Uprooting Trust: Learnings from an Unpatchable Hardware Root-of-Trust Vulnerability in Siemens S7-1500 PLCs

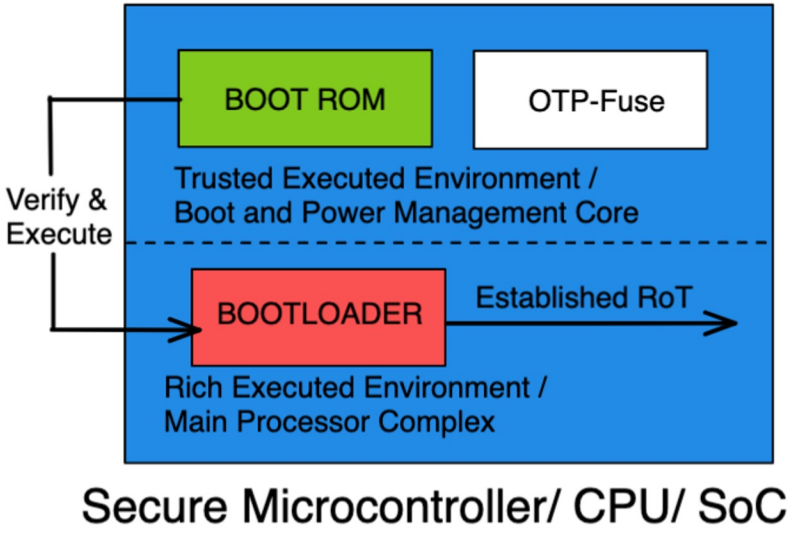
Yuanzhe Wu, Dr. Grant Skipper and Dr. Ang Cui
Red Balloon Security
IEEE International Symposium on Hardware Oriented Security and Trust (HOST) 2023

Hardware Root-of-Trust

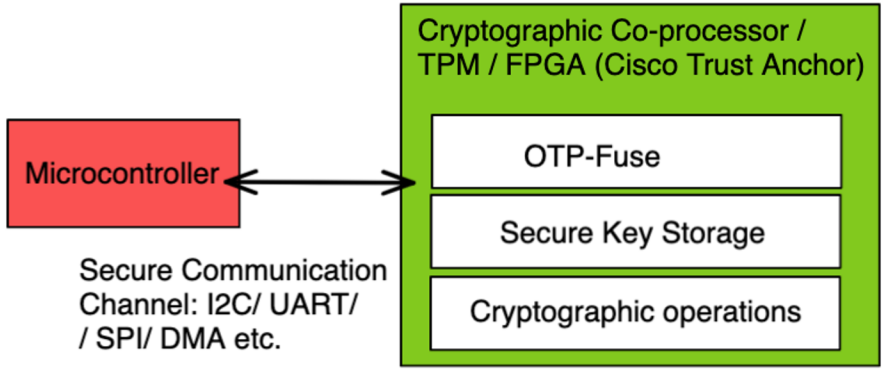
- What is Hardware Root-of-Trust (RoT)[1] for embedded systems?
  - A hardware RoT is a set of functions in a trusted computing module that securely stores cryptographic keys and performs secure operations
  - Secure cryptographic keys storage, cryptographic operations and anti-tampering
  - Provides a foundation for secure boot, secure firmware updates, and authentication
  - Ensures the integrity and security of critical systems

- Challenges in implementing discrete RoT components
  - Integration with the overall system architecture (With legacy systems that require backward compatibility)
  - Ensuring secure communication channels (Physically and protocol vise)
  - Protecting critical materials from exposure (Dedicated Crypto-coprocessor / TPM chip / On die cryptostorage and crypto accelerator)

Hardware RoT examples



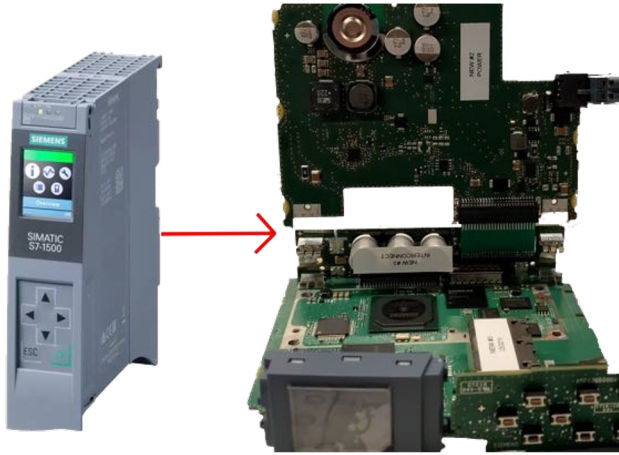
Hardware RoT examples



Dedicated Hardware RoT solutions

Background on Siemens S7-1500 PLCs

- Adoption in critical infrastructures (Critical Systems!)
  - Energy, Water, Transportation, Oil and gas: Nuclear facility
  - Manufactory and Building automation
- Stuxnet Target (early models) [2]
  - Discovered in June 2010
  - Targeted Siemens S7-300 and S7-400 PLCs
  - Altered programmable logic controller (PLC) code to cause physical damage to centrifuges used in uranium enrichment
  - First known malware to cause real-world physical damage to critical infrastructure
- Siemens PLC 31% market share
- Next generation S7-1500 PLCs
  - Hardware RoT implementation detail was NOT known by public



How Secure is Siemens S7-1500?

- Objective:
  - Determine how the S7-1500 PLC is protecting itself from adversarial activity.
- Challenges:
  - Encrypted Firmware!
  - No Debug Access (JTAG[3], Serial[4], etc)
  - **Opaque Boot Process**

Challenge

Limited visibility - need to get creative to understand how RoT is protecting these embedded platforms.

Challenge

Limited visibility - need to get creative to understand how RoT is protecting these embedded platforms.

Hypothesis: Critical information and Firmware decryption key material may exist in volatile memory during the boot process and grant insight into the protection.

Challenge

Limited visibility - need to get creative to understand how RoT is protecting these embedded platforms.

Hypothesis: Critical information and Firmware decryption key material may exist in volatile memory during the boot process and grant insight into the protection.

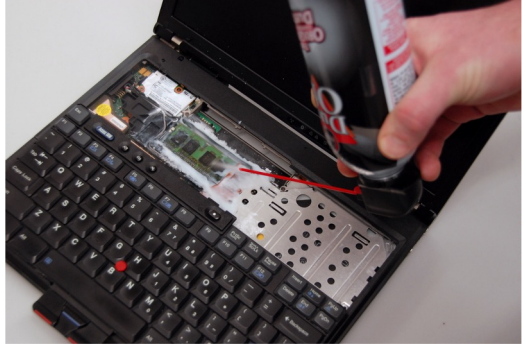
Cryo-mechanical memory extraction

- Data remanence property of DRAM
- Leak secret authentication material in device memory

“Cold boot” => Cryomechanical Memory Extraction

Cold Booting is a reverse-engineering activity that leverages the memory-resonance effect to pin certain bits in memory only during a boot or restart.

Traditional Cold-Boots attacks rely on physical peripherals (DIMM) or Debug access (JTAG).



Traditional Cold boot Attack

“Cold boot” => Cryomechanical Memory Extraction

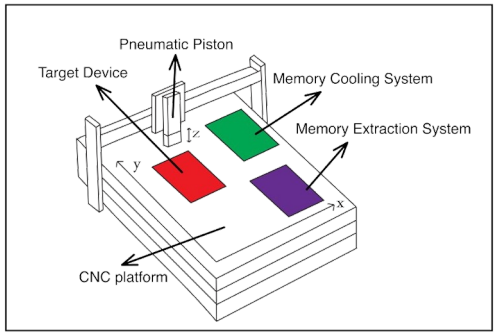
Traditional Cold boot Attack

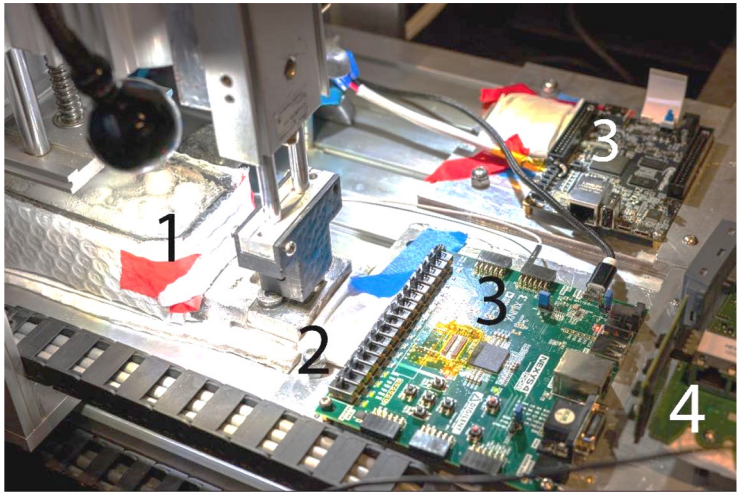
Cold Booting is a reverse-engineering activity that leverages the memory-resonance effect to pin certain bits in memory only during a boot or restart.

Traditional Cold-Boots attacks rely on physical peripherals (DIMM) or Debug access (JTAG).

Engineered a novel CNC-based platform which transferred the physical dedicated RAM integrated circuit to transfer the chip between the target platform, cooling system, and a memory readout device.

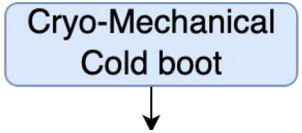
Note: Detail of this work will be provided in a separate paper to appear at WOOT’23 [5]



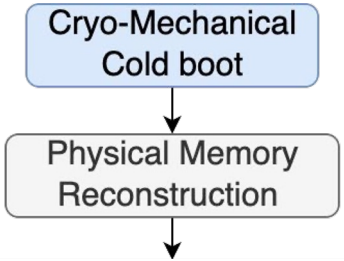


CNC-based Cryo-mechanical memory extraction platform

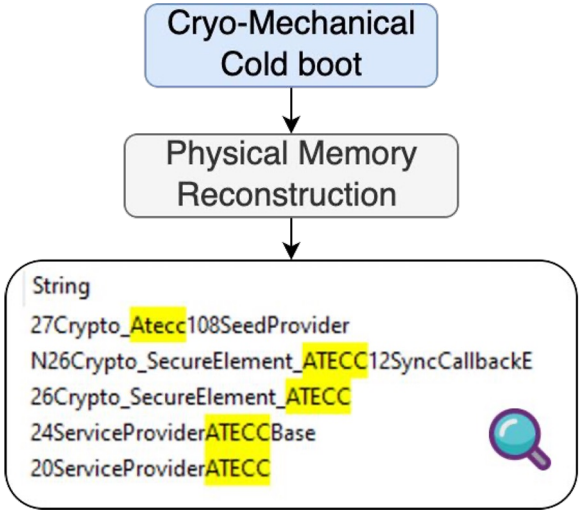
Revealing Trust



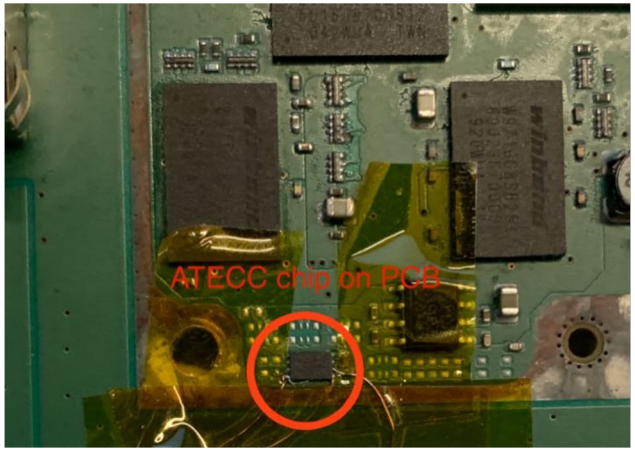
Revealing Trust



Revealing Trust

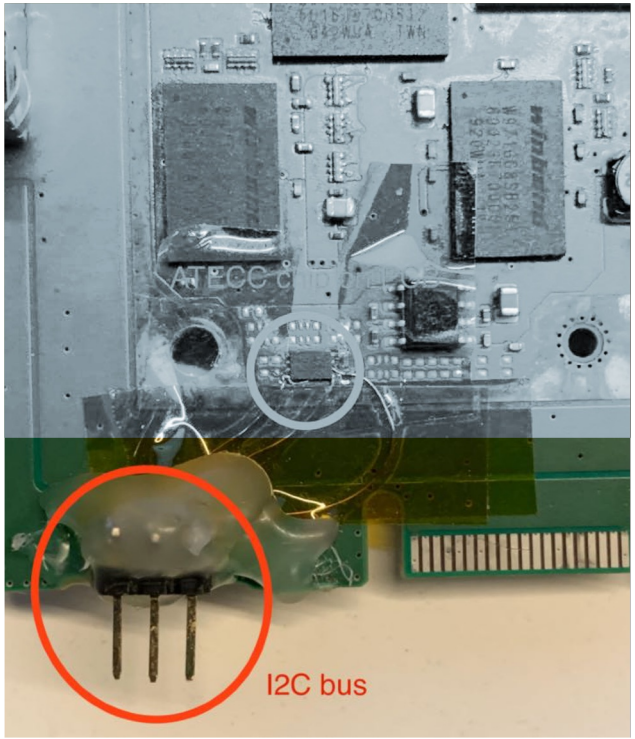


Revealing Trust



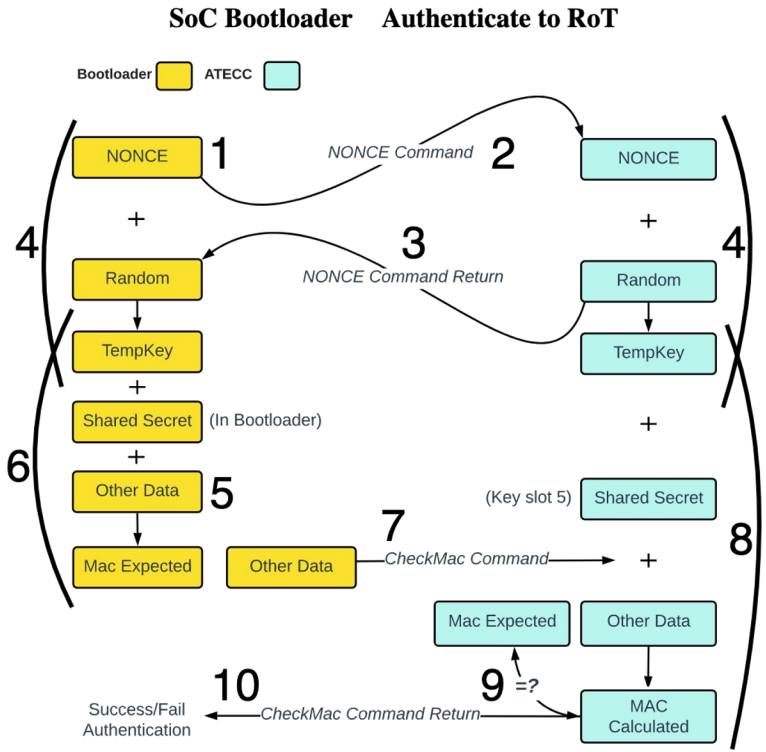
Easily accessible I2C bus for ATECC Crypto Co-Processor

Revealing Trust

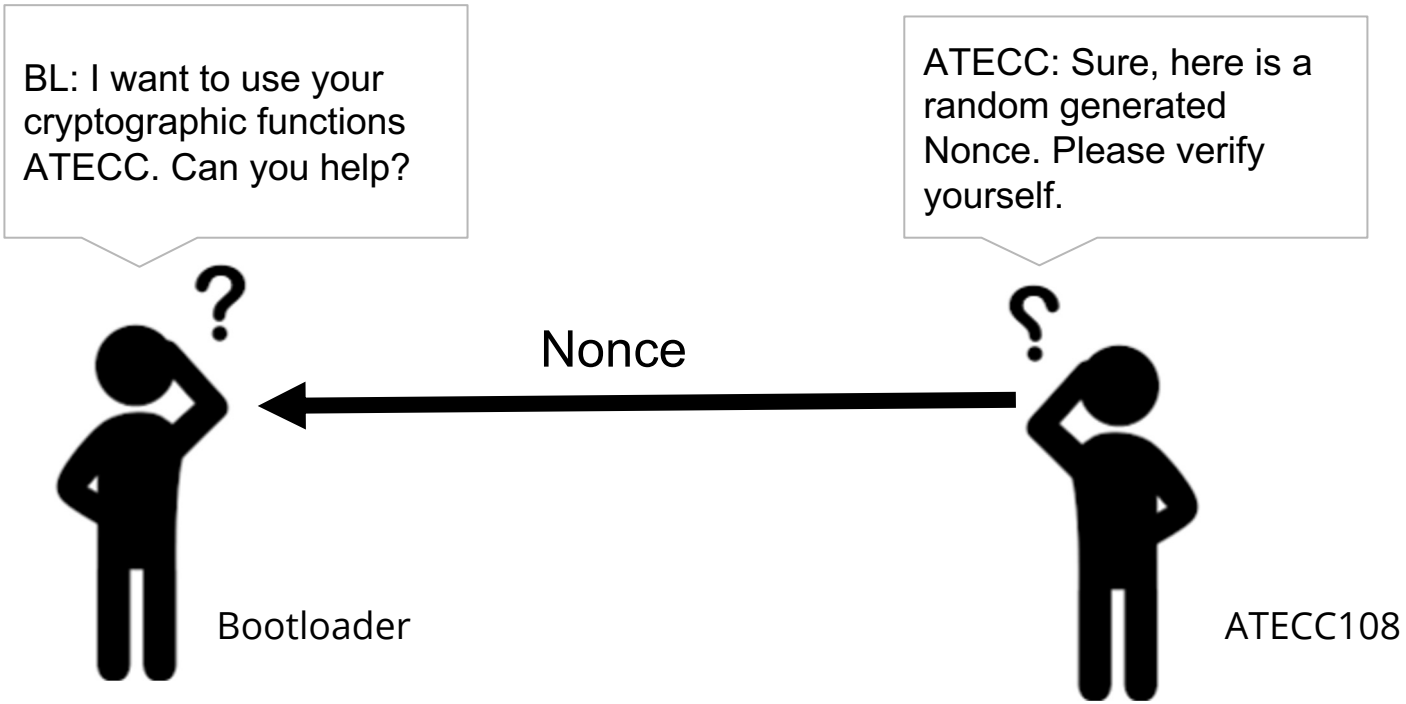


Easily accessible I2C bus for ATECC Crypto Co-Processor

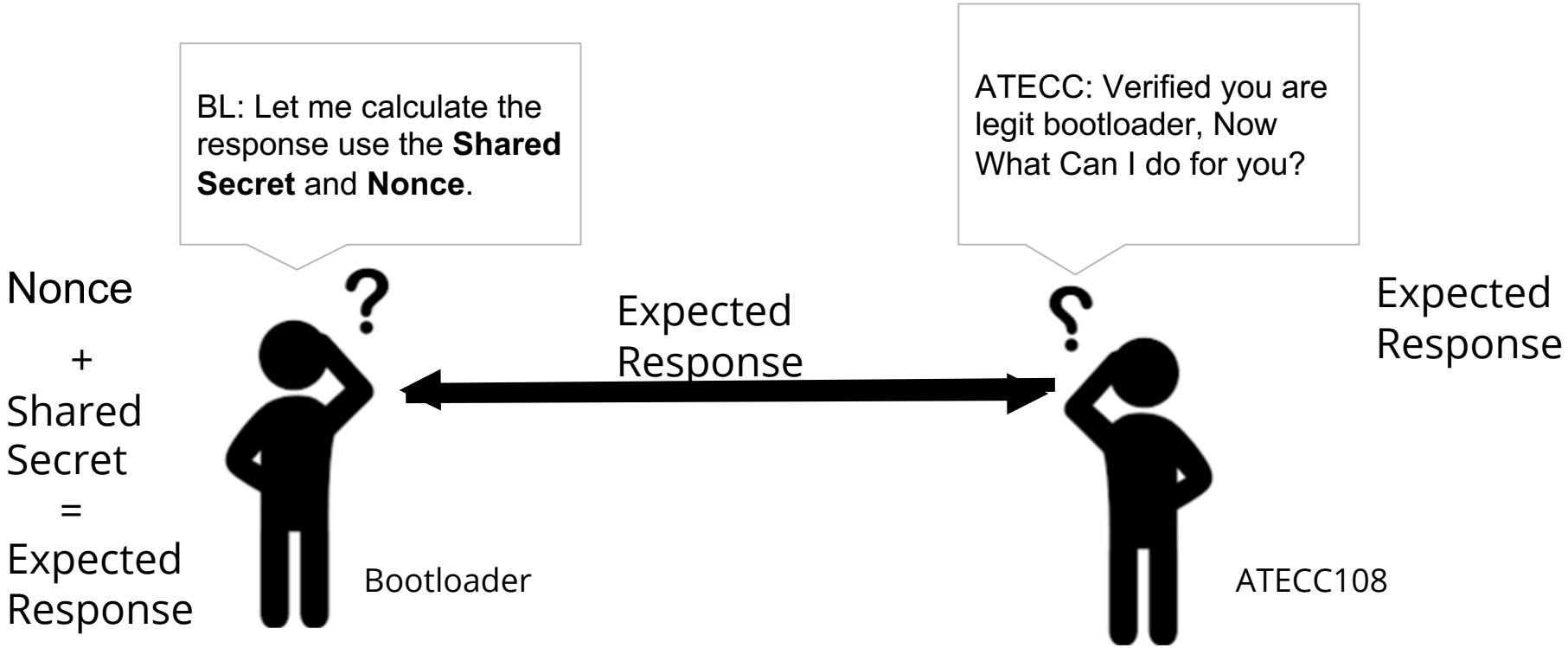
Trusting Trust Step 1: Authenticate the bootloader



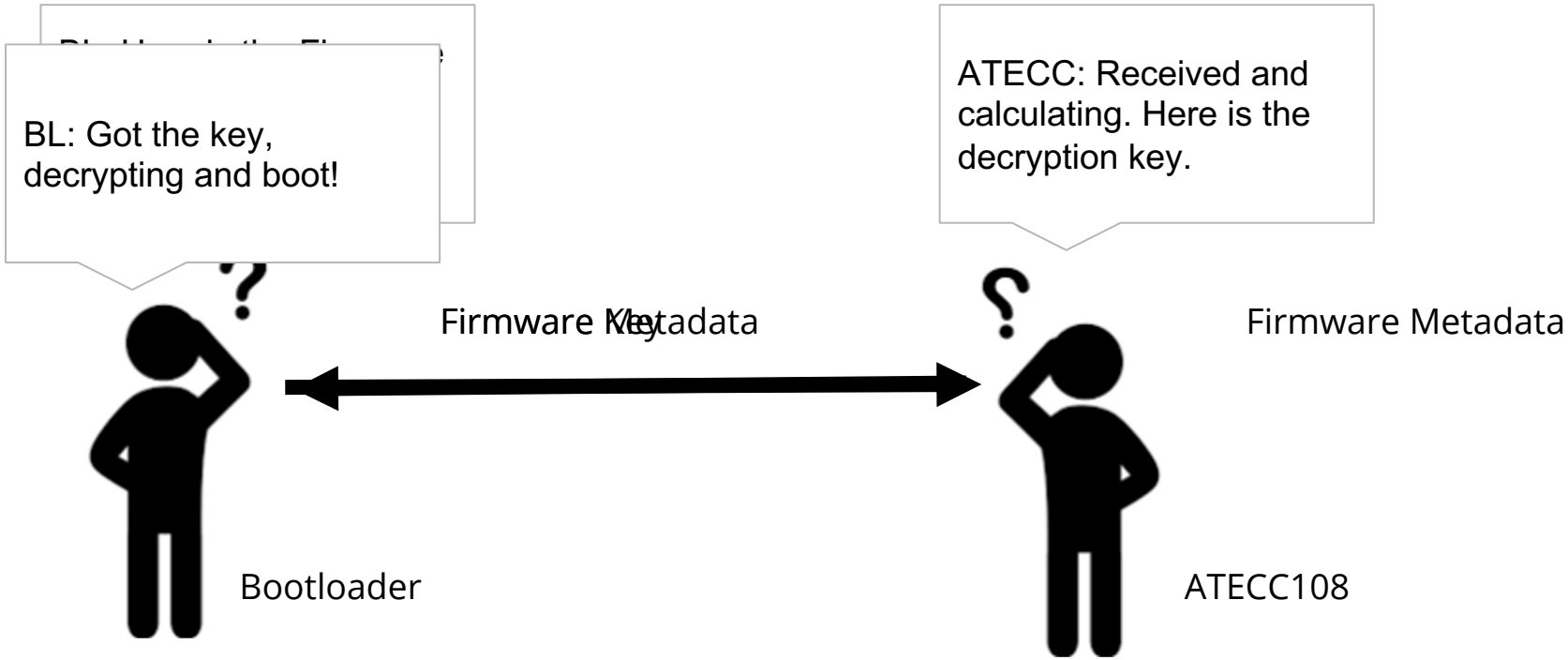
Trusting Trust Step 1: Authenticate the bootloader



Trusting Trust Step 1: Authenticate the bootloader



Trusting Trust Step 2: Decryption key generation



trusting Trust

I2C communication bus is **NOT** vulnerable to replay attack: Nonce is randomly generated each time of the authentication process

Bootloader's authenticity is being verified by the ATECC when establish RoT

Firmware is being protected and verified by symmetric AES-CBC encryption

Firmware Master key material to generate Firmware decryption key is protected by the ATECC chip anti-tampering hardware design.

trusting Trust

I2C communication bus is **NOT** vulnerable to replay attack: Nonce is randomly generated each time of the authentication process

Bootloader's authenticity is being verified by the ATECC when establish RoT

Firmware is being protected and verified by symmetric AES-CBC encryption

Firmware Master key material to generate Firmware decryption key is protected by the ATECC chip anti-tampering hardware design.

Successfully decryption of the Firmware  Firmware can be trusted

The Microcontroller blindly trust and execute whatever data it decrypts.This is **WRONG**.

Mistrusting Trust

I2C communication bus **exposes** the entire authentication process!!

ATECC chip act as a peripheral: **Cannot** verify the integrity of the **bootloader** before execution, instead only Shared Secret is verified.

The **Shared Secret** is **exposed** in the plain text bootloader (thank you cold boot robot!).

Symmetric Firmware verification goes **both way**

ATECC chip doesn't need to be soldered on Siemens S7-1500 PLC PCB to operate.

Mistrusting Trust

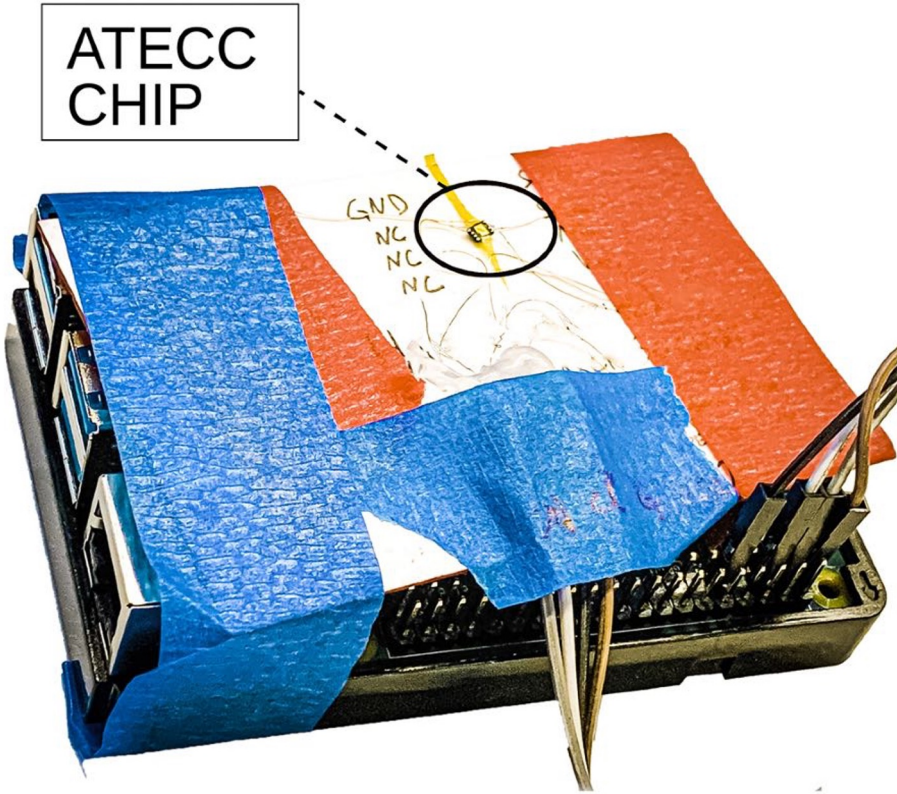
I2C communication bus **exposes** the entire authentication process!!

ATECC chip act as a peripheral: **Cannot** verify the integrity of the **bootloader** before execution, instead only Shared Secret is verified.

The **Shared Secret** is **exposed** in the plain text bootloader (thank you cold boot robot!).

Symmetric Firmware verification goes **both way**

ATECC chip doesn't need to be soldered on Siemens S7-1500 PLC PCB to operate.



ATECC chip as Oracle through Raspberry Pi

Impact

Adversaries may use the hardware RoT protections as a lever against the very same mechanisms meant to thwart attackers.

One single S7-1500 PLC can be used as an oracle to decrypt, re-encrypt, re-authenticate, tampered firmware for an entire generation of devices (Over 100 models affected**)**.

The designs flaws we discovered constitute a physical hardware vulnerability which cannot be patched by vendor security updates.

As embed not frequentl Γ for years to come.

Affected Product and Versions	Remediation
SIMATIC Drive Controller CPU 1504D TF (6ES7615-4DF10-0AB0): All versions	Currently no fix is planned See recommendations from section Workarounds and Mitigations
SIMATIC Drive Controller CPU 1507D TF (6ES7615-7DF10-0AB0): All versions	Currently no fix is planned See recommendations from section Workarounds and Mitigations

Conclusion - What Can We Learn?

100% secure hardware does not exist! Too much trust placed in ‘secure’ hardware is a double-edged sword!

Trusted Peripherals do not exist! Attestation should ideally be both forward and backwards facing to ensure device has not been compromised.

Low-Cost Cryptographic processors are a useful for adding cryptography into systems which would otherwise go without - however, integrators should carefully consider implementation and approach integration from a Zero-Trust perspective.

References

[1] W. A. Arbaugh, D. J. Farber, and J. M. Smith, “A secure and reliable bootstrap architecture,” in Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097). IEEE, 1997, pp. 6571.

[2] N. Falliere, L. O. Murchu, and E. Chien, “W32. stuxnet dossier,” White paper, symantec corp., security response, vol. 5, no. 6, p. 29, 2011.

[3] I. . W. Group et al., “IEEE std. 1149.1 — standard test access port and boundary-scan architecture,” Retrieved March, vol. 9, 2017.

[4] U. Nanda and S. K. Pattnaik, “Universal asynchronous receiver and transmitter (UART),” in 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 01, 2016, pp. 1–5.

[5] Y. Wu, G. Skipper, and A. Cui, “Cryo-mechanical RAM content extraction against modern embedded systems,” In 2023, 17th IEEE Workshop on Offensive Technologies (WOOT’23). To appear.

Acknowledgement

The authors would like to thank Jack Zheng and Aleksey Nogin from Red Balloon Security and HOST anonymous reviewers for their feedback on the early versions of the paper.

Contact Yuanzhe Wu: yuanzhewu@gmail.com | hans@redballoonsecurity.com