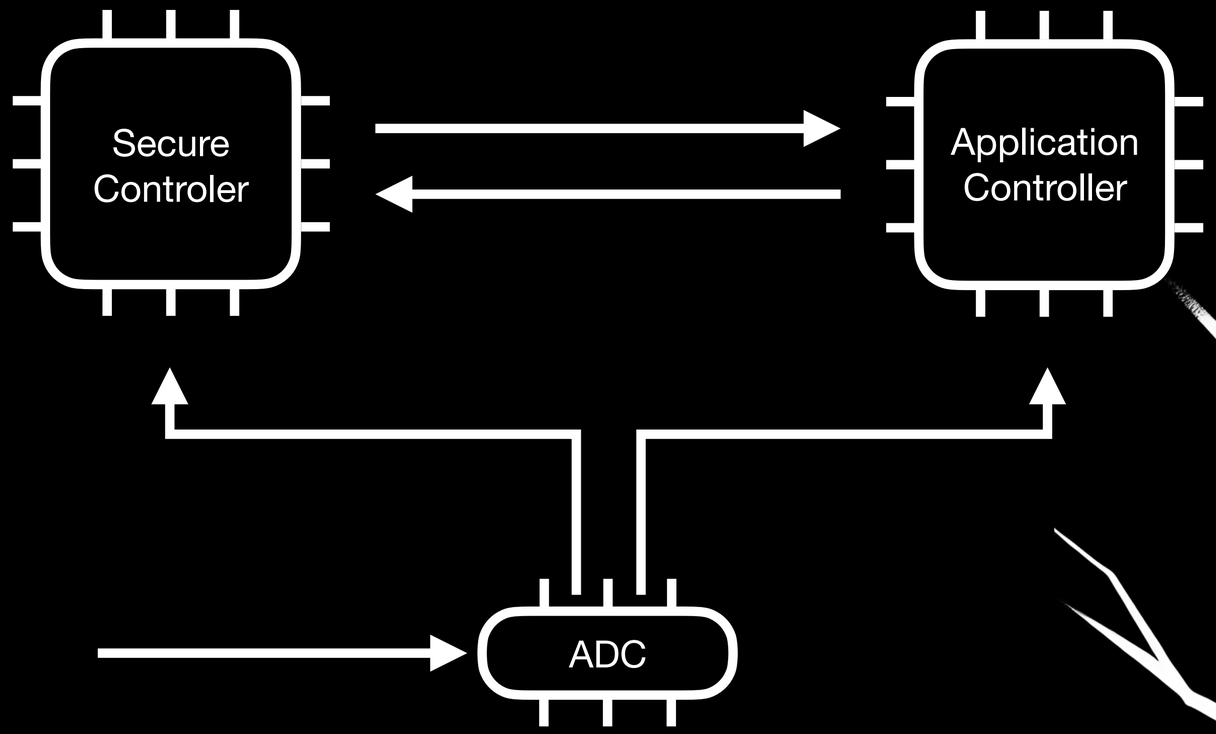
Case I - The story of two SoCs



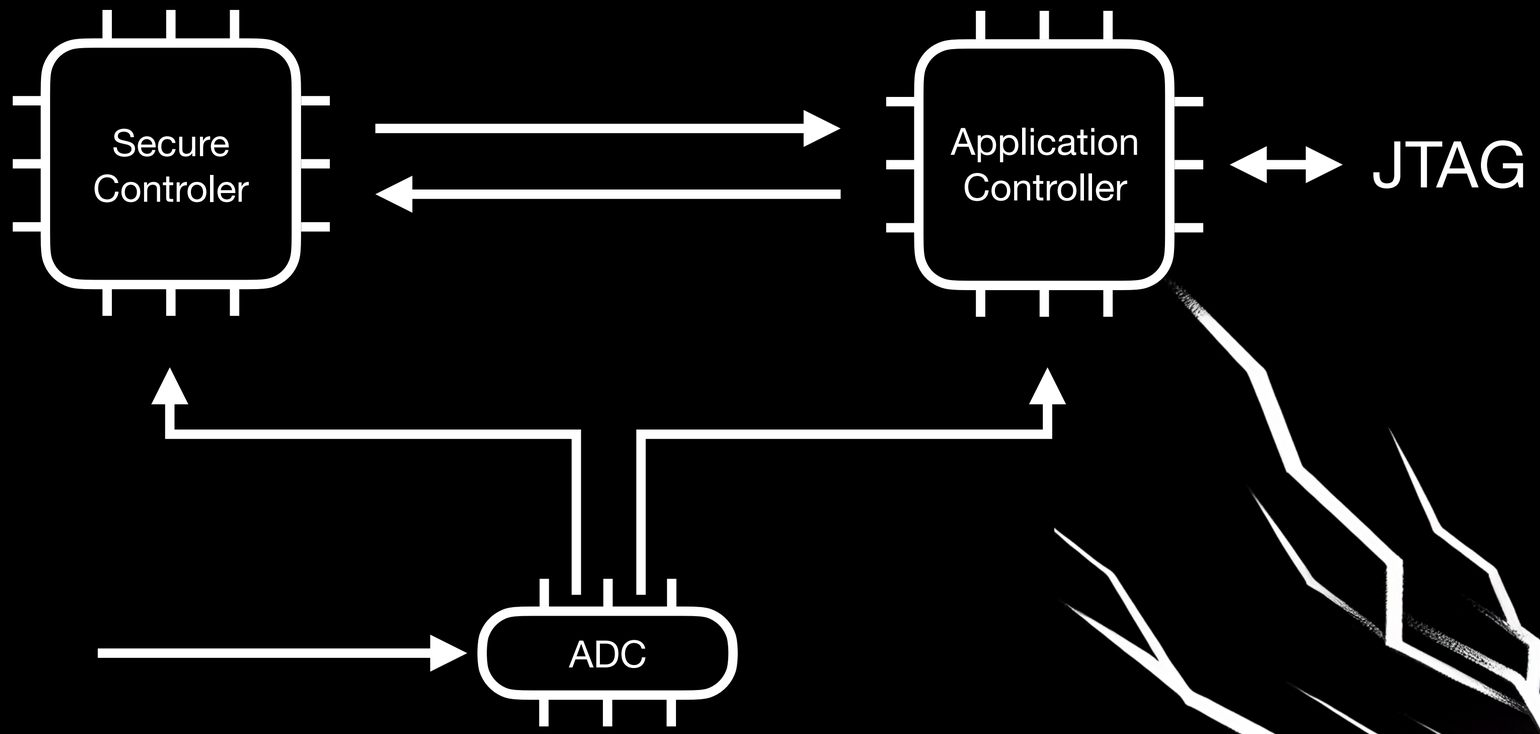
Case I - The story of two SoCs



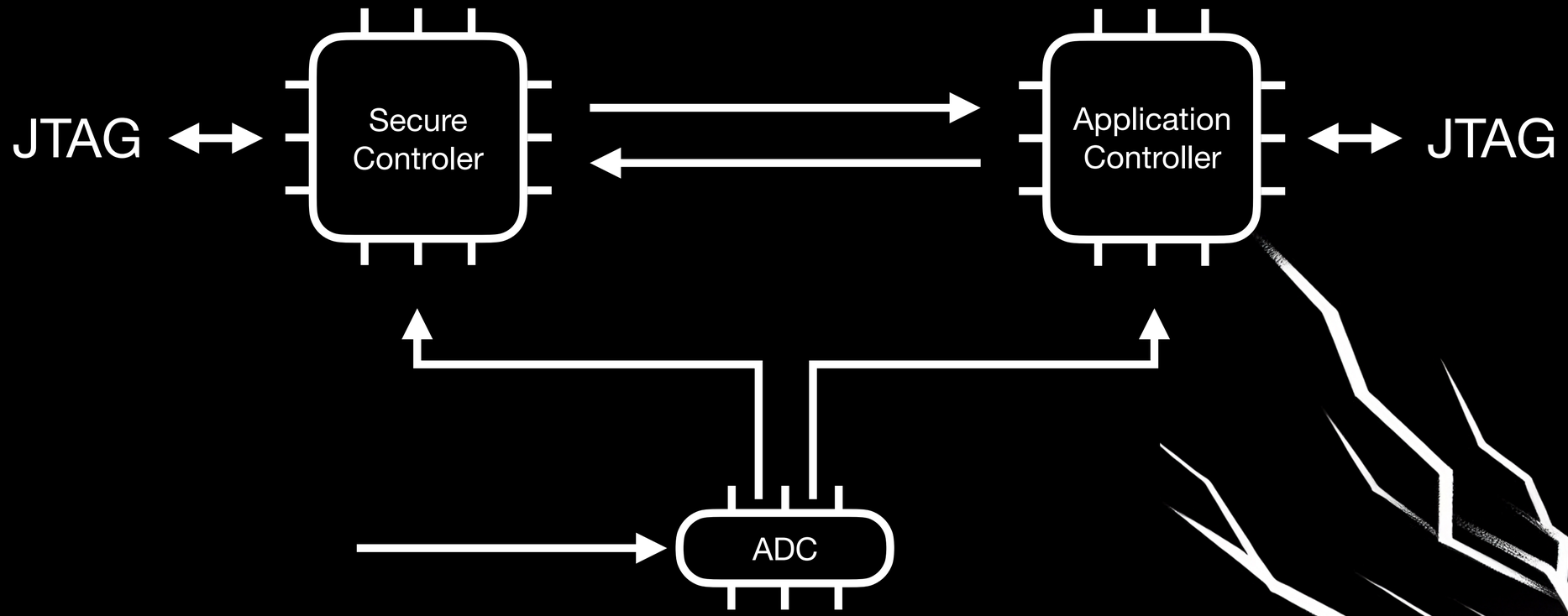
Case I - The story of two SoCs



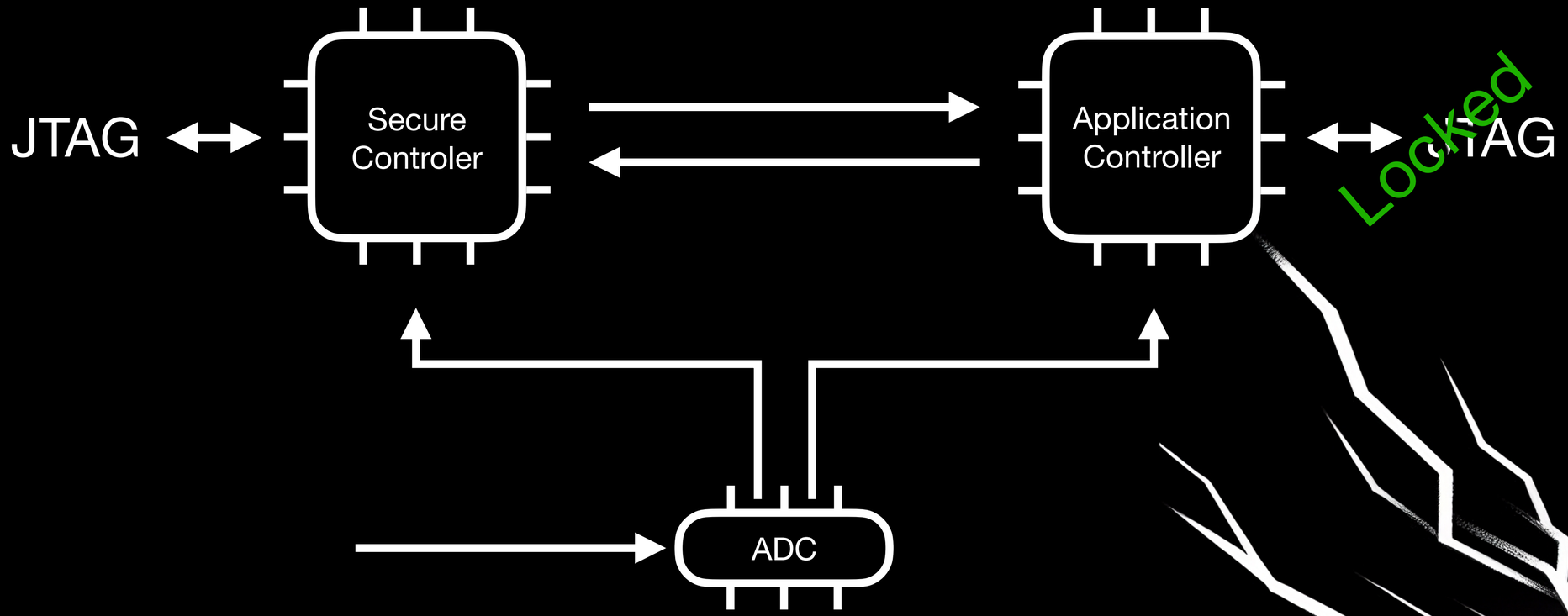
Case I - The story of two SoCs



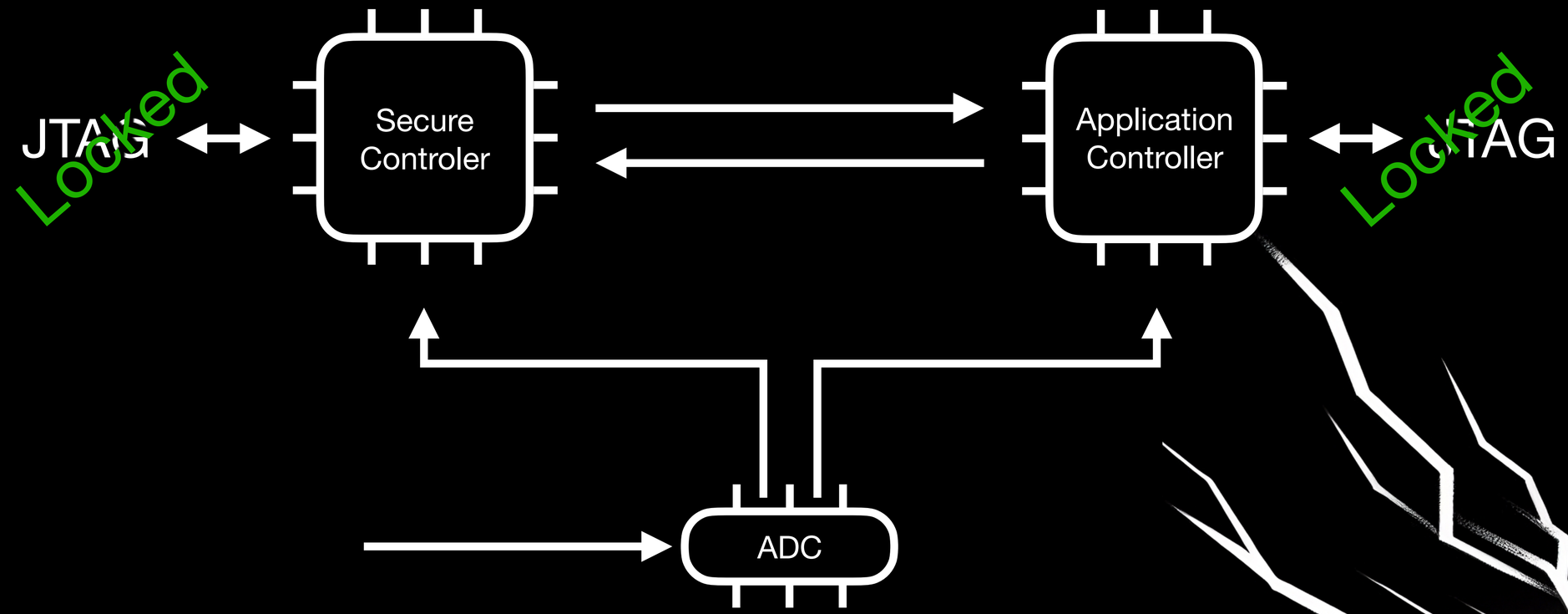
Case I - The story of two SoCs



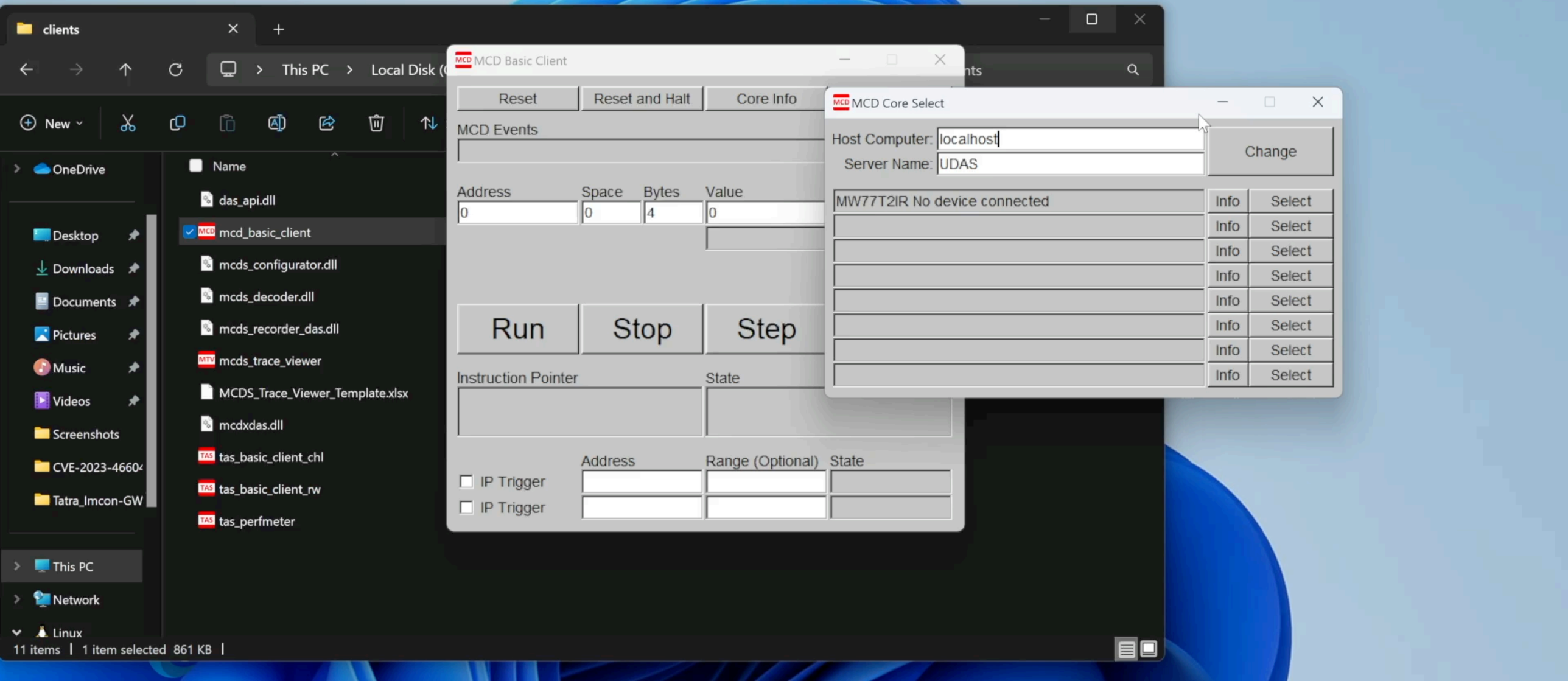
Case I - The story of two SoCs



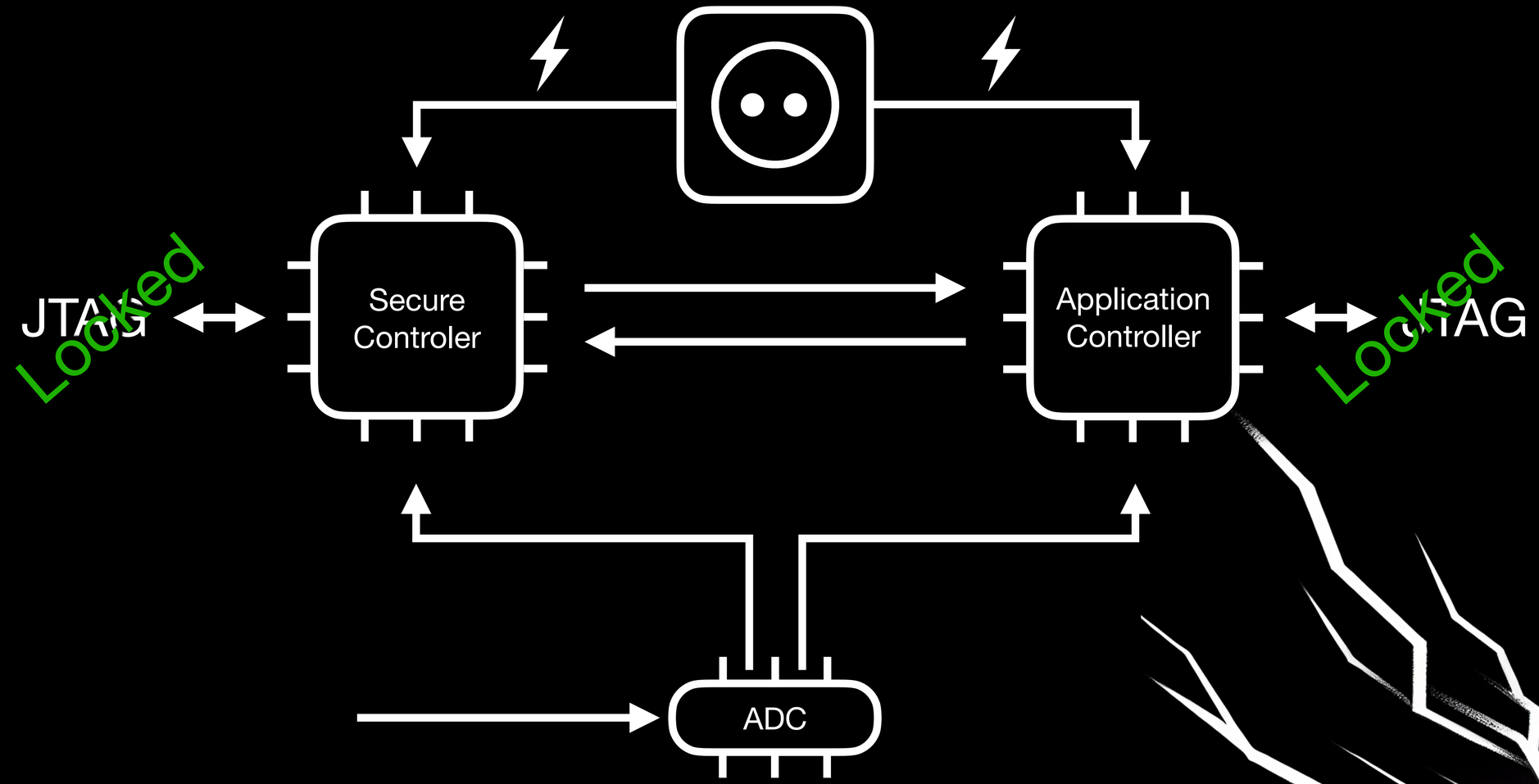
Case I - The story of two SoCs



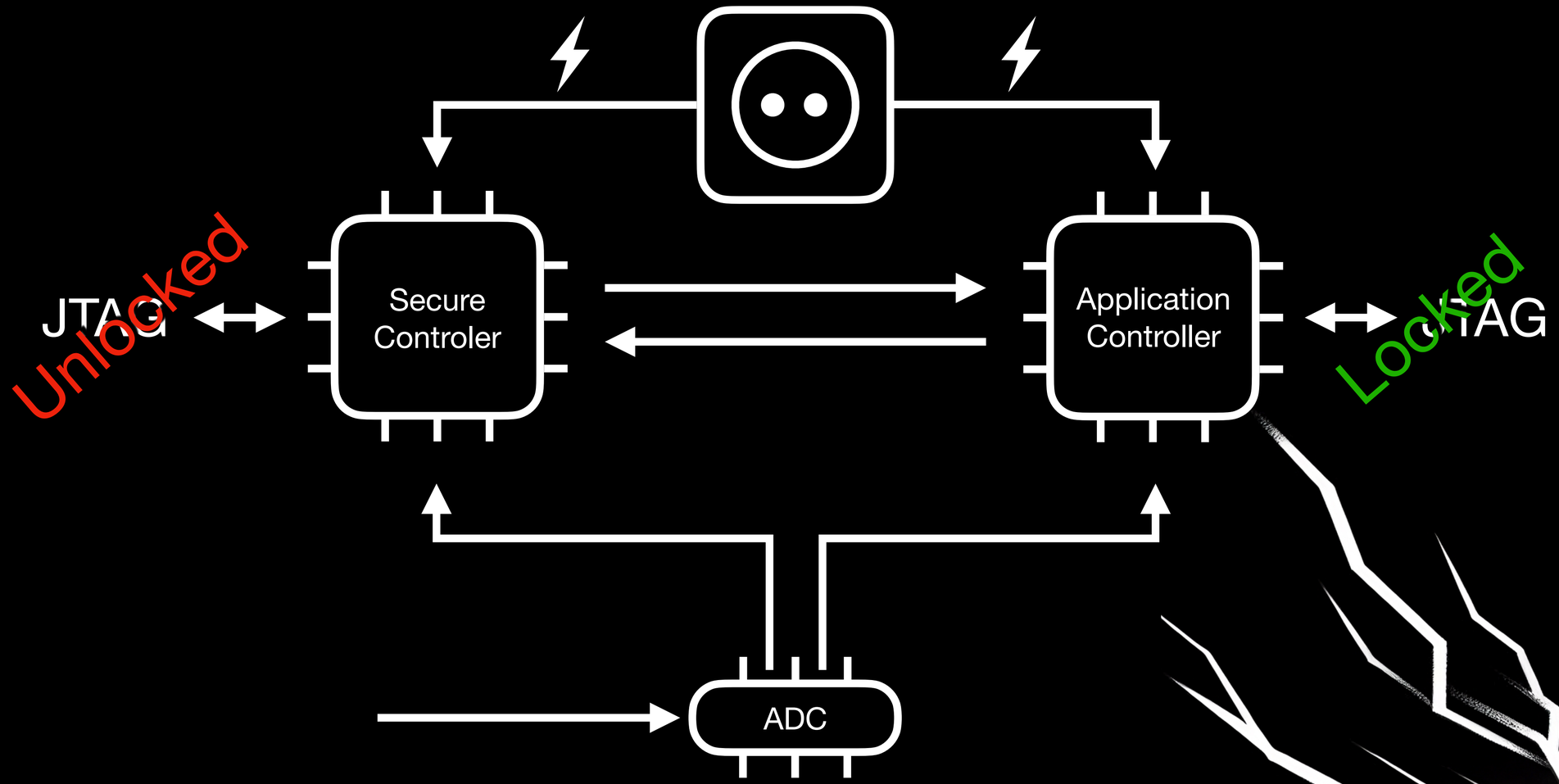
DEMO


clients
This PC
Local Disk
New
OneDrive
Name
das_api.dll
mcd_basic_client
mcds_configurator.dll
mcds_decoder.dll
mcds_recorder_das.dll
mcds_trace_viewer
MCDS_Trace_Viewer_Template.xlsx
mcdxdas.dll
tas_basic_client_chl
tas_basic_client_rw
tas_perfmeter
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
CVE-2023-4660
Tatra_Imcon-GW
This PC
Network
Linux
11 items | 1 item selected 861 KB
MCD Basic Client
Reset
Reset and Halt
Core Info
MCD Events
Address
Space
Bytes
Value
0
0
4
0
Run
Stop
Step
Instruction Pointer
State
Address
Range (Optional)
State
IP Trigger
IP Trigger
MCD Core Select
Host Computer:
localhost
Server Name:
UDAS
Change
MW77T2IR No device connected
Info
Select

Case I - The story of two SoCs



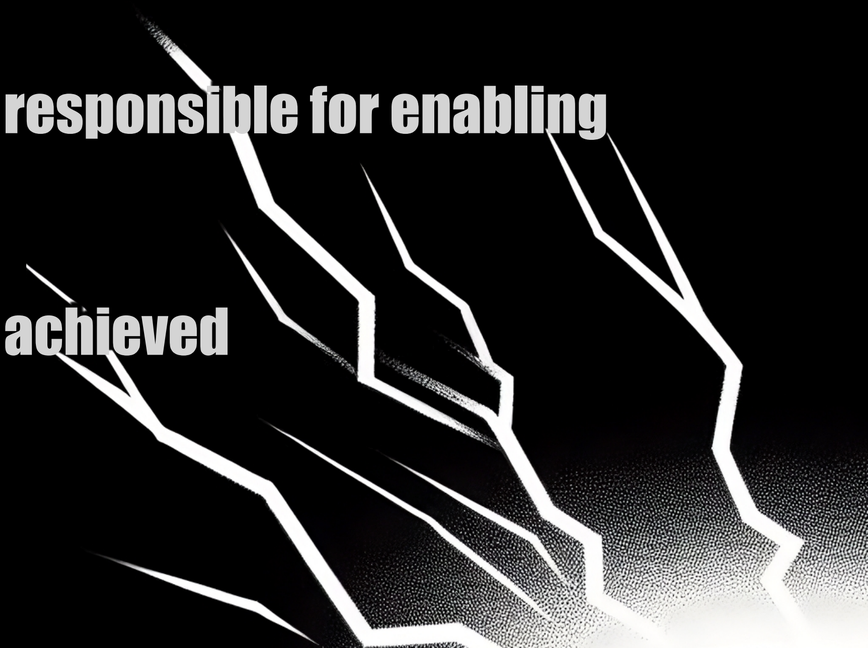
Case I - The story of two SoCs




SECURITY LEVEL?
MAXIMUM!

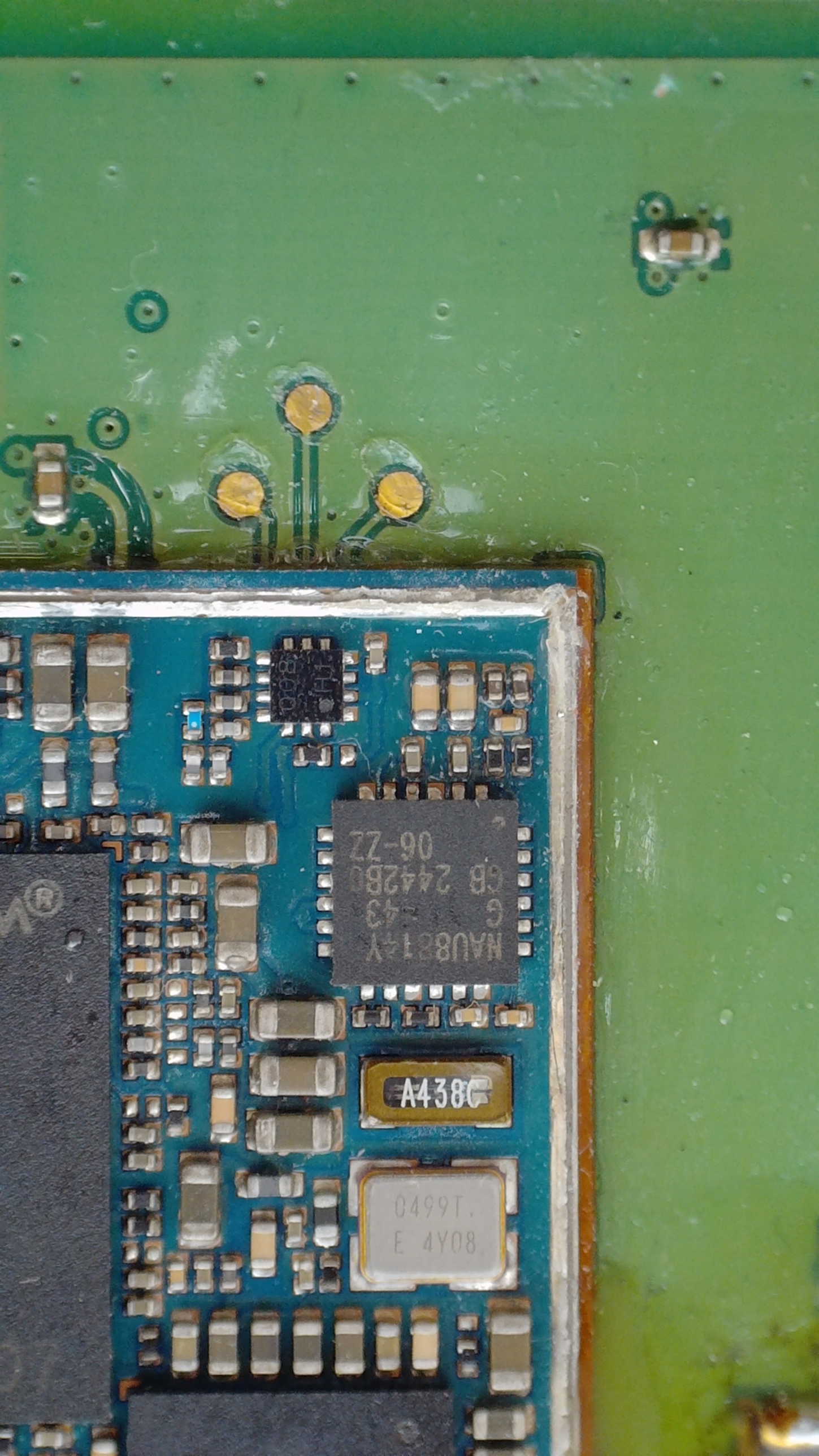
Case II - Secret Pins Leading to Secret Modes

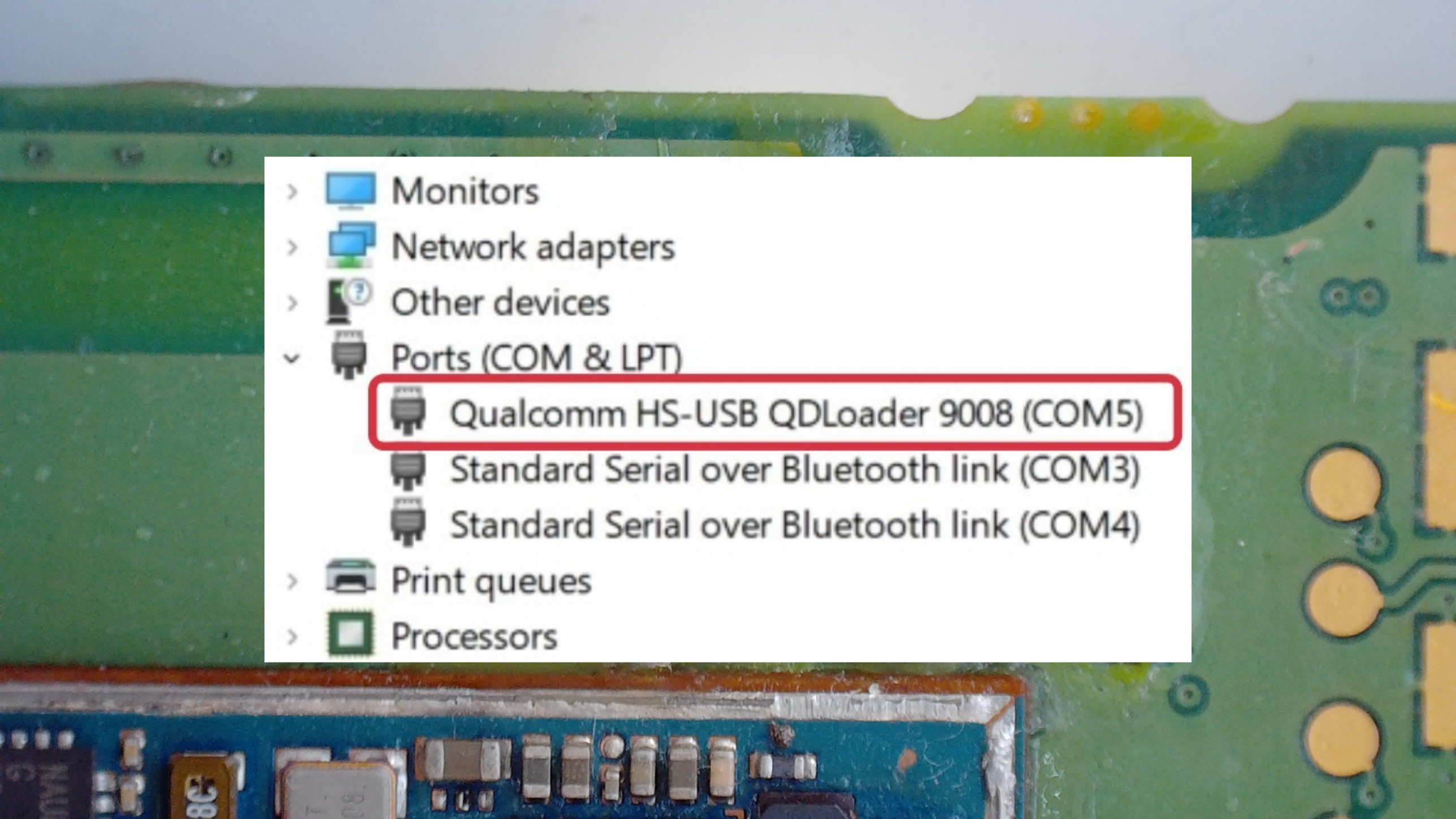
- Security through obscurity is a really common tactic in hardware security mitigations
- Secret pins is a really common way of including debug functionality in the unit, but obscuring it under random pins that need to be shorted
- One such example is Qualcomm EDL Mode
  - To enable, it's as easy as shorting the two dedicated pins responsible for enabling EDL functionality
  - After that, a full dump and re-flash of the firmware can be achieved
    - Main reason behind it - De-bricking



- https://en.wikipedia.org/wiki/Qualcomm_EDL_mode


QUALCOMM®
MDM6200


A438C


Monitors
Network adapters
Other devices
Ports (COM & LPT)
Qualcomm HS-USB QDLoader 9008 (COM5)
Standard Serial over Bluetooth link (COM3)
Standard Serial over Bluetooth link (COM4)
Print queues
Processors

The Hardware - Is that all?

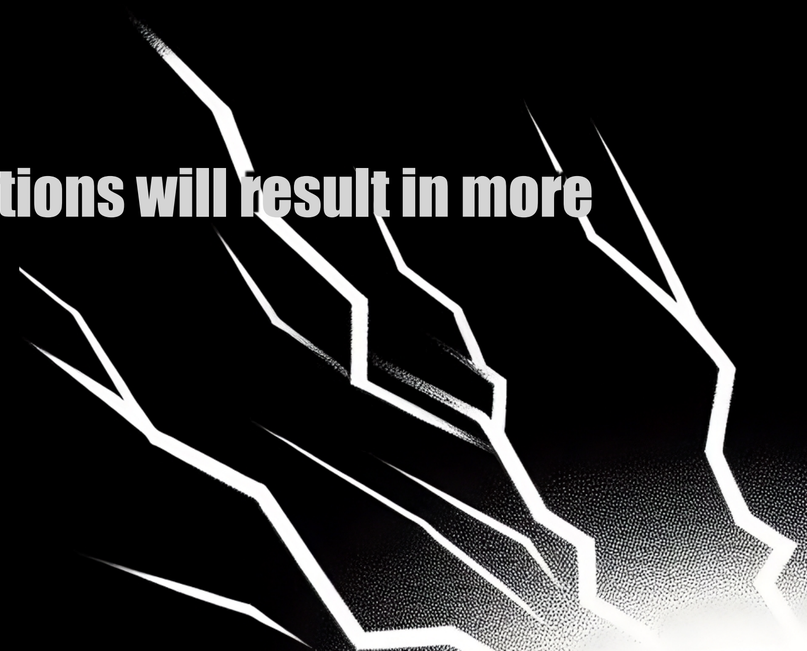
- Many more hardware attack paths exist
- Some are more lab oriented, while others are more applicable to real world
- Debug and other easily accessible interfaces need to be protected
  - e.g. JTAG, UART, SPI, I2C are some of the common attack paths that attackers start with

The Hardware - Is that all?

- Manufacturers need to strictly remove ALL debug interfaces in production units
  - As we saw, even when protected, they can eventually be bypassed
  - Traces and pins should also be removed from production hardware
  - Non-writable fuses (preferably hardware), should be implemented

The Hardware - Is that all?

- While no perfect solution exist, the more steps an attacker needs to gain access to a target, the more expensive it gets
- Same applies to tinkerers and home hackers / tuners
    - While most bypass methods are not "new", more protections will result in more effort needed to achieve the end goal



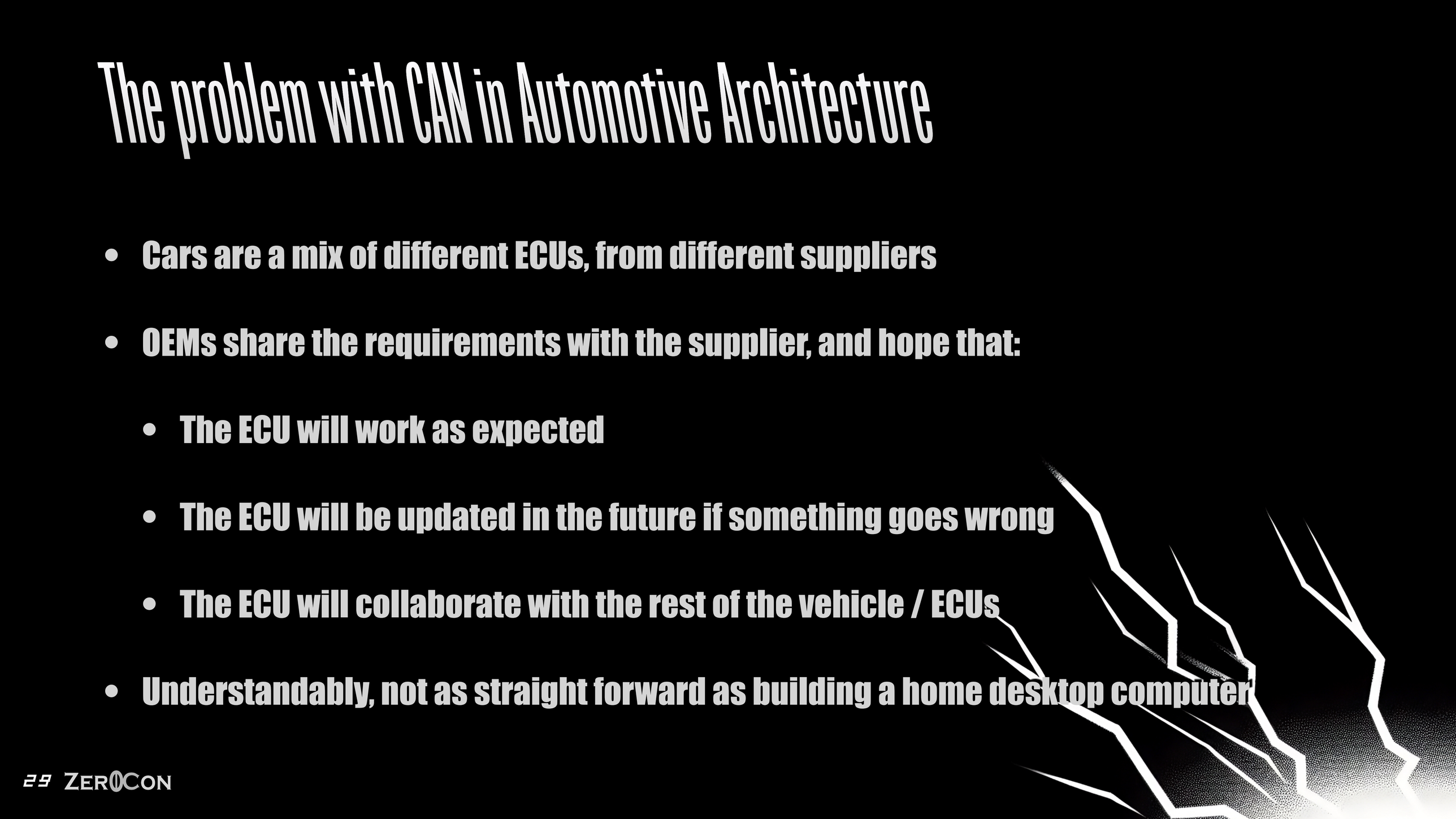
챕터 #2

Automotive Architecture

And Security Through Obscurity

The problem with CAN in Automotive Architecture

- Cars are a mix of different ECUs, from different suppliers
- OEMs share the requirements with the supplier, and hope that:
  - The ECU will work as expected
  - The ECU will be updated in the future if something goes wrong
  - The ECU will collaborate with the rest of the vehicle / ECUs
- Understandably, not as straight forward as building a home desktop computer



The problem with CAN in Automotive Architecture

This is the Worst Car I've Ever Reviewed

4.3M views • 2 weeks ago

AF Auto Focus

Do not buy this version of the Fisker Ocean The dealership: ...

4K

- https://www.youtube.com/watch?v=6xWXRk3yaSw

The problem with CAN in Automotive Architecture

- https://www.youtube.com/watch?v=6xWXRk3yaSw

The problem with CAN in Automotive Architecture

- https://www.youtube.com/watch?v=6xWXRk3yaSw

The problem with CAN in Automotive Architecture

- CAN - Physical and Data link layer communication in ECUs
- On "higher layers", data is exchanged without encryption
- Obscurity of proprietary messages acts as a layer of "protection"
- Protocols on top of CAN are also unencrypted
  - Interception, injection and replication are all applicable
- Protections like SecOC (Message Authentication) can be implemented

The problem with CAN in Automotive Architecture

So, what's the issue here?

The problem with CAN in Automotive Architecture

- Manufacturers keep struggling to implement CAN in a secure way
- Industry is moving towards other solutions (e.g. Automotive Ethernet, FlexRay, etc.) but the lifecycle of the products is really slow
- It's not only up to manufacturers, but also up to suppliers to support it
- Incidents with current gen vehicles are really common
- Will SDVs be the holly grail of Automotive?

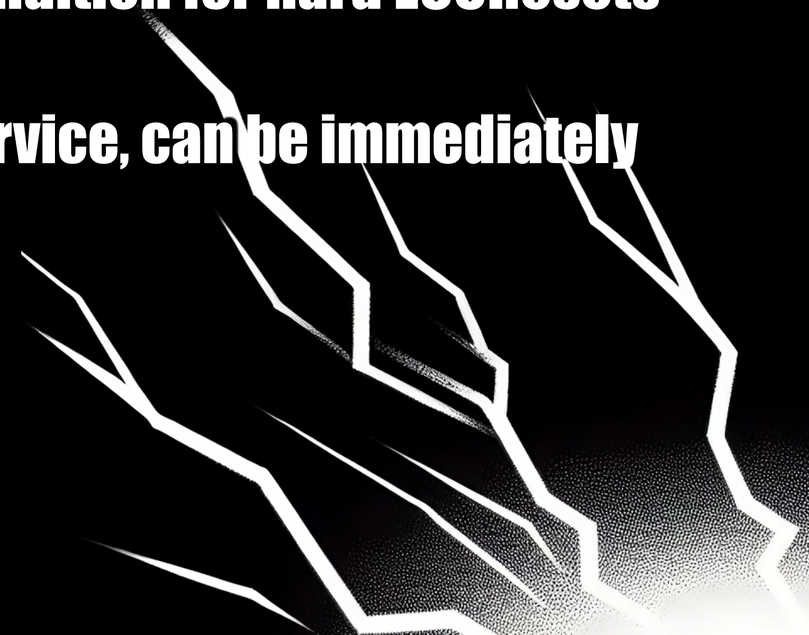
Source: https://www.thedrive.com/news/shadetree-hackers-are-stealing-cars-by-injecting-code-into-headlight-wiring

Case III - UDS Diagnostics and Patches over Patches

- UDS stands for Unified Diagnostic Services, an application layer protocol for communication between electronic control units in automotive electronics
- Allows diagnostic functions such as reading and erasing fault codes, programming, testing, and monitoring of ECUs
- Consists of several “services" which can be used to perform specific actions
- A really common authentication scheme in UDS is the Security Access service (0x27)
  - Allows elevated access to authenticated users

Case III - UDS Diagnostics and Patches over Patches

- Service 0x11 - ECUReset
- 90% of target ECUs, come with no authentication or pre-condition for hard ECUResets
- This means that any ECU which allows execution of this service, can be immediately interrupted by hard reseting it

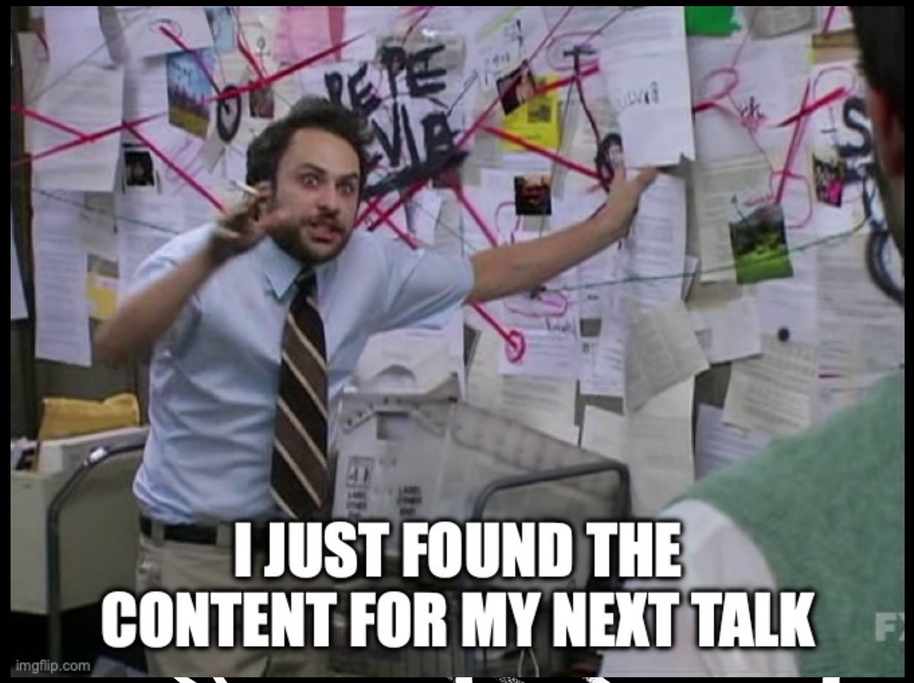


Case III - UDS Diagnostics and Patches over Patches


FIXED
VMware Fusion File Edit View Virtual Machine Window Help
Kali Linux
Sat 10:42
13:42
1 2 3 4
File Actions Edit View Help
cr0wtom@kali-m1: ~/Tools/caringcaribou/tool
(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
$ candump can0,705:7ff
Trash
File System
output

The Aftermath

- Responsibly disclosed finding in one of the biggest OEMs
  - Vulnerability was accepted
  - Directly affects the ADAS system of the vehicle
  - Applicable to 3 different models, and 4-5 years of MY
  - CVEs are yet to be assigned :'(



챕터 #3

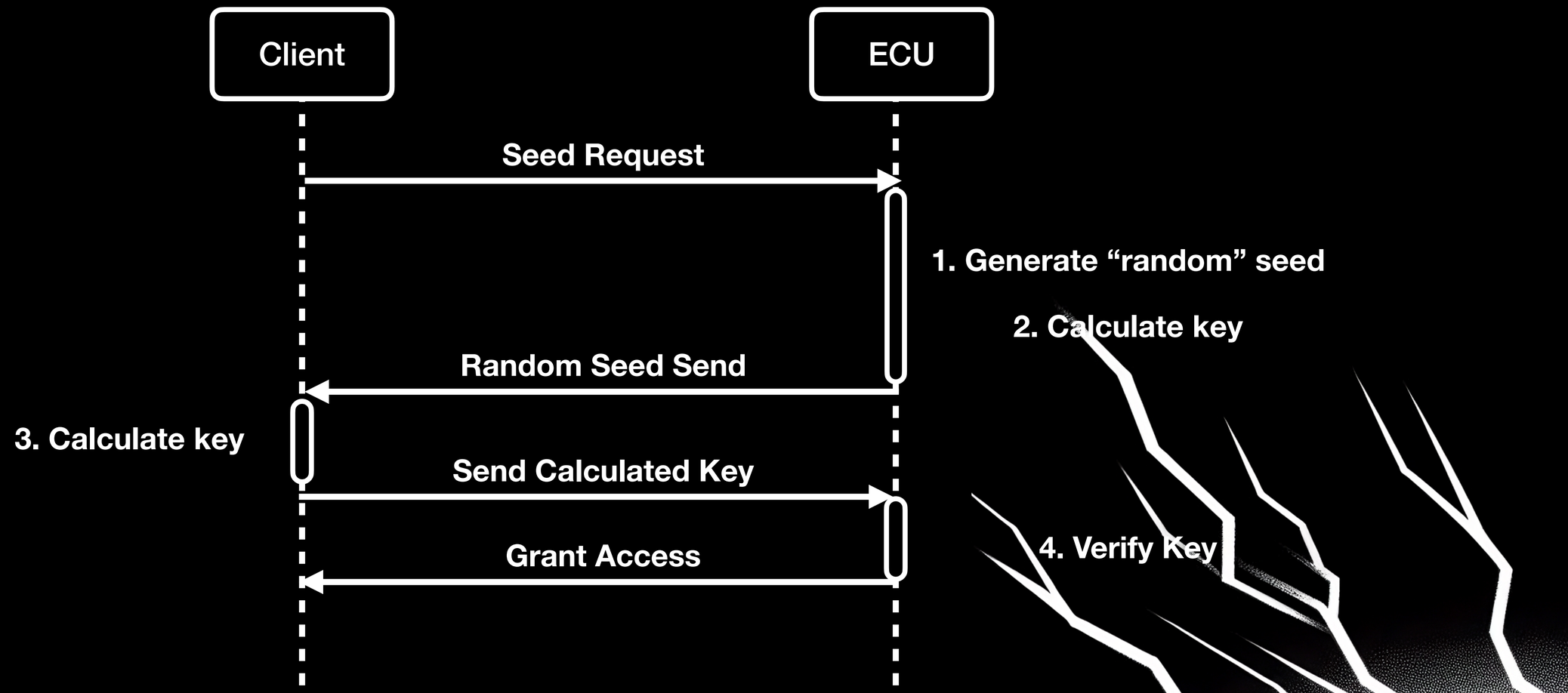
SEEDS

And the challenges of randomness
in an abstract world

Seeds

- The security access service, implements a challenge response mechanism, including:
  - A secret key
  - A secret calculation algorithm
  - A random seed
- Main use for access control purposes
  - Full ECU reprogramming can be considered the holly grail of a security access bypass

Security Access Mechanism

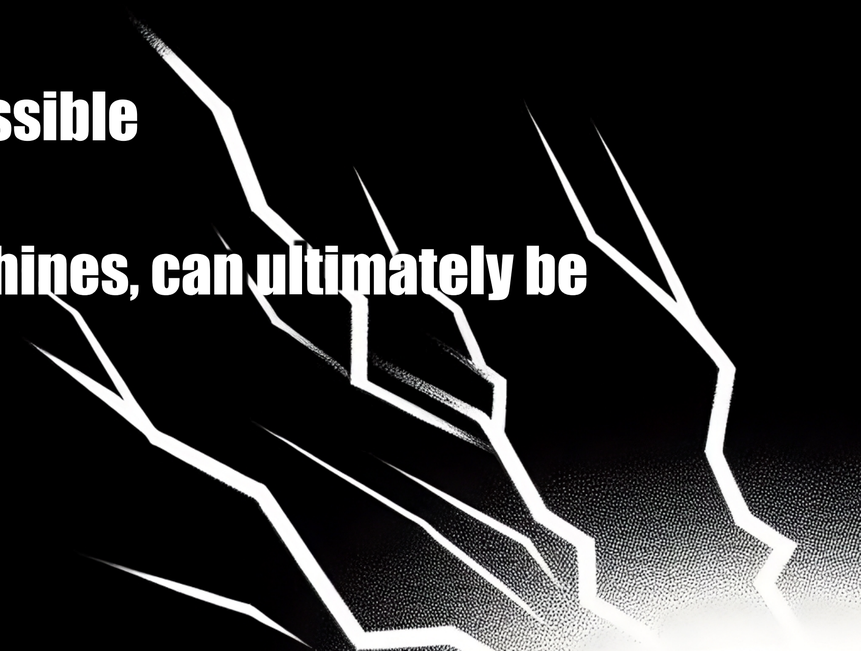


Seeds - Honest Questions

- What is the secret key?
- What is the secret calculation algorithm?
- What is the source of randomness?

"True" Randomness

- While true randomness "might" exist in nature, it's extremely difficult to achieve in deterministic machines
  - Computer systems are deterministic machines
  - Processing power needs to be used as efficiently as possible
  - Even if possible, due to the deterministic aspect of machines, can ultimately be controlled (in some cases)



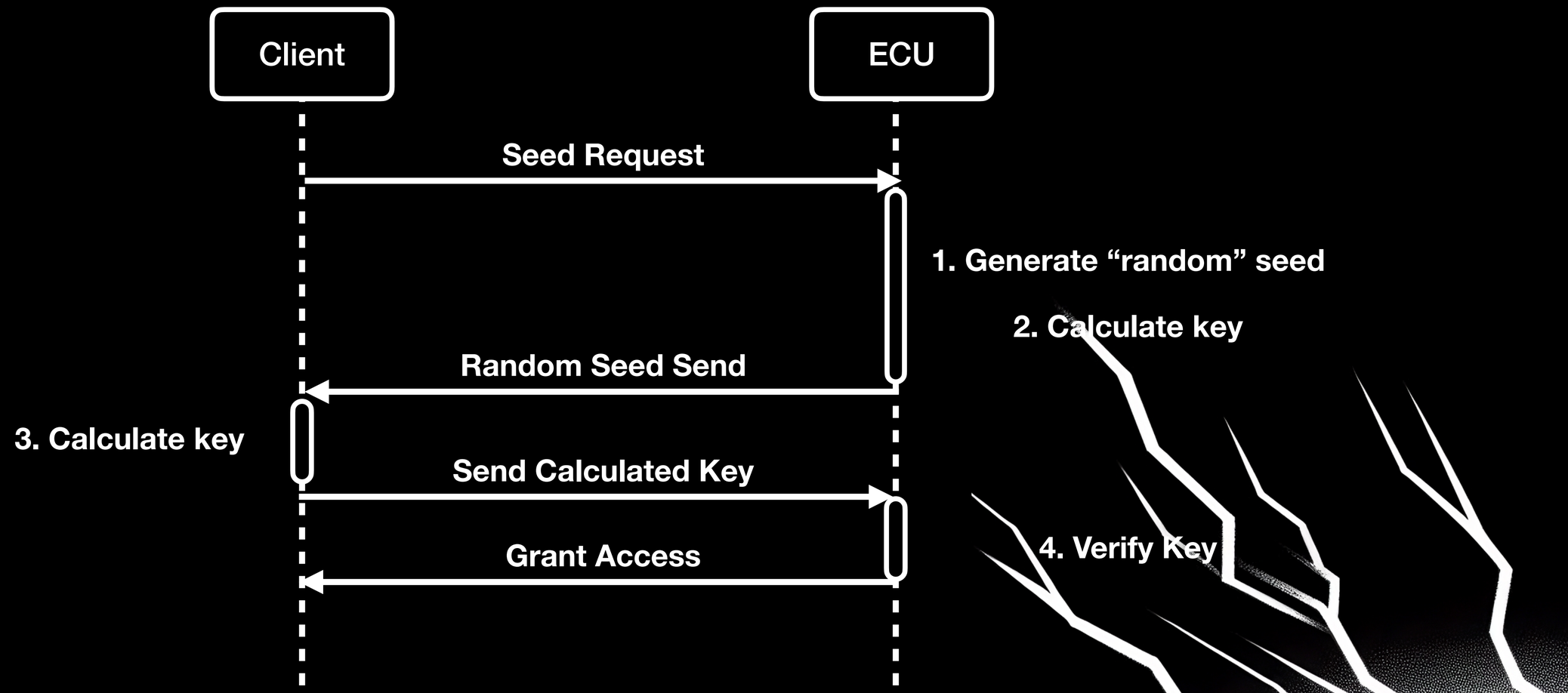
Pseudo-Randomness

- How to solve those issues?
- Pseudo-randomness can be achieved in a level good enough for our every day needs
- Protections have been implemented to avoid manipulation of the source of randomness in a certain/realistic degree

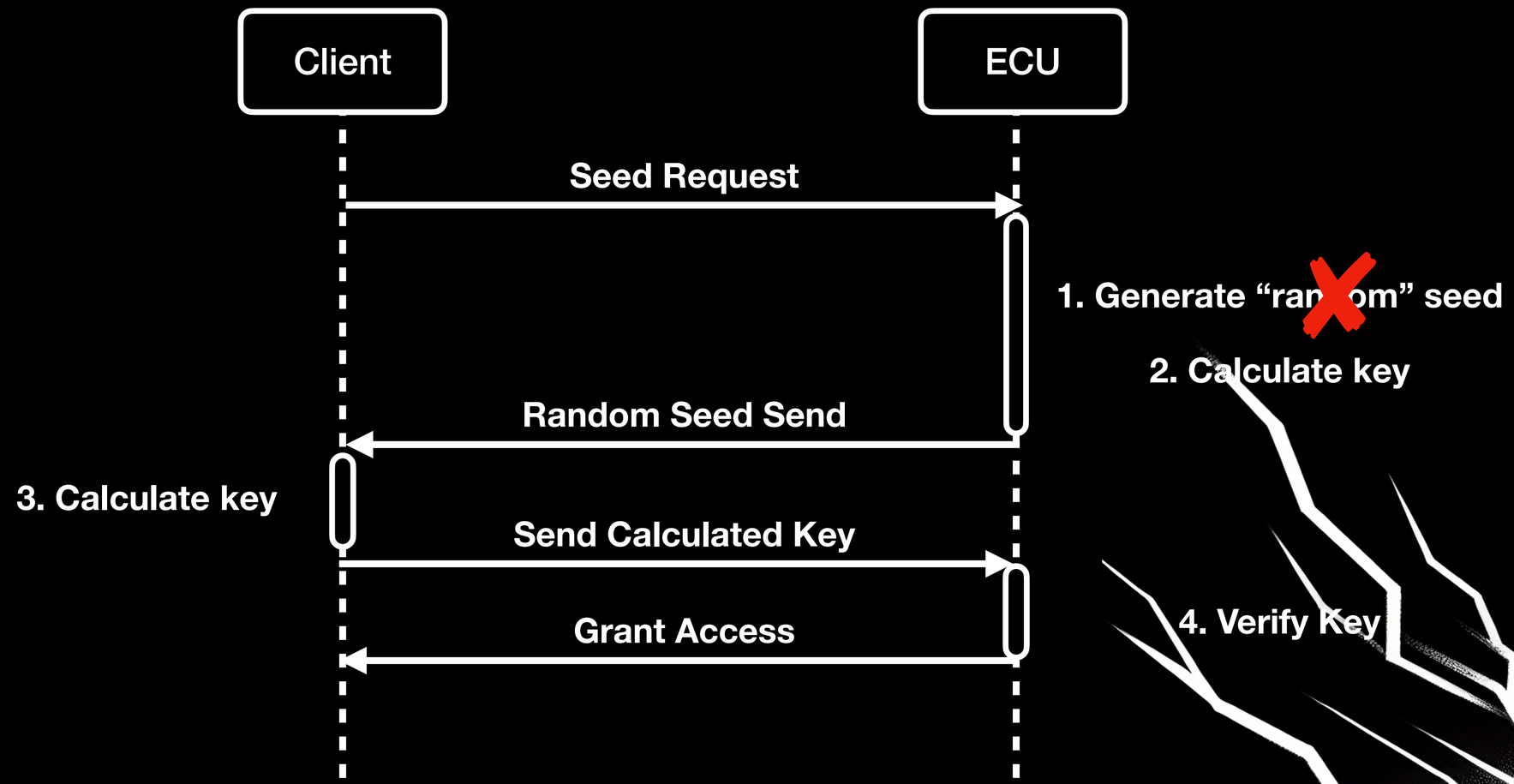
DEMO

```
Security Access Seeds captured:
d  fb  b
e  fc  b
f  fd  b
0  fd  b
1  fe  b
2  ff  b
c  00  b
b  01  c
a  02  b
9  03  b
8  04  b
7  05  b
7  06  5
6  07  5
5  08  6
4  09  5
3  0a  5
2  0b  6
1  0c  5
0  0d  6
f  0d  5
```

Security Access Mechanism



Security Access Mechanism



Sources of Randomness

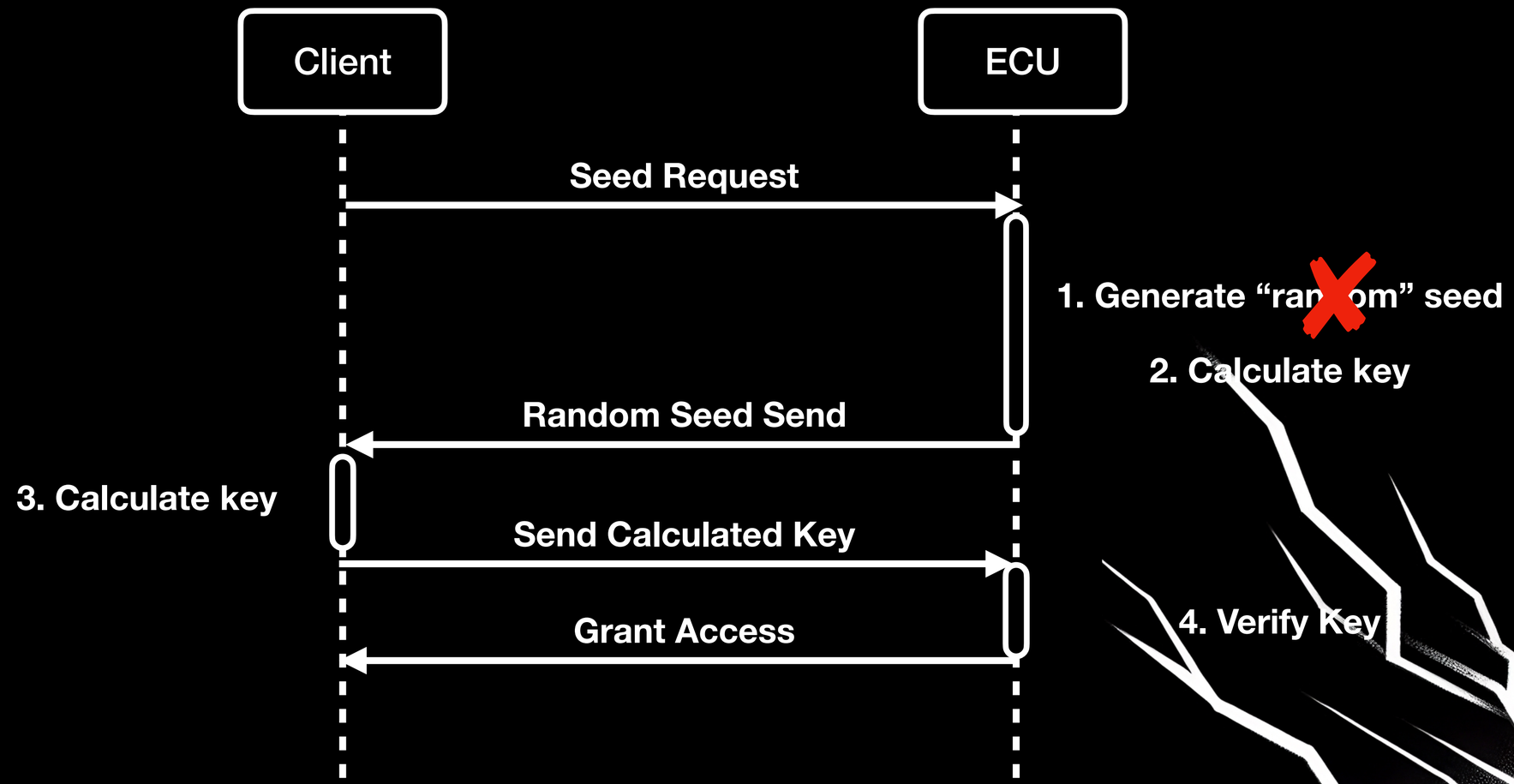
- Analog Noise
- CPU Temperature or Environmental Temperature
- Arbitrary "actions" of sensors

Sources of Randomness

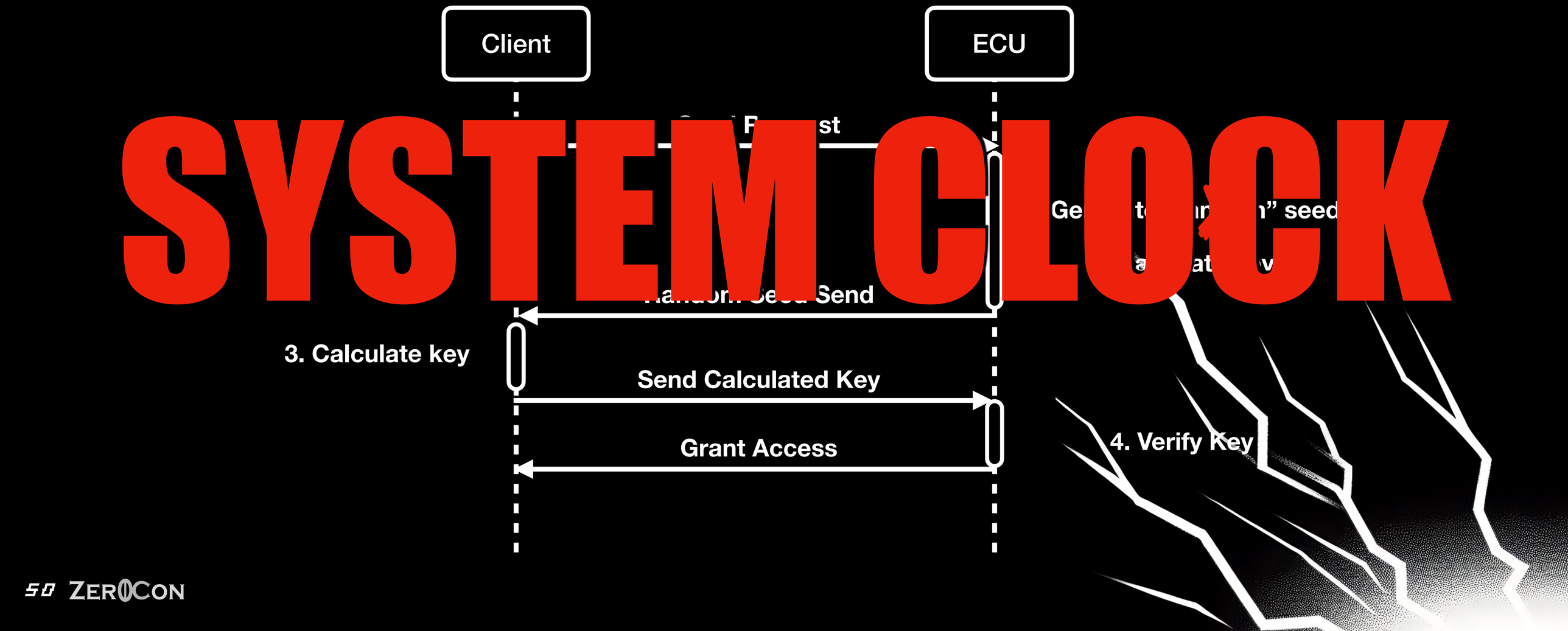
Decompile: FUN_000192f6 – (.bin)

```
1
2  void FUN_000192f6(void)
3
4  {
5    undefined4 uVar1;
6    ushort uVar2;
7    undefined2 uVar3;
8    char cVar4;
9    char cVar5;
10   byte bVar6;
11   int iVar7;
12   int iVar8;
13   uint uVar9;
14
15   cVar4 = DAT_d000df5f;
16   bVar6 = DAT_d000dfd8;
17   cVar5 = DAT_d000df60;
18   if ((byte)(cVar4 - 1U) < 3) {
19     iVar7 = STM_TIM0;
20     iVar8 = STM_CAP;
21     uVar9 = STM_CLC;
22     uVar9 = uVar9 >> 8 & 7;
23     uVar2 = DAT_d000df58;
24     DAT_d000df50 = 0;
25     uVar1 = Ramd0000fcc;
26     DAT_d000df5a = ~uVar2;
27     DAT_d000df54 = uVar1;
28     DAT_d000df40 = CONCAT44(iVar8 * uVar9,iVar7 * uVar9);
29   }
30   uVar3 = DAT_d000df58;
31   DAT_d000df60 = cVar5 + '\x01';
32   DAT_d000dfd8 = bVar6 + 1 & 7;
33   (&DAT_d000dfc8)[bVar6] = uVar3;
34   return;
35 }
```

Sources of Randomness



Sources of Randomness



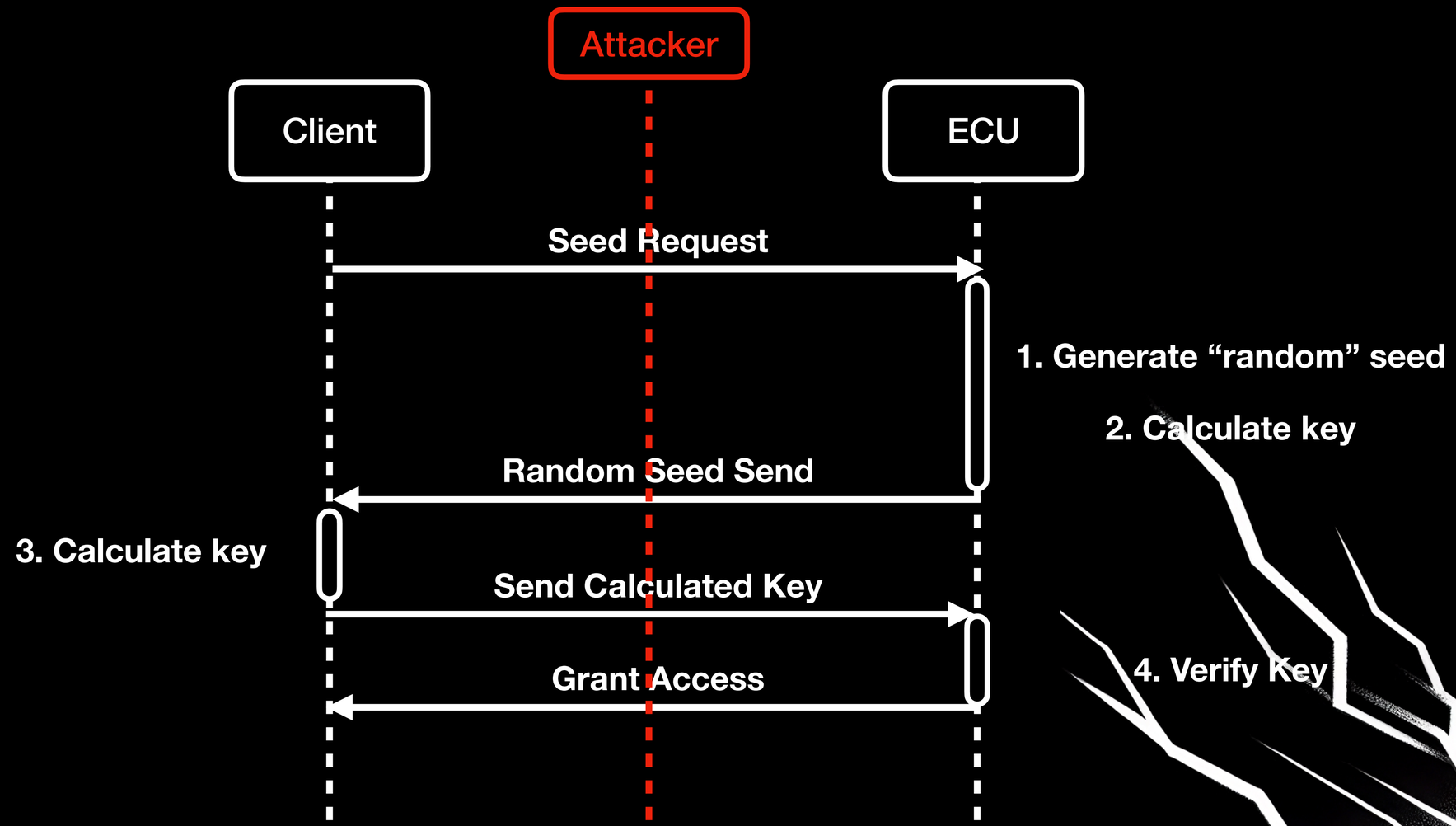
Case IV - Controlling that Seed

Remember CAN?

Case IV - Controlling that Seed

Remember that there is no encryption?

Case IV - Controlling that Seed



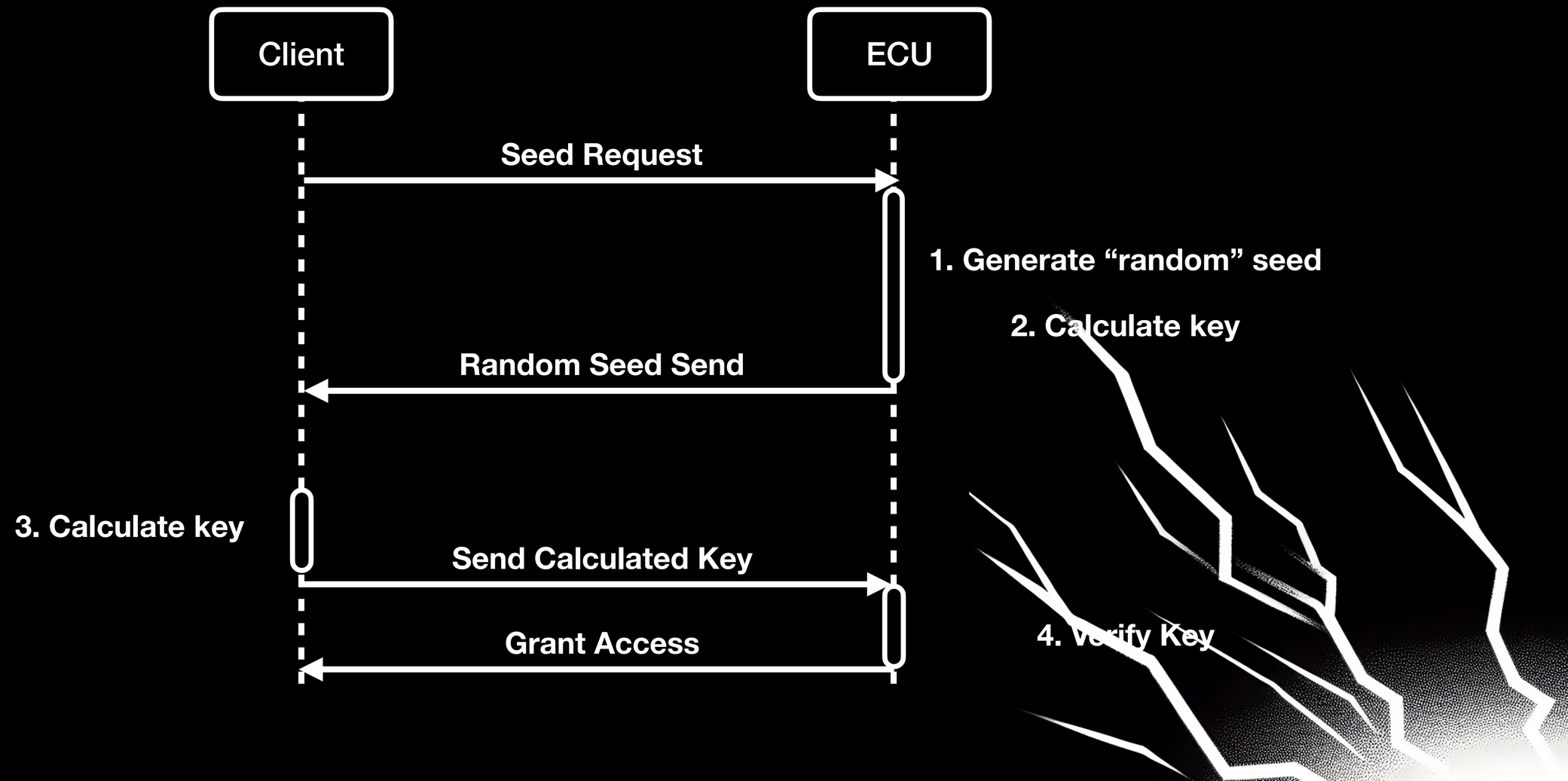
Case IV - Controlling that Seed

- As CAN is unencrypted, an attacker with a MITM position can effectively intercept a seed/key pair
  - OTA Updates
  - Service departments
  - Malicious Taps in between the lines
  - etc.
- Even if we have a seed/key pair, how do we bypass the rest of the restrictions that apply?

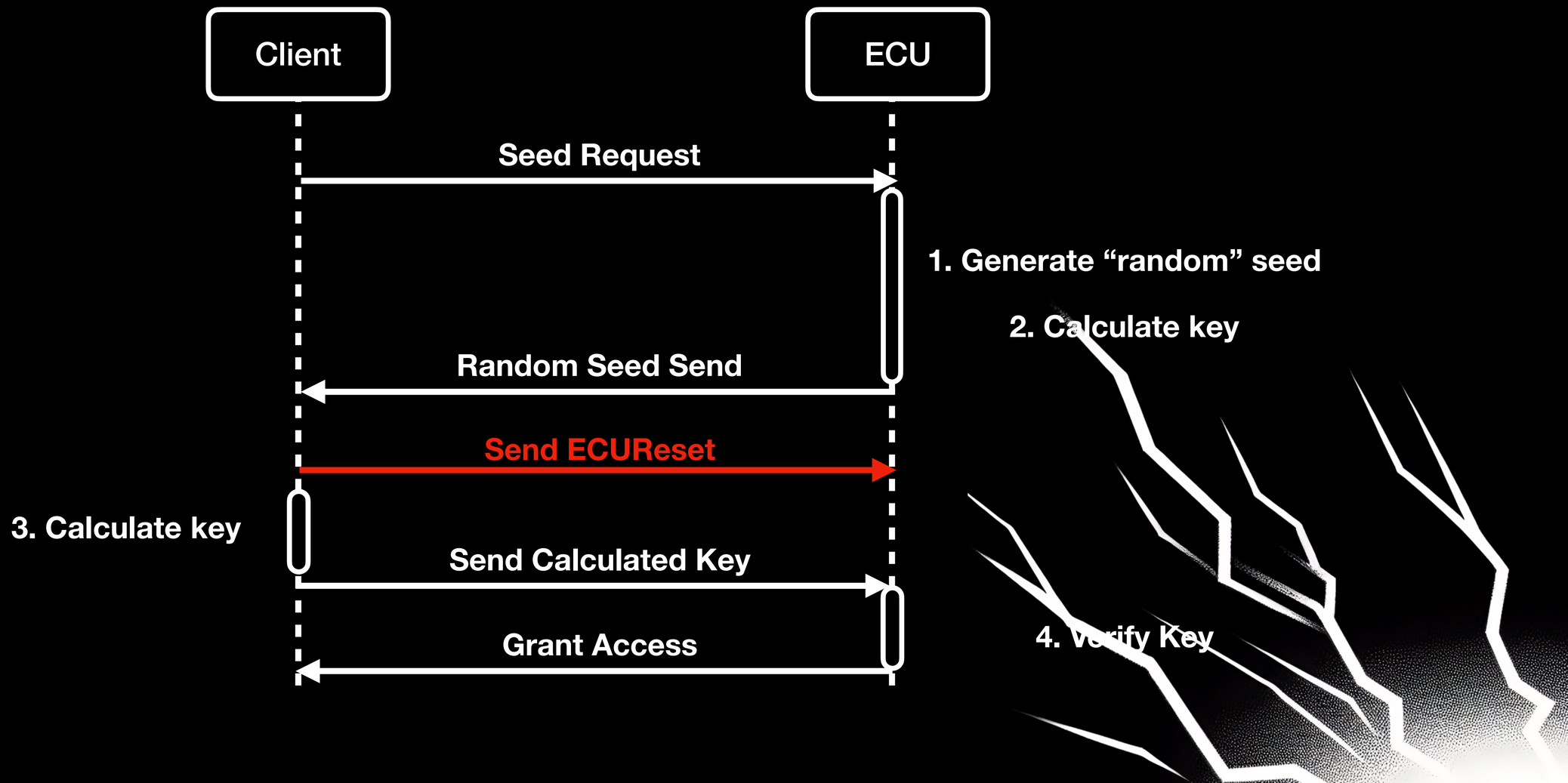
Case IV - Controlling that Seed

- What is the secret key?
- What is the secret calculation algorithm?
- What is the source of randomness?

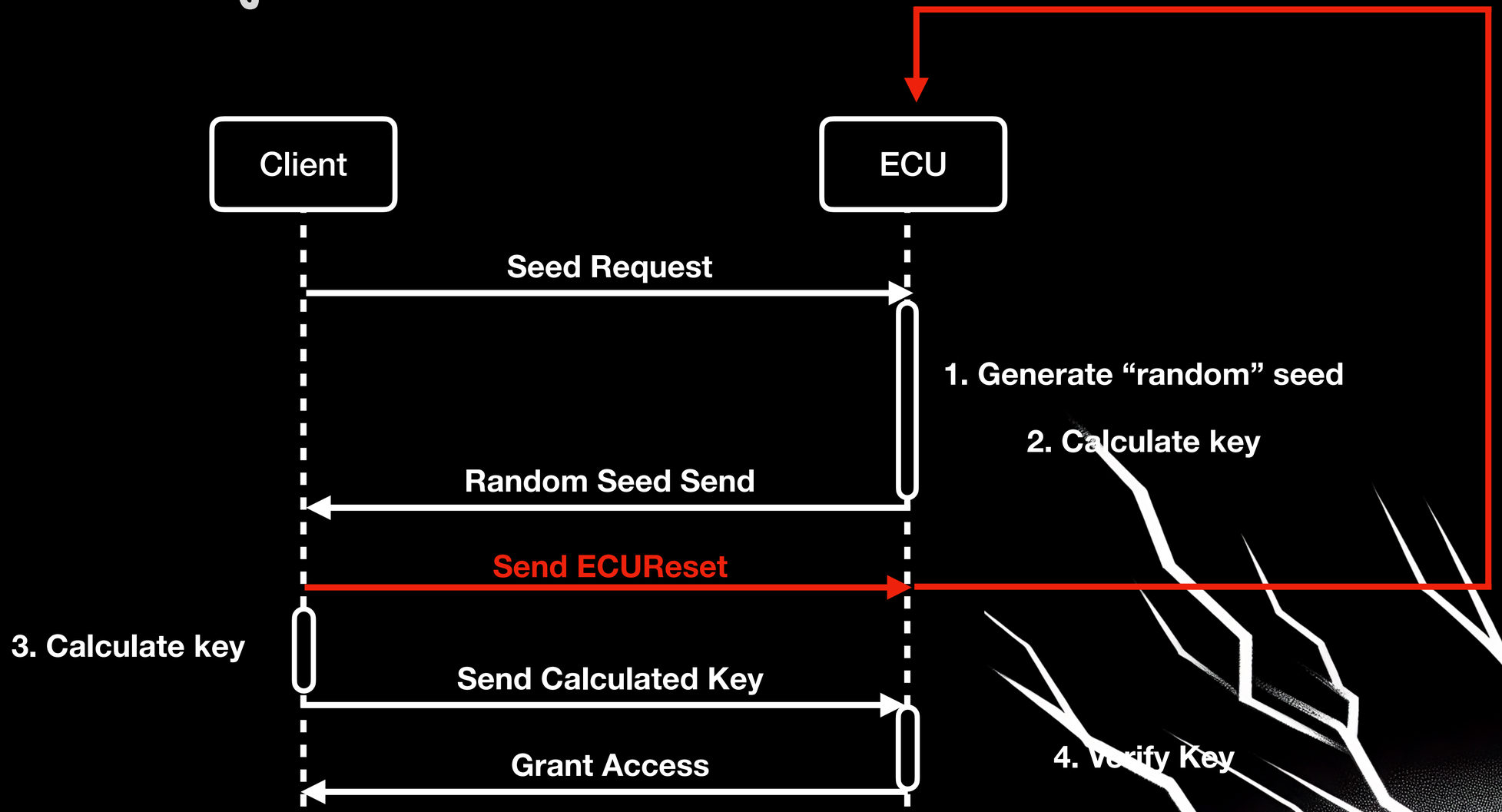
Case IV - Controlling that Seed



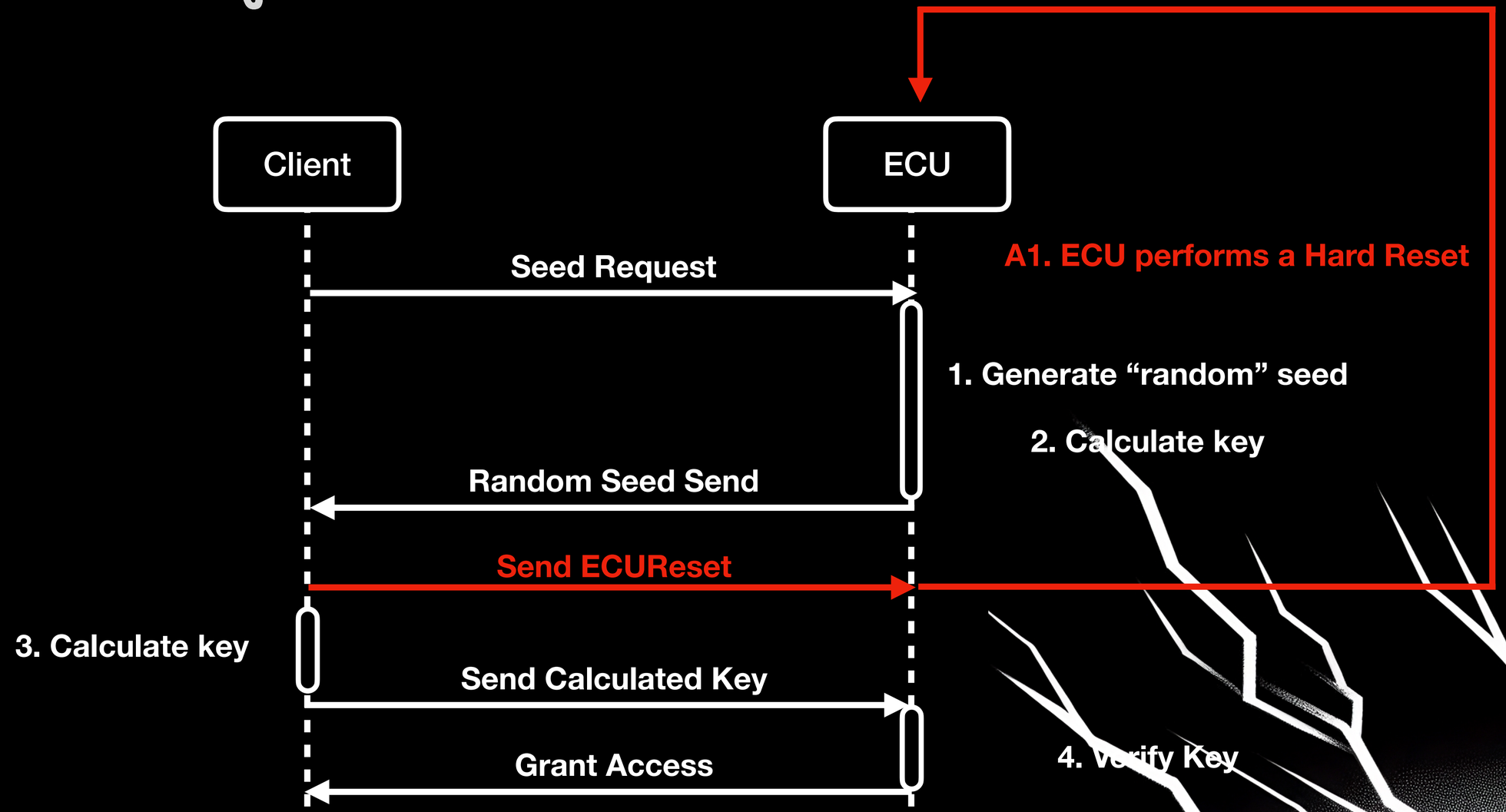
Case IV - Controlling that Seed



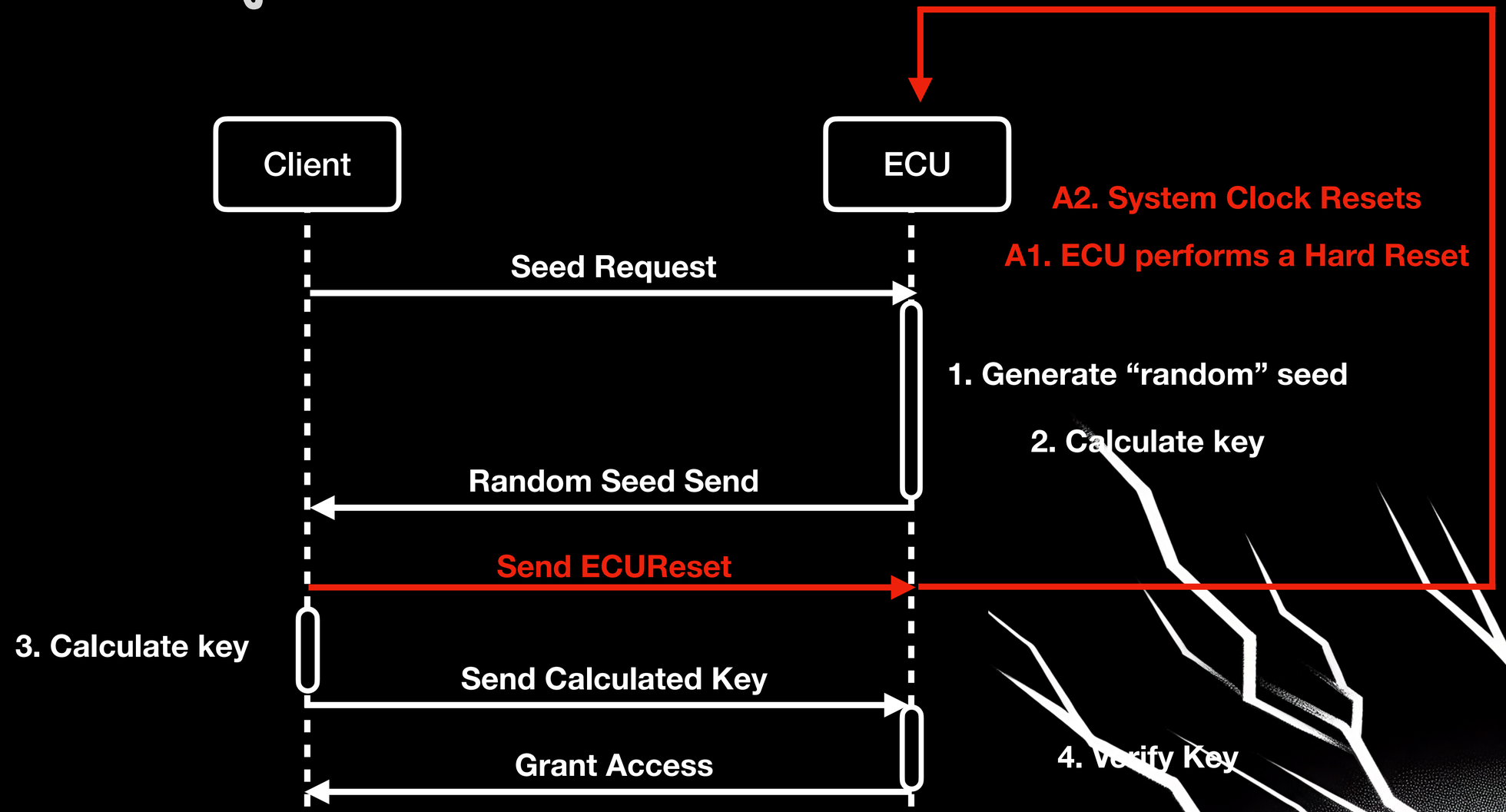
Case IV - Controlling that Seed



Case IV - Controlling that Seed



Case IV - Controlling that Seed



DEMO

Case IV - Controlling that Seed

- First seed capture demo, revealed that seed generation has an iterative algorithm
  - Assumption is that it's based on the system clock of the device
- A hard ECUReset from UDS, should push the unit to perform a full power cycle
  - As a result the system clock should reset to 0
- While our assumptions should result in seed control, we saw no duplicates in our second demo

Case IV - Controlling that Seed

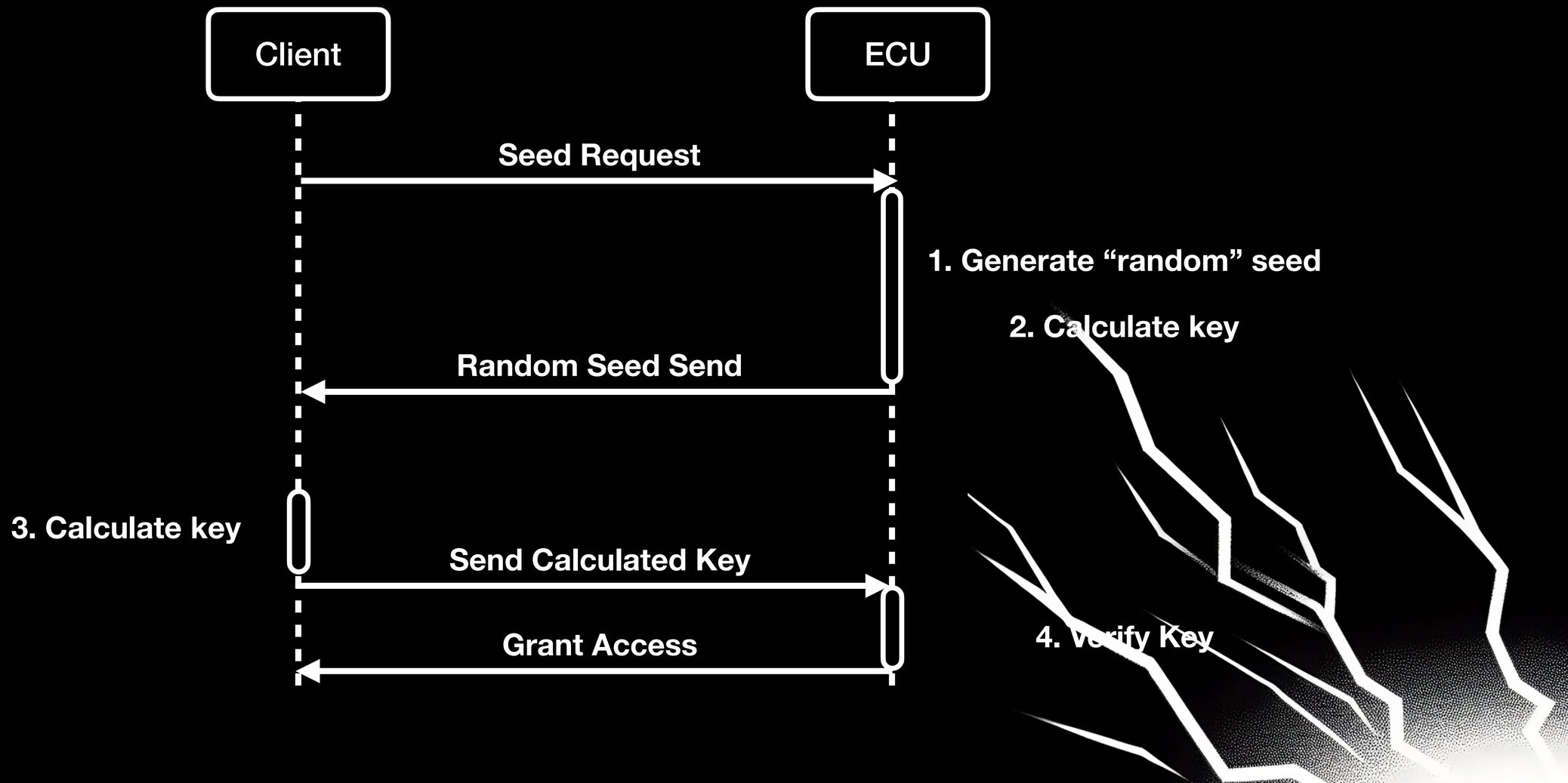
The answer is once again...

Case IV - Controlling that Seed

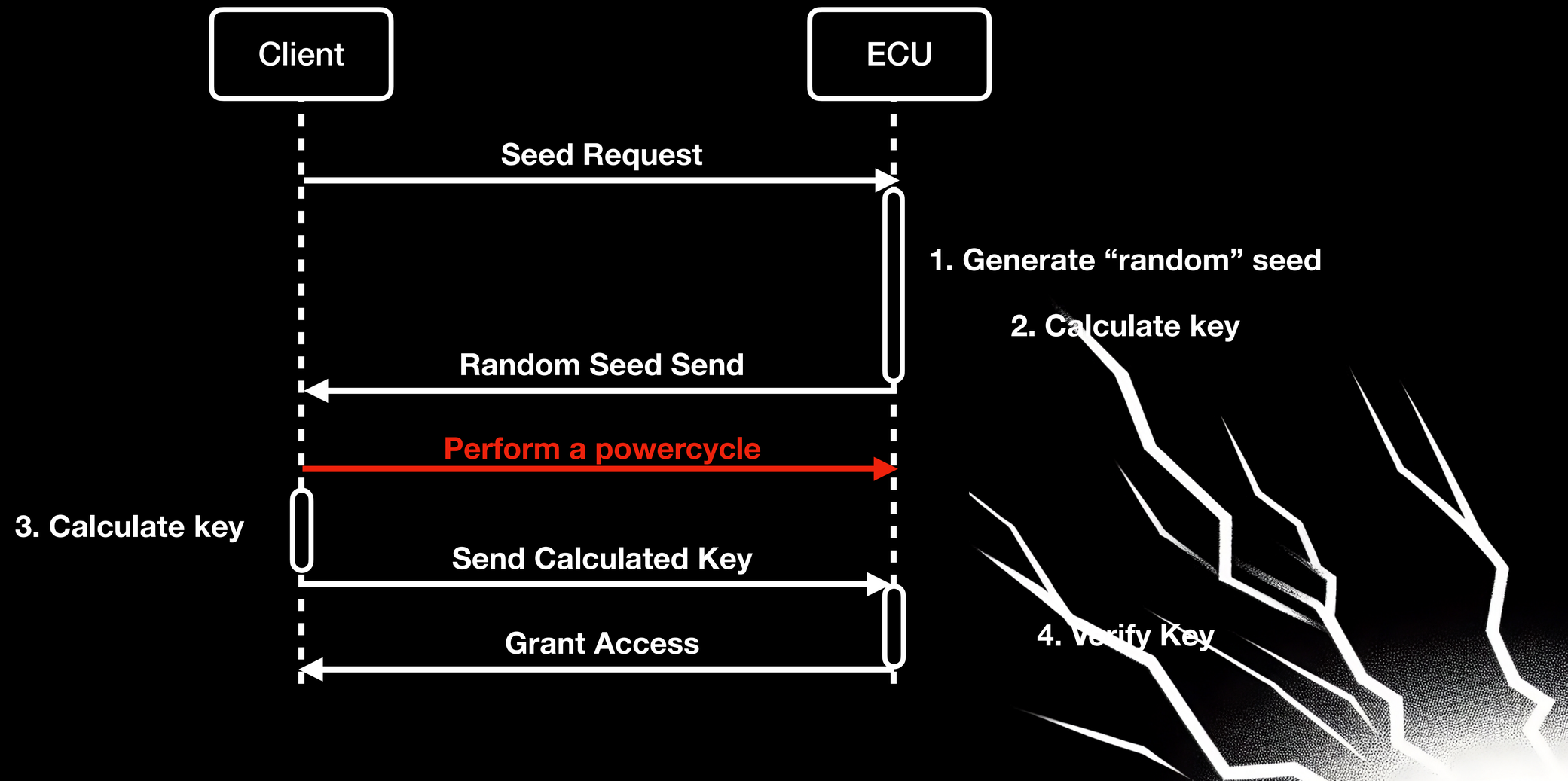
Case V - The Finale

- What if the ECU is restricting software powercycles for safety reasons?
- What if there were more ways to control a powercycle?
- What if we had a nice relay?

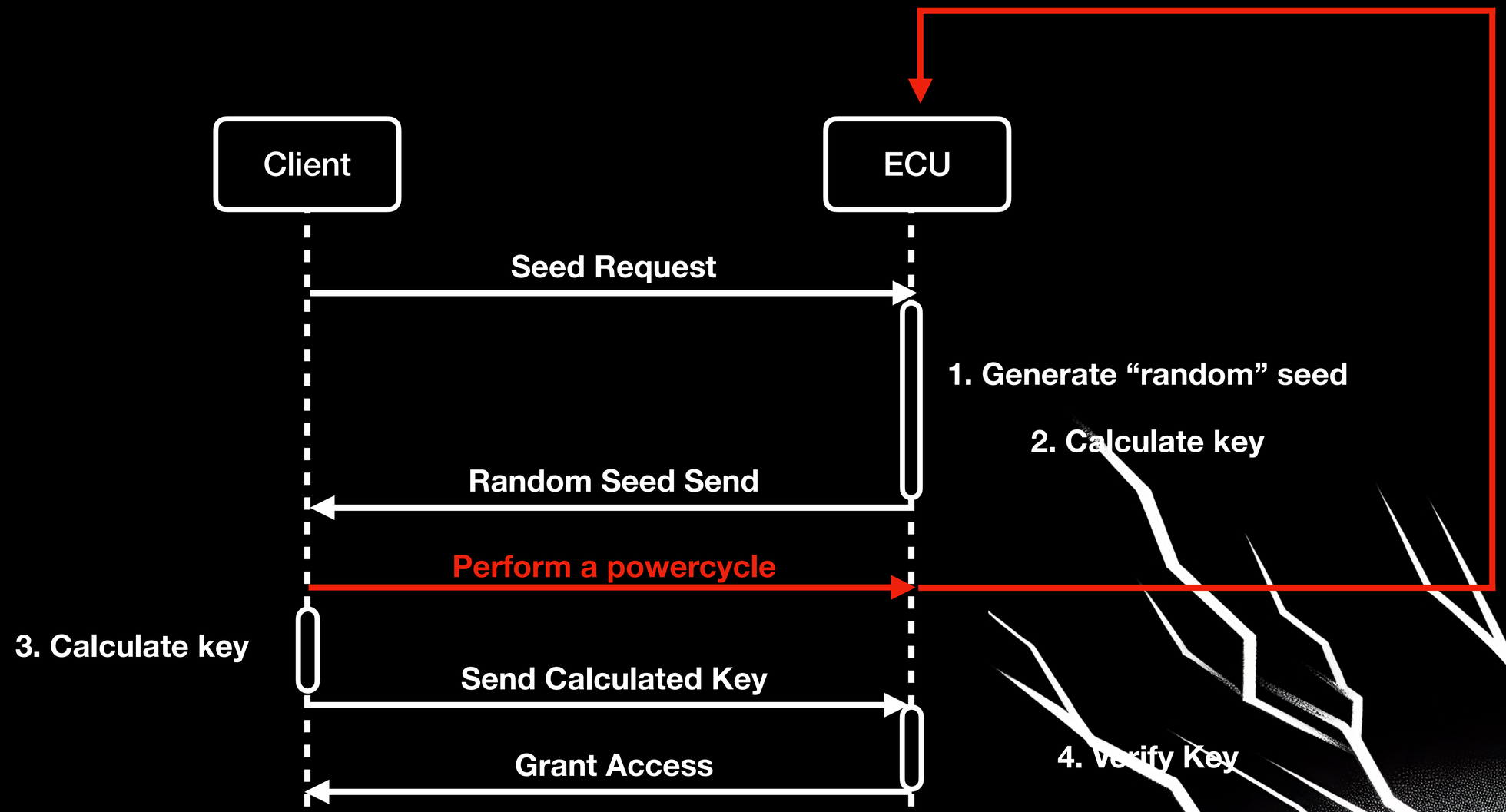
Case V - The Finale



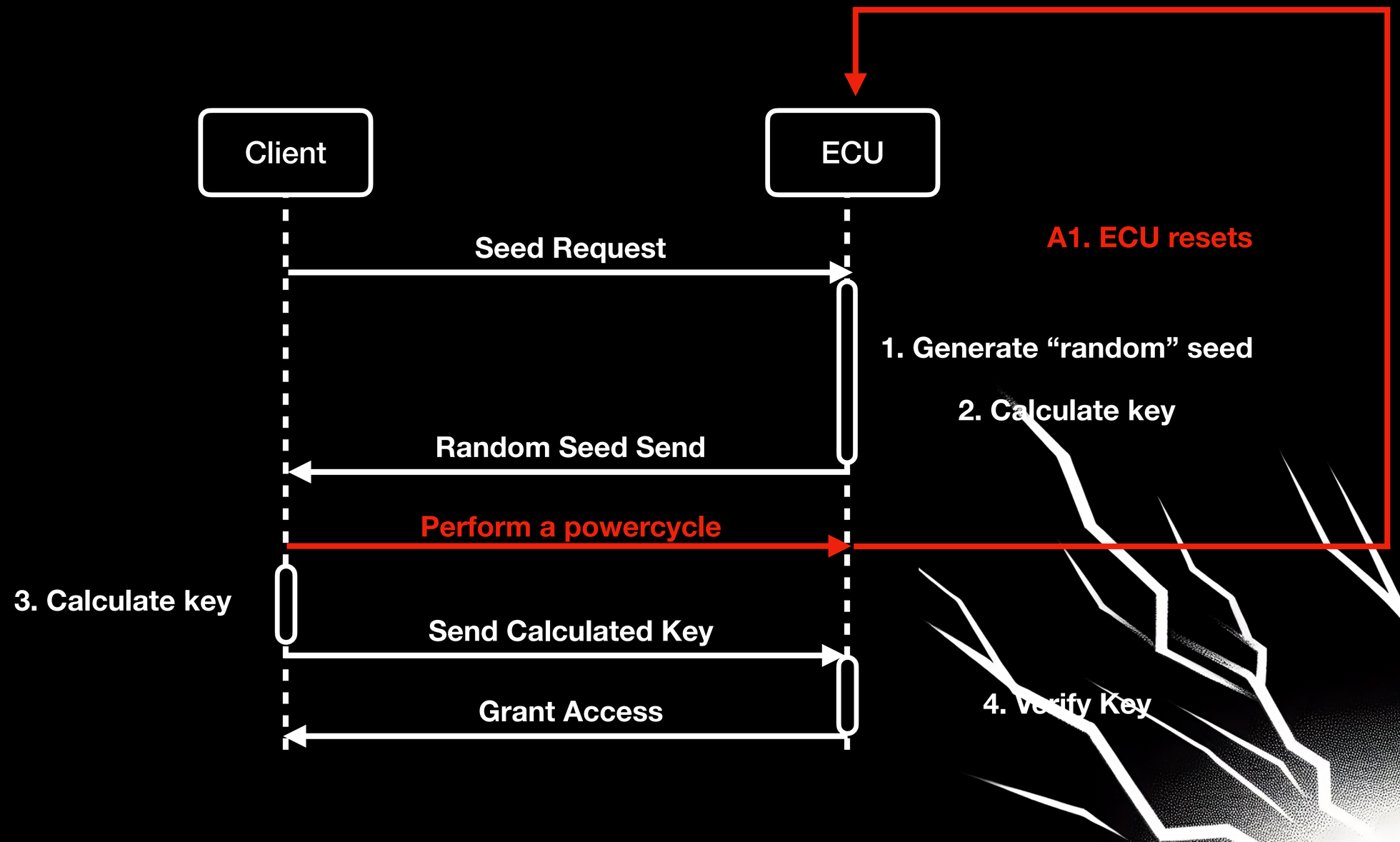
Case V - The Finale



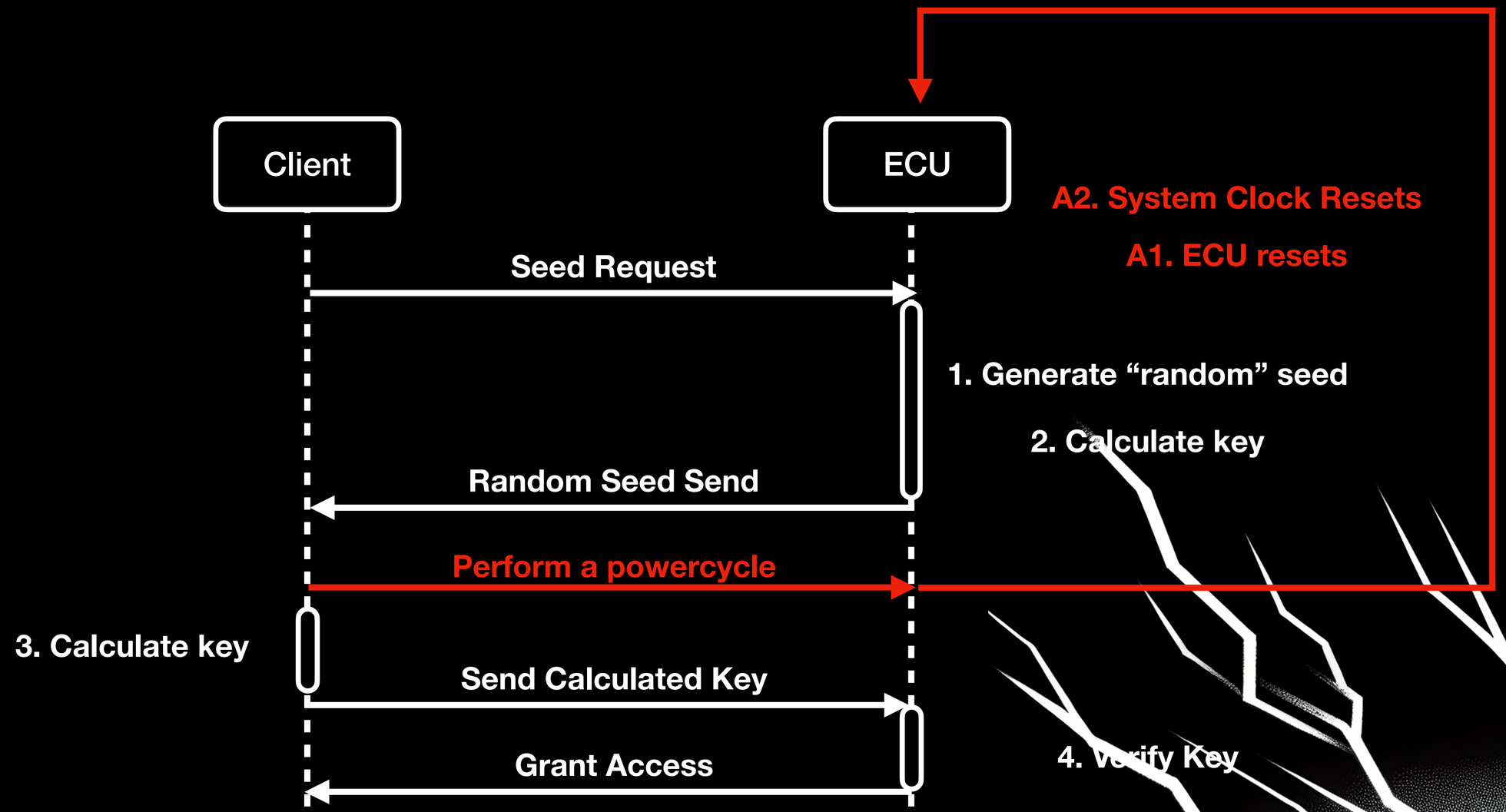
Case V - The Finale



Case V - The Finale



Case V - The Finale



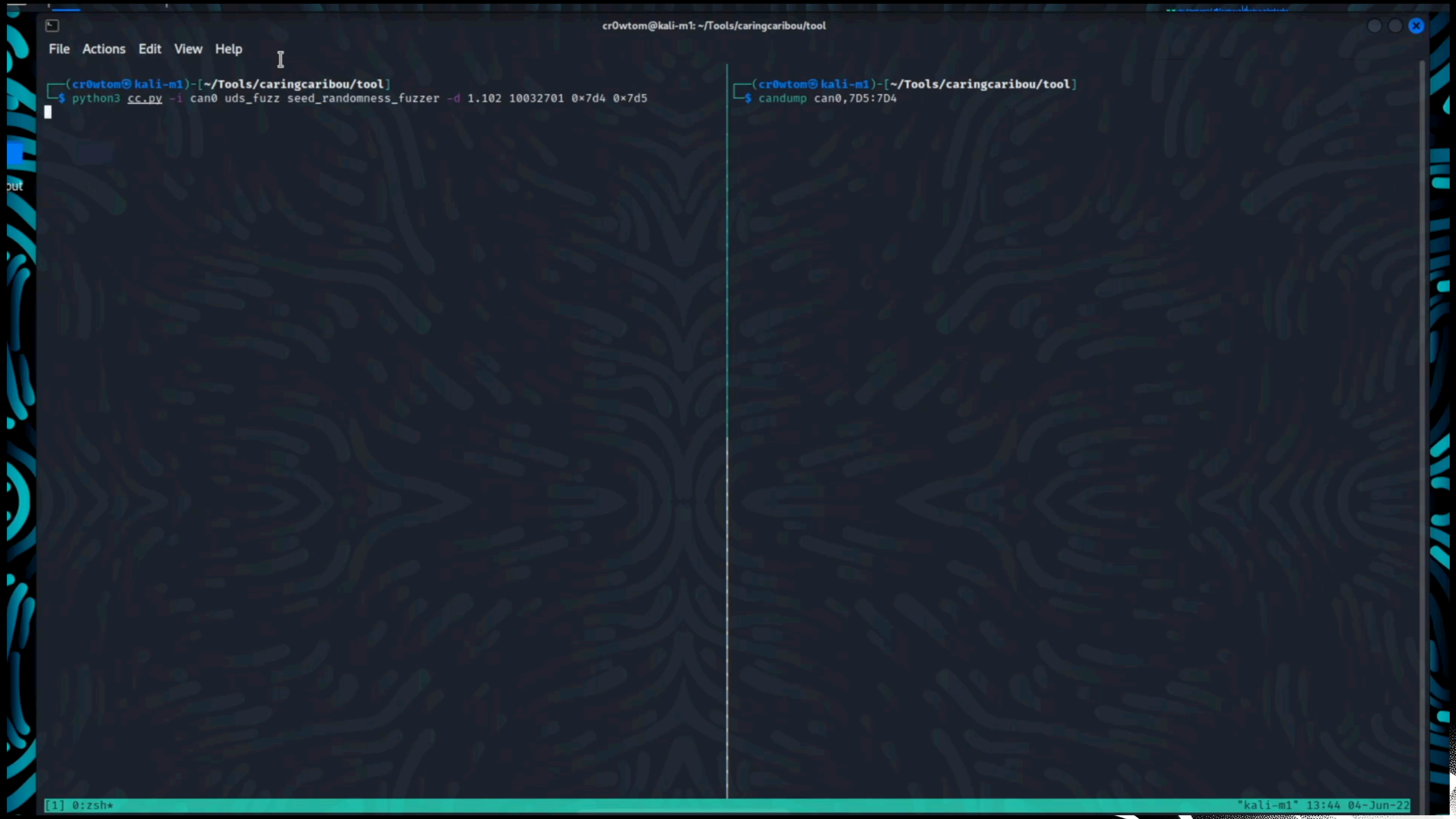
THE FINAL DEMO

And the only one you need

Case V - The Finale

- With this method we can effectively bypass several other mitigations:
  - Restricted ECUReset[1]
  - Restricted bootloader access[1]
  - Restricted debug access ports
- It is mainly applicable to test-bench setups, where we can effectively control the power supply of the ECU
- Can be applied to full vehicle, if it uses battery isolators
  - Mainly utilised in heavy duty vehicles

1.https://troopers.de/troopers22/agenda/tr22-993-uds-fuzzing-and-the-path-to-game-over


cr0wtom@kali-m1: ~/Tools/caringcaribou/tool
File Actions Edit View Help
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz seed_randomness_fuzzer -d 1.102 10032701 0x7d4 0x7d5
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ candump can0,7D5:7D4
[1] 0:zsh*
"kali-m1" 13:44 04-Jun-22

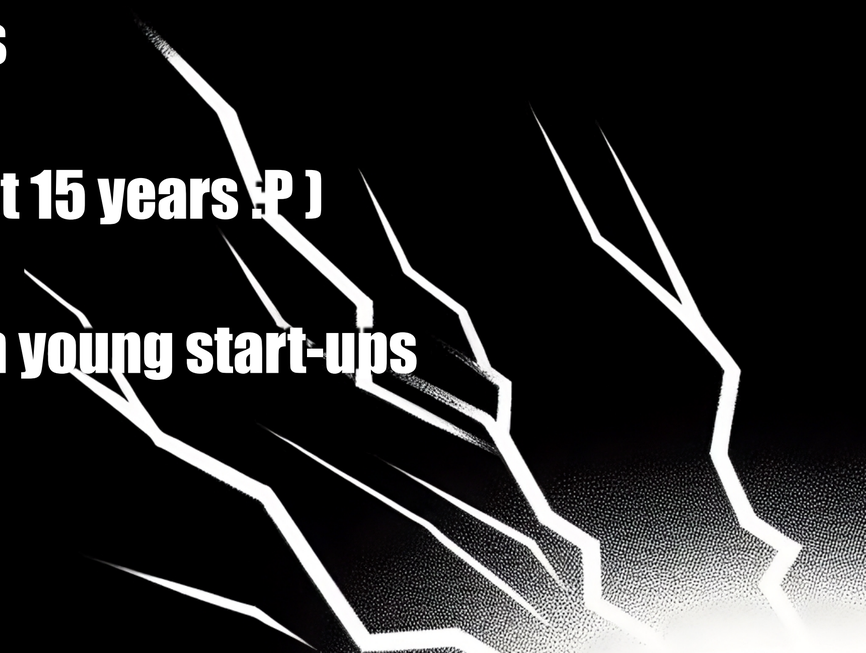
챕터 #4

EPILOGUE

And the struggle of keeping up with the digital world

The Disturbing Truth

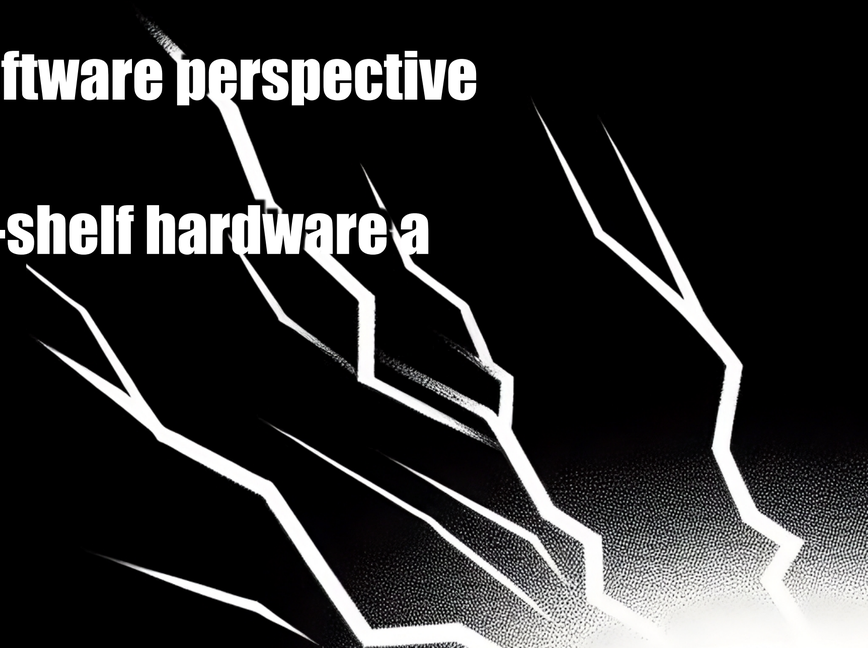
- Only some, out of several other flaws that we see daily, during our testing
- Decades old vulnerabilities and misconfigurations, applicable to modern vehicles
- Several "zero-days" and CVEs are currently "stuck" in VDPs
  - The automotive industry is not ready for this (for the last 15 years :P)
  - Usually, 100+ year old industries, trying to catch up with young start-ups



Do they even care?

Case VI - The Small OEM

- Surprising results from one of the smallest OEMs we worked with
- While testing, none of the "common" vulnerabilities were applicable
- Device was properly protected both from hardware and software perspective
- The underlying hardware was one of the cheapest, off-the-shelf hardware a developer can buy

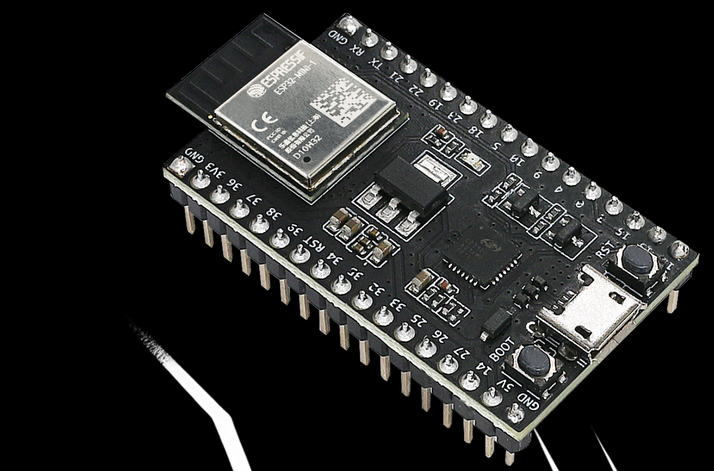


Their magic solution?

One of the cheapest, off-the-shelf hardware a developer can buy :P

Case VI - The Small OEM

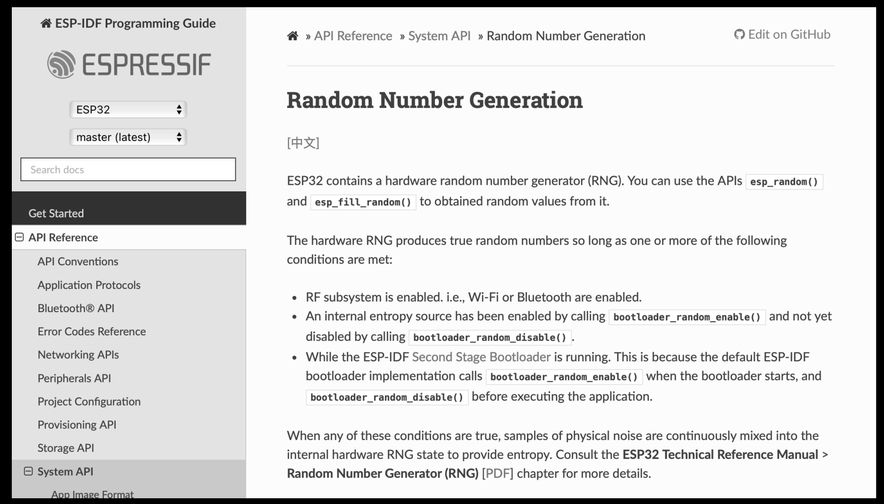
- **ESP32 controller**
  - **Low-cost, low-power system on a chip microcontroller with integrated Wi-Fi and Bluetooth**
  - **Not built with safety-critical functionality**
  - **Mainly targeting the "home automation" industry**



- OEM did not want to be mentioned.
- https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/random.html

Case VI - The Small OEM

- ESP32 controller
  - 10$ ESP32, includes internal RNG[1]
  - Uses random thermal noise, sourced from hardware ADCs[1]
  - Also uses the asynchronous clock mismatch[1]
  - Combines the inputs for a true random number[1]



1.https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf#rng

Case VI - The Small OEM

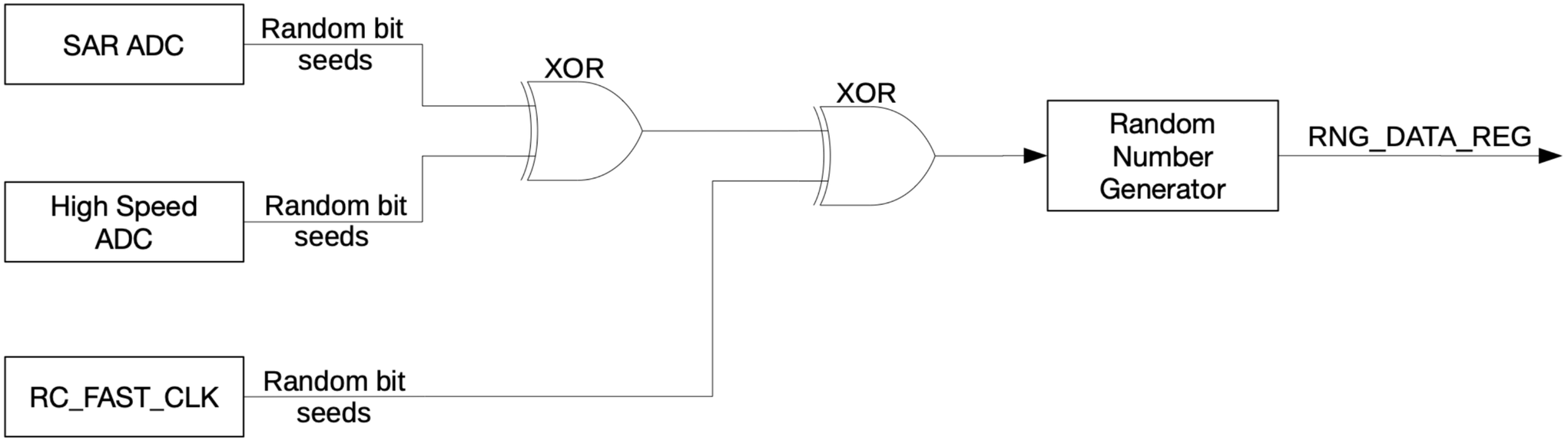


Figure 25-1. Noise Source

• https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf#rng

So, problem solved?

Case VI - The Small OEM

- As said, randomness is not perfect in deterministic machines
- Several steps can be taken to isolate and control the sources of randomness
  - Out of scope of this talk
- Still better than the single, vulnerable source used in expensive safety critical components (especially the system clock)

```
* @warning This function is not safe to use if any other subsystem is accessing the RF subsystem or
*          the ADC at the same time!
```


• https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf#rng

Is there light in the end of the tunnel?

THE END

Thanks to my team:

Phuc Trinh

Diego Ledda (@jbx81)

Martin Pozdena (@pozdemar)

Andre Maia

Ekhi Lopez (@ ekhilopez)

Jarno Pomstra

Pavel Khunt (@ khuntpav)

Francisco Garcia (@frangarcia_16)

Timothee Marion

Teodor Danu

Adam Nastulczyk

Martin Havelka (@mhavelka77)

Wiktor Sedkowski

경청해 주셨어 감사합니다

ZERO CON

Thomas Sermpinis
@CrOwTom | https://crOwsplace.com