Aging-Induced Long-Term Data Remanence in SRAM Cells

by

Joshua Hovanes

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 5, 2023

Keywords: SRAM, volatile memories, aging, data retrieval, IP theft, PUF

Approved by

Ujjwal Guin, Chair, Assistant Professor of Electrical and Computer Engineering
Masoud Mahjouri-Samani, Assistant Professor of Electrical and Computer Engineering
Mehdi Sadi, Assistant Professor of Electrical and Computer Engineering

Abstract

Within the electronics industry, data recovery has been a primary focus of security experts and researches for decades. The vast majority of this research has been performed and realized in the form of hard disk recovery, covering various types of non-volatile memories. In contrast, almost none of this research has considered data recovery from volatile memories. This is because one of volatile memory's innate properties is that the data is lost upon loss of power, leading to the wide assumption that recovering data is impossible. This research discusses the inaccuracies of this assumption, and presents an approach to recovering data from static random access memory, commonly known as SRAM. This conventional wisdom leads to designers not being required to protect sensitive information, such as firmware or secret encryption keys, when the system is retired. Unfortunately, the recycling of these parts means that either intellectual property or security could be compromised, should the data be successfully reconstructed. This thesis presents a novel concept to retrieve previously stored SRAM data, as the aging of SRAM leads to a power-up state with an imprint of the stored values. It then shows that the proposed approaches can partially recover the original SRAM content.

The accuracy and volume of recovered data can be further increased by incorporating multiple chips instead of a single one, as it might be impossible to retrieve data from some cells. Some stable cells in each chip will be moved further towards stability, making a change impossible to detect. However, chip-to-chip process variation means that the change might be detectable in a different, identical chip. After analyzing the power-up states of all cells in all chips, a majority voting algorithm is used to combine the set into the recovered data. This paper presents the experimental results using off-the-shelf SRAM chips, loaded with a binary image and subjected to accelerated aging. It demonstrates the partial recovery of data on SRAM chips that have been aged for as little as four hours at 85°C.

Acknowledgments

I would like to thank my advisor, Dr. Ujjwal Guin, for his support, ideas, and assistance during my time as a graduate student at Auburn. His contributions to my research were very important to my success in both my publications and this thesis.

I would like to extend my gratitude to the committee members of my thesis: Dr. Masoud Mahjouri-Samani and Dr. Mehdi Sadi, for their interest and taking their time to support and acknowledge my work.

I would also like to thank my colleague, Yadi Zhong, for assisting and teaching me a great amount in areas I was unfamiliar with during my time working in Dr. Guin's lab.

My sincere thanks to all of my labmates for making the research experience more enjoyable and rewarding. Your help ranging from brainstorming new ideas to simply assisting with menial tasks when I am busy means very much to me.

Finally, I would like to express my gratitude to my parents for continually uplifting me through my journey through graduate school at Auburn, without your support, I could not have made it this far.

Table of Contents

List of Figures

List of Abbreviations

DRAM	Dynamic Random Access Memory
HCI	Hot Carrier Injection
IC	Integrated Circuit
IP	Intellectual Property
NBTI	Negative Bias Temperature Instability
NVM	Non-Volatile Memory
OEM	Original Equipment Manufacturer
PRNG	Pseudo Random Number Generator
PUF	Physically Unclonable Function
SRAM	Static Random Access Memory
TRNG	True Random Number Generator

Chapter 1

Introduction

In today's interconnected, automated world, integrated circuits (ICs) play a vital role in every part of life. People heavily rely on them for communication and daily tasks, and they form the core of nearly all industries, such as commercial, transportation, industrial, and defense. It is of paramount importance for a nation to maintain technological superiority, especially in semiconductors, over any potential adversaries [4–10]. However, the increasing cost and complexity of semiconductor manufacturing have led to increased centralization and globalization in the electronics supply chain. This has led to the fabrication, testing, assembly, and packaging of many ICs being performed offshore. The Bureau of Industry and Security has reported that over 43% of electronic product assembly is not performed by the original equipment manufacturer (OEM). Instead, it is outsourced to third-party contractors [11] due to superior equipment, lower costs, increased flexibility, proximity to other manufacturers, and potential subsidies [12]. The majority of electronics are now manufactured and assembled in environments with limited trust, government oversight, or visibility, posing a serious threat to protecting intellectual properties (IPs), with much research focused on attempting to re-establish this trust [10, 13, 14]. IP theft, where an adversary obtains an IP illegally, can pose many threats to many parties, ranging from the designer themselves to the consumer, who are using or benefiting from ICs. These threats include IC overproduction, where a manufacturer undercuts a designer by producing extra parts and selling them directly into the supply chain, IP piracy, where IP is distributed, sold, or used without approval from the designers, and design tampering, where chips are made with a design modified by the manufacturer. This has made IP protection a key focus of researchers from

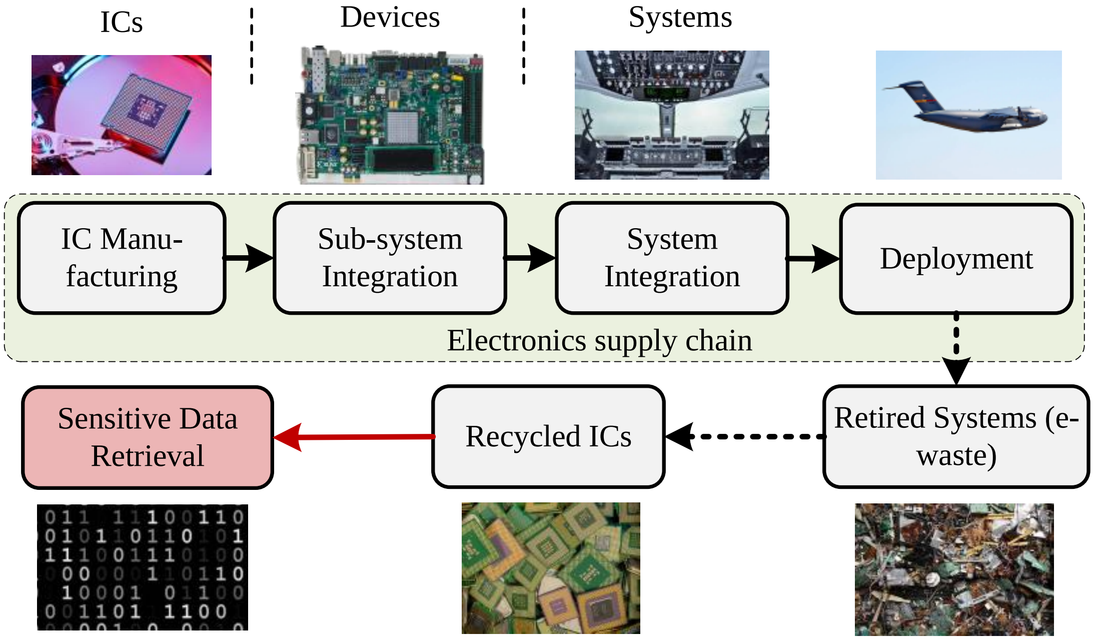


Figure 1.1: The globalized electronics supply chain with threats.

academia, industry, and the government, who have produced a number of different solutions for IP protection [14–18].

In contrast to the IP theft issues at the design stage, the issue of IC recycling poses challenges for end-of-life (EOL) parts. The recycling of ICs from e-waste has challenged researchers to accurately detect parts of inferior quality and quantity [19–21]. In the recycling process, electronic components are taken off from discarded printed circuit boards (PCBs), and resold as new. Typically, these parts are still functional, as the operation life of a chip is generally much larger than its actual time of usage [19, 21]. However, these parts still present large threats when deployed in critical infrastructures. These threats have been widely explored by researchers as the parts may have various defects, fail prematurely, or fail to deliver the necessary performance for the relevant application [21–24]. These threats make recycled IC detection a high priority among researchers, with many different solutions have been proposed [25–28]. These used ICs can also hold sensitive information related to the IP (e.g., the firmware/software, encryption keys, etc.) which may subsequently be exposed to an adversary [29]. Figure 1.1 describes the overall electronics supply chain from IC manufacturing, sub-system/device integration, system integration, and deployment. After the deployment stage, when the ICs are retired and discarded, the adversary recycles the functional parts and may be able to recover previously-used data/programs from the static random access memories (SRAMs). This type of recovery is possible any amount of time after the chip has been retired, as it is a permanent imprint on the chip.

1.1 Motivation

Throughout the last few decades, it has been well established that data recovery from non-volatile memories (NVMs) is possible, and an entire industry has risen from recovering data from broken or erased drives. However, there has also been a concurrent common understanding that all data disappears from volatile memories as soon as they are powered off. Data sanitization for NVMs is well documented as these memories contain the firmware, software, and data for a typical computing system [30]. NIST SP 800-88 Rev. 1 Guidelines for Media Sanitization [31] provides different permanent data erasure processes for NVMs, e.g., hard disk drives (HDD), solid-state drives (SSD), CDs, flash memory, memory sticks, etc. Unfortunately, there are no standard aging-related sanitization processes for volatile memories. SRAMs, a commonly used volatile memory, have a wide range of applications and are prevalent in electronic devices. They can be used as cache memories for processors or block RAMs for FPGAs, which hold sensitive data or programs once they are powered on. As it is a volatile memory, the most common perception is that it lost all its data after powering off [32]. NSA/CSS Storage Device Sanitization manual mentions that the SRAM sanitization is instantaneous after the removal of power supply [32].

Thousands of SRAMs from critical systems exist in e-waste which have never been subjected to any proper sanitation. These are then subject to the IC recycling process, and may then be procured by an adversary. These adversaries can identify the chip and analyze its power-up values. By acquiring the initial data from the untrusted, offshore foundries, the adversary can then attempt to reconstruct the information that was stored on the SRAM.

It is important to study and acknowledge this possibility, as to articulate the potential for secret information to be revealed and develop countermeasures that can improve security and keep critical systems safe. To the best of our knowledge, this research is the first attempt to investigate data retrieval from aged SRAM chips under normal operations.

1.2 Contributions

In this thesis, we present a novel approach for data recovery in SRAM which has previously contained fixed data in fixed locations, such as IPs, firmware, or encryption keys [29]. This is possible due to the effect of negative-bias temperature instability (NBTI) on PMOS transistors within each SRAM cell. This effect causes a change in the power-up state of each cell, biasing it towards the opposite value which is stored. There are three core contributions in the thesis, building upon each other to form the full methodology. These contributions are summarized as follows:

- Firstly, we propose the overall data recovery hypothesis within the SRAM cell. The power-up state of an SRAM cell is based on the difference in threshold voltage (V_{th}) among the two PMOS transistors. When information is stored in the cell, the V_{th} for one PMOS, not both, increases. With this change, the V_{th} difference between the two PMOS transistors deviates from the original difference induced by the process variation, either increasing or decreasing. This can be reflected in a change in power-up state behavior. Our analysis shows that, in each cell, the power-up state is biased toward the complementary value of the stored information.

- Secondly, we present a method to recover the data from the individual SRAM chips by taking advantage of the previous information. Firstly we analyze the initial power-up (IPU) state of a chip before any aging has occurred. This power-up state is represented as a large array of percentage values, representing the likelihood for a cell to initialize to a 1 or 0 value. Then, after data has been stored and accelerated aging has been performed, we record the final power-up (FPU) state of the chip. These are then compared to identify the change in V_{th} difference and consequently to determine the previously written information.

- Thirdly, we present a method to combine the results of multiple chips to drastically improve the recovered data. Within a single chip, there are many cells which have a large V_{th} difference as a result of process variation, which might be too large to overcome with aging. Additionally, there might be cells which begin with a stable power-up state, and the information biases them to become more stable. As information is derived from changes between IPU and FPU states, these cases result in cells which cannot provide evidence of their original contents. To overcome this, we can use multiple chips which have been used in the same system and subjected to the same data. As V_{th} differences due to process variation vary from chip to chip, a cell which is not recoverable in one may be recoverable in another. We demonstrate the effectiveness of this approach using six commercial off-the-shelf SRAM chips aged with an image. coded in binary.

1.3 Organization of the thesis

The rest of the thesis is organized as follows: a description of SRAM, power-up states, and physically unclonable functions is provided in Chapter 2. We describe the proposed method in Chapter 3. We go over the experimental results using a commercial SRAM chip in Chapter 4. Finally, we provide the future possibilities in this research area and conclude the thesis in Chapter 5.

Chapter 2

Background and Related Work

In this chapter, we discuss the fundamentals of SRAM, power-up states, aging, and other concepts fundamental to the methodology described in Chapter 3. We also discuss the relevant research which has been performed in the areas which contribute to our concept.

2.1 Static Random Access Memory

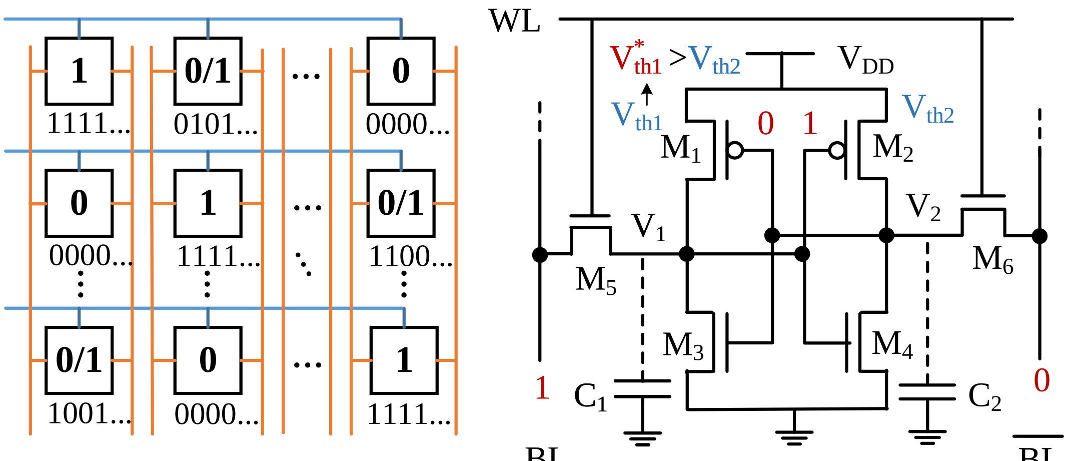
Static random access memory (SRAM) has been in use in computing systems for decades, with its invention dating back to 1963. It can be found in nearly any modern electronic device, typically acting as buffer memory between the processor and dynamic RAM (DRAM). In modern systems, 6T SRAM cells are used, with the namesake six transistors being shown in Figure 2.1(b). The transistors M_1-M_4 make up two inverters, whose outputs are connected to the others' inputs. These inverters make up a bistable latch, with one side reading the opposite value of the other. The values are accessed through transistors M_5 and M_6. These access transistors are turned on by the wordline (WL), and control the ability of the bitlines (BL and $\overline{BL}$) to read or write the internal values.

The static nature of this memory makes it extremely fast to access. However, the number of interconnects and transistors is much greater than its slower alternative DRAM, which uses just one transistor and capacitor per cell. This increase in space increases the utilization of chip area, making it drastically more expensive to implement. In most consumer applications, this leads to giving SRAM the primary role of being a buffer memory on a processor to communicate with the cheaper DRAM. However, in systems where space is not a concern, large quantities of memory are unnecessary, or speed requirements greatly outweigh costs, SRAM can function as the primary RAM for the device. This can range from specialized systems with dedicated SRAM to inexpensive microcontrollers such as the Raspberry Pi Pico [33].

2.1.1 SRAM Power-Up State

Upon a device powerup, all SRAM cells within are initialized to a discrete logic value. In the ideal case, SRAM cells are designed to be equally likely to initialize to either logic 0 or logic 1. However, process variation during the manufacturing process makes this ideal case impossible. These initialization values are unique to each cell, determined by the V_{th} of the PMOS transistors (M_1 and M_2). The transistor with the lower V_{th} determines the power-up state of the chip, meaning that any small mismatch as a result of process variation throws off the balance. For example, if the V_{th} of M_1 exceeds that of M_2, the initialized bit will be logic 0, and vice versa.

This behavior is owed to the ramping of the voltage within the chip, and is visualized in Figure 2.2. This diagram assumes the V_{th} of transistor M_2 is less than that of M_1. Before device powerup, at time $t = t_0$ both inverters are not functioning and both sides of the bistable latch contain logic 0. At this time, M_1 and M_2 are both off, as their V_{gs} is 0. From $t = 500$ to $t = t_1$, the VDD is increasing, and V_1 and V_2 are increasing along with it. However, as the threshold voltage of M_2 is lower, its V_{gs} is closer to its V_{th}, and V_2 increases marginally faster. Between $t = t_1$ and $t = t_2$, the lower threshold voltage of M_2 has enabled V_2 to rise above V_1, allowing it to enter saturation first. This in turn causes V_2 to continue to climb, which continues to force off M_1, decreasing V_1. After $t = t_2$, V_2 enters linear operation and continues to climb to VDD. V_1 reaches and remains at 0. Note that in Figure 2.1(b), V_1 is connected to BL, giving us the expected output of logic 0.



(a) SRAM power-up state

(b) SRAM cell aging

Figure 2.1: An abstract view of an SRAM array and an SRAM cell. (a) SRAM power-up state for creating unique ID [1, 2] and recycled IC detection [3], (b) Effect of aging on a SRAM cell.

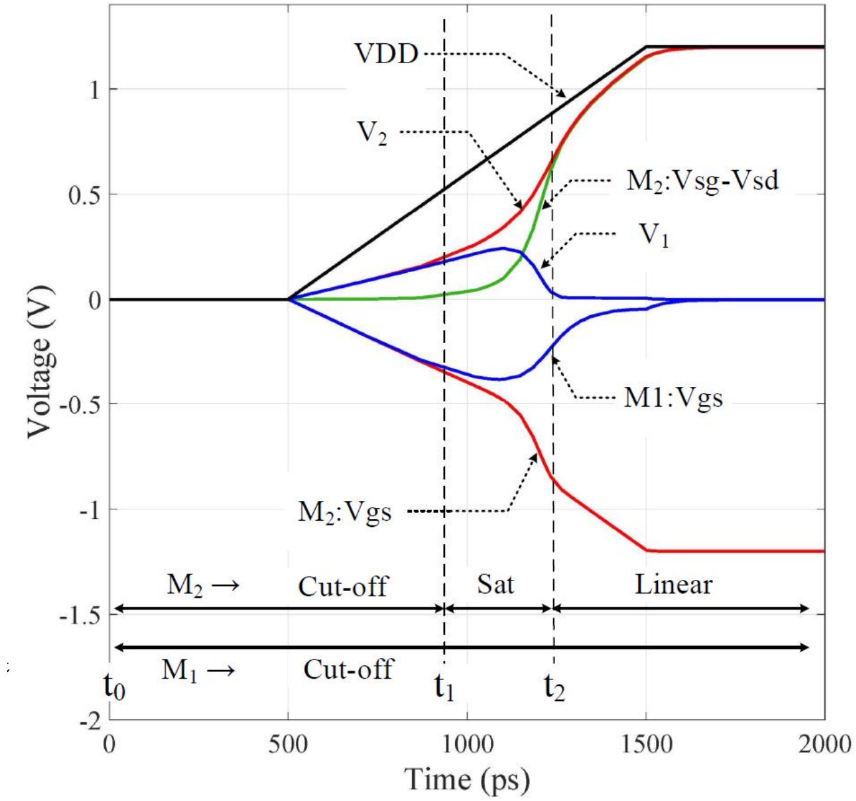


Figure 2.2: Timing diagram of SRAM initialization. [3]

2.1.2 SRAM PUFs and TRNGs

SRAM power-up behavior have allowed researchers to develop various physically unclonable functions (PUFs) and true random number generators (TRNGs) using SRAM architecture [1, 2, 34–40]. SRAM cells are designed to be perfectly symmetrical to achieve the maximum static noise margin. However, process variation makes all cells biased towards one inverter, which results in a consistent value across multiple power-ups. As a result of this bias in combination with external noise, three types of power-up values can be observed. These consist of cells that are: stable at logic 1 (S_1), e.g. cells with '111...' power-up states in Figure 2.1(a); stable at logic 0 (S_0), e.g. cells with '000...' in Figure 2.1(a); and unstable cells e.g. cells with '010...' in Figure 2.1(a). Stable cells are present in the case that the difference in V_{th} between M_1 and M_2 is greater than the external thermal noise, while unstable cells represent that the difference is small enough to still be overcome by outside noise.

PUFs are well-known and researched hardware security primitive that has many applications. A PUF is defined as a function which, while knowing the architecture and having physical access to the function, cannot be modeled or recreated on identical architectures. These are generally created through abusing the uncontrollable and immeasurable process variation of transistors. PUF unclonability allows them to be utilized as signatures from a device, allowing for authentication of hardware authenticity or identity [41–43]. Over the past decades, researchers have identified several different architectures for PUFs, such as the arbiter PUF or ring oscillator PUF (ROPUF) [44–46]. These PUFs, while powerful, cannot be applied in most integrated circuits, as they require special architecture within the chip for their specific use. However, SRAM is present in nearly every modern, and PUF, and a PUF can be realized from SRAM behavior when the stable bits are extracted [1,2,35,36,47]. This can create a unique, unclonable ID, as these stable cells vary in location and have unique biases from chip to chip due to process variation. In the case of SRAM PUFs, it is often required to incorporate error correction mechanisms [39, 40, 47, 48] so that the ID remains stable as the chip ages, as some of the internal biases, and thus the initialization values, may change as a result of aging. These chips may also be subjected to higher levels of external noise, causing previously stable cells to become unstable in extreme environmental conditions.

TRNGs are a powerful cryptographic primitive used for generating random values for use in cryptographic keys or other seeds [49, 50]. TRNGs differ from the similarly named, frequently used primitive pseudo RNGs (PRNGs) in a way that makes them far more reliable for secret key generation. PRNGs rely on an extremely obtuse algorithm to procedurally generate numbers which may seem random when analyzed in isolation, but are actually based on previous outputs. This predictability makes their use in cryptographic key generation less desirable. TRNGs, however, use outside, unpredictable, and usually immeasurable inputs, such as thermal noise, to generate random numbers. TRNGs can be created in an SRAM architecture by isolating the unstable bits in the array [35, 36]. The outcome of these bits is more dependent on external thermal and power supply noise than the internal biases of the cell.

2.1.3 SRAM Cell Aging

As transistors are used, their behavior changes and degrades over time due to various effects. This degradation extends to multiple facets of the transistor, but, in the case of SRAM cells, the most important factor is the change in the PMOS threshold voltage. These changes are what influence the power-up state, and significantly affect the reliability of SRAM PUFs and TRNGs. There are a number of effects which change the threshold voltage over long periods of time. These include hot carrier injection (HCI) [51, 52] and bias temperature instability (BTI) [53–56]. The most significant effect is a subset of BTI, negative bias temperature instability (NBTI).

The effects of NBTI have been widely studied, as it has presented itself as a serious reliability concern as transistors have decreased in size [53–55, 57]. NBTI describes positive charges getting stuck over time in the gate oxide. NBTI affects multiple parameters of a PMOS transistor, such as drain current, transconductance, and, most importantly, threshold voltage. Threshold voltage is affected as, when positive charges become embedded in the oxide, the amount of voltage required for V_{gs} must not only surpass the innate threshold voltage, but also the positive voltage provided by the extra positive charges. NBTI takes root when the V_g of a transistor is 0, and $V_g = V_s = VDD$. In the context of the SRAM cell in Figure 2.1 with stored logic 1, this means that the M_1 transistor will be subjected to NBTI. This raises the threshold voltage over a long period of time, leading to a change in bias of the power-up state described in Section 2.1.1. This phenomenon is visually described in Figure 2.1(b). The red numbers represent data which has been stored for long periods of time within the cell, and the V_{th1}, which is originally equal to V_{th2} increases to V^*_{th1}.

2.2 SRAM for detecting recycled ICs

As described in Section 1.1, IC recycling has become a prominent threat to the supply chain and many critical infrastructures. However, we can take advantage of the used nature of these devices in conjunction with the previously described behaviors to identify recycled SRAM. Guin et al. recently showed that the power-up state of an SRAM cell depends on the threshold voltages of the PMOS transistors, and the power-up state depends on the content that an SRAM cell is aged with, also described in Section 2.1.1 [3]. Over a large number of cells, the power-up states should balance to an equal distribution of $1s$ and $0s$. This is due to the fact that, even while an individual cell can be profoundly affected by process variation, it by nature is Gaussian over a large sample. As SRAMs are subjected to programs, which contain substantially more $0s$ than $1s$, the power-up state distribution will be noticeably skewed towards containing more $1s$ than $0s$, due to the NBTI-related aging described in Section 2.1.3. These chips can subsequently be identified as counterfeit parts as a result of their SRAM biasing, and safely removed from the supply chain. This method expands beyond identifying just standalone SRAM, as most modern systems contain SRAM which can be checked with this method.

2.3 Cold Boot Attack

While we believe our work to be the first research into the recovery of data from SRAM over a long period, there has been previous research into short-term data remanence in volatile memories [58, 59]. These methods have been frequently known as cold boot attacks, as lower temperatures increase the amount of time during which the values can be accurately determined. In these methods, the contents of RAM slowly dissipate after power-off, and can be extracted up until the point that the values completely dissipate. Skorobogatov determined that many SRAM chips can be read from minutes to hours after their power-off at -50°C [58]. However, while significant, this type of data remanence cannot be compared to our method, as our method involves permanent imprints on the part, rather than the slowly dissipating charge on the internal capacitors. Furthermore, our method does not rely on any environmental factors after the chip is decommissioned, as the parts are recycled from discarded e-waste.

Chapter 3

Aging Influenced SRAM Data Retrieval Approach

This section presents our novel methodology to reverse engineer the data originally stored within a used SRAM chip. As the aging increases the PMOS threshold voltage, the power-up state of each SRAM cell shifts towards the opposite of its stored information. An adversary with access to both the initial state and the chip may be able to retrieve the previously stored sensitive data by reading the power-up state of the recycled SRAM chips. We propose two primary approaches to partially recover sensitive information that is statically stored in SRAMs. The first approach targets data retrieval for individual SRAM chips. The second approach uses the first approach on multiple chips, and uses a majority voting system to merge the collective findings and improve both the quantity and the quality and accuracy of the information retrieved. This section begins with the establishment of the threat model.

3.1 Threat model

This threat model assumes that an adversary has very long-term malicious intentions [8, 9] to retrieve and exploit sensitive information from critical applications. This threat model describes the capabilities and available resources to the adversary for sensitive data retrieval from a recycled chip with on-chip SRAMs or commercial off-the-shelf chips. Additionally, this threat model makes clear the assumptions of the security of the supply chain.

- *Untrusted manufacturing:* The globalization of the electronics supply chain has led to unsupervised, offshore production of ICs and other electronic devices. We consider these manufacturers as untrusted entities as these parts are manufactured with limited trust and oversight.

- *Recycling of e-waste:* E-waste is often sent to offshore locations, and an adversary might be able to recover the e-waste themselves or procure recycled devices. These used parts could be used to read the SRAM power-up states as they are still functional.

- *Access to initial power-up state information of SRAM:* The adversary has access to the initial power-up (IPU) state for the SRAM arrays which are meant for use in critical systems. As manufacturing, assembly, test, and packaging are performed at offshore, untrusted facilities, the initial power-up data of a new chip may be provided by these institutions or collected before the chip is returned to the original designer.

- *SRAM Usage:* It is understood that the secret information, such as the program firmware or keys, resides within the same locations in the SRAM array throughout the life of the chip. As a result, these cells are aged with consistent information throughout their lifetime.

- *Data Sanitization:* Once the device with SRAM is retired from normal operation, it is discarded with the assumption that, as a volatile memory, SRAM does not need to be subjected to any special sanitization procedure, and powering down the chip is sufficient [32].

3.2 Hypothesis for Individual SRAM Cell Data Recovery

As described in Section 2.1.1, the SRAM power-up state is biased inversely to the stored binary data. As a result, it is possible to derive information about the actual SRAM content by observing the difference in the initial power-up state (IPU) and final power-up state after aging (FPU). To identify the state accurately, the SRAMs are powered on multiple times, as an SRAM array can contain both stable (S) and unstable (U) cells. By doing this, we generate a distribution of $0s$ and $1s$ for each cell. Recalling the information from Section 2.1.1, if a cell has consistently powered up with a logic 0 (or 1), we call it either a

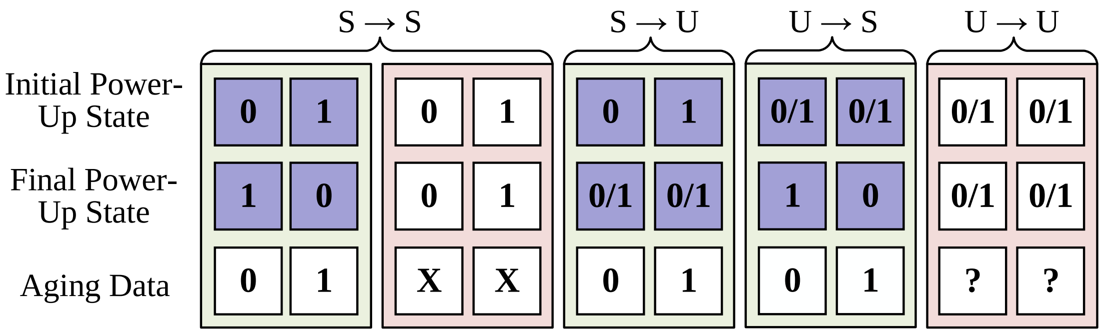


Figure 3.1: Aging data recovery from SRAM cells.

stable-0 (S_0) or stable-1 (S_1) cell. If the power-up content appears both as logic 0 and logic 1 across multiple power-ups, the cell is identified as an unstable cell.

We divide our analysis of the change between IPU and FPU states into four main categories. Figure 3.1 shows all four possible scenarios, with the relevant combinations of logic 0 and logic 1. These four categories can be explained as follows:

- *Stable remains stable (S→S):* An SRAM cell has a stable value both before, and after aging. This category has two scenarios within it. The first scenario is that the FPU state (S_1 or S_0) is complementary to the initial state (S_0 or S_1). If this occurs, it represents that the aging has overcome the initial threshold voltage difference (ΔV_{th}) of the PMOS transistors and has significantly skewed towards the opposite value. This can only happen when the aging data is 0 for S_0 (becomes S_1) or 1 for S_1 (becomes S_0). These are represented in the first two columns of Figure 3.1. Conversely, if an SRAM cell remains at the same stable value as the initial state, we cannot uniquely determine the aging content. For example, if both the IPU and FPU states remain in an S_0 state, it is possible that the cell has been aged with: (*i*) logic 0 but is not sufficient to offset the initial ΔV_{th} bias caused by process variation, or (*ii*) logic 1 that further increases the initial bias. Identical, but complementary analysis can be done for logic 1 stable cells. These are represented by the third and fourth columns of Figure 3.1, which shows that no data can be determined from these states.

- *Stable becomes unstable (S→U):* An SRAM cell has a stable value before aging, and an unstable value after aging. In this case, the values loss of stability indicates that the ΔV_{th} has been skewed towards the opposing value. As aging's effect on the is complementary, and the indication of a change in ΔV_{th} is explicit, there is no ambiguity in determining the aging data. It must be the same as the initial state of the SRAM cell, indicated in the fifth and sixth column of Figure 3.1.

- *Unstable becomes stable (U→S):* An SRAM cell has an unstable value before aging, and a stable value after aging. An unstable cell becomes more stable when a ΔV_{th} with a small magnitude is increased, thus increasing the noise threshold enough to be stable. As before, the change in ΔV_{th} is explicit, and is complementary to the actual stored value, so it can be determined. The aging data must be complementary to the final state of the SRAM cell, indicated in the seventh and eighth columns of Figure 3.1.

- *Unstable remains unstable (U→U):* An SRAM cell has an unstable value before aging, and the value remains unstable after aging. In this case, the aging may or may not have changed the ΔV_{th} between the PMOS transistors. In order to determine whether we can extract information from these cells, we must analyze the distribution of the instability from the IPU and FPU. When measuring the IPU and FPU, we measure multiple times to note stability or instability. We can then record the distribution of logic 1 and logic 0 in the states before and after aging, and compare them. If the distribution has shifted towards producing more $1s$ than before, then it is likely that the complementary (logic 0) value has been stored. Conversely, if the distribution has shifted towards producing more $0s$ than before, it is likely that a logic 1 value has been stored. If the distribution changes very slightly or none at all, then no information can be extracted from these cells.

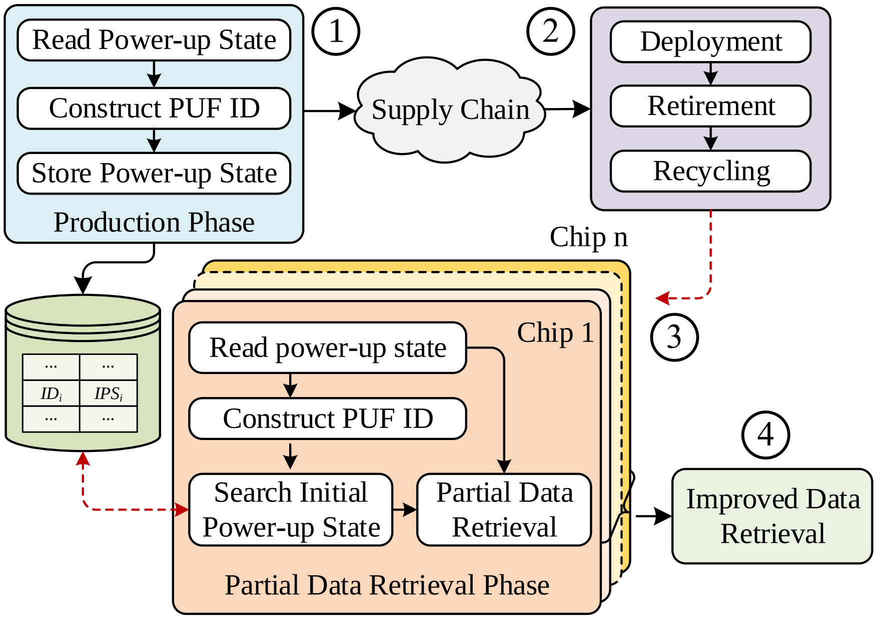


Figure 3.2: The proposed approach for recovering IP data.

3.3 Proposed Flow for SRAM Data Retrieval

Since the majority of ICs and electronic devices are manufactured offshore, extra care must be taken to protect sensitive information once the system is retired. Even if it has been erased permanently from all non-volatile memories, an impression of the stored data in a volatile memory, such as SRAM, may be retrieved by an adversary. Figure 3.2 shows the overall flow that describes data recovery from a used SRAM chip, presented through the following steps:

- *Preservation of the initial power-up states:* As described in Section 3.2, it is vital to have access to the IPU in order to accurately determine the change in ΔV_{th}. To identify the IPU for any given chip, it must be assigned a relevant ID. The ID can be constructed from the stable bits from the same IPU, acting as a PUF as described in Section 2.1.2. The untrusted manufacturer or adversary can maintain a database of $\{ID, IPU\}$ pairs. Step 1 in Figure 3.2 describes the unique ID generation, IPU measurement, and storage for later data recovery use.

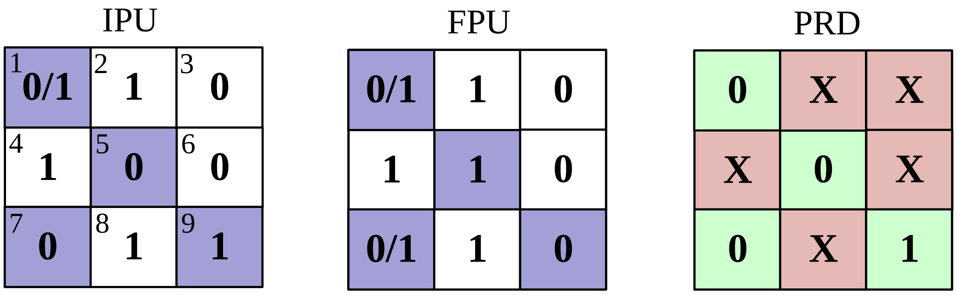
- *Device deployment, retirement, and recycling:* After the chips leave the hands of the manufacturer or adversary, they enter the supply chain as genuine, new parts. They are then purchased and deployed in critical infrastructure which may use proprietary firmware, programs or keys which are loaded into the same location in the SRAM. As a result, the chip is aged consistently, with the same sensitive content. As many of these devices share the same function and are used in identical applications, they will be subjected to the same aging parameters as other devices. After the electronics are retired or replaced, any volatile memories are simply discarded rather than destroyed. They are then sent as e-waste to an untrusted location, where the adversary can acquire the chips once again, as shown in step 2 of Figure 3.2.

- *Partial data retrieval:* Once the adversary again has access to the physical device, they first power up the chip and read the FPU state. From this, they reconstruct the device ID to identify the chip. The adversary can then recover the IPU states from the database by querying the device ID. The adversary can now directly compare the IPU and FPU states and partially reconstruct the sensitive data which was aged in the SRAM chip. The partial data retrieval phase is shown in step 3 of Figure 3.2, and will be described in detail in Section 3.3.1.

- *Improved data retrieval:* Following the first partial data recovery, the adversary can continue to collect and identify chips used in identical applications. After the same partial data recovery is performed on multiple chips, the adversary can then combine the results using a majority voting algorithm to construct the complete aging data. Step 4 in Figure 3.2 shows the improved data retrieval, which will be described in detail in Section 3.3.2.

3.3.1 Partial Data Retrieval

Once the adversary is in possession of both the IPU and FPU states, they will apply the data recovery methodology presented in Section 3.2 to each SRAM cell to retrieve aging information. The SRAM data for IPU to FPU states which change from stable 0 to stable 1 ($S_0 \rightarrow S_1$) or vice versa ($S_1 \rightarrow S_0$), stable to unstable ($S \rightarrow U$), or unstable to stable ($U \rightarrow S$) can be uniquely determined. Any cell which is unstable to unstable ($U \rightarrow U$) with a power-up state change greater than the desired threshold (T_H) can also be used as information. However, if the IPU and FPU states remain stable, as the same value, or remain unstable without a big enough change to cross the threshold, the cell is indeterminate and cannot be recovered.



IPU

1 0/1	2 1	3 0
4 1	5 0	6 0
7 0	8 1	9 1

FPU

0/1	1	0
1	1	0
0/1	1	0

PRD

0	X	X
X	0	X
0	X	1

Figure 3.3: An example of partial data retrieval with an SRAM chip.

Consider the example which is displayed in Figure 3.3. This represents a 3-by-3 SRAM cell from the adversary perspective, given only the IPU and the FPU. First, consider cell 5, which becomes S_1 from S_0 after aging. This is only possible when the cell is aged with 0. Second, consider that cell 1's IPU is an unstable cell with approximately 10% of its results returning logic 1 and the remainder returning logic 0. However, while the FPU is also unstable, approximately 70% of its results return logic 1. The aging data of this cell can only be logic 0, as it has biased the cell's power-up state towards 1. Finally, both the IPU and FPU for cell 3 are S_0. As a result, we cannot make any decision about its aging content, as any range of ΔV_{th} may have happened without our observation. This logic is extended to all cells in the array, whose results form the partially recovered data (PRD).

3.3.2 Improved Data Retrieval using Multiple SRAM Chips

Combining multiple PRDs can be an effective way to increase the total recovered information from the SRAM chips aged with the same content. One of the major contributors to the need for multiple chips is the existence of cells which are being made more stable. Practically half of the cells are aged with contents that help increase the initial threshold voltage bias ΔV_{th} of the PMOS transistors due to Gaussian random process variation [3]. It is practically impossible to determine the aging content of these cells on a small aging scale, due to the fact that the stable cells are becoming more stable, and our only insight from the outside into the ΔV_{th} is the changing power-up state. As a result, one cannot completely recover aging data from a single chip. If we combine the data of multiple chips, there is a high probability that many of the cells which were unrecoverable in one SRAM might be recoverable in another.

Another improvement made by multiple chips is the ability to use majority voting to improve the quality of the data. After aging, it is possible that the recovered content from one cell location will vary across two more chips, which will create an error. Considering multiple chips, we can apply majority voting (a binary decision rule that selects the outcome based on the majority) to remove the conflicts.

In the event that majority voting cannot find determine a clear outcome, such as in the case of a tie, we can apply more nuanced decision methods, such as distribution analysis to make the decision. This type of analysis looks at the overall magnitude of the change in the FPU across all three chips, and makes its decision based on which direction overall the FPU shifted more strongly in.

Figure 3.4 shows an example to explain our proposed scheme for improved data retrieval. First, the PRD of cell 2 of chips 1, 2, and 3 contains ’X’, ’0’ and ’X’. This signifies that we can recover the cell 2 value from the change in chip 2, whereas the aging reinforces stability in chips 1 and 3. Second, for cell 1, the PRD has conflicting recovered data. The data recovered from chips 1 and 2 is ’0’ while chip 3 reports ’1’. The majority voting looks at this

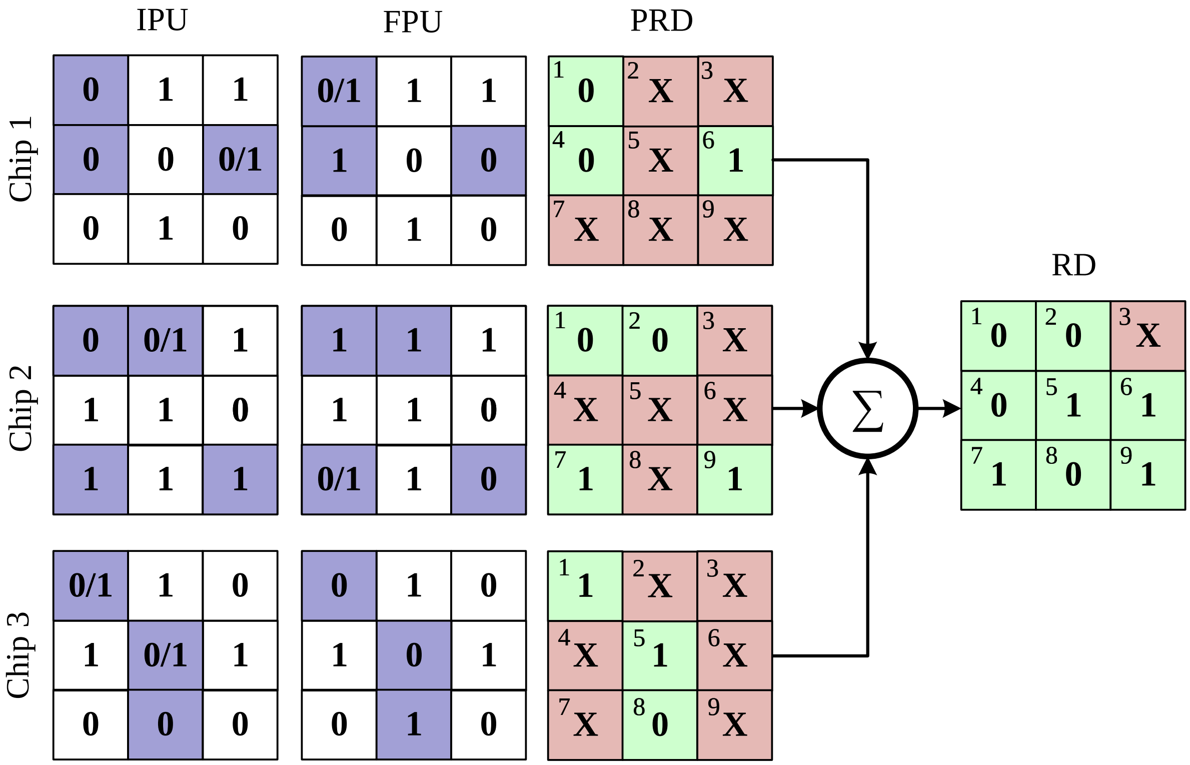


Figure 3.4: An example of the improved data retrieval using multiple SRAM chips.

and decides that the recovered value of this cell is logic '0'. This analysis can be performed throughout all of the SRAM cells to create the complete, improved recovered data.

3.3.3 Algorithm for Partial and Improved Data Retrieval

Algorithm 1 and Algorithm 2 show the overall improved data retrieval process. Algorithm 1 shows the overall partial data retrieval process (described in Section 3.3.1) for multiple individual (N) SRAM chips. For a recycled chip going through the recovery process, multiple power-ups for both IPU and FPU states are recorded to capture the stable and unstable cells, along with the distribution of logic 1 and logic 0 values. We denote the number of power-ups as M, the collection of all M IPUs and FPUs of the same chip as $\{IPU\}_M$, and $\{FPU\}_M$, respectively. The partial data retrieval (`part-data-rec`) function takes $\{IPU\}_M$ and $\{FPU\}_M$ as inputs, Line 2. The initial and final state (IS and FS) of an SRAM chip can be understood as the element-wise addition of all power-up states within $\{IPU\}_M$ and $\{FPU\}_M$, Lines 4-5. IS and FS are now representations of how many

Algorithm 1: The proposed partial data retrieval.

```
   Input  : Sets of IPU states ($\{IPU\}_M$), and FPU states
            ($\{FPU\}_M$) for $N$ chips
   Output: Partially Recovered Information ($\{H\}^n$), PU Difference ($\{D\}^n$)
1  
2  function part-data-rec ($\{IPU\}_M$, $\{FPU\}_M$) is
3      $IS \leftarrow \varnothing$; $FS \leftarrow \varnothing$ ;
4      $IS \leftarrow$ element-wise-addition($\{IPU\}_M$) ;
5      $FS \leftarrow$ element-wise-addition($\{FPU\}_M$) ;
6      $D \leftarrow$ diff($IS, FS$) ;
7      $H \leftarrow$ hyp-bit-array($D$) ;
8      return $H$ ;
9  end
10 function hyp-bit-array ($D$) is
11     foreach (element $[i, j] \in |D|$) do
12         if $D[i, j] > T_H$ then
13             $H[i, j] \leftarrow 1$ ; /* – logic 1 – */;
14         else if $D[i, j] < -T_H$ then
15             $H[i, j] \leftarrow -1$ ; /* – logic 0 – */;
16         else
17             $H[i, j] \leftarrow 0$ ; /* - insufficient information - */;
18         end
19         return $H$, $D$ ;
20     end
21 end
22 for $k \leftarrow 0$ to $N$ do
23     $H^k$=part-data-rec($\{IPU\}^k_M, \{FPU\}^k_M$);
24     $D^k$=part-data-rec($\{IPU\}^k_M, \{FPU\}^k_M$);
25 end
26 return $\{H\}^N$, $\{D\}^N$
```

times (out of M total power-ups) cells in the SRAM array powered into a logic 1 state. As described in the data retrieval concept in Section 3.2, a chip's aging bias is the complementary imprint of the IP data written to SRAM cells. If a chip is aged with logic 0, the final power-up will lean towards more 1s, e.g. a change of 5 to 9 from IS to FS under $M = 10$, opposite to the programmed data bit. This shift in the SRAM ΔV_{th} can be constructed by computing the difference between the accumulated initial and final states, IS and FS, Line 6. The hypothesis bit array (H), or the partially reconstructed data from a single chip, can be computed with the function `hyp-bit-array`, Lines 10-21. This function compares

Algorithm 2: Majority voting algorithm for improved data retrieval.

```
Input  : Partially Recovered Information ($\{H\}^N$), PU Difference ($\{D\}^N$)
Output: Recovered data ($RD$)
______________________________
$RD \leftarrow \varnothing$ ;
$PRD \leftarrow$ element-wise-addition($\{H\}^N$) ;
$SUM \leftarrow$ element-wise-addition($\{D\}^N$) ;
foreach (element $[i, j] \in |PRD|$) do
    if ($PRD[i, j]$) $\geq +1$ then
        $RD[i, j] \leftarrow 1$ ;
    else if ($PRD[i, j]$) $\leq -1$ then
        $RD[i, j] \leftarrow 0$ ;
    else
        if $SUM[i, j]) > T_H$ then
            $RD[i, j] \leftarrow 1$ ;
        else if $SUM[i, j]) < -T_H$ then
            $RD[i, j] \leftarrow 0$ ;
        else
            $RD[i, j] \leftarrow X$ ;
        end
    end
end
return $RD$ ;
```

the difference between the IS and FS to a sample-size-dependent positive threshold T_H. If the difference is positive and greater than T_H, we can determine that the amount of logic 0 returns in the FPU has increased, and that the cell has been aged with data opposite to the aging bias of 0, a logic 1. This information is saved in an array H, Lines 12-13. Conversely, if the difference is less than the threshold $-T_H$, an increase in logic 1 returns in the FPU indicates that the cell must have been aged with logic 0, Lines 14-15. We mark these cells as -1 in array H. If the difference ends up being either 0, or below the threshold in magnitude, that value is indeterminate, and is marked as 0 in array H, Lines 16-17. The constructed bit array along with the thresholds applied, along with the difference array is returned once the information for all SRAM cells has been generated, Line 19. This process is repeated for each chip in the set to provide the complete data for majority voting, Lines 22-25. These arrays are then returned for use in the majority voting algorithm, Line 26.

Algorithm 2 describes the improved retrieval process by combining data from multiple (N) SRAM chips. Each chip contributes to the data retrieval process as the result of certain cells which remain stable or become more stable after aging in other chips. Thus quite a few indeterminate bits will be recovered when an additional chip is added. To account for conflicting values between multiple chips, e.g. one chip determines a cell with data 0 while another chip calls it data 1, the majority voting takes place and will help to resolve the conflict. All of the partially recovered data and difference data goes through element-wise addition, Lines 3-4. Any cells with a positive final PRD value are passed through as a 1 to the final data, Lines 6-7. Any cells with a negative final value are passed through as a 0 to the final data, Lines 8-9. Any cells with a net zero sum will go through further analysis of the power-up state change distributions, Lines 10-17. If the sum of the power-up state for a particular cell's distribution is greater than the threshold T_H, a 1 is passed through to the data, Lines 11-12. Conversely, if the sum of the power-up state for a cell's distribution is lower than the negative threshold $-T_H$, a 0 is passed through to the data, Lines 13-14. If the value's magnitude is 0 or falls below both positive and negative thresholds, the data is considered inconclusive, which we marked as X, Lines 15-16. The final recovered SRAM data (RD is returned once the program completes the combination of the partially retrieved data from multiple SRAM chips, Line 20.

Chapter 4

Experimental Results

Using the techniques explained in Chapter 3, the sensitive information stored in the SRAM during its operation can be leaked to an adversary once they read and compare the initial and final power-up states. In this section, we demonstrate actual data retrieval using these methods from commercial, off-the-shelf SRAM chips. We simulate statically positioned sensitive data by storing a black-and-white, binary-coded image on the SRAM, shown in Figure 4.2. We then use accelerated aging of the chip to simulate use in a critical application.

The reconstruction of SRAM data (Algorithm 1 and 2) is processed via in-house MATLAB scripts. Our implementation of the proposed approaches can retrieve an imprint of the image in as little as four hours of accelerated aging at 85°C.

4.1 Experimental Setup

Figure 4.1 shows our experimental setup for the accelerated aging of SRAMs. The SRAM chips we used were six Microchip 23A640 64 Kbit SRAMs [60] with a serial interface. This was ideal for our needs, as the image we were storing was just under 40 Kb. It also represents an easy target for recycling, as a standalone chip would be easy to remove from a PCB for analysis. The SRAM chips are aged with the contact probe of Temptronic ThermoSpot DCP-201 system [61] at a constant 85°C. All chips were subjected to the same aging conditions, and all had the same black-and-white image stored. Multiple power-up states from the SRAM were recorded after every four hours of aging. Voltage shifters ensure voltage compatibility between the SRAM chips (1.65V) and Raspberry Pi (3.3V) during the reading and writing of SRAM data.

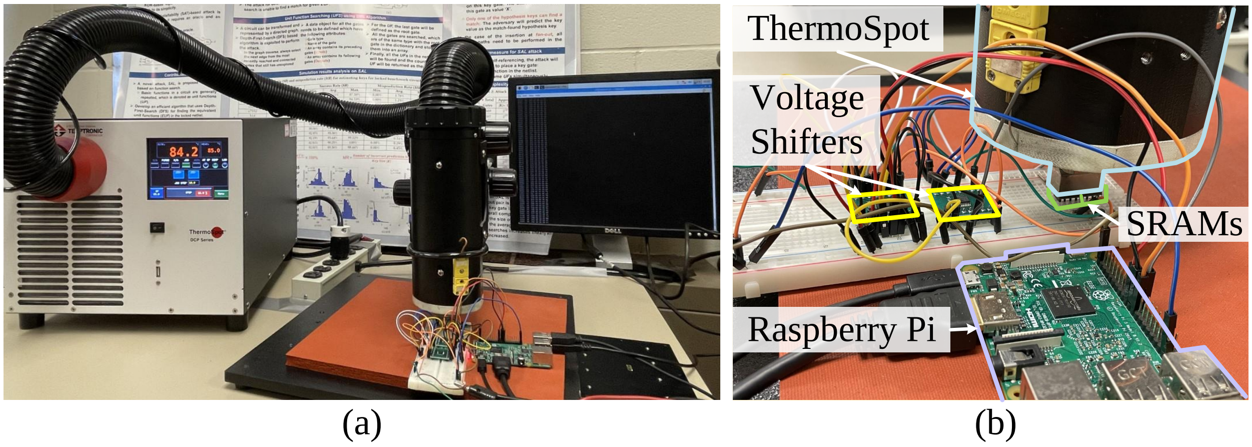


Figure 4.1: Experimental setup. (a) Accelerated aging set-up using ThermoSpot, (b) zoomed-in view of SRAM aging.

4.2 Measurement

SRAM power-up states were taken at four-hour intervals, via communication with the Raspberry Pi. The Raspberry Pi uses an SPI interface to both read from and write data to the chip. When writing to the chip, we used a C program which read the binary image from a text file, and sent the information to the chip. For reading the power-up states, using an in-house C program, the Raspberry Pi would automatically read the power-up state and dump it to a text file in hexadecimal format every 5 seconds. This text file was then processed by our aforementioned MATLAB scripts. These readings were done under a number of different conditions. We performed the power-up readings at high temperature, room temperature, and under different voltage conditions. Throughout, we took 25 samples per chip, per interval, per condition, leading to 100 samples per chip per interval.

4.3 Data Retrieval

In order to present data in the best way that lines up with our hypotheses, we collected and analyzed data by changing two main parameters. Firstly, we vary the chip count, being that in Section 3.3.2, we state that the recovery improves with a higher amount of chips. To demonstrate this, we analyzed the data at intervals of two, four, and six chips. Secondly, we vary the aging time, being that in Section 2.1.3, we state that as an SRAM is aged, it

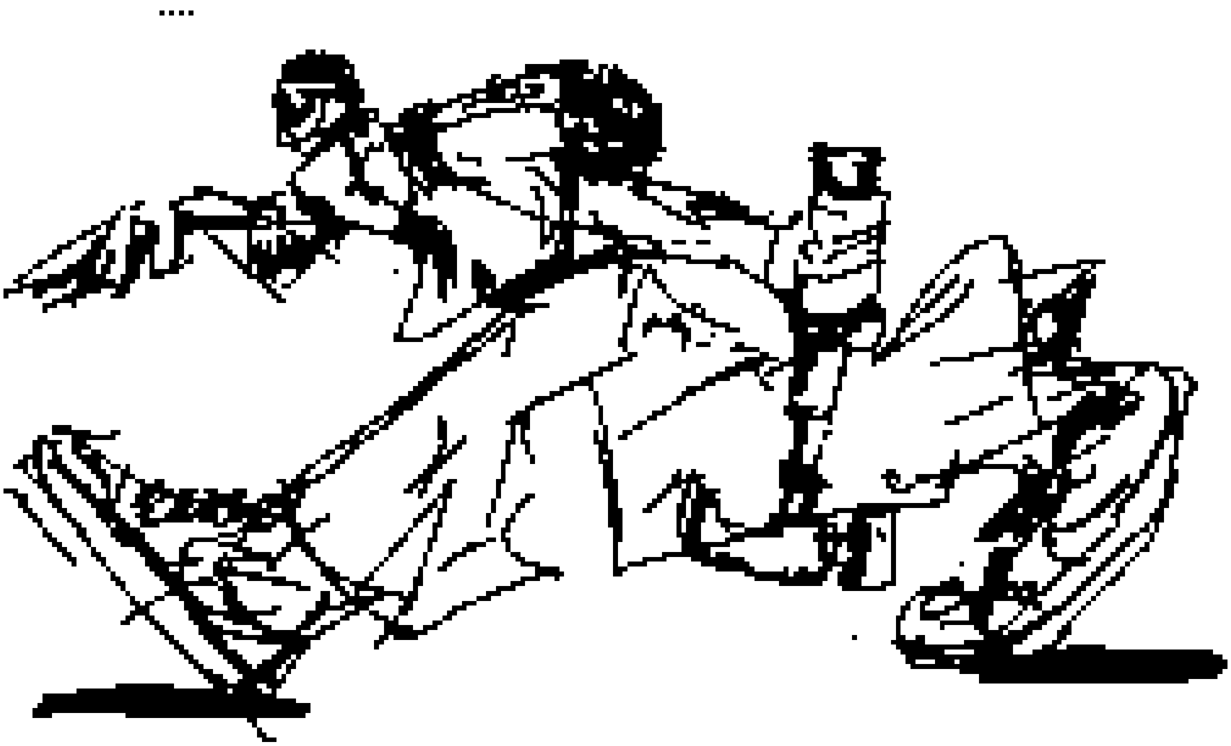


Figure 4.2: Original image used for SRAM aging.

progressively becomes more biased. Therefore, we analyzed data at intervals of four, eight, and twelve hours of accelerated aging.

The results of the analysis being performed under these different conditions are shown in Figure 4.3. The white and black pixels correspond to the respective data retrieved, while the grey pixels indicate that no information could be recovered. When compared to the original in Figure 4.2, it is clear that those with the higher chip counts and aging time recovered data more accurately and in higher volume. This is clearly shown in Figure 4.4, which shows both the accurate data recovered, along with the percentage of the data recovery noise. In our experiment, these figures range from 49.5% recovery with 4.8% noise at 6 chips/12 hours, to 16.5% recovery with 10.3% noise with 2 chips/4 hours.

4.3.1 Noise Induced Improvement in Data Retrieval

As described in Section 3.3.2, many cells when performing a partial data retrieval are simply impossible to recover data from, because the data stored within them pushes stable

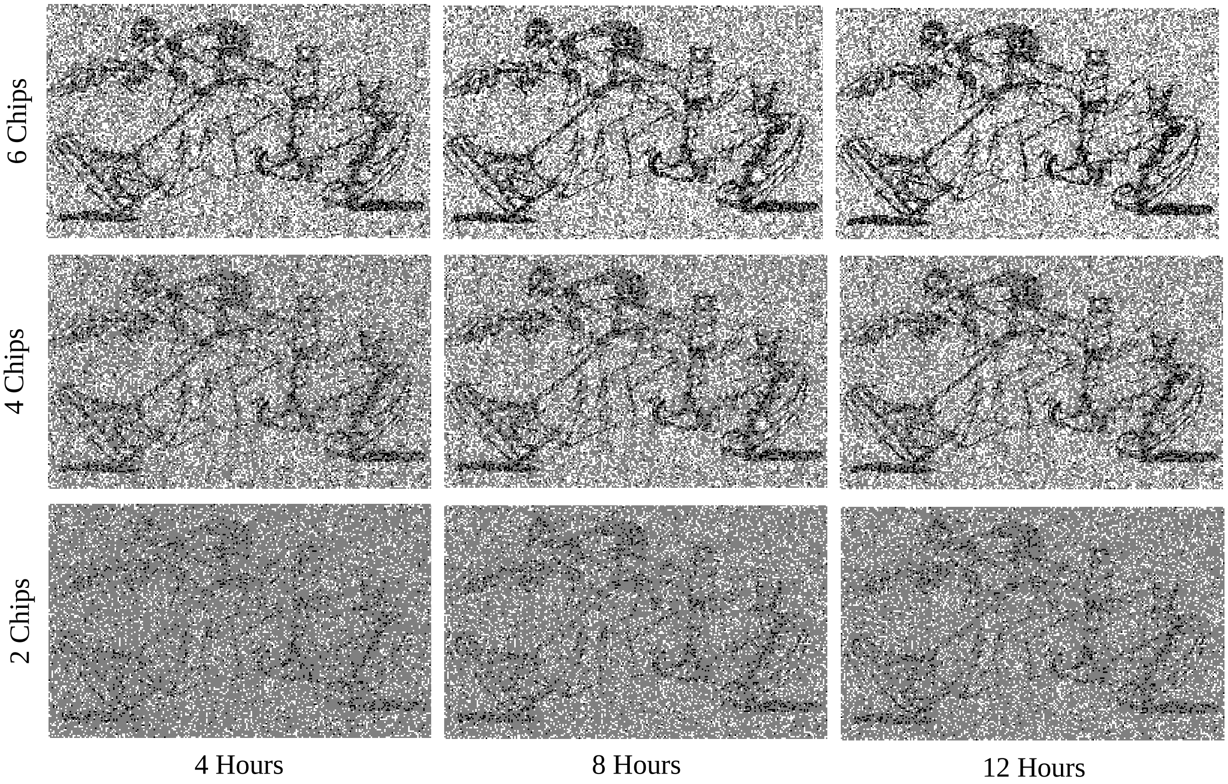


Figure 4.3: Data recovery from SRAM at varying parameters.

cells further towards stability. This leads to the logical conclusion that unstable cells are more likely to provide information about the ΔV_{th} than stable cells, for a few primary reasons. Firstly, the ΔV_{th} is already very close to 0, so any change will be reflected more strongly than that of a change when the ΔV_{th} is of a higher magnitude. Secondly, and more importantly, an unstable cell will be able to reflect any change in the ΔV_{th}, as it can represent it as a change in the distribution of logic 1 and logic 0 results.

We noticed, when performing our data acquisition at high temperatures (125°C), a higher portion of the cells were unstable as a result of the thermal noise. Many of these cells were stable at room temperature, and did not provide any relevant information about the stored data. However, many of the cells which were stable became unstable and were able to provide information at these higher temperatures.

Theoretically, with a large enough sample size, this could work with arbitrarily large amounts of noise in order to make every cell unstable. These would be able to provide

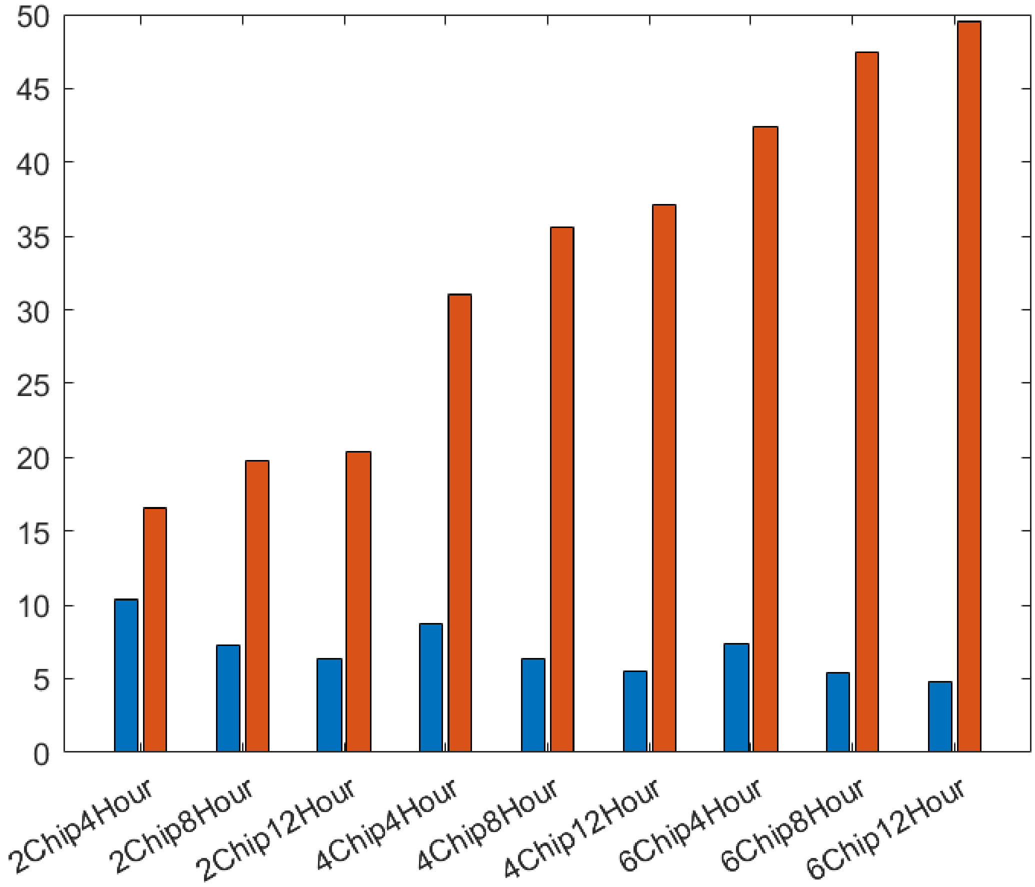


Figure 4.4: Noise and data recovery values at varying parameters.

explicit information, as the difference in the IPU and FPU distributions would demonstrate a tangible change in the ΔV_{th}.

Chapter 5

Future Research Direction and Conclusion

The conventional wisdom that volatile memories can be disposed of without any risk of data leakage is both incorrect, and potentially dangerous. Adversaries can take advantage of this assumption to potentially steal vital trade secrets or secret encryption keys, both of which threaten our economy and our daily well-being. We must proceed with a more cautious approach in our attitude towards the potential data still present in some form or fashion within volatile memories, and look past the fact that the data is not preserved upon loss of power.

This thesis has laid the groundwork for lots of potential future work in this area. In Section 4.3.1 we discuss the use of high thermal noise to encourage cell instability, leading to higher overall data recovery. This improvement is not bounded by thermal noise, and any type of noise injection could improve the overall data recovery from single chips by reducing, or eliminating, the problem of stable cells remaining stable being impossible to recover.

Another key area in which this research can improve is in the removal of the need for an initial power-up state. In this research, it is key to being able to identify the change in ΔV_{th}, however, if the change could be detected without it, it would enable many more attacks. For example, retroactive analysis of SRAM could be performed on devices that never had the chance to prevent it. It would also open the door for much a more flexible class of adversaries. This research requires a heavy amount of coordination and connection to the manufacturer, a requirement that could be eliminated if the initial power-up was unnecessary.

The final key way in which this research can be carried forward is prevention. New strategies are going to have to be developed and carried out in order to prevent what was once thought not possible from happening in the future. Techniques such as scrambling data in use, calibrating the cells, or simply developing more robust sanitization methods would all be beneficial given what we have proposed.

In conclusion, in this thesis, we presented a new method of recovering data over long periods of inactivity from SRAM chips which have been subjected to aging. The method we presented took advantage of the effects of aging on the power-up state of SRAM cells, analyzing the differences between initial and final power-up states to identify the data stored within. The proposed approach proved effective in practice, clearly showing an imprint of previously stored data even after the device had been powered off. This research has been very important to both increase our understanding of what can be recovered from volatile memories and also to help dispel the notion that parts, besides the hard disk, that have been exposed to sensitive information can be safely discarded without substantial sanitization procedures.

Bibliography

[1] W. Wang, A. D. Singh, and U. Guin, "A Systematic Bit Selection Method for Robust SRAM PUFs," *Journal of Electronic Testing*, pp. 1–12, 2022.

[2] W. Wang, U. Guin, and A. Singh, "Aging-Resilient SRAM-based True Random Number Generator for Lightweight Devices," *Journal of Electronic Testing*, vol. 36, pp. 301–311, 2020.

[3] U. Guin, W. Wang, C. Harper, and A. D. Singh, "Detecting Recycled SOCs by Exploiting Aging Induced Biases in Memory Cells," in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 72–80, 2019.

[4] The Challenge of Technological Superiority, 2015, `https://apps.dtic.mil/sti/pdfs/AD1016314.pdf`.

[5] Congressional Res. Service, "Semiconductors: U.S. Industry, Global Competition, and Federal Policy," 2020. `https://fas.org/sgp/crs/misc/R46581.pdf`.

[6] US Government, "H.R.7178 - CHIPS for America Act," 2022.

[7] Semiconductor Industry Association, Government Incentives and U.S. Competitiveness in Semiconductor Manufacturing, September, 2020.

[8] S. Ezell, "Moore's Law Under Attack: The Impact of China's Policies on Global Semiconductor Innovation," 2021.

[9] White House, Building Resilient Supply Chains, Revitalizing American Manufacturing, and Fostering Broad-Based Growth, 100-Day Reviews under Executive Order 14017, June, 2021.

[10] D. DiMase, Z. A. Collier, J. Muldavin, J. A. Chandy, D. Davidson, D. Doran, U. Guin, J. Hallman, J. Heebink, E. Hall, Honorable A. R. Shaffer, "Zero Trust for Hardware Supply Chains: Challenges in Application of Zero Trust Principles to Hardware," *NDIA*, 2021.

[11] Assessment of the Critical Supply Chains Supporting the U.S. Information and Communications Technology Industry, prepared by U.S. Departments of Commerce and Homeland Security, 2022.

[12] International Labour Office, The distribution of value added among firms and countries: The case of the ICT manufacturing sector, 2017, `https://www.ilo.org/wcmsp5/groups/public/---dgreports/---inst/documents/publication/wcms_544190.pdf` .

[13] U. Guin, Z. Zhou, and A. Singh, "Robust Design-for-Security Architecture for Enabling Trust in IC Manufacturing and Test," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 5, pp. 818–830, 2018.

[14] U. Guin, Q. Shi, D. Forte, and M. M. Tehranipoor, "FORTIS: A Comprehensive Solution for Establishing Forward Trust for Protecting IPs and ICs," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 21, no. 4, pp. 1–20, 2016.

[15] U. Guin, Z. Zhou, and A. Singh, "A Novel Design-for-Security (DFS) Architecture to Prevent Unauthorized IC Overproduction," in *VLSI Test Symposium (VTS)*, pp. 1–6, 2017.

[16] U. Guin, S. Bhunia, D. Forte, and M. M. Tehranipoor, "SMA: A System-Level Mutual Authentication for Protecting Electronic Hardware and Firmware," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 265–278, 2016.

[17] A. Stern, D. Mehta, S. Tajik, U. Guin, F. Farahmandi, and M. Tehranipoor, "SPARTA-COTS: A Laser Probing Approach for Sequential Trojan Detection in COTS Integrated Circuits," in *IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pp. 1–6, 2020.

[18] A. Jain, Z. Zhou, and U. Guin, "Survey of Recent Developments for Hardware Trojan Detection," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2021.

[19] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain," *Proc. of the IEEE*, pp. 1207–1228, 2014.

[20] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit Integrated Circuits: Detection, Avoidance, and the Challenges Ahead," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 9–23, 2014.

[21] M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits: Detection and Avoidance*. Springer International Publishing, 2015.

[22] Committee on Armed Services, House of Representatives, "National Defense Authorization Act for Fiscal Year 2011," 2010.

[23] U. Guin, D. DiMase, and M. Tehranipoor, "A Comprehensive Framework for Counterfeit Defect Coverage Analysis and Detection Assessment," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 25–40, 2014.

[24] M. Leblanc and C. Abesamis, "NASA Counterfeit Parts Awareness and Inspection," 2016.

[25] Y. Zhang and U. Guin, "End-to-End Traceability of ICs in Component Supply Chain for Fighting Against Recycling," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 767–775, 2019.

[26] P. Chowdhury, U. Guin, A. D. Singh, and V. D. Agrawal, "Two-pattern Δ_{IDDQ} Test for Recycled IC Detection," in *International Conference on VLSI Design and International Conference on Embedded Systems (VLSID)*, pp. 82–87, 2019.

[27] M. Alam, S. Chowdhury, M. M. Tehranipoor, and U. Guin, "Robust, Low-Cost, and Accurate Detection of Recycled ICs using Digital Signatures," in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 209–214, 2018.

[28] W. Wang, U. Guin, and A. Singh, "A Zero-Cost Detection Approach for Recycled ICs using Scan Architecture," in *VLSI Test Symposium (VTS)*, pp. 1–6, 2020.

[29] J. Hovanes, Y. Zhong, and U. Guin, "Beware of Discarding Used SRAMs: Information is Stored Permanently," in *IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pp. 1–7, 2020.

[30] M. M. Hasan and B. Ray, "Data recovery from {"Scrubbed"}{NAND} flash storage: Need for analog sanitization," in *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1399–1408, 2020.

[31] A. R. Regenscheid, L. Feldman, G. A. Witte, *et al.*, "NIST Special Publication 800-88, Revision 1: Guidelines for Media Sanitization," 2015.

[32] NSA/CSS Storage Device Sanitization Manual, NSA/CSS Policy Manual 9-12, 2014, `https://www.nsa.gov/portals/75/documents/resources/everyone/media-destruction/storage-device-declassification-manual.pdf`.

[33] Raspberry Pi Ltd, *Raspberry Pi Pico W*, 2023.

[34] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *International workshop on cryptographic hardware and embedded systems*, pp. 63–80, Springer, 2007.

[35] D. E. Holcomb, W. P. Burleson, and K. Fu (2009), "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1198–1210.

[36] V. van der Leest, E. van der Sluis, G.-J. Schrijen, P. Tuyls, and H. Handschuh (2012), "Efficient Implementation of True Random Number Generator based on SRAM PUFs," in *Cryptography and Security: From Theory to Applications, springer*, pp. 300–318.

[37] A. Aysu, E. Gulcan, D. Moriyama, P. Schaumont, and M. Yung, "End-to-end design of a PUF-based privacy preserving authentication protocol," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 556–576, Springer, 2015.

[38] S. K. Mathew, D. Johnston, S. Satpathy, V. Suresh, P. Newman, M. A. Anders, H. Kaul, A. Agarwal, S. K. Hsu, G. Chen, *et al.*, "μ RNG: A 300–950 mV, 323 Gbps/W All-Digital Full-Entropy True Random Number Generator in 14 nm FinFET CMOS," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 7, pp. 1695–1704, 2016.

[39] R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications.* Springer Science & Business Media, 2013.

[40] K. Xiao, M. T. Rahman, D. Forte, Y. Huang, M. Su, and M. Tehranipoor, "Bit selection algorithm suitable for high-volume production of SRAM-PUF," in *2014 IEEE international symposium on hardware-oriented security and trust (HOST)*, pp. 101–106, IEEE, 2014.

[41] M. J. a. Mahmod and U. Guin, "A Robust, Low-Cost and Secure Authentication Scheme for IoT Applications," *Cryptography*, vol. 4, no. 1, p. 8, 2020.

[42] Y. Zhong, J. Hovanes, and U. Guin, "On-demand device authentication using zero-knowledge proofs for smart systems," in *Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 1–6, 2023.

[43] J. Hovanes, Y. Zhong, and U. Guin, "A Novel IoT Device Authentication Scheme Using Zero-Knowledge Proofs," *GOMACTech*, 2023.

[44] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon Physical Random Functions," in *ACM Conf. on Computer and Communications Security*, pp. 148–160, 2002.

[45] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in *Design Automation Conference*, pp. 9–14, 2007.

[46] X. Xin, J.-P. Kaps, and K. Gaj, "A configurable ring-oscillator-based puf for xilinx fpgas," in *IEEE Euromicro conf. on digital system design*, pp. 651–657, 2011.

[47] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[48] G.-J. Schrijen and V. Van Der Leest, "Comparative analysis of SRAM memories used as PUF primitives," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1319–1324, IEEE, 2012.

[49] Y. Zhong and U. Guin, "Chosen-Plaintext Attack on Energy-Efficient Hardware Implementation of GIFT-COFB," in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 1–4, 2022.

[50] Y. Zhong and U. Guin, "Fault-injection based chosen-plaintext attacks on multicycle aes implementations," in *Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 443–448, 2022.

[51] K.-L. Chen, S. A. Saller, I. A. Groves, and D. B. Scott, "Reliability Effects on MOS Transistors Due to Hot-Carrier Injection," *IEEE Transactions on Electron Devices*, vol. 32, no. 2, pp. 386–393, 1985.

[52] S. Mahapatra, D. Saha, D. Varghese, and P. B. Kumar, "On the generation and recovery of interface traps in MOSFETs subjected to NBTI, FN, and HCI stress," *IEEE Tran. on Electron Devices*, pp. 1583–1592, 2006.

[53] D. K. Schroder and J. A. Babcock, "Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing," *Journal of applied Physics*, vol. 94, no. 1, pp. 1–18, 2003.

[54] V. Reddy, A. T. Krishnan, A. Marshall, J. Rodriguez, S. Natarajan, T. Rost, and S. Krishnan, "Impact of negative bias temperature instability on digital circuit reliability," *Microelectronics Reliability*, pp. 31–38, 2005.

[55] W. Wang, S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu, and Y. Cao, "The impact of NBTI effect on combinational circuit: Modeling, simulation, and analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 2, pp. 173–183, 2009.

[56] P. Chowdhury, U. Guin, A. D. Singh, and V. D. Agrawal, "Estimating Operational Age of an Integrated Circuit," *Journal of Electronic Testing*, pp. 1–16, 2021.

[57] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of pmos nbti effect for robust nanometer design," in *Proceedings of the 43rd annual Design Automation Conference*, pp. 1047–1052, 2006.

[58] S. Skorobogatov, "Low temperature data remanence in static RAM," tech. rep., University of Cambridge, Computer Laboratory, 2002.

[59] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, "Lest we remember: cold-boot attacks on encryption keys," *Communications of the ACM*, vol. 52, no. 5, pp. 91–98, 2009.

[60] SRAM Datasheet, `https://ww1.microchip.com/downloads/en/DeviceDoc/22126E.pdf`.

[61] Temptronic ThermoSpot DCP-201 Bench Top Temperature Forcing System, `https://www.intestthermal.com/temptronic/thermospot`.