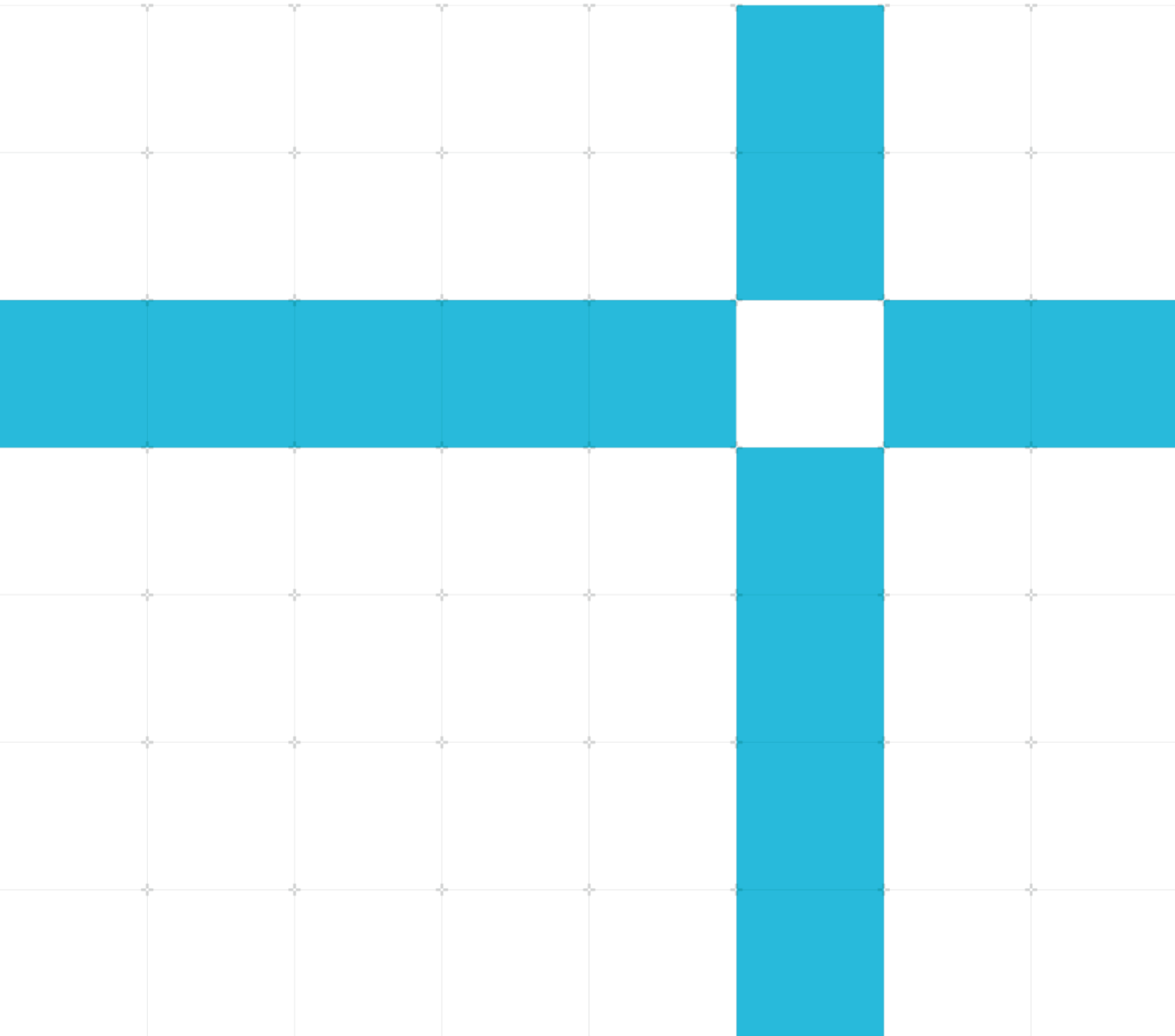
Architecture Security Advisory ASA

Version: 1.0

SRAM Aging Attacks

Non-Confidential



Architecture Security Advisory

SRAM Aging Attacks

Arm Non-Confidential Document Licence ("Licence")

This Licence is a legal agreement between you and Arm Limited ("Arm") for the use of Arm's intellectual property (including, without limitation, any copyright) embodied in the document accompanying this Licence ("Document"). Arm licenses its intellectual property in the Document to you on condition that you agree to the terms of this Licence. By using or copying the Document you indicate that you agree to be bound by the terms of this Licence.

"Subsidiary" means any company the majority of whose voting shares is now or hereafter owner or controlled, directly or indirectly, by you. A company shall be a Subsidiary only for the period during which such control exists.

This Document is NON-CONFIDENTIAL and any use by you and your Subsidiaries ("Licensee") is subject to the terms of this Licence between you and Arm.

Subject to the terms and conditions of this Licence, Arm hereby grants to Licensee under the intellectual property in the Document owned or controlled by Arm, a non-exclusive, non-transferable, non-sub-licensable, royalty-free, worldwide licence to:

(i) use and copy the Document for the purpose of designing and having designed products that comply with the Document;

(ii) manufacture and have manufactured products which have been created under the licence granted in (i) above; and

(iii) sell, supply and distribute products which have been created under the licence granted in (i) above.

Licensee hereby agrees that the licences granted above shall not extend to any portion or function of a product that is not itself compliant with part of the Document.

Except as expressly licensed above, Licensee acquires no right, title or interest in any Arm technology or any intellectual property embodied therein.

THE DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. Arm may make changes to the Document at any time and without notice. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, third party patents, copyrights, trade secrets, or other rights.

NOTWITHSTANING ANYTHING TO THE CONTRARY CONTAINED IN THIS LICENCE, TO THE FULLEST EXTENT PETMITTED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, IN CONTRACT, TORT OR OTHERWISE, IN CONNECTION WITH THE SUBJECT MATTER OF THIS LICENCE (INCLUDING WITHOUT LIMITATION) (I) LICENSEE'S USE OF THE

DOCUMENT; AND (II) THE IMPLEMENTATION OF THE DOCUMENT IN ANY PRODUCT CREATED BY LICENSEE UNDER THIS LICENCE). THE EXISTENCE OF MORE THAN ONE CLAIM OR SUIT WILL NOT ENLARGE OR EXTEND THE LIMIT. LICENSEE RELEASES ARM FROM ALL OBLIGATIONS, LIABILITY, CLAIMS OR DEMANDS IN EXCESS OF THIS LIMITATION.

This Licence shall remain in force until terminated by Licensee or by Arm. Without prejudice to any of its other rights, if Licensee is in breach of any of the terms and conditions of this Licence then Arm may terminate this Licence immediately upon giving written notice to Licensee. Licensee may terminate this Licence at any time. Upon termination of this Licence by Licensee or by Arm, Licensee shall stop using the Document and destroy all copies of the Document in its possession. Upon termination of this Licence, all terms shall survive except for the licence grants.

Any breach of this Licence by a Subsidiary shall entitle Arm to terminate this Licence as if you were the party in breach. Any termination of this Licence shall be effective in respect of all Subsidiaries. Any rights granted to any Subsidiary hereunder shall automatically terminate upon such Subsidiary ceasing to be a Subsidiary.

The Document consists solely of commercial items. Licensee shall be responsible for ensuring that any use, duplication or disclosure of the Document complies fully with any relevant export laws and regulations to assure that the Document or any portion thereof is not exported, directly or indirectly, in violation of such export laws.

This Licence may be translated into other languages for convenience, and Licensee agrees that if there is any conflict between the English version of this Licence and any translation, the terms of the English version of this Licence shall prevail.

The validity, construction and performance of this Licence shall be governed by English Law.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-21585 version 4.0

Web Address

http://www.arm.com

Contact

arm-security@arm.com

Contents

1 Introduction.

Arm is aware of a research paper, *UnTrustZone: Systematic Accelerated Aging to Expose On-chip Secrets*, that demonstrates physical attacks to recover remanent data in Static Random Access Memory (SRAM). The techniques employed in the research paper leverage data-remanence due to Bias Temperature Instability (BTI) to obtain information across privilege levels given time and unrestricted physical access to a target device.

Despite the name of the paper, BTI affects all digital components using SRAM to store internal state. SRAM is used to implement "fast memory" on devices across the semiconductor industry and the demonstration does not depend on Arm TrustZone. This threat is outside the scope of the TrustZone threat model but there may be products implementing TrustZone for which the threats described in the paper could be relevant.

Data-remanence in SRAMs due to BTI and other physical effects (e.g., electromigration) has been known for several decades [1] and are often collectively called "aging" effects. The research paper advances the state of the art by demonstrating a technique to both accelerate aging and recover data robustly from SoCs (Systems on Chip) and microcontrollers on the market today.

Data is recovered from uninitialized SRAMs, i.e., burned-in values can be read when an SRAM is powered on before any other values have been written. Data-remanence in this case can be eliminated by overwriting the affected memories at power-on with other data, and in security-relevant scenarios this is recommended. Incomplete erasure has enabled attacks both historical and recent [2, 3] and is defined as a *Common Weakness Enumeration* (CWE) maintained by MITRE [4].

1.1 Transistor Bias.

Field-Effect Transistors (FETs) that compose the logic gates and storage elements in digital electronics behave like switches: there is an *on* state, and an *off* state. When a FET is in an on or off state for an extended time it becomes *biased*, requiring slightly more or less voltage to begin a state transition than it did in the past. BTI is one of the causes of this bias. This has little to no impact on the function of modern commercial devices when they are used in their intended operating conditions but is a reliability concern as semiconductors age.

1.2 Data Retention in Static Random Access Memory.

SRAM cells are constructed from varying numbers of transistors in different topologies, and all types may be present in a single device. Despite this diversity, all SRAM types are constructed by cross-coupling sets of transistors such that when one set of transistors is on, the other is forced to be off, and vice versa. This is how an SRAM retains state through time [5].

Due to the bias effect in FETs, the longer that an SRAM cell holds a given single-bit value, the more likely it is for that SRAM cell to have a discernable bias. This bias persists even after

a device is powered off. When power is restored, the SRAM cell will default to the complement of its former state with increasing probability the longer the state was held. Bias occurs more rapidly at higher temperatures and voltages [6]. Once this bias occurs, it remains observable under nominal operating conditions when the SRAM is first powered on.

1.3 Threat Model.

The threat models used in the three scenarios described in the research paper all assume that an adversary has board-level debug access and can read the uninitialized power-on state of an SRAM that has previously held confidential data. Physical access was used to accelerate aging. Each of the three attacks described in the paper also have specific limitations.

2 Attacks.

The three attacks in the paper all follow the steps below.

Preparation (device off):

1. Modify the Printed Circuit Board (PCB) to allow an adversary to increase the supply voltage of the SRAM using an adversary-controlled power supply.

2. Increase the device's external temperature e.g., with a heating element or oven.

With the device running:

3. Secure information is moved into a target SRAM and the system is put into a state such that the information remains there.

4. Over-volt the SRAM and apply heat over several hours or days to accelerate aging.

Power off and restore the device to a nominal operating environment:

5. Extract uninitialized power-on state from the target SRAM.

6. Apply an error correction technique to recover data from the uninitialized state.

2.1 Exfiltrating Secrets from TrustZone Secure World.

In this scenario, an adversary runs non-secure software that calls secure cryptographic functions from the Secure World through the Non-Secure Callable (NSC) interface. Secure World software moves cryptographic secrets into SRAM for rapid access during execution. Then the adversary stresses the device for several hours before overwriting the boot firmware. Changing the tested devices' boot firmware allows the adversary to obtain debug access. Immediately after a reboot, the adversary can dump the uninitialized contents of SRAM where the secret key was stored. An adversary then does an exhaustive search through all keys within a fixed Hamming distance.

2.1.1 Limitations.

This attack assumes that erasing a device's boot firmware allows uninitialized power-on state to be read from target SRAM regions on power up, in this case via debug privilege escalation, and that an adversary can trigger cryptographic NSC functions with chosen plaintext to detect when the recovered secret is correct. Pre-stress profile data reduces error but is not required.

2.2 Exfiltrating Proprietary Firmware from a TrustZone Enabled Device.

In this scenario, firmware is loaded from flash regions marked as *Secure* into SRAM prior to execution. During execution, the firmware remains in SRAM while the adversary stresses the device for 24 hours. Then the adversary erases the device and dumps the contents of the SRAM containing firmware for analysis.

2.2.1 Limitations.

As in exfiltrating secrets, this attack assumes that a device's boot firmware allows uninitialized power-on state to be read from SRAM via debug privilege escalation. An adversary must also be able to move firmware into SRAM. Note that many microcontrollers execute firmware directly from flash memory and are not vulnerable.

2.3 Exfiltrating Secrets from Cache.

In many Arm processors, the data values in cache lines are unchanged when the cache line is invalidated. Data values in cache lines are updated when overwritten by new data. For debugging purposes, Arm processors often have an implementation defined method to allow direct access to internal memory regardless of validity [7]. This method is only enabled at higher privilege.

The data an adversary wishes to extract is read, or forced to be read, into a cache and the device is stressed without disturbing those memories. After a period of continuous stress, the adversary erases and reprograms the device with a custom boot stage that enables user software to execute in the most privileged mode. The adversary executes privileged code to read the uninitialized power-on state of the target cache.

2.3.1 Limitations.

An adversary must be able to move or influence the victim to move the desired data into a cache and prevent its eviction during stress. They must also be able to read uninitialized power-on state from this buffer.

3 Are Arm processors affected by SRAM aging attacks?

Most semiconductors use SRAM for fast memories and may use SRAM for on-chip main memory as well. Because BTI stems from physical properties of the materials used in FETs, retention affects all devices containing SRAM memories, including devices that contain Arm IP. Even if an Arm processor or component does not contain SRAM, it is likely that SRAM is used elsewhere in the device. These SRAM memories may be vulnerable to this and similar threats.

While physical attacks are outside the intended scope of many processor security features such as TrustZone, logical separation of privilege, and virtualization, products that use these features may consider these threats relevant. SRAM aging attacks are particularly relevant when secrets that are valuable to a physical adversary are repeated across many devices, and when these devices also permit long-term physical access. Devices that never allow access to uninitialized power-on SRAM state are not vulnerable.

4 Recommendations.

4.1 Mitigations.

To reduce the likelihood of success of this or similar attacks (e.g., cold-boot attacks on SRAM) at least one of the following mitigations should be implemented. If neither are implemented, then a device is vulnerable to SRAM data retention attacks.

- Devices should prevent all access to uninitialized SRAM state when those SRAM regions have ever been used to store secure or privileged data. Debug and test ports and functionality are a common attack vector and access should be restricted either through secure authentication and/or physical means (e.g., blown fuses) that are not easily reversable by an adversary.

- SRAM and internal caches should be zeroed, not just invalidated, on restart and power transition events where uninitialized SRAM state may be exposed. If a dedicated hardware component or instruction is not available, then firmware may zero memories and memory-mapped registers. In this case, it should not be possible to disable or erase this firmware once it is enabled.

To counter stress-induced silicon aging, a device may include voltage and thermal sensors to detect conditions described in the paper and respond with mechanisms to zero affected SRAMs. While this will counter attempts to accelerate SRAM aging it will not prevent data retention due to natural silicon aging.

4.2 Hardware Root of Trust.

Many modern devices include a *Hardware Root of Trust* (HRoT) with dedicated resources to isolate trusted computations from the rest of the system and manage secure system operations such as secure boot. In many systems this will be the only defense against data remanence threats. These resources should remain inaccessible to the rest of the system, including debug and test channels such as JTAG, memory test, and scan-chain, even if these system operations can be deactivated or bypassed.

The HRoT should initialize SRAMs and supply firmware to the CPU such that memories and caches that are potential targets are overwritten with valid non-secret data prior to allowing debug access or any user-provided firmware to execute on the CPU. The HRoT should also verify the integrity of CPU firmware to prevent hijacking the boot process or forcing debug-specific boot flows as was done in the paper to exfiltrate data from cache.

Some recent Arm-based systems contain a *Runtime Security Engine* (RSE) that implements the required isolated resources and preserves the integrity of the initial boot stages when used correctly. The RSE serves as the HRoT in such systems.

5 Conclusion.

SRAM aging attacks may or may not be critical to a product depending on that product's environment and the information it contains. The countermeasures should match the risk of loss for both product manufacturers and end-users. The following examples illustrate products and relevance of the threat: 1) An IoT device where the only secret stored in SRAM is a Wi-Fi password. In this case the threat is likely not relevant. An adversary with physical access to the device inside a home can probably access the network more simply by other means. 2) A subscription TV smart card. This threat is likely very relevant as these cards may contain keys that are shared across a large number of devices and are in the physical possession of potential adversaries.

SRAM aging attacks fall outside the scope of many architectural security features like TrustZone because the weakness is inherent to SRAM memories, and the logical isolation provided by the architecture is not intended to cover these physical effects. Systems may be vulnerable to these threats depending on their implementation of mitigations and countermeasures within the system as a whole.

Due to the complexity and invasiveness required, the likelihood of this attack is considered low relative to other attacks. Future iterations of this technique could evolve and progressively remove certain limitations to make it more applicable to other threat scenarios. In that case, further evaluation of security critical systems should be considered.

6 References.

1. “Data Remanence in Semiconductor Devices”. Gutmann. 2001

2. “Low Cost Attacks on Tamper Resistant Devices”. Anderson et al. 1997

3. “Beware of Discarding Used SRAMs: Information is Stored Permanently”. Hovanes et al. 2022

4. “CWE-1330: Remanent Data Readable after Memory Erase – MITRE Common Weakness Enumeration”. https://cwe.mitre.org/data/definitions/1330.html

5. *Robust SRAM Designs and Analysis*. Singh et al. Springer 2012

6. “Bias Temperature Instability analysis of FinFET based SRAM cells”. Kahn et al. 2014

7. *Arm Cortex-A75 Core Technical Reference Manual r3p1*
https://developer.arm.com/documentation/100403/0301/functional-description/level-1-memory-system/direct-access-to-internal-memory