

Partial Key Overwrite Attacks in Microcontrollers: a Survey

CASCADE 2025

pcy Sluys, Lennert Wouters, Benedikt Gierlichs, Ingrid Verbauwhede

Outline

- 1 Introduction
- 2 PKO Attacks
- 3 Survey
- 4 Conclusion

Outline

1 Introduction

2 PKO Attacks

3 Survey

4 Conclusion

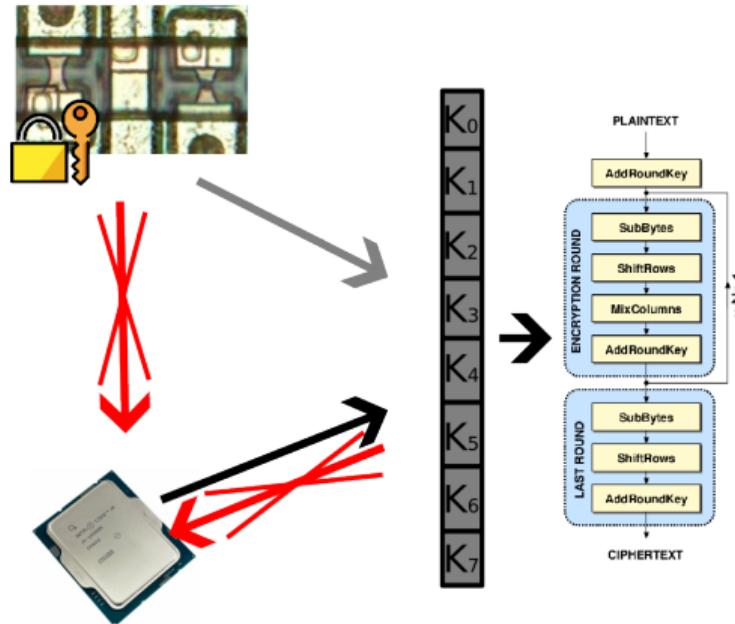
What are PKO attacks?

- ▶ Chips often have cryptographic accelerators^{CITATION NEEDED}
- ▶ In some implementations, the key is kept separate from the CPU
 - PUF
 - OTP/fuses
 - ...
 - Hardware KDF
 - TEE

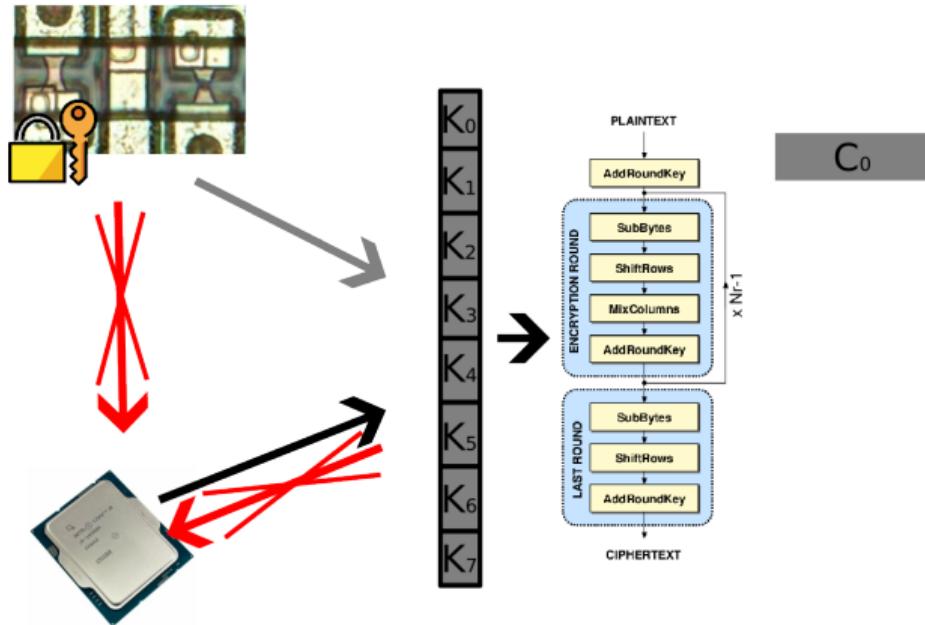
What are PKO attacks?

- ▶ Chips often have cryptographic accelerators^{CITATION NEEDED}
- ▶ In some implementations, the key is kept separate from the CPU
 - PUF
 - OTP/fuses
 - ...
 - Hardware KDF
 - TEE
- ▶ If **partial** overwrite of a **write-only** key register allowed: key leakage!
 - ⇒ Partial Key Overwrite attack

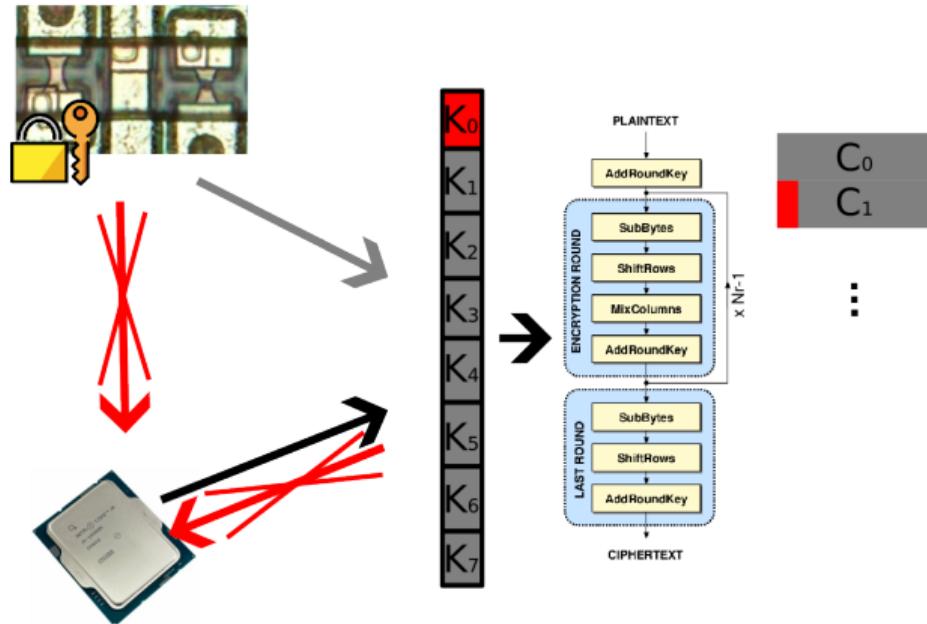
An example



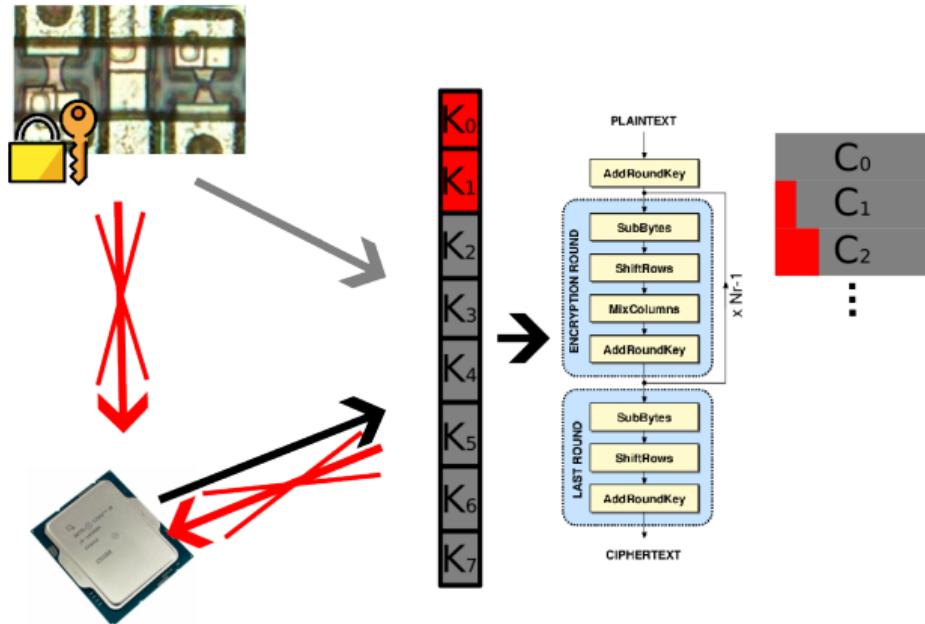
An example



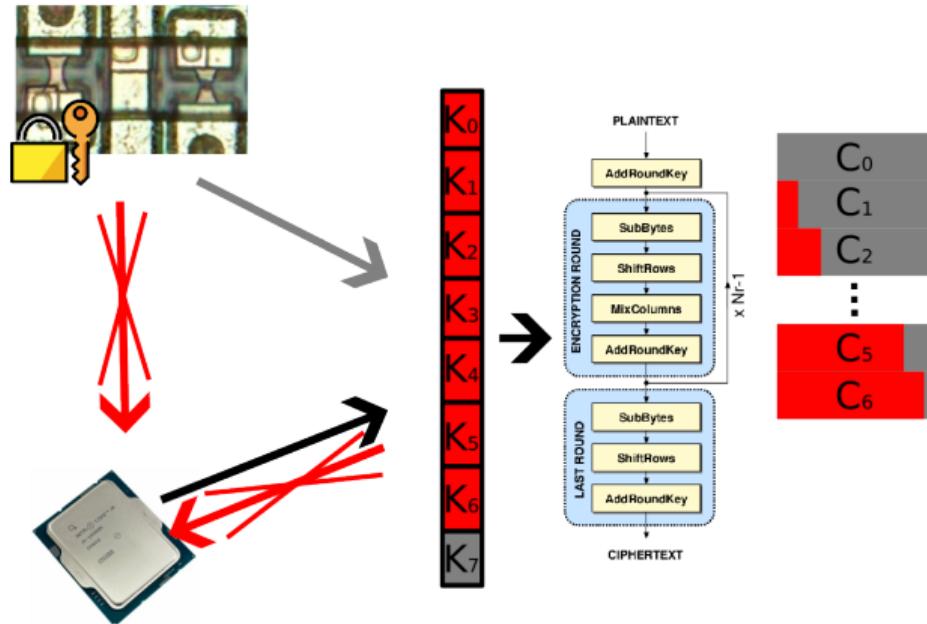
An example



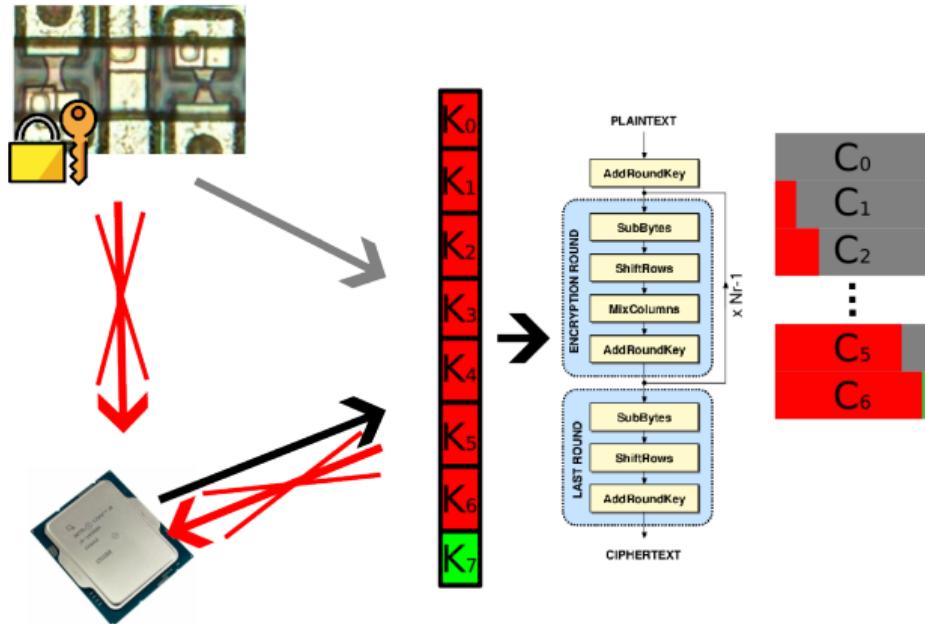
An example



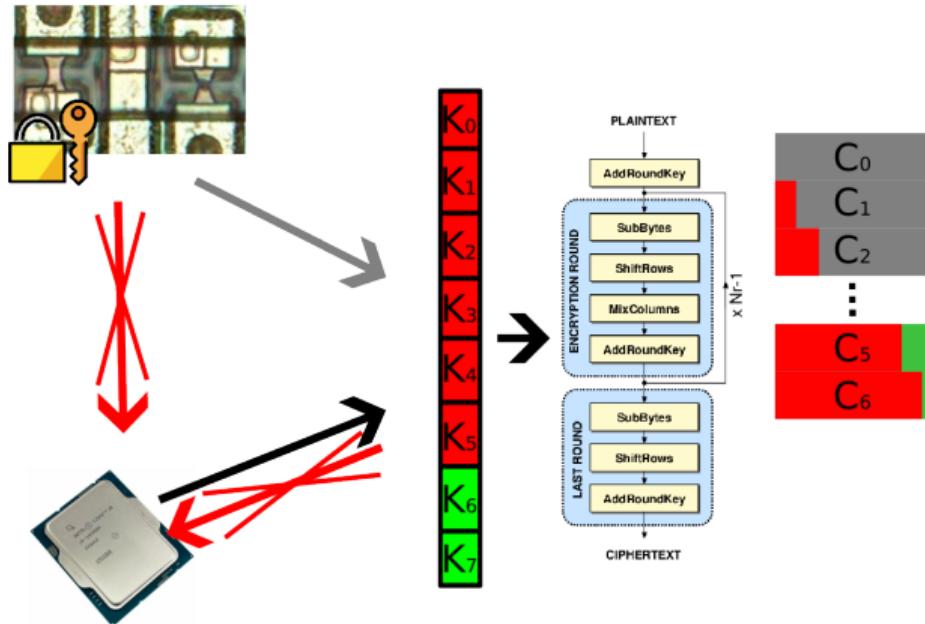
An example



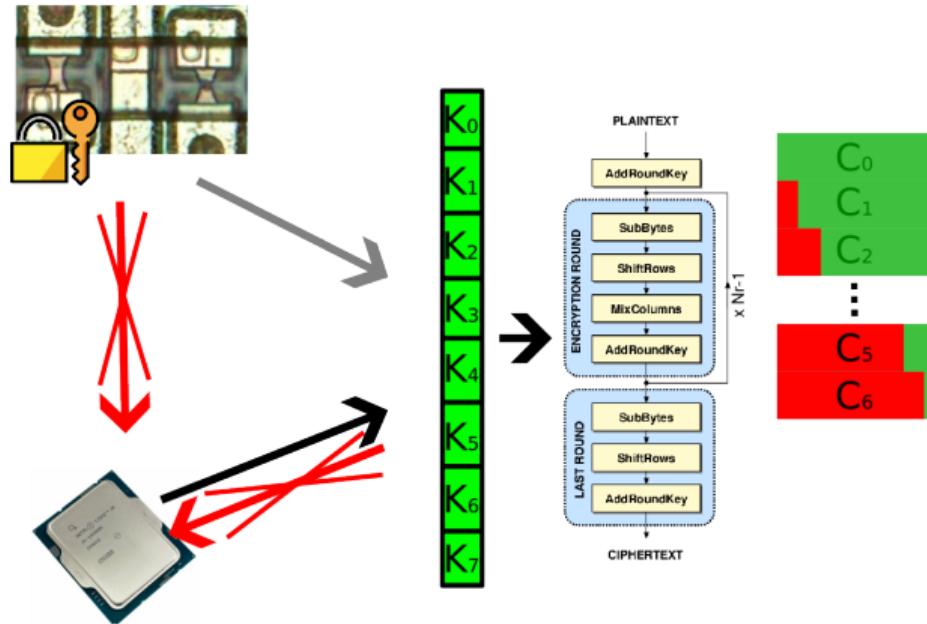
An example



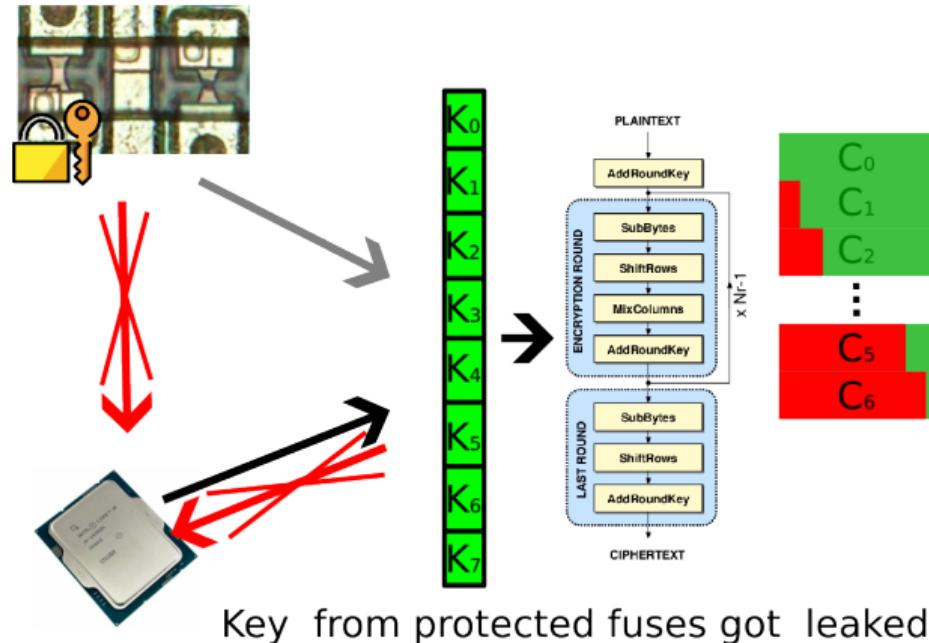
An example



An example



An example



Outline

1 Introduction

2 PKO Attacks

3 Survey

4 Conclusion

Attacker model

- ▶ Attacker can query cryptographic module
- ▶ Key not exposed to attacker
- ▶ Attacker can overwrite **parts of** the key

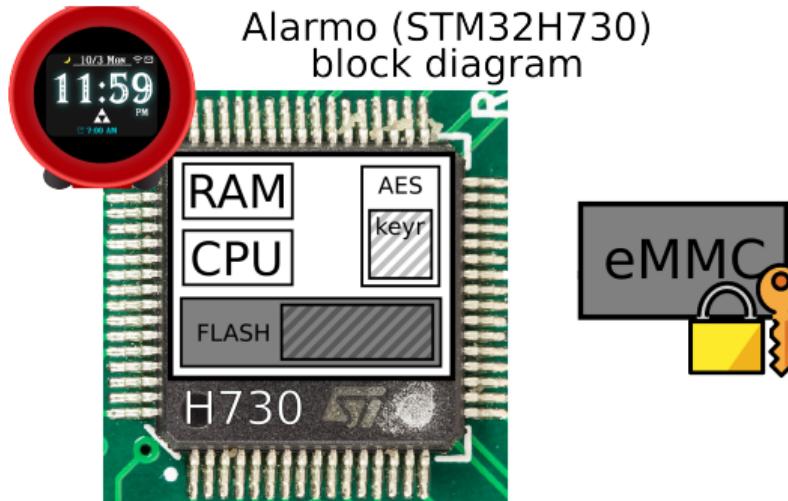
Attacker model

- ▶ Attacker can query cryptographic module
- ▶ Key not exposed to attacker
- ▶ Attacker can overwrite **parts of** the key

Embedded context:

- ▶ Query cryptographic module using low-privilege code execution or debug access
- ▶ Key secured using TEE, protected fuses, boot ROM, ...

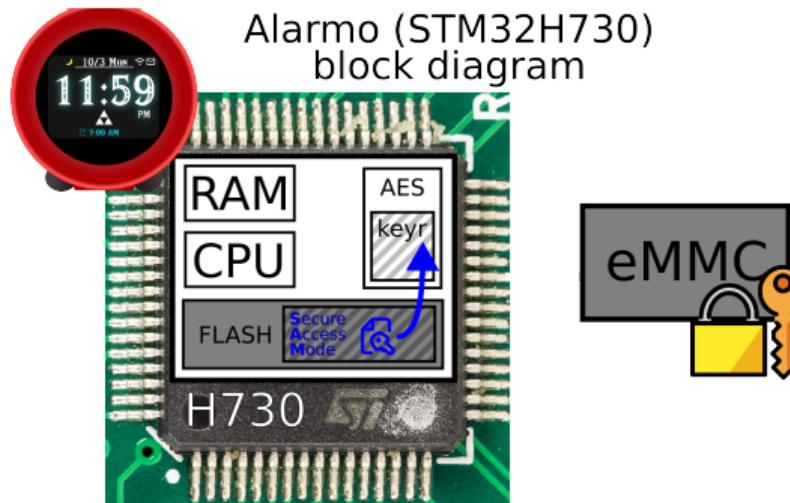
Example: Alarmo¹



Based on CC BY-SA 4.0 images by Raimond Spekking and PantheraLeo1359531

¹<https://garyodernichts.blogspot.com/2024/10/looking-into-nintendo-alarmo.html>

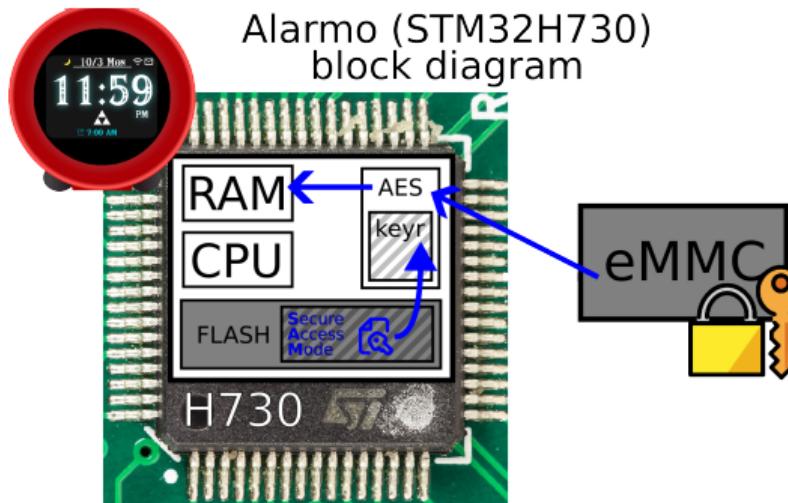
Example: Alarmo¹



Based on CC BY-SA 4.0 images by Raimond Spekking and PantheraLeo1359531

¹<https://garyodernichts.blogspot.com/2024/10/looking-into-nintendo-alarmo.html>

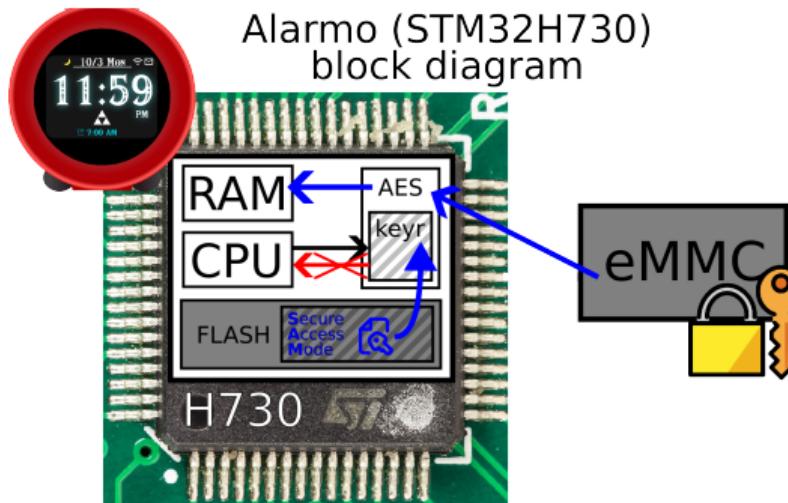
Example: Alarmo¹



Based on CC BY-SA 4.0 images by Raimond Spekking and PantheraLeo1359531

¹<https://garyodernichts.blogspot.com/2024/10/looking-into-nintendo-alarmo.html>

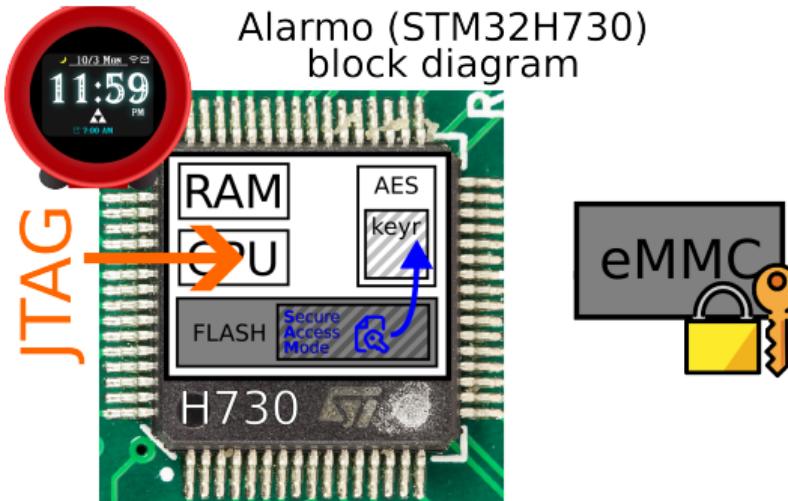
Example: Alarmo¹



Based on CC BY-SA 4.0 images by Raimond Spekking and PantheraLeo1359531

¹<https://garyodernichts.blogspot.com/2024/10/looking-into-nintendo-alarmo.html>

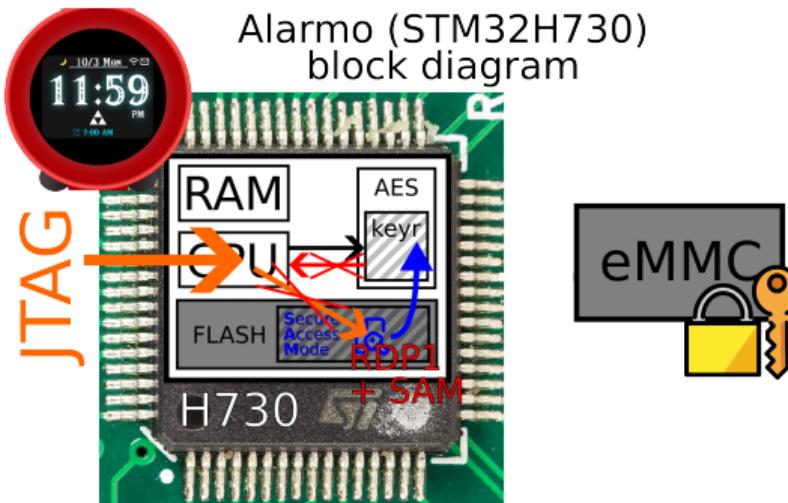
Example: Alarmo¹



Based on CC BY-SA 4.0 images by Raimond Spekking and PantheraLeo1359531

¹<https://garyodernichts.blogspot.com/2024/10/looking-into-nintendo-alarmo.html>

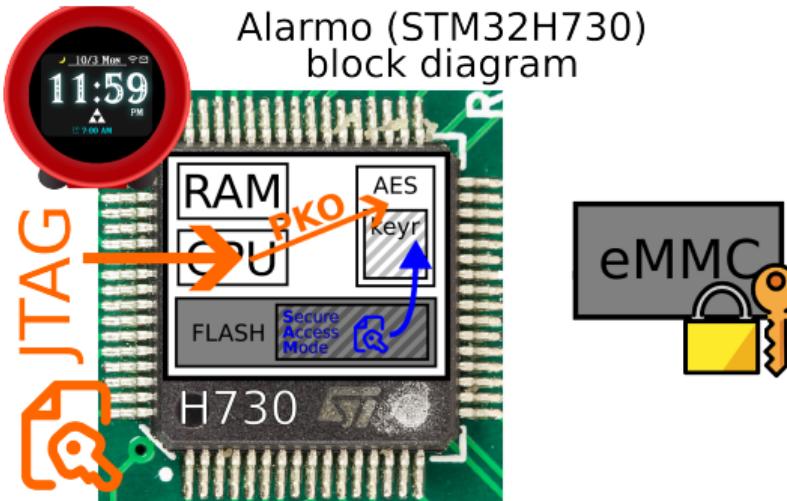
Example: Alarmino¹



Based on CC BY-SA 4.0 images by Raimond Spekking and PantheraLeo1359531

¹<https://garyodernichts.blogspot.com/2024/10/looking-into-nintendo-alarmo.html>

Example: Alarmo¹



Based on CC BY-SA 4.0 images by Raimond Spekking and PantheraLeo1359531

¹<https://garyodernichts.blogspot.com/2024/10/looking-into-nintendo-alarmo.html>

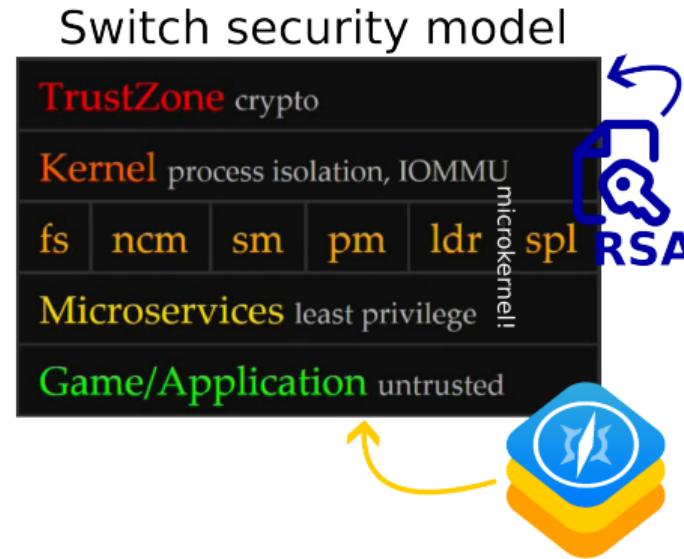
Example: Nintendo Switch²



Based on slide from "Console Security - Switch" by plutoo, derrek and naehrwert. CC BY 4.0

²https://switchbrew.org/wiki/Switch_System_Flaws#TrustZone

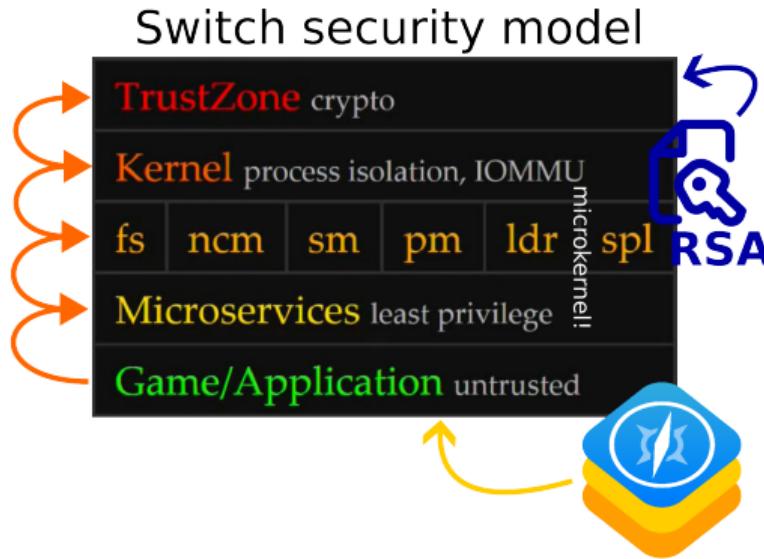
Example: Nintendo Switch²



Based on slide from "Console Security - Switch" by plutoo, derrek and naehrwert. CC BY 4.0

²https://switchbrew.org/wiki/Switch_System_Flaws#TrustZone

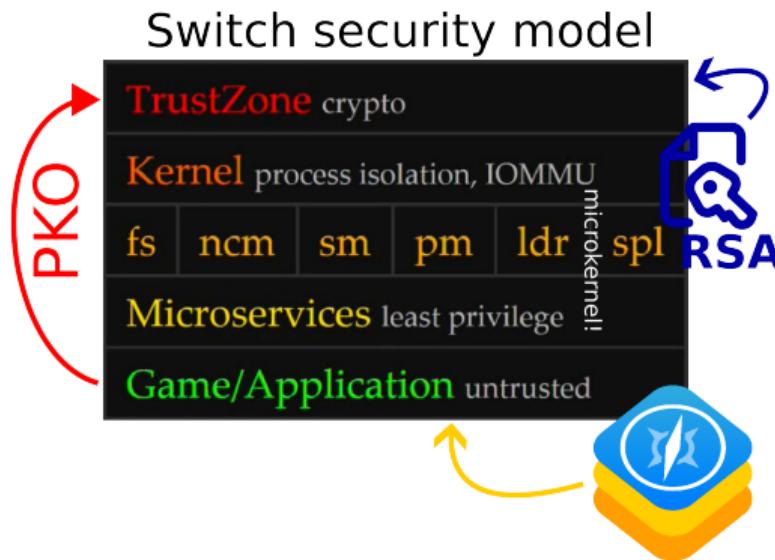
Example: Nintendo Switch²



Based on slide from "Console Security - Switch" by plutoo, derrek and naehrwert. CC BY 4.0

²https://switchbrew.org/wiki/Switch_System_Flaws#TrustZone

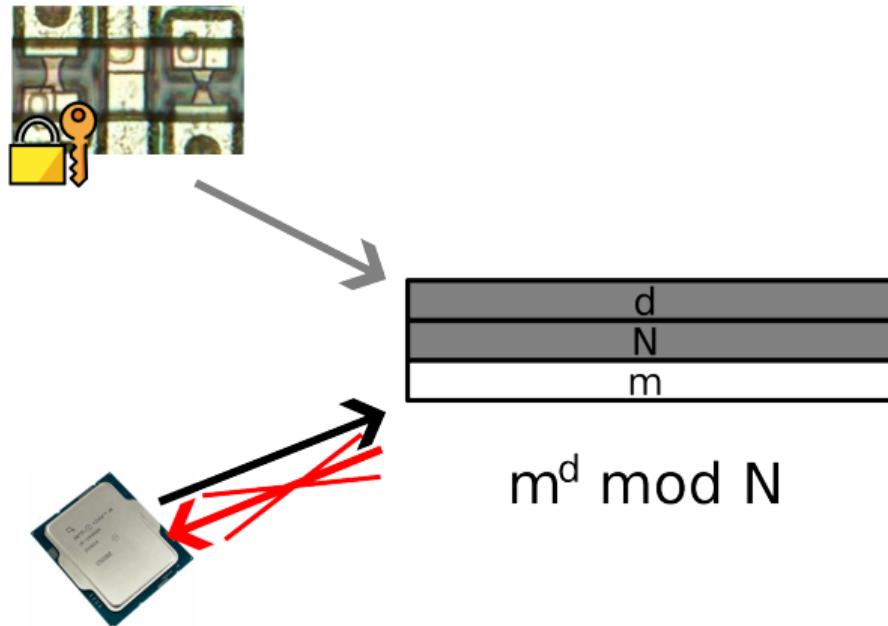
Example: Nintendo Switch²



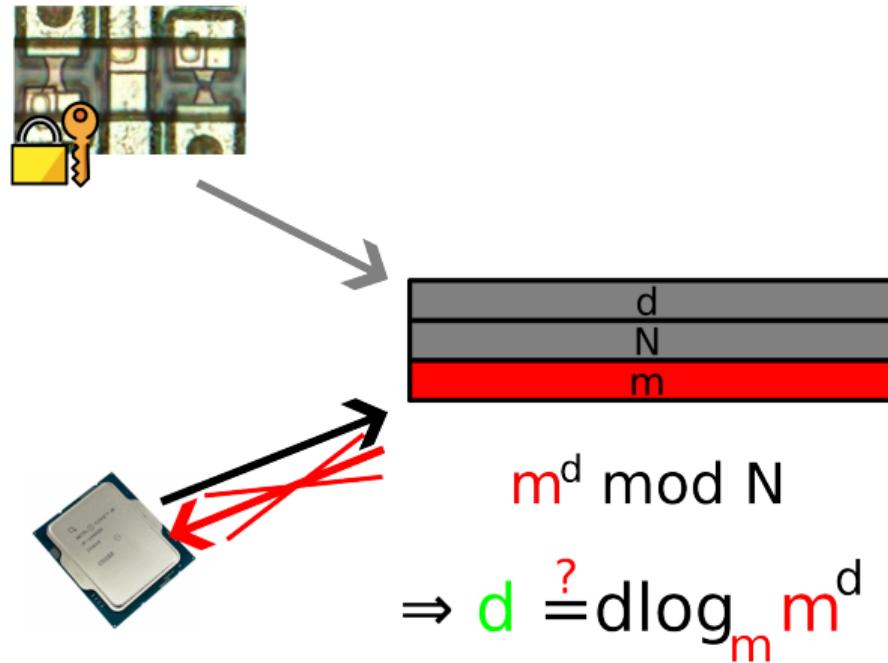
Based on slide from "Console Security - Switch" by plutoo, derrek and naehrwert. CC BY 4.0

²https://switchbrew.org/wiki/Switch_System_Flaws#TrustZone

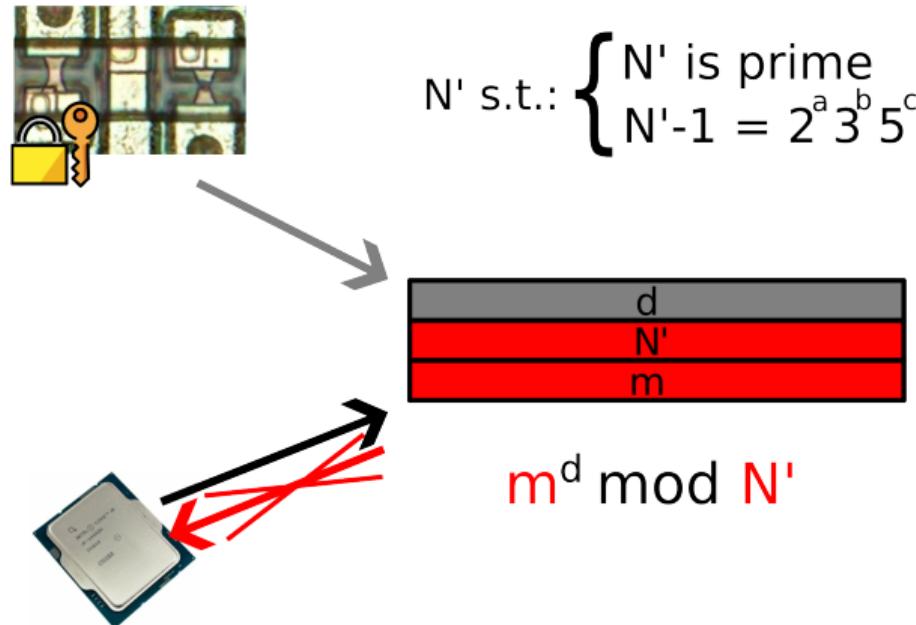
Not just block ciphers



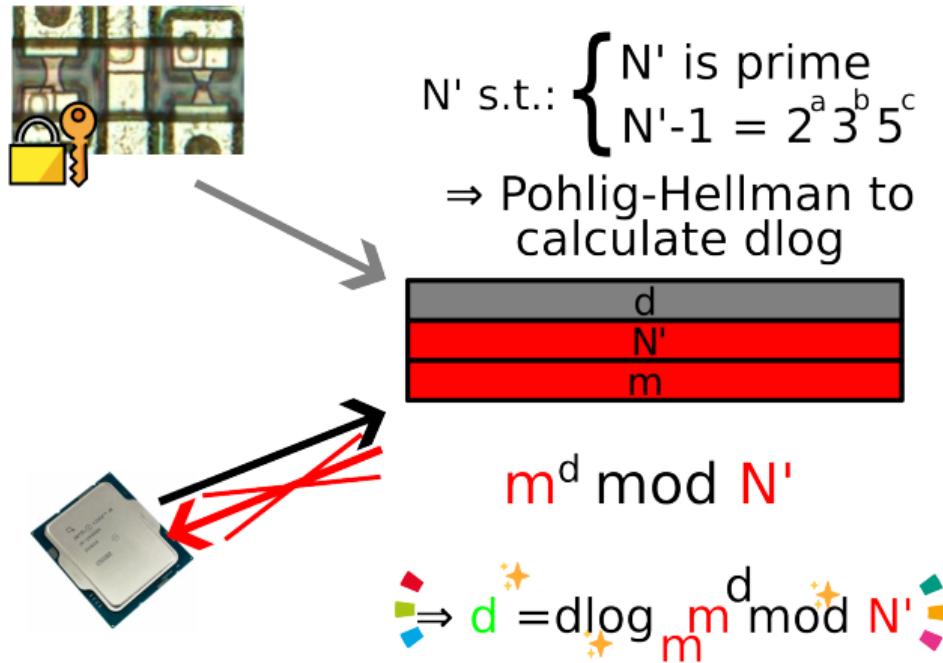
Not just block ciphers



Not just block ciphers



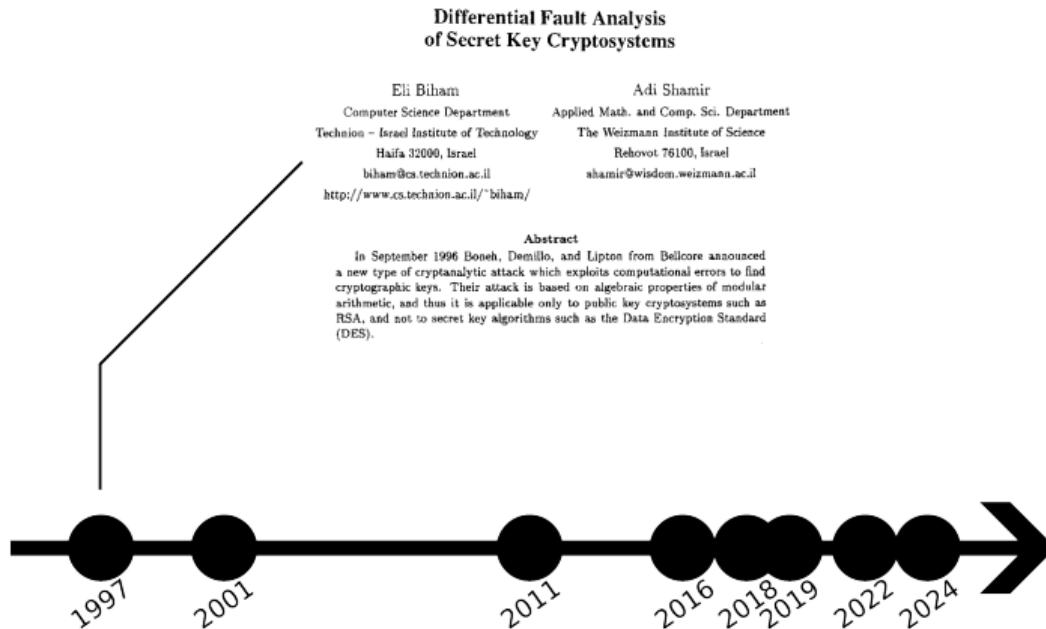
Not just block ciphers



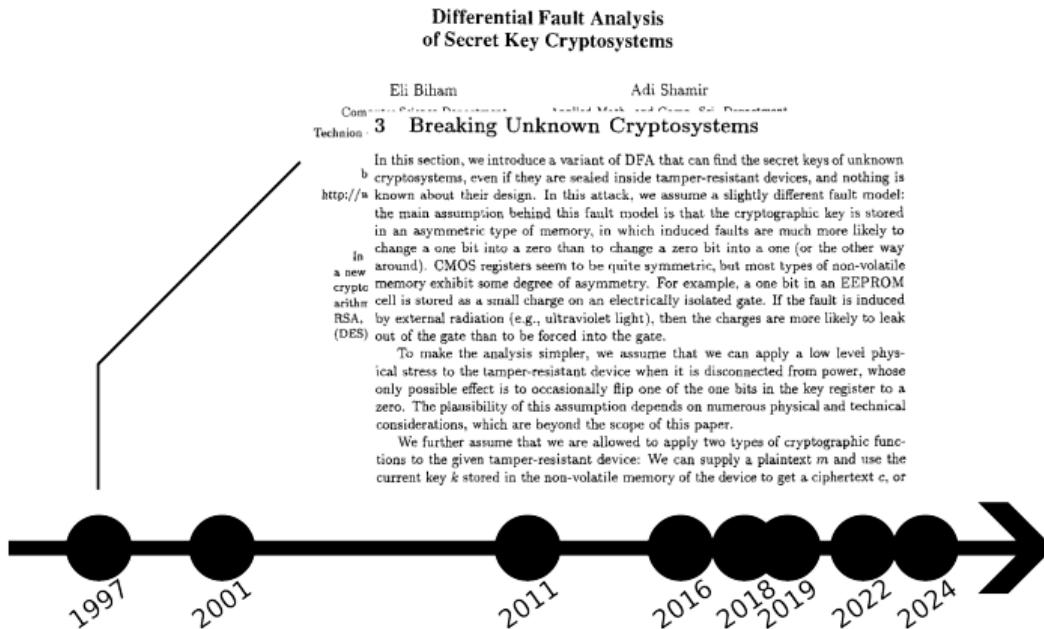
Timeline of PKO attacks



Timeline of PKO attacks



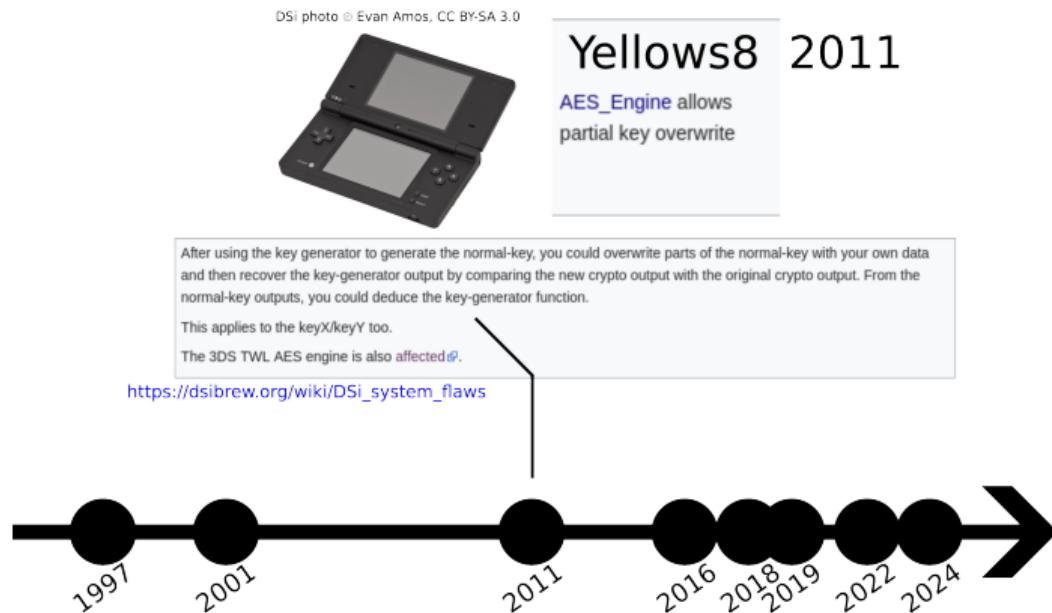
Timeline of PKO attacks



Timeline of PKO attacks



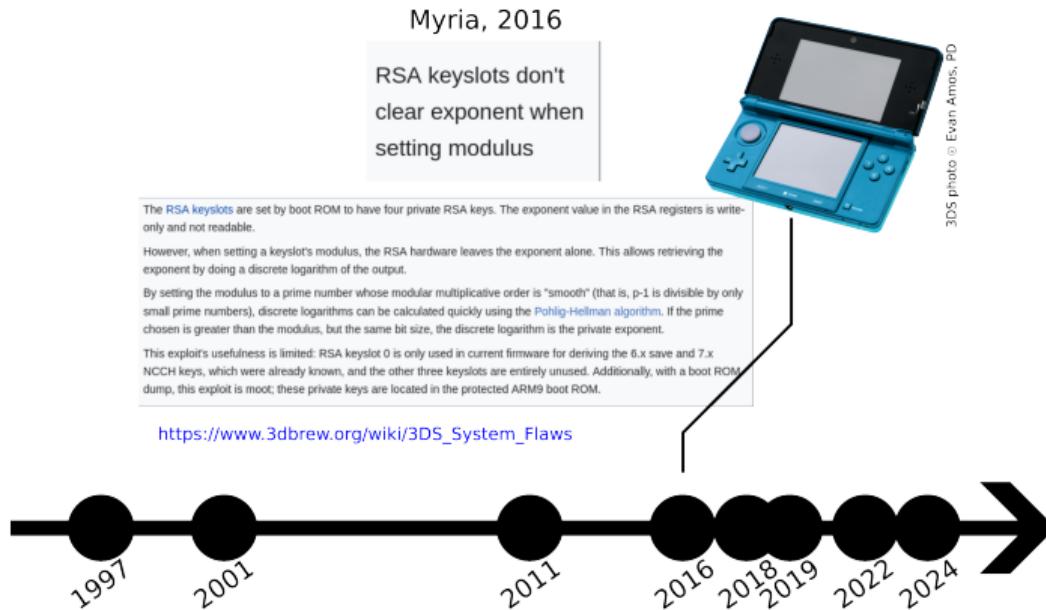
Timeline of PKO attacks



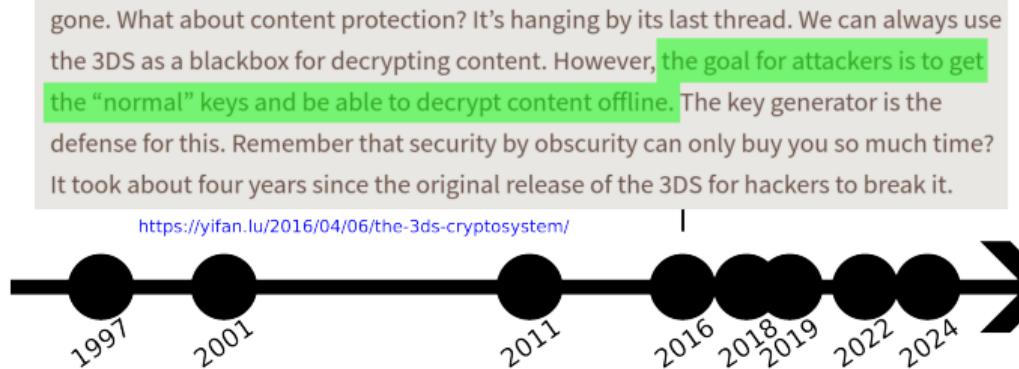
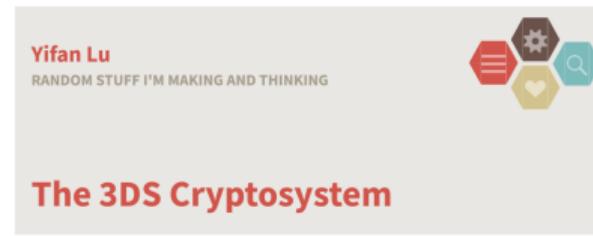
Timeline of PKO attacks



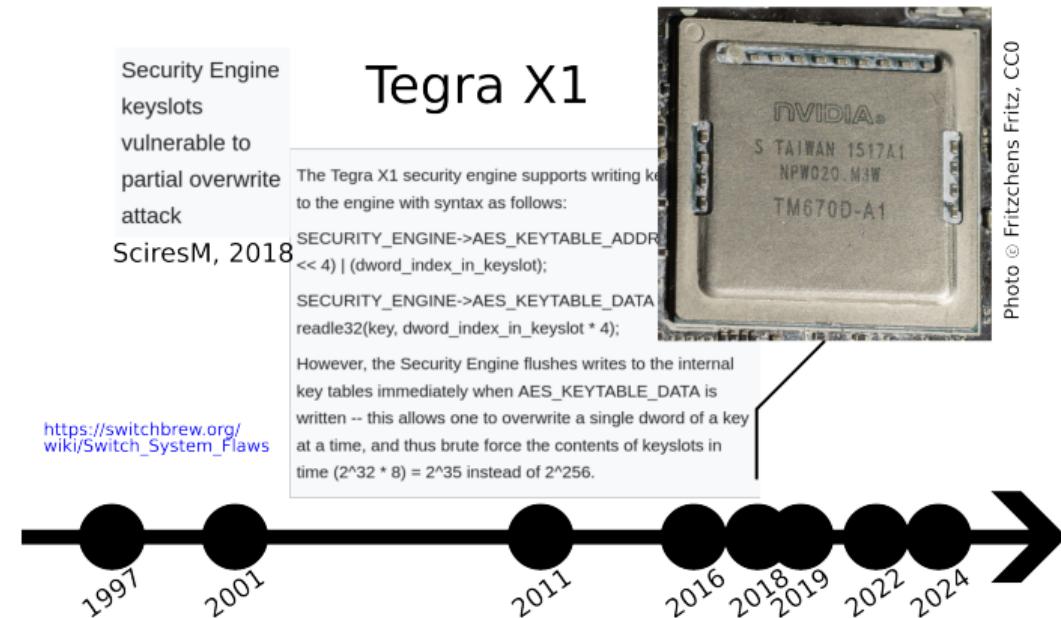
Timeline of PKO attacks



Timeline of PKO attacks



Timeline of PKO attacks



Timeline of PKO attacks

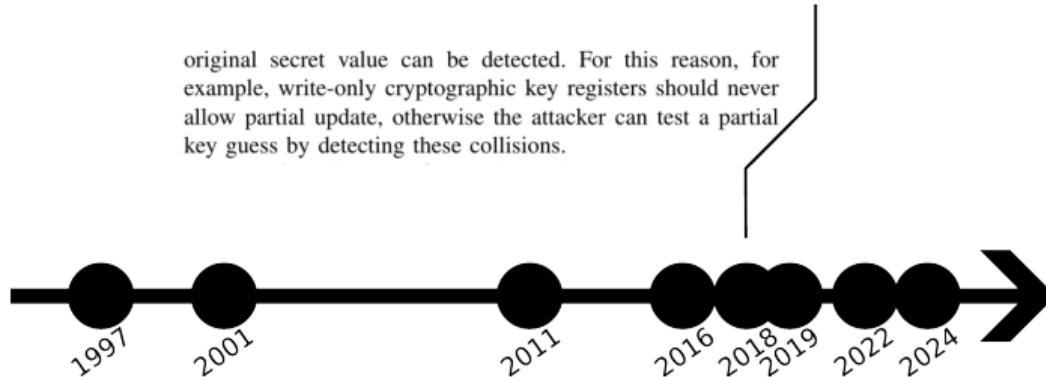
Fault Attacks on Secure Embedded Software: Threats, Design and Evaluation

Bilgiday Yuce
Virginia Tech
Blacksburg, VA
bilgiday@vt.edu

Patrick Schaumont
Virginia Tech
Blacksburg, VA
schaum@vt.edu

Marc Wittenman
Riscure – Security Lab
Delft, Netherlands
wittenman@riscure.com

original secret value can be detected. For this reason, for example, write-only cryptographic key registers should never allow partial update, otherwise the attacker can test a partial key guess by detecting these collisions.



Timeline of PKO attacks

TrustZone
allows using
imported RSA
exponents
with arbitrary
modulus
**SciresM,
2019**

TrustZone supports "importing" RSA private keys encrypted with TrustZone only keydata in console that has compromised userland / calculations with them offline. In practice and SSL (console client cert communication). However, the actual SMC API only imports separately by userland in each call. This means that userland can pass in any message private exponent) % modulus back from the API.

By choosing a prime number modulus P such that P has "smooth" order ($\text{totient}(P)$) only by "small" primes), one can efficiently use the [Pollard-Rho algorithm](#) to calculate the discrete logarithm of such a result directly, and thus obtain the private exponent.

This is mostly useless in practice, given the general availability of other exploits to obtain these decrypted exponents.

This was fixed in 10.0.0 by importing the modulus in addition to the exponent for the ES device key and ES client cert key. For backwards compatibility reasons the SSL key and Lotus key still only import the exponent.

StorageExpMod also now validates that the exponentiation of "DDDD..." about the provided modulus by the imported exponent and then the fixed public exponent returns "DDDD...", and returns invalid argument if validation fails.

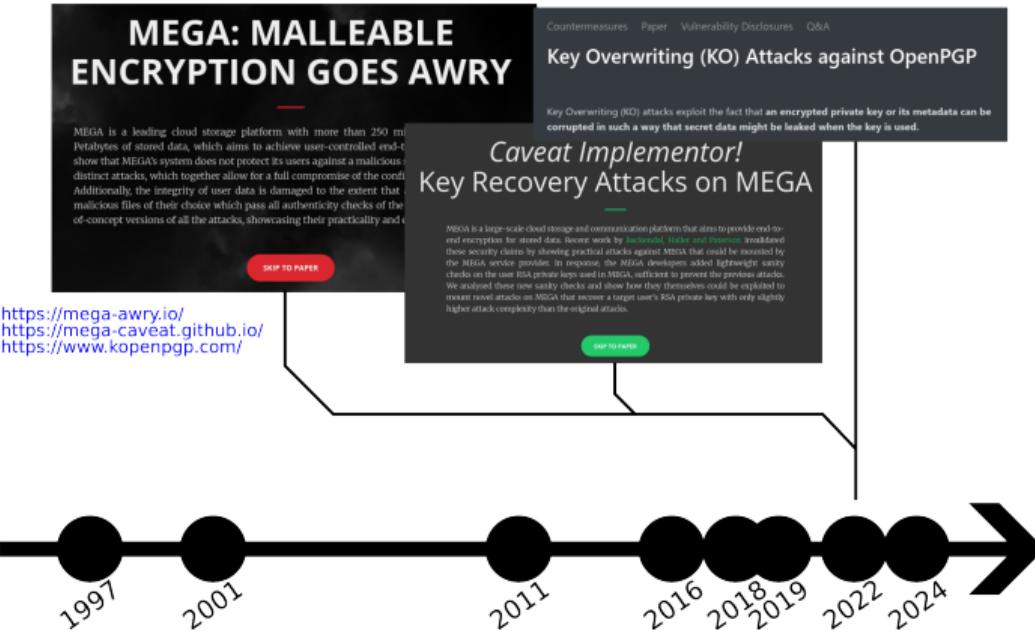


Switch photo © Evan Amos, PD

[https://switchbrew.org/
wiki/Switch_System_Flaws](https://switchbrew.org/wiki/Switch_System_Flaws)



Timeline of PKO attacks



Timeline of PKO attacks



Timeline of PKO attacks



Attack comparison and context

- ▶ Mounted using software exploits:
 - PGP attacks
 - MEGA attacks
- ▶ Mounted using invasive circuit-level attacks:
 - Partial EEPROM wipe in [1]
- ▶ Mounted using fault injection:
 - Safe Error Analysis

Attack comparison and context

- ▶ Mounted using software exploits:
 - PGP attacks
 - MEGA attacks
- ▶ Mounted using invasive circuit-level attacks:
 - Partial EEPROM wipe in [1]
- ▶ Mounted using fault injection:
 - Safe Error Analysis
- ▶ Mounted using [??]:
 - PKO in microcontroller

“Mounted using [???”?

Exact attack method?

- ▶ Not software vuln: attacking bug in **hardware** state machine
- ▶ Not side channel or fault attack
- ▶ Not an invasive attack

“Mounted using [???”?

Different attack:

- ▶ Hardware components work fine: AES, CPU, DMA, IRQ, system bus, timer, ...

Exact attack method?

- ▶ Not software vuln: attacking bug in **hardware** state machine
- ▶ Not side channel or fault attack
- ▶ Not an invasive attack

“Mounted using [???”?

Exact attack method?

- ▶ Not software vuln: attacking bug in **hardware** state machine
- ▶ Not side channel or fault attack
- ▶ Not an invasive attack

Different attack:

- ▶ Hardware components work fine: AES, CPU, DMA, IRQ, system bus, timer, ...
- ▶ *Composition of blocks* ⇒ *hardware bugs*

“Mounted using [???”?

Exact attack method?

- ▶ Not software vuln: attacking bug in **hardware** state machine
- ▶ Not side channel or fault attack
- ▶ Not an invasive attack

Different attack:

- ▶ Hardware components work fine: AES, CPU, DMA, IRQ, system bus, timer, ...
- ▶ *Composition of blocks* ⇒ *hardware bugs*

Examples:

- ▶ ‘Execute-only memory’ readout using timer interrupts [3, 5]
- ▶ BUSted & Nemesis
- ▶ More: [2], [4, ntrcardhax], [6], [7]

“Mounted using [???”?

Exact attack method?

- ▶ Not software vuln: attacking bug in **hardware** state machine
- ▶ Not side channel or fault attack
- ▶ Not an invasive attack

Different attack:

- ▶ Hardware components work fine: AES, CPU, DMA, IRQ, system bus, timer, ...
- ▶ *Composition of blocks* ⇒ *hardware bugs*

Examples:

- ▶ ‘Execute-only memory’ readout using timer interrupts [3, 5]
- ▶ BUSted & Nemesis
- ▶ More: [2], [4, ntrcardhax], [6], [7]

Nameless...

“Mounted using [???]”?

- ▶ “Standalone components work fine”

E

gs

Nameless...

“Mounted using [???]”?

- ▶ “Standalone components work fine”
- ▶ “Interfacing of blocks is complex, bugs arise”

E

gs

Nameless...

“Mounted using [???]”?

- ▶ “Standalone components work fine”
- ▶ “Interfacing of blocks is complex, bugs arise”

E

gs

(with help from people on Mastodon...)

VULNERABLE **R**ESULT OF
INDIVIDUALLY **S**ECURE
COMPONENTS **A**TTACK

Nameless...

“Mounted using [???”?

- ▶ “Standalone components work fine”
- ▶ “Interfacing of blocks is complex, bugs arise”

E

gs

(with help from people on Mastodon...)

VULNERABLE RESULT OF
INDIVIDUALLY SECURE
COMPONENTS ATTACK



VRISKA (from *Homestuck*)

Nameless...

Outline

1 Introduction

2 PKO Attacks

3 Survey

4 Conclusion

This raises the question,

Is this still a problem nowadays?

How many off-the-shelf chips are vulnerable?

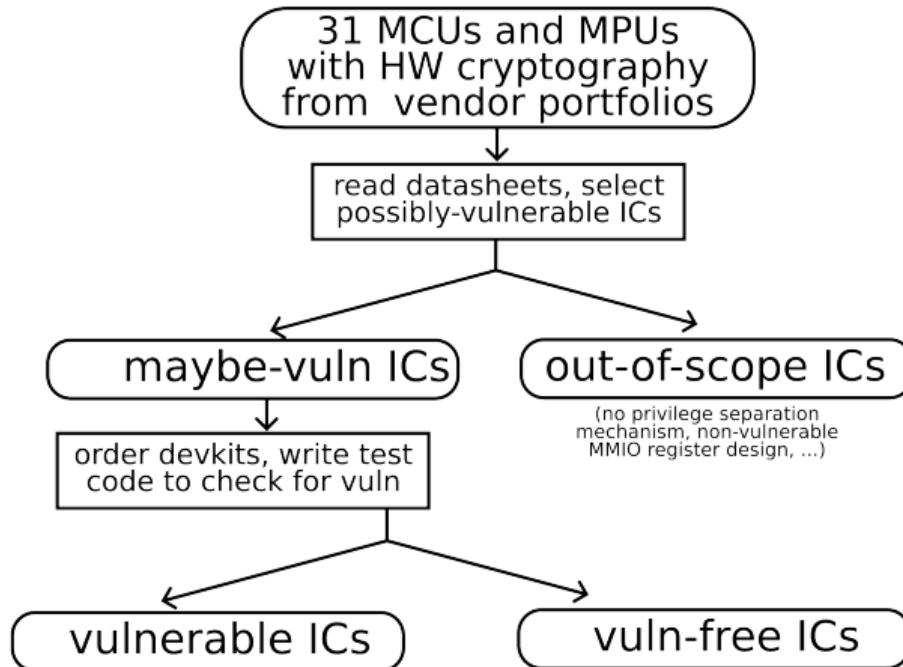
This raises the question,

Is this still a problem nowadays?

How many off-the-shelf chips are vulnerable?

⇒ **Time for a survey**

Method



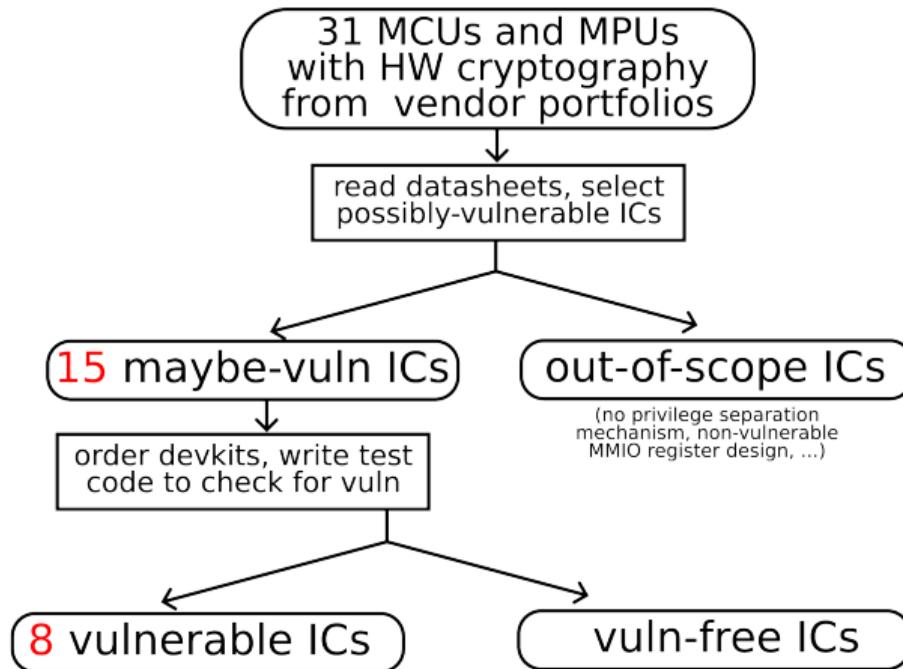
Focus

- ▶ Skews towards SoCs with documented accelerators (i.e. **microcontrollers**, not **microprocessors**)
 - Please publish documentation
 - Attempted reverse-engineering two anyway
(Renesas RA2E1, Microchip SAML11)

Focus

- ▶ Skews towards SoCs with documented accelerators (i.e. **microcontrollers**, not **microprocessors**)
 - Please publish documentation
 - Attempted reverse-engineering two anyway (Renesas RA2E1, Microchip SAML11)
- ▶ Survey of **SoCs**, not *end-user products*
 - Latter not practical

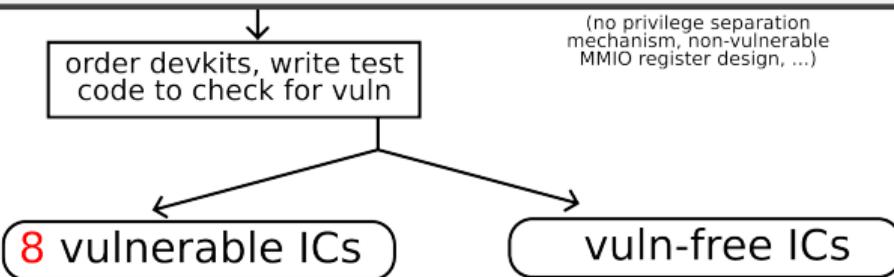
Results



Results

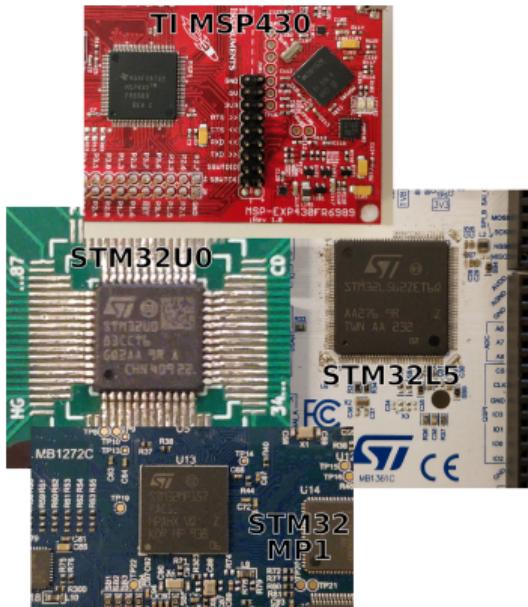
31 MCUs and MPUs
with HW cryptography
from vendor portfolios

- ▶ 3 different vendors with vulnerable devices
- ▶ For every vendor: also have a product **with** countermeasures
- ▶ RSA accelerators: hardware bugs??

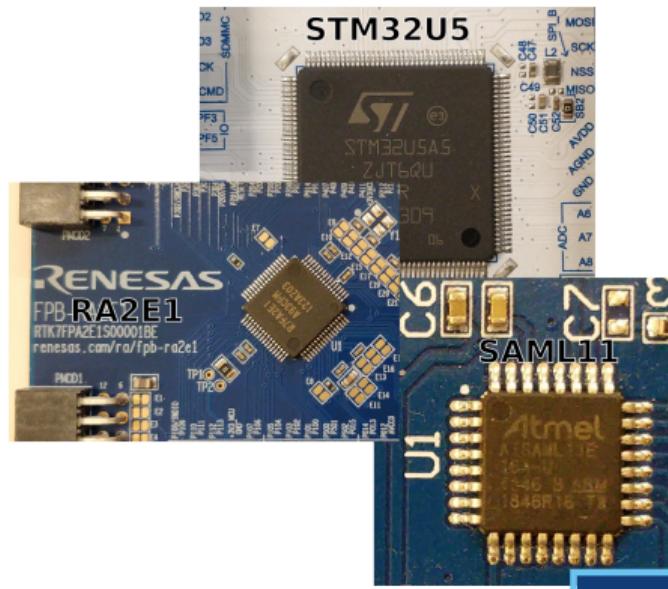


Details

Vulnerable



Not vuln.



Details

ESP32x3:

Vulnerable



It's Complicated

Not vuln.



Outline

1 Introduction

2 PKO Attacks

3 Survey

4 Conclusion

Conclusion

- ▶ ‘Simple’ but overlooked attack
- ▶ Caused not by single faulty component, but complex interaction between components
- ▶ Still important to real-world attackers
- ▶ Seems to be *slowly* on its way out?
(Correlation with introduction of Arm PSA Certified?)

Questions?

Bibliography

- [1] Eli Biham and Adi Shamir. "Differential Fault Analysis of Secret Key Cryptosystems". In: *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*. Ed. by Burton S. Kaliski. CRYPTO '97. Berlin, Heidelberg: Springer-Verlag, 1997, pp. 513–525. ISBN: 978-3-540-69528-8. URL: <https://doc.lagout.org/security/Papers/DFA%20of%20Secret%20Key%20Cryptosystems.pdf>.
- [2] Márton Bognár, Jo Van Bulck, and Frank Piessens. "Mind the Gap: Studying the Insecurity of Provably Secure Embedded Trusted Execution Architectures". In: *2022 IEEE Symposium on Security and Privacy (SP)*. 2022, pp. 1638–1655. DOI: 10.1109/SP46214.2022.9833735. URL: <https://mici.hu/papers/bognar2022gap.pdf>.
- [3] Márton Bognár et al. "Intellectual Property Exposure: Subverting and Securing Intellectual Property Encapsulation in Texas Instruments Microcontrollers". In: *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 2155–2172. ISBN: 978-1-939133-44-1. URL: <https://www.usenix.org/conference/usenixsecurity24/presentation/bognar>.
- [4] plutoo, derrek, and smea. *Console Hacking: Breaking the 3DS*. Dec. 27, 2015. URL: https://media.ccc.de/v/32c3-7240-console_hacking (visited on 12/16/2024).
- [5] Marc Schink and Johannes Obermaier. "Taking a look into execute-only memory". In: *Proceedings of the 13th USENIX Conference on Offensive Technologies*. WOOT'19. Santa Clara, CA, USA: USENIX Association, 2019, p. 1. URL: https://www.usenix.org/system/files/woot19-paper_schink.pdf.
- [6] SciresM and hexkyz. *Je Ne Sais Quoi - Falcons over the Horizon*. Archived at <https://archive.is/wNT42>. Nov. 19, 2021. URL: <https://hexkyz.blogspot.com/2021/11/je-ne-sais-quoi-falcons-over-horizon.html> (visited on 12/06/2024).
- [7] Samuel Junjie Tan, Sergey Bratus, and Travis Goodspeed. "Interrupt-Oriented Bugdoor Programming: A Minimalist Approach to Bugdooring Embedded Systems Firmware". In: *Proceedings of the 30th Annual Computer Security Applications Conference*. ACSAC '14. New Orleans, Louisiana, USA: Association for Computing Machinery, 2014, pp. 116–125. ISBN: 9781450330053. DOI: 10.1145/2664243.2664268. URL: <https://doi.org/10.1145/2664243.2664268>.

Other image attributions

- ▶ zest (Encryption key icon, MIT)
- ▶ Andrew Hussie (Vriska Serket, Sweet Bro & Hella Jeff)
- ▶ mikeazo on Stack Overflow (AES diagram, CC BY-SA 3.0)
- ▶ Apple (WebKit logo, CC BY-SA 4.0)
- ▶ ArtyomK2707 (Intel i9-14900K, CC BY-SA 4.0)
- ▶ InfoSecDJ (microscopic fuses, CC BY-SA 4.0)
- ▶ Lisa Schulz (lock and key icons, CC BY-SA 4.0)