An Exploration of Embedded Memories Start-Up Patterns as Physical Unclonable Functions

Khairul Shazwan Mamat [1], Chia Yee Ooi [2*]
[1, 2] *Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia*
Email: [1] kshazwan3@graduate.utm.my, [2] ooichiayee@utm.my
*Corresponding Author

***Abstract*—Memory-based Physical Unclonable Functions (PUFs) are normally used for authentication and key generation for hardware security. Various types of memories, such as SRAM, DRAM, and flash, are explored for their effectiveness as PUFs. SRAM (Static Random-Access Memory) has been a prominent choice for PUF applications due to its reliable start-up pattern, which exhibits variability across different chips due to inherent manufacturing differences. DRAM (Dynamic Random-Access Memory) PUF can be based on start-up pattern, DRAM related latency parameters, or DRAM retention. In this paper, we explored both embedded SRAM and SDRAM of an FPGA to be utilized as PUFs. SRAM is, as expected, shown to be effective and reliable as a PUF. Its start-up pattern is well-established for generating unique identifiers. Although SDRAM shows promising randomness, its bit error rate (BER) is high (exceeding 50%), which significantly impacts its reliability as a PUF. A high BER means that the data obtained from the SDRAM can be unreliable, which is critical for PUF applications where consistency is key. Since most of the system has both SRAM and SDRAM, we recommend that SRAM can be utilized continuously for its established reliability as a PUF. Meanwhile, SDRAM with its strong randomness attributes make it a good candidate for a different role, which is a seed for data obfuscation to scramble or obscure data, especially in scenarios where authentication fails, thus enhancing overall security.**

Keywords*—authentication; physical unclonable function; reliability; robustness; randomness

I. Introduction

Memory-based Physical Unclonable Functions (PUFs) play a crucial role in authentication and key generation; they leverage the inherent manufacturing variations in memory cells to generate unique and unpredictable responses. PUFs are embedded in hardware, which provides a layer of physical security that software-based solutions cannot match. This inherent security helps protect against cloning and replay attacks. PUFs can be used to generate cryptographic keys on-demand. These keys are derived from the PUF’s response dynamically and do not need to be stored permanently, reducing the risk of key exposure or theft. In addition, the approach has lower hardware overhead.

Start-up based SRAM PUFs exploit the unique start-up values of SRAM cells when power is first applied. Due to manufacturing process variations, each SRAM cell may start in a different state, creating a unique pattern of 0s and 1s [1,2]. When powered off and then on again, a good SRAM PUF exhibits the same start-up pattern due to inherent cell characteristics [3]. DRAM (Dynamic Random-Access Memory) PUFs can also use the initial state of memory cells, but DRAM cells require periodic refreshing to maintain data, which adds complexity. Moreover, the need for power cycles and a waiting period before capturing the PUF responses to obtain reliable signatures is another primary challenge [4]. Besides start-up patterns, DRAM PUFs can be explored based on latency (DRAM-related timing parameters) and retention (how long data remains stable after refresh is stopped) [5]. These factors can contribute to variability used in PUF applications.

However, SRAM PUFs can be unstable due to environmental factors and inherent circuit mismatches. Several studies highlighted the need to improve their stability and robustness. Methods like error correction code (ECC) and fuzzy extractors enhance PUF reliability by correcting errors found in PUF responses. Kim et al. demonstrated that ECC could reduce error rates to less than 10^{-6} [6]. Chen et al. achieved a failure rate lower than 10^{-9} using polar codes [7], while other studies proposed integrating ECC with masking methods to minimize data requirements [8-10]. Techniques such as Von Neumann extractors, fuzzy extractors, and Linear Shift Register extractors [11] improved SRAM PUF performance. Liu et al.'s improved Von Neumann extractor reduced errors to under 1% [12]. While fuzzy extractors ensure stable key generation, Ali Pour et al. introduced a masking mechanism on top of the extractors to protect helper data [13]. Physically altering SRAM cells can also enhance stability but may affect regular memory operations. These techniques include using dual-mode SRAM cells, Hot Carrier Injection (HCI), and various transistor adjustments. Liu et al. achieved a 100% stable PUF with HCI [14], while Chang et al. designed a dual-mode SRAM cell optimization using word-line voltage modulation and dynamic voltage scaling to improve reliability [15]. Research into alternative transistor technologies, like FinFETs, shows promise for SRAM PUFs. Zhang et al. [16] and Narasimham et al. [17] found that FinFET SRAM PUFs offer stable performance despite aging

https://www.academicedgepub.co.uk/journals/journal-of-computer-science-and-engineering-research-jcser
Received: 30-08-2024
Revised: 15-09-2024
Published: 26-09-2024

challenges. Methods for selecting stable SRAM bits include analyzing spatial dependencies, discharge inversion effects, and power supply variations. Liao et al. [18-20] and Lee et al. [21] developed techniques to identify and use stable SRAM cells for better reliability. Data remanence-based technique [22-23] uses retained data after power-off to select stable SRAM cells, improving key consistency. Studies suggest optimizing power-off durations and using data retention metrics for better PUF performance.

Kumar et al. investigated the use of Dynamic Random-Access Memory (DRAM) startup values as a PUF response [24]. The process involves precharging the DRAM, allowing it to settle, and then activating the sense amplifier to read the DRAM values. Some cells discharge slower and remain at 1, while others discharge faster and switch to 0. The captured value serves as a seed for a Linear Feedback Shift Register (LFSR) to generate a random number as the PUF response. Key findings include approximately 50% uniqueness, 99% reliability under voltage and temperature variations, 47% bit aliasing, and 48% uniformity. Zheng proposed using Pico-Physical Unclonable Functions (Pico-PUFs) to supply a random initial pattern as a challenge to DRAM, reducing the size of the PUF challenge and response during enrollment [25]. The process involves initializing DRAM with a random pattern, disabling refresh operations, and later recharging the DRAM. The PUF response is read out based on the precharge time (TRP). Post-processing involves using Error-Correcting Codes (ECC) to generate a codeword from the PUF response, which is then XORed with the response to create helper data. This data is stored for reconstruction. In PUF reconstruction, H (a secret key) is XORed with the generated PUF response, decoded by ECC, and compared with the original golden key. The uniqueness achieved is 45%, and the bit error rate (BER) is 0.44%. [24] initialized the DRAM pattern with a precharging process while [25] initialized the DRAM pattern with an external PUF, namely Pico-PUF so these two work are not based on the original startup patterns of DRAM.

Our study compares both startup-based SRAM PUF and startup-based DRAM PUF. We explore the characteristics of DRAM PUF which is based on purely the startup values and discusses the effectiveness of DRAM PUF based on the original startup values. The rest of the paper is outlined as follows. Section II discusses the performance metrics related to DRAM PUFs. Section III describes our methodology to setup the comparison between the SRAM PUF and DRAM PUF utilizing the embedded memories on FPGA boards. We also detail out the analysis steps on the characteristics of both PUFs. Section IV discusses the results of performance metrics of both PUFs. Section V concludes the study.

II. PERFORMANCE METRICS OF DRAM PUFS

The effectiveness of DRAM PUF mainly hinges on three key quality parameters: uniqueness, reliability, and randomness. Besides, bias measures the deviation of the distribution of binary outputs (0s and 1s) from an ideal 50/50 distribution in PUF responses. It quantifies how skewed the response values are; uniformity measures how evenly the binary outputs (0s and 1s) are distributed across all responses of the PUF. It reflects the degree to which the PUF outputs are balanced. Evaluating these parameters is essential for ensuring that PUFs meet contemporary cryptographic security standards. This research assesses various PUF metrics to gauge how effectively PUFs deliver secure and reliable hardware-based security solutions.

A. Uniqueness

Uniqueness measures the degree of difference in PUF responses across different instances of the same type of PUF. The satisfactory uniqueness value should be greater than 0.5 and approaching 1. A high uniqueness value suggests that each PUF instance generates a unique response, enhancing security by ensuring that no two instances produce the same response. For optimal security, uniqueness should be as high as possible, ideally close to 1, indicating that responses are nearly completely distinct. The uniqueness is evaluated as in Equation (1). Where N is the total number of devices (PUFs) being compared. HD(R_i, R_j) is the Hamming distance between the PUF responses R_i and R_j from devices i and j. n is the number of bits in each PUF response.

$$\text{Uniqueness} = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{HD(R_i, R_j)}{n} \qquad (1)$$

B. Bit Error Rate (BER)

BER measures the error rate in the PUF responses compared to the expected or reference responses. The ideal value of BER should be close to 0. A low BER indicates minimal error rates between the PUF output and the reference or between different instances. For secure applications, BER should be as close to 0 as possible, indicating high consistency and reliability. BER can be measured using Equation (2). Where x_i and y_i are the bits at position i in the two binary strings. n is the length of the binary strings.

$$\text{BER} = \sum_{i=1}^{n} \frac{(x_i \oplus y_i)}{n} \qquad (2)$$

C. Randomness

Randomness measures how unpredictable and uniformly distributed the responses of a Physical Unclonable Function (PUF) are. This is assessed using statistical tests to determine if the responses exhibit characteristics of random sequences. High randomness is approaching the maximum entropy value. Randomness, or entropy, quantifies the unpredictability of the PUF output as described in Equation (3). For binary data, the maximum entropy value is 1 bit per bit of output. High randomness ensures that the PUF responses are difficult to predict, enhancing security. Ensuring that the PUF responses exhibit high randomness is crucial for reliable and secure hardware-based security solutions.

$$\text{Randomness (Entrophy)} = -\sum_{i} p_i \cdot \log_2(p_i) \qquad (3)$$

where p_i is the probability of occurrence of the i-th response value.

D. *Bias*

Bias, as described in Equation (4), measures the deviation of the PUF output from a uniform distribution. Specifically, it quantifies how far the proportion of 1s or 0s is from an equal distribution. Ideal Value: Low Bias, ideally close to 0 Explanation: Bias measures how uniformly the PUF output bits are distributed. Ideally, bias should be close to 0, meaning that the number of 1s and 0s are nearly equal. This ensures that the output is uniformly distributed and avoids predictability.

$$\text{Bias} = \left|\frac{P_1 - P_0}{P_1 + P_0}\right| \tag{4}$$

where P_1 is the proportion of 1s and P_0 is the proportion of 0s.

E. *Uniformity*

Uniformity measures the balance of '0's and '1's in a single PUF response using Equation (5).

$$\text{Uniformity} = \left(\frac{1}{L}\sum_{i=1}^{L} R_i\right) \text{X } 100\% \tag{5}$$

where R_i is the frequency of the i-th distinct response value and L is the total number of distinct response values. Uniformity should be close to 1, indicating that the distribution of 0s and 1s is as even as possible across the PUF responses.

III. METHODOLOGY

First, the initial startup values are collected from both the SRAM and SDRAM on FPGA boards. A total of 10 boards are used for this purpose. Next, based on the startup values and the characteristics of the Physical Unclonable Function (PUF), the optimal addresses and bit positions to be included in the PUF is determined; this step is required for SDRAM and is called SDRAM segment selection. Then, PUF characteristic analysis is performed to understand its performance.

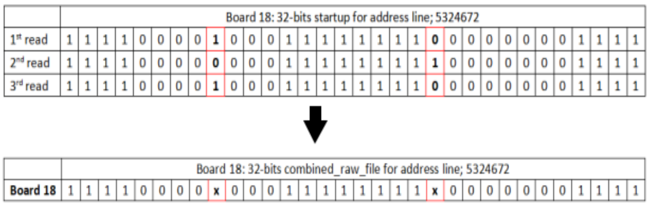
To ensure the quality of the SDRAM PUF, it is crucial to select appropriate segments of SDRAM to be included as the PUF. Given the substantial size difference between SRAM and SDRAM (which is much larger), we need to identify and choose specific SDRAM segments that possess good PUF characteristics. This process involves selecting and analyzing different segments to ensure accurate and reliable PUF performance evaluation based on the PUF evaluation metrics described in Section II. There are two methods used for selecting these SDRAM segments: majority voting and pattern analysis.

A. *SDRAM Segment Selection using Majority Voting*

Majority voting is used in the methodology to select stable bits of SDRAM to be used as PUF response. The method is as detailed in the following steps:

- Initial comparison of the data within the same board: For each board, collect multiple readings of the data bits at specific address lines. Compare these readings to detect any bit errors or inconsistencies. For an address line, if its data shows significant variability or changes across readings, consider it unreliable and eliminate it from further consideration as a potential PUF challenge line.
- Comparison of the data across the boards: Apply the same comparison and elimination process across all boards for the stable address lines identified in the first step. The stable address lines which show the lowest error rates on their data will form a set of reliable candidates for the PUF challenge.
- Pruning based on bit-level stability: Within the selected address lines from the previous step, assess the stability of each bit by evaluating its error rate. Choose the most stable bits (those with the lowest error rates) from these address lines to be used as the final PUF challenge. This ensures that the selected bits provide reliable and consistent PUF responses.

The raw data consists of 27 files of SDRAM data, one for each of the 9 boards, each of which is tested 3 times. These files are then combined into one file for each board to ease the comparison between the bit values at the same position obtained from the three repetitions of the test. If all three bits are the same, that value is written to the new file, otherwise, an "x" is used to indicate a mismatch. This process results in 9 files of combined SDRAM data, namely combined_raw_file and Fig. 1 illustrates the example of this step. This step is repeated for 9 boards.



	Board 18: 32-bits startup for address line; 5324672																															
1st read	1	1	1	1	0	0	0	0	**1**	0	0	0	1	1	1	1	1	1	1	1	**0**	0	0	0	0	0	0	0	1	1	1	1
2nd read	1	1	1	1	0	0	0	0	**0**	0	0	0	1	1	1	1	1	1	1	1	**1**	0	0	0	0	0	0	0	1	1	1	1
3rd read	1	1	1	1	0	0	0	0	**1**	0	0	0	1	1	1	1	1	1	1	1	**0**	0	0	0	0	0	0	0	1	1	1	1

	Board 18: 32-bits combined_raw_file for address line; 5324672																															
Board 18	1	1	1	1	0	0	0	0	**x**	0	0	0	1	1	1	1	1	1	1	1	**x**	0	0	0	0	0	0	0	1	1	1	1

Fig. 1. Generating combined_raw_file.

Next, the majority voting method is employed based on the PUF metric of BER to identify stable bits for use in challenges. The average number of "x" in each board and bit position is calculated, with the requirement that the average number of "**x**" in the same board is less than 6, "**a**" and the average number of "**x**" in different boards, "**b**" is less than 4. Table I shows the example of the placement of bits and the count of 'x' for one of the address lines (line: 5324672). The average of **a** and **b** are 6.33 and 1.73, respectively, therefore, was exampted from selection due to value **a** exceed the limit. This ensures that the selected bits are stable and reliable across multiple PUF responses. The process produces 1173 lines of 32-bit data for each board for each repetition. The next step involves selecting the bit positions to be used as PUF response for each of the address lines. This is done by comparing the mismatch count across all the boards for each bit position. This results in 32 tables, with each table corresponding to a bit position and the rows being the address lines. The tables are organized such that the address lines with the least number of "x" (mismatch count) values are at the top, while those with the most "x" values are at the bottom.

TABLE I. MISMATCHES FOR AN ADDRESS LINE OF DIFFERENT BOARDS.

	Bit Position																																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	a
Board 18	1	1	1	1	0	0	0	0	x	0	0	0	1	1	1	1	1	1	1	1	x	0	0	0	0	0	0	0	1	1	1	1	**2**
Board 47	0	0	0	1	0	0	1	0	1	1	0	0	1	1	0	1	0	1	0	1	x	1	0	x	0	0	0	0	1	1	x	0	**3**
Board 48	0	0	0	1	1	1	0	x	0	0	0	0	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	**1**
Board 49	x	1	x	0	0	0	0	x	x	1	x	x	0	0	0	0	1	x	1	1	x	x	0	x	1	1	1	1	0	x	x	x	**13**
Board 50	x	0	0	0	1	x	x	0	0	x	x	0	1	1	1	1	x	0	0	0	0	x	x	x	0	0	0	0	x	0	0	1	**10**
Board 51	1	1	1	1	0	x	x	x	x	1	x	x	0	1	0	0	1	0	1	x	1	1	1	1	1	1	x	1	1	1	1	1	**8**
Board 53	x	1	x	x	0	1	0	x	1	1	x	1	1	0	x	x	1	1	1	x	0	0	0	0	x	1	1	x	0	0	0	0	**10**
Board 54	1	1	x	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	x	0	0	1	x	1	1	0	0	0	0	**3**
Board 55	0	0	1	1	x	x	1	1	0	0	0	0	1	0	1	1	0	0	0	x	1	1	1	1	x	0	0	0	x	x	1	x	**7**
b	**3**	**0**	**3**	**1**	**1**	**3**	**2**	**4**	**3**	**1**	**4**	**2**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**3**	**3**	**3**	**1**	**3**	**2**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	

For each bit position, the address lines that have zero "x" values in every board are selected as the challenges for the PUF. In cases where there are multiple addresses with zero "x" values in the same line, one address is selected randomly. Table II shows the example of the addresses with the possible bit position to be used as challenges source. The addresses 25722602, 25751978 and 25722602 have zero mismatch for more than two bits but only one bit for the respective address will be selected. For instance, for address 25722602, bit at position 18 is selected while for address 25751978, bit at position 11 is selected.

TABLE II. BIT POSITIONS WITH ZERO MISMATCH FOR EACH ADDRESS.

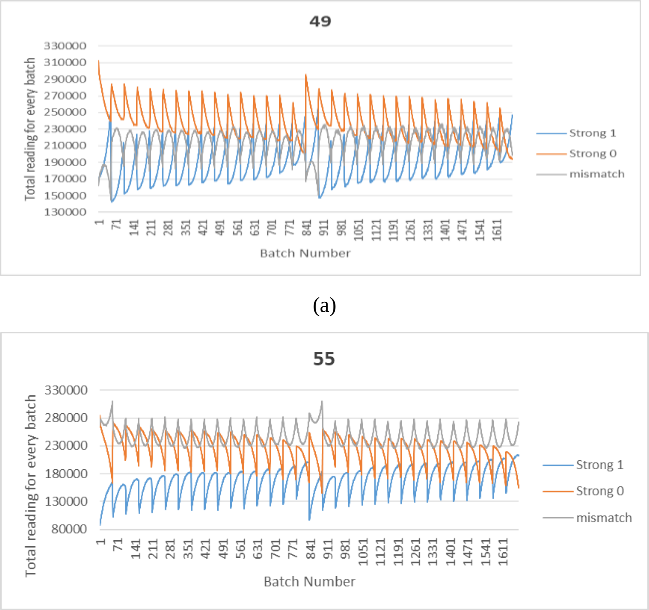
Address	Bit Position				
25722602	13	17	18	20	21
25751978	5	11	22	23	29
25754561	18	19	20		
25768384	13	23			
25770217	9				
25790378	5	22	30		
25722602	13	17	18	20	21

Finally, the challenges and their corresponding stable bit positions are compiled into a single file. There are 1079 challenges in total, which can be used to evaluate the quality of the PUF.

B. *SDRAM Startup Pattern Analysis*

Pattern analysis is performed on the start-up values of each board to evaluate the randomness and the uniqueness of the data and to determine the minimum number of address lines to be used in extracting the start-up. First, SDRAM start-up data is collected and analyzed to identify their recurring features or trends that can guide towards effective decisions about SDRAM segment selection, ensuring that the segments chosen for testing are representative and meaningful.

The SDRAM readings were conducted multiple times to ensure accuracy and reliability. Each reading was then grouped into batches of 200,000 lines each for systematic analysis; batch 1 starts from address 1 to address 200,000 and batch 2 starts from address 200,001 to address 400,000 and so on. As shown in Fig. 2, this process revealed distinct cyclical and repetitive patterns within the data, focusing on the sums of all ones, all zeros, and mismatches within each batch. These recurring patterns suggest systematic structures or behaviors in the SDRAM data. The data displayed a recurring structure with two main cycles, each comprising 15 smaller sub-cycles; for each of the sub-cycles, it consists of 71 batches. The graph shows that the pattern repeats for every sub-cycle. While the general pattern remained consistent across different boards, the specific shape of each cycle was uniquely characteristic of each individual board. This unique cyclic signature underscores SDRAM's potential to serve as a PUF source, highlighting its capacity to produce distinct and identifiable patterns suitable for secure key generation and authentication purposes.



(a)

(b)

Fig. 2. (a) Total bits of strong 1, strong 0 and mismatch of every batch for Board 49. (b) Total bits of strong 1, strong 0 and mismatch of every batch for Board 55.

When comparing the patterns of the two main cycles of Boards 50 and 51 side by side as shown in Fig. 3, focusing on the sum of mismatches (X), an interesting observation emerges; while each sub-cycle shows an almost similar pattern, there are clear differences between the first main cycle and the second main cycle. Further analysis involves comparing all sub-cycles side by side, examining the average, maximum, and minimum values for the sum of mismatches in Boards 50 and 51. This detailed comparison reveals different behaviors between the two boards. For Board 51, the pattern of maximum and minimum mismatches stays relatively constant throughout the cycles, indicating a stable formation. In contrast, board 50 starts with a large difference between maximum and minimum mismatches, which gradually converges to a smaller difference over batch. The address range used as challenges can be based on the sub-cycle size, as the pattern of the remaining sub-cycles corresponding to the remaining regions across the SDRAM show a similar pattern. Therefore, for PUF characteristic testing, the data tested are from address 211*200,000 to address 281*200,000 (a total of 71*200,000 addresses) which represent the average length of one sub-cycle.

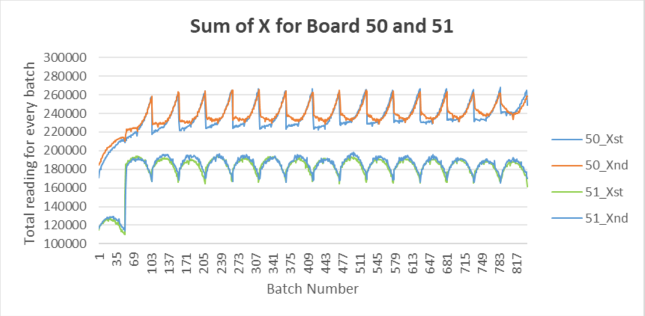


Fig. 3. Sum of mismatch bits for Boards 50 and 51 for two repetitions where 50_Xst denotes the first main cycle for Board 50 and 50_Xnd denotes the second main cycle for Board 50; similar goes to Board 51.

These findings highlight that each board has a unique and random startup value pattern, which can be used for PUF applications. The distinct cycle signatures and their variability across different boards show the potential of using SDRAM for secure key generation and hardware authentication. By using these cycle patterns, segmentation can be done by focusing on the range of one complete cycle to check the PUF quality. This approach ensures that the selected segments capture the unique and stable characteristics necessary for reliable PUF performance, enhancing the overall security and robustness of the system.

IV. Experimental Results and Discussions

This section discusses the reliability and robustness of the startup-based SRAM PUF and startup-based DRAM PUF using the PUF performance metrics described in Section II.

A. *Reliability and Robustness of the startup-based SRAM PUF*

Table III tabulates the computed PUF metrics for different SRAM on ATXmega128A1 labelled as 49, 50 and 120 respectively. The startup value readings of SRAM reveal that most quality metrics are close to their ideal values, highlighting the potential of SRAM for PUF applications. SRAM stands out due to its unique properties and advantages, especially when integrated into compact devices with limited space. Utilizing the existing RAM embedded within a device allows for the creation of a reliable and secure PUF without the need for additional hardware, thereby minimizing area overhead and reducing costs. One of the key benefits of SRAM-based PUFs is their inherent randomness and uniqueness, which are critical for generating secure cryptographic keys. Unlike other PUF methods that may require significant modifications or additions to the hardware, SRAM-based PUFs leverage the naturally occurring variations in the manufacturing process of the SRAM cells. Similar trends of results were obtained from the performance evaluation using Microchip ATmega328P.

TABLE III. PUF Performance for SRAM (8 KB) on Microchip ATXMega128A1 microcontroller boards 49, 50 and 120.

Board	49	50	120
Bias	0.24129	0.23879	0.24034
Randomness	0.95758	0.95847	0.95792
Uniformity	0.62065	0.61940	0.62017
BER	0.05071	0.05017	0.05489
Uniqueness	0.45537	0.45516	0.45997

TABLE IV. PUF Performance for SRAM (4 KB) on Microchip ATmega328P microcontroller boards A0, A1, A4 and A5.

Board	A0	A1	A4	A5
Bias	0.18938	0.14392	0.18599	0.18828
Randomness	0.97390	0.98491	0.97486	0.97427
Uniformity	0.59469	0.57196	0.59299	0.59414
BER	0.01924	0.01813	0.01829	0.01899
Uniqueness	0.48888	0.49402	0.49155	0.48020

The search space of a PUF, often referred to as its "challenge-response space" or "CRP space," plays the role of enhancing security. A vast CRP space makes it exceedingly difficult for attackers to hack the PUF, as the length of the key can be significantly increased by using longer bit sequences. The immense CRP space provided by SRAM-based PUFs is crucial for protecting sensitive information in modern electronic devices. Considering the theoretical potential, SDRAM, with its significantly larger size, can offer an even greater CRP space than SRAM. This means that SDRAM-based PUFs could theoretically provide an even higher level of security due to the larger number of possible responses. Therefore, we explored the possibility of startup-based SDRAM PUF in the experiment elaborated in the following section.

B. *Reliability and Robustness of the startup-based SDRAM PUF: Majority Voting*

SDRAM, known for its high density and speed, differs significantly from SRAM in that it relies on capacitors and requires periodic refreshing and initialization. This fundamental difference presents a challenge when evaluating SDRAM's potential as a PUF source. To address this, we first perform PUF quality tests on SRAM to establish a controlled reference. Given the significant size difference between SRAM and SDRAM, this study aims to determine an appropriate range or space within SDRAM for testing, ensuring a fair comparison and assessing SDRAM's viability as a PUF source.

Using the majority voting method, the segmentation process can be refined to identify 1079 key segments. These segments, or potential bits, were selected based on their reliability and consistency across different readings and boards. The average number of mismatches ("x") in each board and bit position was calculated with two specific criteria: the average number of mismatches within the same board must be less than 6, and the average number of mismatches between different boards must be less than 4. This precise approach ensures that the identified segments are highly reliable and exhibit minimal variability, making them ideal candidates for PUF testing. The results of BER and Bias in Table V demonstrate the quality of the SDRAM PUF based on the first three readings to evaluate the robustness of the selected SDRAM segments. Table VI displays the results in uniqueness and randomness of the SDRAM PUF across the different boards. The readings were repeated 10 times across 10 different boards, focusing on previously identified key positions.

TABLE V. PUF PERFORMANCE IN BER AND BIAS FOR SDRAM OF 10 DIFFERENT BOARDS.

Board	BER	Error Rate (%)	Board	Biasness	Error Rate (%)
18	0.10078	10.1	**18**	0.39844	20.3
47	0	0.0	**47**	0.49219	1.6
48	0.06094	6.1	**48**	0.44531	10.9
49	0.00313	0.3	**49**	0.34375	31.3
50	0.00703	0.7	**50**	0.55469	10.9
51	0.07578	7.6	**51**	0.38281	23.4
52	0.08047	8.0	**52**	0.40625	18.8
53	0.02891	2.9	**53**	0.52344	4.7
54	0.00703	0.7	**54**	0.4375	12.5
55	0.03281	3.3	**55**	0.28125	43.8
Ave.	0.04	3.969	**Ave.**	0.427	14.688

TABLE VI. PUF PERFORMANCE IN UNIQUENESS AND RANDOMNESS FOR SDRAM OF 10 DIFFERENT BOARDS.

Repetition	Uniqueness	Error Rate (%)	Repetition	Random-ness	Error Rate (%)
1	0.51769	3.5	**1**	0.90426	9.6
2	0.52760	5.5	**2**	0.87373	12.6
3	0.52418	4.8	**3**	0.89381	10.6
4	0.52371	4.7	**4**	0.89181	10.8
5	0.53344	6.7	**5**	0.87868	12.1
6	0.50300	0.6	**6**	0.84990	15.0
7	0.53411	6.8	**7**	0.84334	15.7
8	0.53810	7.6	**8**	0.82825	17.2
9	0.53846	7.7	**9**	0.81981	18.0
10	0.53238	6.5	**10**	0.87168	12.8
Ave.	**0.527**	**5.454**	**Ave.**	**0.866**	**13.447**

The results show that SDRAM PUF is good in randomness and uniqueness, and satisfactory in biasness. However, BER varied significantly across the boards, as illustrated in Table V. Being consistently low BER is a crucial quality for PUFs, which are intended to generate secure keys. Inconsistent BER quality can compromise the system's reliability and security, potentially causing vulnerabilities even before an attacker attempts to exploit them. This inconsistency highlights the need for a thorough analysis and optimization of SDRAM segments to ensure robust and dependable PUF performance.

C. *Reliability and Robustness of the startup-based SDRAM PUF: Pattern Analysis*

The repeated patterns of ones, zeros, and mismatches reveal that certain characteristics of SDRAM memory can be leveraged for effective segmentation. This detailed analysis deepens the understanding of SDRAM's randomness and uniqueness, highlighting any periodic behaviors or anomalies that could affect testing accuracy and reliability. The results presented below showcase the quality of the PUF using pattern analysis segmentation. The analysis indicates high-quality segments that, in principle, could be used as PUF sources as confirmed again in the segmentation results in Table VII. However, the Bit Error Rate (BER) values present a significant issue. Some boards exhibit BER values exceeding 50%, indicating that the readings vary substantially with each repetition. This inconsistency in BER values is like the findings from the SDRAM segmentation, which also revealed significant variability. These results suggest that while SDRAM shows potential in terms of pattern consistency, the high BER values undermine its reliability as a PUF candidate.

TABLE VII. PUF PERFORMANCE FOR SDRAM OF BOARDS 49, 52, 54 AND 55.

File	49	52	54	55
Bias	0.13614	0.05306	0.01783	0.14844
Randomness	0.98552	0.99796	0.99968	0.98370
Uniformity	0.43193	0.47347	0.49303	0.42578
Uniqueness	0.45211	0.50286	0.45999	0.48430
BER	**0.25425**	**0.57936**	**0.26882**	**0.44306**

D. *SDRAM PUF as Obfuscation Seed*

Utilizing the strengths of SDRAM PUF (strong uniqueness and randomness) and the weakness (poor BER), we recommend SDRAM PUF to be used in generation of obfuscation seeds. The high BER can be leveraged to increase the unpredictability of the obfuscation seeds in a secure system to jumble the important or secret information. This increases the difficulty of attackers stealing the information. In some secure applications such as secure scan architecture, the scan data which is obfuscated when a wrong key is provided does not need a reversed engineering to obtain the original scan data. Thus, higher BER can improve the robustness of the obfuscation while reducing the recognizability.

V. CONCLUSION

This study evaluated the reliability and robustness of startup-based SRAM and DRAM PUFs by analyzing metrics including bias, randomness, uniformity, and uniqueness.

SRAM-based PUFs demonstrate strong performance with metrics close to ideal values, offering excellent randomness and uniqueness that are critical for secure key generation. In contrast, while SDRAM PUFs also show strong randomness and uniqueness, their high Bit Error Rate (BER) across different boards presents a significant challenge, impacting their reliability and security. Despite this, SDRAM PUFs can be effectively used for obfuscation purposes, leveraging high BER to increase unpredictability and enhance the robustness of security mechanisms. Both SRAM and SDRAM PUFs have unique advantages and can be strategically applied in various security contexts.

Acknowledgment

This research is partly supported by Fundamental Research Grant Scheme, FRGS/1/2022/TK07/UTM/02/23 under Malaysian Ministry of Higher Education.

References

[1] P. Ahr, M. Noushinfar, and C. Lipps, ‘RAM-Based PUFs: Comparing Static-and Dynamic Random Access Memory’, in Workshop on Next Generation Networks and Applications, 2021.

[2] W. Wang, A. D. Singh, and U. Guin, “A Systematic Bit Selection Method for Robust SRAM PUFs,” Journal of Electronic Testing, vol. 38, no. 3, pp. 235–246, Jun. 2022, doi: https://doi.org/10.1007/s10836-022-06006-x.

[3] G. Torrens, A. Alheyasat, B. Alorda, and S. A. Bota, “SRAM-Based PUF Reliability Prediction Using Cell-Imbalance Characterization in the State Space Diagram,” Electronics, vol. 11, no. 1, p. 135, Jan. 2022, doi: https://doi.org/10.3390/electronics11010135.

[4] F. Najafi, M. Kaveh, D. Martín, and M. Reza Mosavi, “Deep PUF: A Highly Reliable DRAM PUF-Based Authentication for IoT Networks Using Deep Convolutional Neural Networks,” Sensors, vol. 21, no. 6, p. 2009, Mar. 2021, doi: https://doi.org/10.3390/s21062009.

[5] J. Miskelly and M. O’Neill, “Fast DRAM PUFs on Commodity Devices,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 11, pp. 3566–3576, Nov. 2020, doi: https://doi.org/10.1109/tcad.2020.3012218.

[6] M.-S. Kim et al., “Error reduction of SRAM-based physically unclonable function for chip authentication,” International Journal of Information Security, vol. 22, no. 5, pp. 1087–1098, Feb. 2023.

[7] Chen, B., et al. A Robust SRAM-PUF Key Generation Scheme Based on Polar Codes. in GLOBECOM 2017 - 2017 IEEE Global Communications Conference. 2017.

[8] S. S. Kudva et al., "16.4 High-Density and Low-Power PUF Designs in 5nm Achieving 23× and 39× BER Reduction After Unstable Bit Detection and Masking," 2024 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2024, pp. 302-304.

[9] M. Laban and Milos Drutarovsky, “Improved Efficiency of PUF Response Reconstruction Method,” Apr. 2020.

[10] Neale, A. and M. Sachdev. A low energy SRAM-based physically unclonable function primitive in 28 nm CMOS. in 2015 IEEE Custom Integrated Circuits Conference (CICC). 2015.

[11] M. Gong, H. Zhang, C. Wang, Q. Tong, and Z. Liu, “Design and implementation of robust and low-cost SRAM PUF using PMOS and linear shift register extractor,” Microelectronics Journal, vol. 103, pp. 104844–104844, Sep. 2020.

[12] Liu, H., et al., *Methods for Estimating the Convergence of Inter-Chip Min-Entropy of SRAM PUFs.* IEEE Transactions on Circuits and Systems I: Regular Papers, 2018. 65(2): p. 593-605.

[13] A. Ali Pour et al., "Helper Data Masking for Physically Unclonable Function-Based Key Generation Algorithms," in IEEE Access, vol. 10, pp. 40150-40164, 2022.

[14] K. Liu, X. Chen, H. Pu and H. Shinohara, "A 0.5-V Hybrid SRAM Physically Unclonable Function Using Hot Carrier Injection Burn-In for Stability Reinforcement," in IEEE Journal of Solid-State Circuits, vol. 56, no. 7, pp. 2193-2204, July 2021.

[15] C.-H. Chang, Chao Qun Liu, L. Zhang, and Zhi Hui Kong, “Sizing of SRAM Cell with Voltage Biasing Techniques for Reliability Enhancement of Memory and PUF Functions,” Journal of Low Power Electronics and Applications, vol. 6, no. 3, pp. 16–16, Aug. 2016.

[16] Zhang, S., et al. Evaluation and optimization of physical unclonable function (PUF) based on the variability of FinFET SRAM. in 2017 International Conference on Electron Devices and Solid-State Circuits (EDSSC). 2017.

[17] Narasimham, B., et al. SRAM PUF quality and reliability comparison for 28 nm planar vs. 16 nm FinFET CMOS processes. in 2017 IEEE International Reliability Physics Symposium (IRPS). 2017.

[18] Liao, Z. and Y. Guan. The Cell Dependency Analysis on Learning SRAM Power-Up States. in 2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). 2018.

[19] Liao, Z., et al. The impact of discharge inversion affects learning SRAM power-up statistics. in 2017 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). 2017.

[20] Z. Liao and Y. Guan, ‘Rudba: Reusable user-device biometric authentication scheme for multi-service systems’, in 2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2021, pp. 214–225.

[21] J. Lee, D.-W. Jee, and D. Jeon, ‘Power-up control techniques for reliable SRAM PUF’, IEICE Electron. Express, vol. 16, p. 20190296, 2019.

[22] Pyi Phyo Aung, Nordinah Ismail, Chia Yee Ooi, Koichiro Mashiko, Hau Sim Choo, and Takanori Matsuzaki, “Data Remanence Based Approach towards Stable Key Generation from Physically Unclonable Function Response of Embedded SRAMs using Binary Search”, *J. Adv. Res. Appl. Sci. Eng. Tech.*, vol. 35, no. 2, pp. 114–131, Dec. 2023.

[23] A. Santana-Andreo, P. Saraza-Canflanca, R. Castro-Lopez, E. Roca, and F. V. Fernandez, “Reliability improvement of SRAM PUFs based on a detailed experimental study into the stochastic effects of aging,” AEU - International Journal of Electronics and Communications, vol. 176, pp. 155147–155147, Mar. 2024.

[24] A. Kumar, None Manoj Sindhwani, and S. Sachdeva, “An Innovative Architecture of DRAM PUF,” Journal of Integrated Circuits and Systems, vol. 18, no. 2, pp. 1–9, Sep. 2023, doi: https://doi.org/10.29292/jics.v18i2.675.

[25] Y. Zheng, Z. Huang, L. Li, C. Xie, Q. Wang, and Z. Wu, “Implementation and Analysis of Hybrid DRAM PUFs on FPGA,” Oct. 2021, doi: https://doi.org/10.1109/nana53684.2021.00074.