Secure Intermittent Computing with ARM TrustZone on the Cortex-M

Pietro Chiavassa
Filippo Gandino
Renato Ferrero
Jan Tobias Mühlberg

Index

- Background
- State of the Art
- Proposed Solution
- Implementation
- Future works

Background

Intermittent computing (ImC)

- Energy harvesting systems
  - Small capacitor
  - Charge/Operate/Die cycle
- Checkpointing
  - To ensure forward progress
  - Save volatile MCU state to non-volatile memory (NVM)

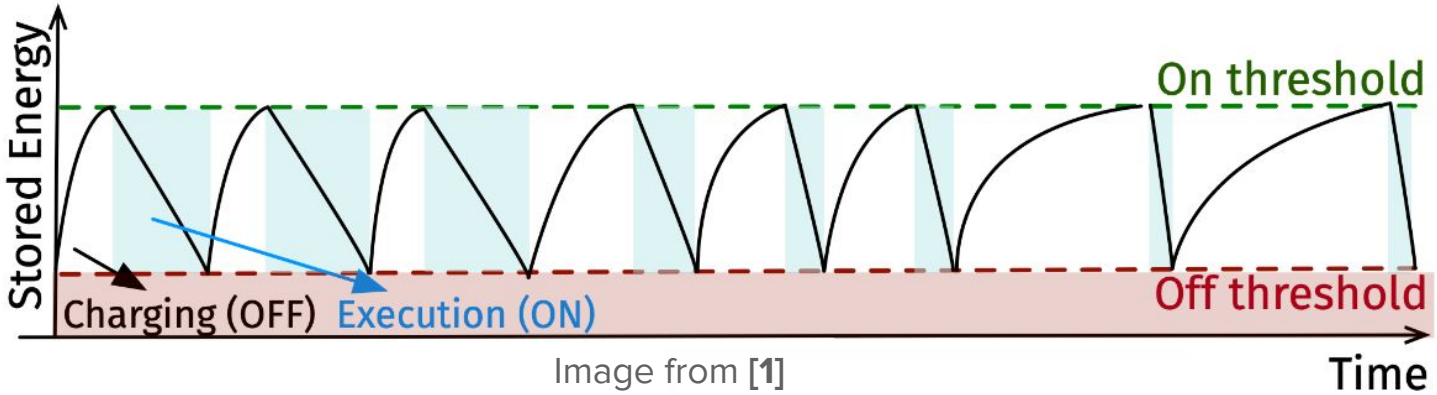


Image from [**1**]

Security issues[2]

- Checkpoint snooping
  - Secret keys
  - Intermediate state of crypto primitive
- Checkpoint spoofing
  - Change checkpoint content
  - Control device operations
- Checkpoint replay
  - Replace checkpoint with a previous one
  - Jump to any point in the software program

Arm Trustzone for Cortex-M

- Hardware isolation
  - Secure (S) / non secure (NS) world
  - Memory mapped division
  - Orthogonal to MPU (privilege level)
- Access rule
  - Secure code can access all memory
  - Non secure code only non-secure memory
- Flow
  - Boot in secure world
  - Perform security configurations
  - Switch to non secure
  - Calls to secure world for secure services

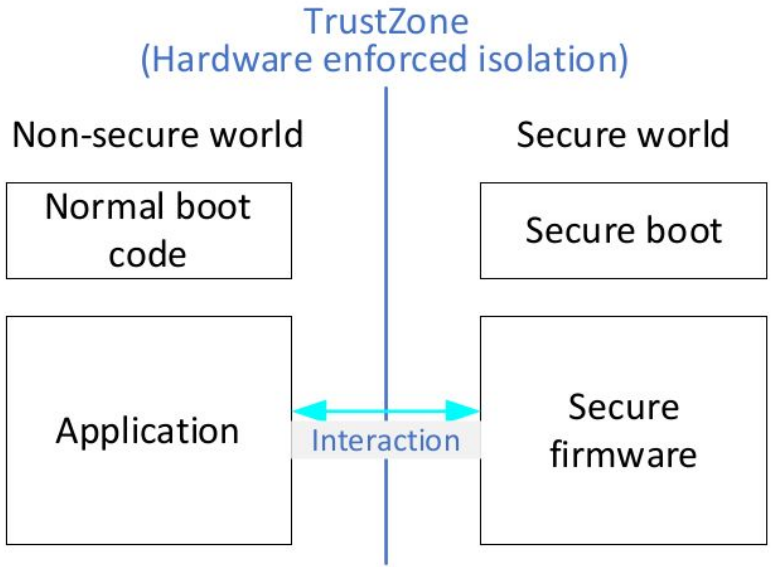


Image from **[3]**

State of the Art

On Securing Persistent State in Intermittent Computing[4]

- Checkpoint of non-secure world
  - Data copied in secure world
  - Stored in FLASH memory
  - No cryptographic protection of checkpoints
- Limitations
  - MCU state not checkpointed (only arbitrary data)
  - Device lifetime limited by FLASH wear

Secure Intermittent Computing Protocol (SICP)[5, 6]

- MSP430
  - Internal FRAM memory (non-volatile)
- Implementation
  - hardware attacks on internal memory
  - cryptographic protection of checkpoints
  - Secret key and nonce in Tamper-Free memory
- Limitations
  - Intellectual property (IP) encapsulation to simulate Tamper-Free memory
  - Full security chain is not discussed based on attacker capabilities
    - e.g. changes to firmware

Contribution

- Trustzone isolation
  - Protection against firmware attacks
  - Smaller TCB
- Target platform
  - Consider full security chain
  - Use only commercially available hardware
- Ensure device lifetime
  - FLASH wears out
- Checkpoint utility
  - Real implementation

Proposed solution

Target platform

- STM32U5 MCU
  - Cortex-M33 core
- Security features
  - ARM TrustZone
  - Random number generator
  - AES hardware accelerators
  - Readout protection (debug lock)
- External FRAM memory
  - For checkpoint storage
  - SPI connection

Image from **[7]**

Security objectives

- Information security
  - Integrity, authenticity, and confidentiality of checkpoints
- Freshness
  - To prevent checkpoint replay
- Atomicity
  - Possible power loss during checkpoint creation
  - System in **cannot** enter undefined state at startup
- Disruption of forward progress is **out-of-scope**

Attacker model

- Assumption
  - Checkpoint stored on external memory chip
- Attacker
  - Can read/write content in external memory
  - Can run arbitrary code in non-secure world
  - Can control device power (start - stop device)
- Limitations
  - Cannot read/write internal memory via hardware attacks
  - Cannot run arbitrary code in secure world
  - No side channel attacks (e.g. differential power analysis)

Security chain

- Root of trust
  - hardware key in system FLASH (internal)
- Firmware protection
  - lock debug features (RDP=2)
  - secure boot **(optional)**
- Checkpoint of secure world **(only)**
  - Handled by secure firmware
  - On external memory (FRAM)
  - Cryptographic protection of checkpoints
  - Checkpoint key (encrypted) and nonces in secure FLASH

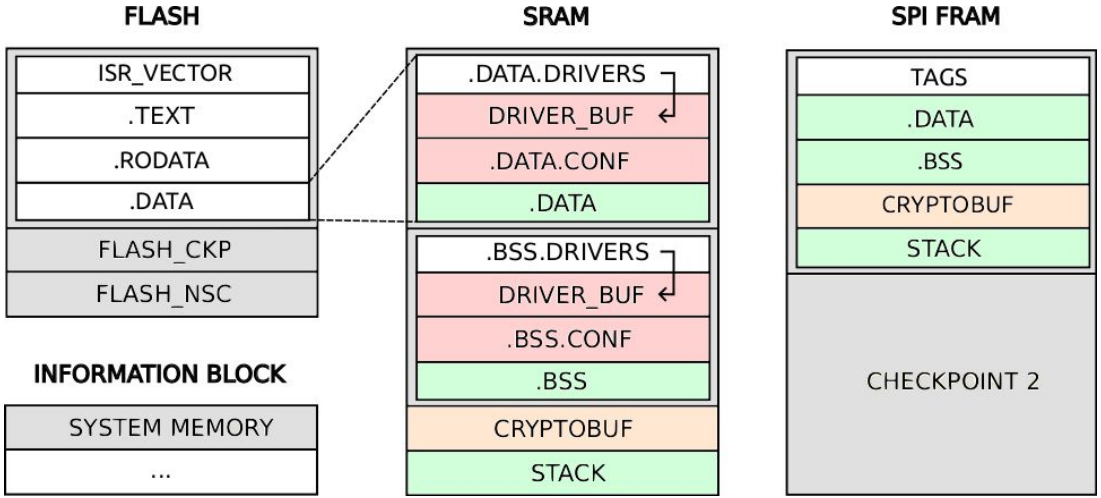
Implementation

Checkpoint utility

- Routines
  - SAVE and RESTORE
  - Similar to SICP
- Entered via system call (SVC)
  - Switch to privileged mode
  - Important CPU registers saved in stack
- SAVE
  - Creates checkpoint (stack + global variables)
- RESTORE
  - Checks and loads checkpoint in memory after reboot
  - Simulates exception exit from SAVE

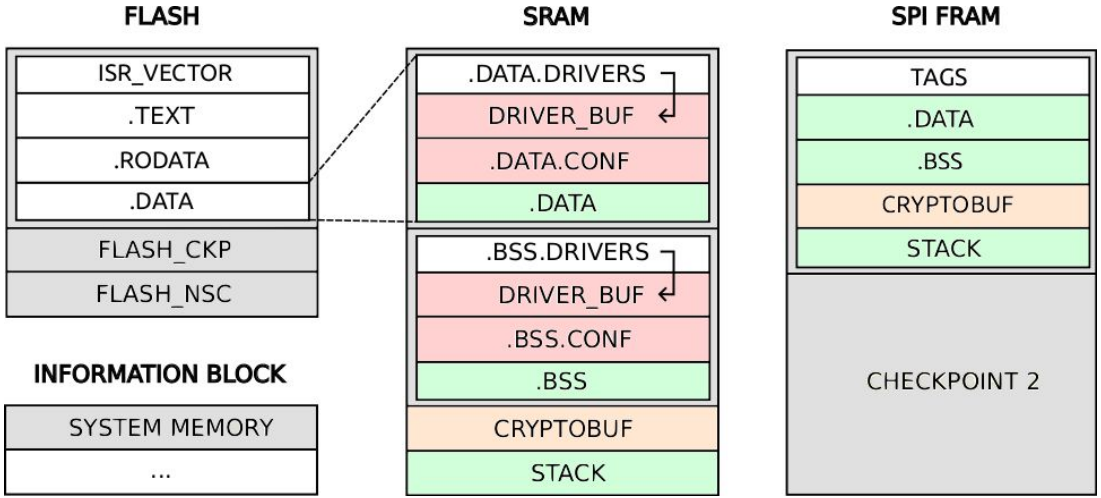
Custom memory layout

- FLASH_CKP
  - circular buffer to store nonces
  - erase operation expensive
  - also avoids FLASH wear
- .DATA divided into
  - .data (associated data)
  - .data.conf (plaintext)
  - .data.drivers
  - driver_buf stores copy of drivers (plaintext)
- same layout for .BSS
- CRYPTOBUF for ciphertext



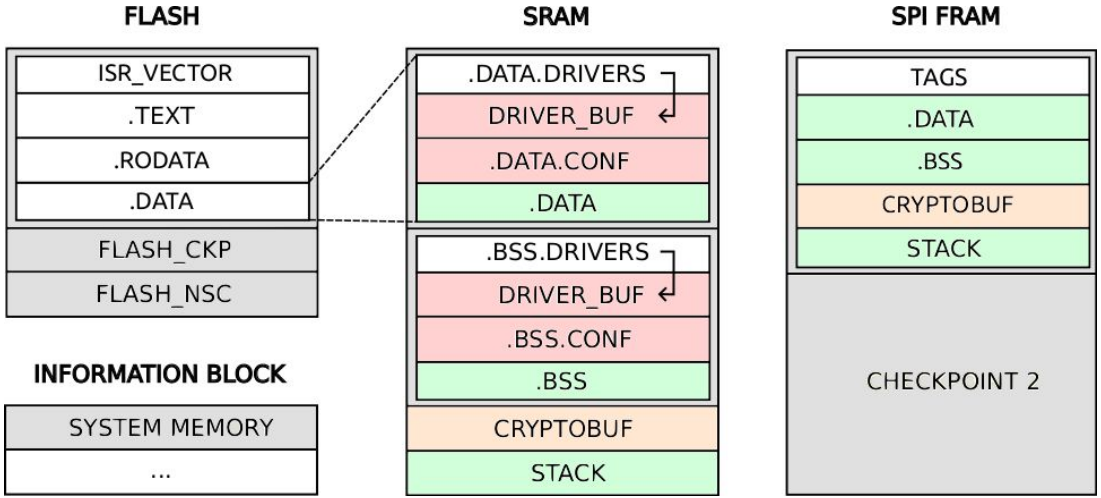
SAVE routine (checkpoint creation)

1. Drivers copied to buffers
   - Routine may change them
2. Random 96 bit value
   - Initialization vector (IV) for AES-GCM
3. AES-GCM (IV, ad, pt)
   - 3 calls with increasing IV
   - for all .data, .bss and stack
   - Stack only as ad
4. Result
   - Ciphertext in CRYPTOBUF
   - 3 authentication tags



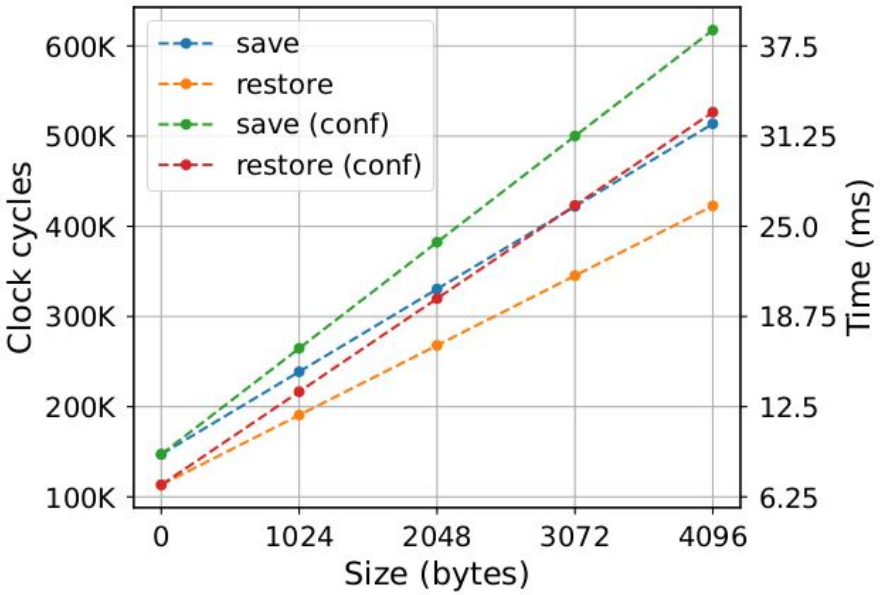
SAVE routine (checkpoint creation)

5. Save checkpoint to FRAM
   - 3 authentication tags
   - Associated data
   - CRYPTOBUF (ciphertext)
6. Save nonce in FLASH
   - IV (96 bit) + address of checkpoint in FRAM (32 bit)
   - This validates new checkpoint and invalidates previous one
7. Manage nonce buffer
   - Crear previous FLASH page when starting new one



Experiment 1

- Variable application size
  - Simulated with buffer in .bss
- Confidentiality
  - Either full confidential or non confidential
- Checkpoint overhead
  - Linear increase with application size
  - Encryption adds overhead
  - SAVE slower than RESTORE



Experiment 2

- Contribution of single operations
  - With fixed application size (2048 bytes)
  - Either full confidential or non-confidential
- Results
  - Main overhead from SPI R/W
  - SAVE slower due to memory programming and external memory operations
  - Slow driver copy (poorly optimized)

TABLE 1. BENCHMARK OF SAVE

Operation	Cycles	Time (ms)
Authenticated encryption	51777 (103745)	3,236 (6,484)
Copy drivers	11366	0,710
Write nonce	20409 (20408)	1,276
IV generation	451	0,028
Page erase*	274	0,017
SPI R/W	245883 (245884)	15,368
Others	275 (277)	0,017
Total	330435 (382405)	20,652 (23,9)

TABLE 2. BENCHMARK OF RESTORE

Operation	Cycles	Time (ms)
Authenticated encryption	52077 (104044)	3,255 (6,503)
Copy drivers	11325	0,708
Decrypt key	2697	0,169
Read nonce	586	0,037
Page erase*	274	0,017
SPI R/W	200677 (200675)	12,542
Others	378 (334)	0,024 (0,021)
Total	268014 (319935)	16,751 (19,996)

( ) means confidential results

* averaged over # of nonces in page

Conclusion

- SICP (save + restore)
  - No protection: < 1 ms (data in FRAM)
  - Protection: comparable to our results (44 ms)
- Energy consumption would be a better metric
  - We use higher clock cycles (160 MHz vs 8MHz)
  - But waste cycles waiting for peripherals (RNG, crypto unit, SPI)
- Device lifetime
  - **52 years** (nonces in FLASH, 1 checkpoint/s)

Future works

Future works

- Save non-secure world
  - requires design of secure service
- Saved state
  - Confidentiality for stack
  - Save Heap
  - Save Peripheral state
- Performance optimization
  - DMA to parallelize SPI operations
- Power analysis
  - Also in real use case

Thanks for your attention

References

- [1] *K. S. Yıldırım, “The today and future of intermittent computing for sustainable sensing,” https://community.arm.com/arm-research/m/resources/990.*
- [2] *S. Krishnan, A., Schaumont, P. (2018). Exploiting Security Vulnerabilities in Intermittent Computing. In: Chattopadhyay, A., Rebeiro, C., Yarom, Y. (eds) Security, Privacy, and Applied Cryptography Engineering. SPACE 2018. Lecture Notes in Computer Science(), vol 11348. Springer, Cham.* https://doi.org/10.1007/978-3-030-05072-6_7.
- [3] *Trusted firmware-m technical overview. https://www.trustedfirmware.org/docs/TrustedFirmware-MTechnicalOverviewQ1-2023.pdf.*
- [4] *Hafiz Areeb Asad, Erik Henricus Wouters, Naveed Anwar Bhatti, Luca Mottola, and Thiemo Voigt. 2020. On Securing Persistent State in Intermittent Computing. In Proceedings of the 8th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems (ENSsys '20). Association for Computing Machinery, New York, NY, USA, 8–14. https://doi.org/10.1145/3417308.3430267.*
- [5] *A. S. Krishnan, C. Suslowicz, D. Dinu and P. Schaumont, "Secure Intermittent Computing Protocol: Protecting State Across Power Loss," 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 2019, pp. 734-739, https://doi.org/10.23919/DATE.2019.8714997.*
- [6] *Archanaa S. Krishnan and Patrick Schaumont. 2022. Benchmarking and Configuring Security Levels in Intermittent Computing. ACM Trans. Embed. Comput. Syst. 21, 4, Article 36 (July 2022), 22 pages. https://doi.org/10.1145/3522748.*
- [7] *STMicroelectronics. B-U585I-IOT02A - Discovery kit for IoT node with STM32U5 series. https://www.st.com/en/evaluation-tools/b-u585i-iot02a.html.*