Systematic Correlation and Cell Neighborhood Analysis of SRAM PUF for Robust and Unique Key Generation

M. Tauhidur Rahman[1] · Alison Hosey[3] · Zimu Guo[2] · Jackson Carroll[2] · Domenic Forte[2] · Mark Tehranipoor[2]

Received: 27 July 2017 / Accepted: 15 August 2017 / Published online: 5 September 2017
© Springer International Publishing AG 2017

Abstract A physical unclonable function (PUF) is a structure that produces a unique response, with an issued challenge (input), which can be used as an identifier or a cryptographic key. SRAM PUFs create unique responses upon power up as certain SRAM cells output a "1" or "0" with high probability due to uncontrollable process variations. A current challenge in SRAM PUFs is their sensitivity to temperature and voltage variations as well as aging. It is always challenging to make SRAM PUFs reliable and unique with algorithms that isolate stable and uncorrelated bits quickly with minimal testing (enrollment). In this paper, we explore the selection of stable and uncorrelated bits through enrollment under different conditions (temperature and voltage) and also by exploiting previously undiscovered interactions between neighboring SRAM cells. We propose neighbor influenced cell selection algorithm (NICSA) with the help of metrics that analyze the impact of each neighboring cell and each enrollment condition. The proposed NICSA helps to identify the "best" cells and conditions for stable bit selection. Besides reliability, SRAM PUF can be less unique due to systematic correlation among chips. We study the systematic correlation between SRAMs power-up values to find the uncorrelated cells among chips for better uniqueness. We have analyzed data from 5 ISSI, 3 IDT, and 3 Cypress SRAMs and our metrics identify the best neighborhood size (16 stable neighbors) and best enrollment condition pair high temperature, high voltage, and low temperature for NICSA.

Keywords SRAM-PUF · Low-cost PUF · SRAM PUF reliability · SRAM PUF neighborhood · Robust SRAM PUF

1 Introduction

A physical unclonable function is an emerging potential security block for generating volatile secret keys in cryptographic applications [1, 2, 4–8]. A physical unclonable function (PUF) is an umbrella term used for hardware primitives that use their physical characteristics to perform authentication, identification, counterfeit detection, and volatile cryptographic key generation [2, 4–8]. A PUF is described as unclonable due to its uniqueness being derived from the uncontrollable variations introduced during the manufacturing process. PUFs offer a high level of protection in cryptographic applications with strong volatile key storage. PUFs are issued a challenge and (ideally) produce a unique and reliable response in return. Since this response is unique to the device, it can, therefore, be used

✉ M. Tauhidur Rahman
tauhidur.rahman@uah.edu

Alison Hosey
alison.hosey@uconn.edu

Zimu Guo
zimuguo@ufl.edu

Jackson Carroll
jacksonecarroll@ufl.edu

Domenic Forte
dforte@ece.ufl.edu

Mark Tehranipoor
tehranipoor@ece.ufl.edu

[1] University of Alabama in Huntsville, Huntsville, AL, USA

[2] University of Florida, Gainesville, FL, USA

[3] University of Connecticut, Storrs, CT, USA

as a device ID or key. Unlike previous methods, PUFs are less vulnerable to attacks, and also require no additional manufacturing steps. A variety of different types of PUFs has been explored recently, including Arbiter PUF, ring-oscillator (RO)-PUF, SRAM PUF, Latch PUF, and many more [2, 3, 18, 19, 21–23, 43]. This paper explores utilizing the popular SRAM PUF for identification and cryptographic key generation. SRAM PUF offers the convenience of using commonly available and integrated SRAM (instead of including a dedicated hardware in the circuit) as well as the capacity to provide large enough outputs for identifier/key generation/storage [5, 8].

SRAM is ubiquitous, and a critical block in modern FPGA and system-on-chip use the smallest possible device sizes in any given technology [5, 8]. Although SRAM PUF offers many appealing features, there are two main challenges to its current application. First, SRAM PUF is quite sensitive to noise generated by temperature and voltage level variations [4, 8]. This sensitivity is not unique to SRAMs and can be seen in a variety of electronics due to physical phenomena which alter threshold voltages and other properties [3, 7, 20]. Second, SRAM PUF also experiences the effects of aging on the reliability of the output [8]. Previously, error correcting code (ECC) [7, 8, 20, 33, 34] had been used to repair errors in the SRAM output prior to use as a cryptographic key. Unfortunately, ECC creates a substantial amount of overhead while also can be manipulated for extracting keys [20, 32, 33]. However, it has been proved in [35, 36] that the fuzzy extractors can generate information theoretically secure cryptographic keys even if the helper data leaks information. Alternatively, exhaustive measurements of the SRAM PUF can be used to identify the most reliable cells [4, 8]. Eiroa et al. [4] is a SRAM cell selection algorithm where cells are selected in two steps. Most stable words are selected in the first step. In the second step, the most stable cells are selected from each selected stable word. However, the result shows that a media of only 2 out of the 14 words are chosen. In other words, around 86% of total resources are wasted. Also, $\sim$ 3.51% of total cells from selected words also flip due to environmental variations. A lot of resources are wasted due to no cell-by-cell analysis in their proposed method ([4]). They also lack reliability analysis after aging the SRAMs. On the other hand, a PUF also has to be unique for high-volume production. Systematic and/or spatial correlation [26, 28] can hamper the uniqueness of a PUF. Different PUFs might produce the almost same response (key) for a given challenge due to systematic and/or spatial correlation [30].

In our previous work, we have identified that neighboring SRAM cells can be used to determine the more reliable SRAM PUF cells [8]. Based on our observation, we presented a simple bit selection algorithm. While the SRAM PUF reliability improved dramatically compared to random bit selection, our approach was mostly operating in a blind fashion. Besides, we lack a technique to select the uncorrelated cells among different chips that maximize the PUF uniqueness. In this paper, we perform more advanced analysis to identify the "best" cells and conditions. Our main contributions include:

1. Development of three new metrics to analyze the relationships between neighboring SRAM cells in one dimension – 1D (i.e., assuming that the SRAM data is organized as one long array) and the influence of environmental conditions.
2. Utilization of these proposed metrics, real SRAM data (over 1 billion measurement bits) are examined in greater detail, and the effect of different temperature, voltage, and aging conditions on reliability can be more easily categorized. We determine the optimal window and threshold sizes. We also identify the most effective measurement conditions for initial enrollment.
3. We propose a neighbor influenced cell selection algorithm (NICSA) by choosing the optimal window size and threshold with minimal test time to obtain low-cost, robust SRAM PUF.
4. We study the SRAM cells' power-up values across different SRAM boards to make the PUF unique by selecting uncorrelated cells. The selection of uncorrelated cells makes the PUF unique (inter-hamming distance is 47.34%, close to ideal value 50%).
5. We also explore low-cost enrollment with the help of NICSA.
6. We evaluate our proposed algorithm for three SRAM vendors (ISSI (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf), IDT (http://www.idt.com/), and Cypress (http://www.cypress.com/file/43006/download)).

The rest of the paper is organized as follows. In Section 2, we describe the details of SRAM PUF, its application, cell characteristic, PUF quality metrics for robust and unique PUF, and existing work to make SRAM PUF robust and unique. SRAM PUF reliability factors and the proposed metrics for neighboring SRAM cell analysis in 1D are described in Section 3. We discuss the systematic correlation and uncorrelated cell selection method in Section 4. In Section 5, we present our proposed NICSA-based bit selection approach for robust key generation. Section 6 will provide the results of our experimental test setup. The conclusions are drawn from this work as well as intended future work will be given in Section 7.

2 Preliminaries

2.1 SRAM PUF Architecture

In recent years, there has been much investigation into SRAM PUF as a simple but effective form of hardware-based identification and key generation/storage. The usage of SRAM as the PUF medium is appealing for a variety of reasons. Most notably, SRAM is commonly available in most systems and therefore does not require additional hardware. The uniqueness of output from one SRAM compared to another is also the largest among existing PUFs [3, 4]. For a standard 6T SRAM, every memory cell is composed of six transistors that are two cross-coupled CMOS inverters and two access transistors. The inverters are designed to be symmetric, a match in size, etc., but random variations incurred during manufacturing will result in random mismatches. SRAM PUFs exploit the mismatch which results in each SRAM cell being biased (or skewed) toward a zero or one at power-up. Due to uncontrollable variations in the manufacturing process, different CMOS devices have different physical parameters (e.g., doping-levels, transistor oxide thickness, etc.). When a SRAM is powered-up, these variations affect the power-up state of their associated cells. It has been observed that certain cells have a strong "preference" to power-up to a "1" or "0' state. Cells that have no "preference" are deemed neutral and power-up at random depending upon the influences of system noise [5]. The more useful cells for PUF output are the ones that strongly prefer "0" or "1." However, it has been shown in [37] that not all platforms can be used as PUF.

By examining the power-up state of a SRAM, a unique identifier can be created because the process variations and resultant preferences are truly random (being entirely dependent upon a physical anomaly). A SRAM PUF identification system would be similar to the fingerprint security measures found in biometrics. When fingerprints are used for identification, the fingerprint is a unique identifier which is compared to other fingerprints in a database and subsequently accepted or rejected based upon its authenticity. The SRAM PUF output (which is ideally a unique and reliable response) would then be compared to a known SRAM PUF response database and verified. Alternatively, the SRAM PUF response can be used as a volatile cryptographic key, which is that target of this paper.

It is required that the SRAM PUF output must remain unchanged over time in different operating conditions (change in voltage and temperature). The robustness of SRAM PUF depends on the mismatch in strength between two cross-coupled inverters. The SRAM cells can be divided into two categories based on the power-up value which solely depends on the strength between two cross-coupled inverters in SRAM cell [5]:

- Neutral cell: The difference of strength between two cross-coupled inverters in a SRAM cell is almost negligible. The power-up value of a cell might depend on the measurement noise and random, "1"/"0" (as shown in Fig. 1).
- Skewed cell: The difference in strength between two cross-coupled inverters in a SRAM cell is significant to produce "0" or "1" during power-up. Some cells possess little process variation and produce weak "0" or weak "1." Among many technology parameters, threshold voltage V_{th} of transistors has the most impact on the start-up value of a SRAM cell [17]. The intra-die variability increases process variability as CMOS technology node is shrinking. The intra-die variability vs. CMOS technology node has been reported in [25]. This variability in the manufacturing processes is responsible for increasing the difference of strength between two cross-coupled inverters and improving SRAM PUF quality. These cells are known as partially skewed cells. Besides, the power-up values of these cells are impacted by the measurement noise, temperature and voltage fluctuations, and aging. These cells can be used for TRNG and PUF for applications in identification where error can be tolerated. On the other hand, a strong mismatch between cross-coupled inverters in a SRAM cell might produce a strong "0" or a strong "1" (as shown in Fig. 1). These cells can tolerate more noise and are ideal candidates when a PUF is used for cryptographic keys.

The major difference between skewed cells and neutral cells is that the difference of strength between two cross-coupled inverters. Skewed SRAM cell has a preference to "1" or "0" because one inverter is stronger than the other. The robustness of a cell depends on the difference in strength between cross-coupled inverters. On the other hand, the neutral cells do not have any preference. The neutral cells are usually used to generate the true random number.

2.2 PUF Quality Metrics

Reliability, *uniqueness*, *randomness*, and *bit-aliasing*, are the popular metrics to assess PUF quality. Fractional Hamming distance, the ratio between the number of bit differences between two same-length bit strings and the length of the string, is used to measure the quality of PUFs.

- The *reliability* of a PUF determines how often a PUF can generate the same response to a given challenge.

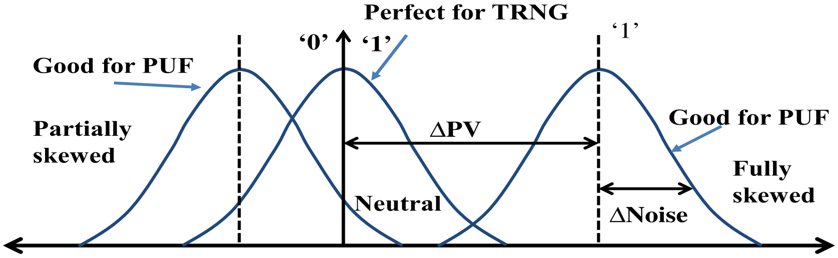


Fig. 1 Characteristic of SRAM cells' power-up state

A PUF must generate the same response at all operating conditions in every power-up cycle during its entire lifetime.

- *Uniqueness* measures how well a single PUF is differentiated from other PUFs based on its challenge-response pair. Different PUFs must generate different responses to a given challenge in order to separate one from another. The average inter-chip fractional Hamming distance for an ideal PUF must be 0.5. Different chips may produce nearly identical PUF responses due to systematic variations and this is measured by *bit-aliasing*. The bit-aliasing rate measures the distribution of "0"s and "1"s at particular bit position among PUFs. We estimate the bit-aliasing of the lth bit as the percentage Hamming weight (HW) of the lth bit across M devices:

$$(bit - aliasing)_l = \frac{1}{M}\sum_{i=1}^{M} r_{i,l} \qquad (1)$$

where $r_{i,l}$ is the lth binary bit of an n-bit response from a chip i. Average bit-aliasing for an ideal PUF is 0.5. *Diffusion* is used to measure the uniqueness of a strong PUF. This metric measures the variance of a CRP space of a strong PUF [30].
- *Randomness* measures the unpredictability of the responses of a PUF. *Randomness* also helps to determine whether a PUF is biased or not. For an unbiased PUF, changing one bit in a challenge should alter nearly half of the bits of the response. NIST's statistical test suite is popularly used to evaluate the quality of randomness for random and pseudorandom number generators designed for cryptographic applications, and can also be used for PUF [23, 29].

More details on PUF quality metrics can be read in [30].

2.3 Related Work

To ensure the reliability of the SRAM PUF response, the output needs to either be error corrected in post-processing or analyzed in such a way that only stable cells are used. ECC uses specialized encoding and decoding processes to ameliorate data instability. Unfortunately, such processes create considerable overhead for implementation. Soft-decision coding scheme, a low-cost ECC scheme for SRAM-PUF, requires lesser area than hard-decision coding [10, 11]. In soft-decision scheme, reliability information is provided with bit-value in order to improve the error correcting capability. But, soft-decision ECC schemes require large amount of PUF bits (e.g., 128-bit key requires $\sim 5K$ SRAM cells [10, 11]). The number of PUF bits increase significantly with the number of errors (e.g., the PUF size and complexity increase by $\sim\ x2.6$ if the error probability changes to 5 from 15% [17]). The repetition codes are heavily used in soft-decision fuzzy extractor due to their simplicity [40]. However, repetition codes are over-optimistic in entropy estimation and may not be suitable for high-security applications [40]. In fuzzy extractors, significant amount of entropy is lost from the helper data during recovery of PUF response in fuzzy extractors [41]. Losing a large amount of entropy, by correcting a large amount of errors, can make the PUF useless [41, 42]. However, it is claimed in [35] that the fuzzy extractors are able to generate information theoretically secure cryptographic keys even if the helper data leaks information. To reduce the overheads, an alternative is to reduce sensitivity to temporal variations. For instance, a reliability enhancement [16] was developed to reinforce the preferred value of SRAM cell by inducing accelerated aging. But performing burn-in stress for 120 h for one SRAM PUF is prohibitively expensive, time-consuming, and in turn degrades the SRAM. Another proposed approach modifies the V_{DD} ramp-up time to make cells more reliable [17], but this approach requires special circuitry not present in standard SRAM. RESP, proposed in [15], utilizes voltage scaling induced access failures in SRAM array to generate a large set of robust signatures. Garg et al. proposed a technique, in [14], that controls the polarity of the aging in SRAM arrays to make the key uniform. Their proposed technique controls the reliability by further injecting aging to the SRAM arrays after achieving target uniformity. Hofer et al. proposed a technique that selects cells that provide a high mismatch between their crucial transistors are selected for lowering the error rate [13]. The robustness of DRV-based SRAM PUF was proposed in [45]. The robustness of design dependent SRAM PUF was presented in [45]. Data-dependent (anti-) aging was proposed in [12] for robust keys. The robustness of glitch-PUF was proposed in [43].

We originally proposed a SRAM bit selection algorithm in [8]. The basis of this algorithm is to identify the most stable bits based upon the performance of their neighboring cells in a series of enrollment tests (testing under some extreme corner conditions). The algorithm finds the cells which consistently only output a value of “0” or “1” and then ranks their overall stability using a weighting algorithm. The weighting algorithm takes into account the number of stable and unstable neighbors a cell has. Essentially, the farther away a cell is from the nearest unstable cell (weight 0), the higher weighted value it is assigned. As shown in Fig. 2, stable bits (labeled S) and unstable bits (labeled U) are assigned based upon these calculated stability weights. The cells that have a weight greater than a predefined threshold are selected as the most stable ones. While the measurements used in the algorithm rely only upon fresh (un-aged) SRAM, results showed that the algorithm was most successful in identifying cells that were robust against aging (stable over the lifetime of the device).

3 SRAM PUF Reliability

In this section, we explain SRAM PUFs’ neighborhood-based dependency observed in [8, 9]. We explore the reasons of correlation among neighbor cells. We model SRAM PUF error and uniqueness based on spatial and systematic correlations. We also explore the bit-selection metrics for robust SRAM PUF in this section.

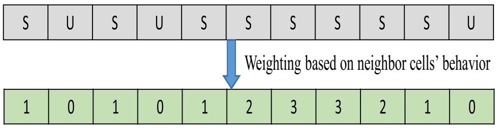
3.1 Neighborhood-based Error Mechanism

The physical distance between adjacent memory cells is decreasing dramatically due to the demand of low-area and low-power systems. Technology is shrinking, and decreasing space between cells makes the value of coupling capacitance dominant [26, 28]. This coupling capacitance causes two types of capacitive cross-talk: cross-talk noise and cross-talk delay. A cross-talk effect can introduce a negative/positive glitch in logic “1”/“0” in the victim lines due to the transition at the aggressor lines. The magnitude of this glitch can cause the cell flips. Besides the magnitude, a wider glitch can drive the load capacitance to a different logic value. The effect of cross-talk is aggravated by a higher operating speed.

Ideally, a PUF has to be unique and robust/reliable overall operating conditions in the entire chip lifetime. One cell’s power-up value can be impacted by the neighbor cells due to cross-talk noise among SRAM cells. The systematic correlation can make the PUF less unique. However, the output of a SRAM cell should not be correlated to neighbor cells. A PUF output might be erroneous due to system noise and aging. Cross-coupling and spatial correlation can degrade the quality of a PUF even more. The PUF output might be erroneous because a cell might be influenced by noisy neighbor cells. On the other hand, spatial correlation can make the output of neighbor cells correlated. This spatial correlation can hamper both intra-chip and inter-chip randomness.

Figure 3 shows that two physically adjacent SRAM cells are coupled through a coupling capacitor, Cap_c. However, the cell coupling capacitor can be present at any location. The top SRAM cell represents the victim cell, and the bottom SRAM cell is an aggressor cell. Most noisy cells can be considered as aggressor cells, and the other cells (target cells) can be considered as victim cells. The instantaneous voltage change across Cap_c between aggressor cell and victim cell causes current i_{cap}. The current through victim and aggressor cell can be expressed as [27]:

$$i_{cap} = Cap_c \frac{d(V_{Q_{VR}} - V_{Q_{AR}})}{dt} \quad (2)$$



S	U	S	U	S	S	S	S	S	S	U
1	0	1	0	1	2	3	3	2	1	0

Weighting based on neighbor cells’ behavior

Fig. 2 Bit selection weighting algorithm (top: stable (S) and unstable (U) bits; bottom: associated weighted value)

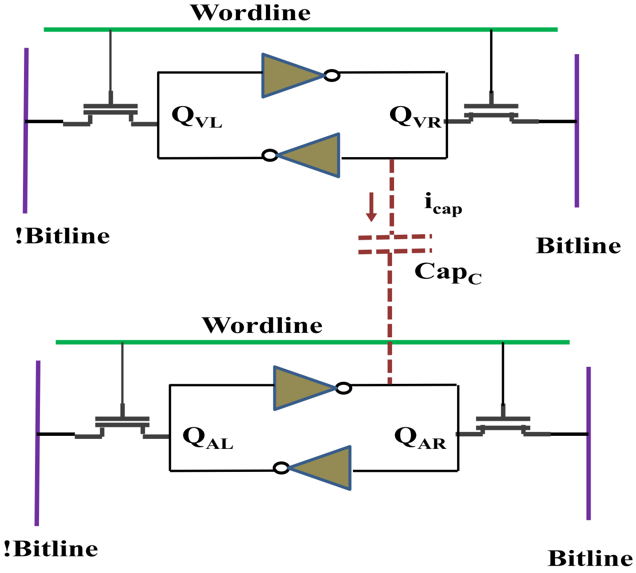


Fig. 3 Coupling capacitance impacts the behavior of physically adjacent cells

where $V_{Q_{VR}}$ is the voltage of victim cell's right side node Q_{VR} and $V_{Q_{AR}}$ is the voltage of aggressor cell's right-side node Q_{VR}. The current through coupling capacitor Cap_c. Equation 2 shows that the current through a coupling capacitor depends on the coupling capacitance Cap_c and can change the power-up value of a victim cell if the coupling capacitance is larger than a certain threshold value. This threshold value changes technology to technology [27].

Table 1 summarizes the mechanism of flipping a target (victim) cell's power-up value by neighbor (aggressive) cell(s). The power-up values of a target cell might change due to the noisy behavior of neighbor cells. The neighborhood dependency can be summarized with the help of cross-coupling capacitance among the neighbor cells. The power-up value of $Q_{VR} = 0$ might change to "1" when the power-up value of Q_{AR} changes from "0" to "1" from one measurement to next (see Fig. 3) because of cross-talk energy through Cap_c. Similarly, the power-up value of $Q_{VR} = 1$ might change to "0" when the power-up value of Q_{AR} changes from "1" to "0" from one measurement to next.

In addition, the bit-line load, charge injection into the SRAM cell, leakage, and noise coupling can make the start-up values of the SRAM PUF less robust [44]. The length of the wordline can cause the errors in memory [44, 46]. Besides, changing the voltage of a wordline could inject noise into an adjacent wordline through electromagnetic coupling as well [44, 46]. All of these could impact the PUF response.

3.2 Bit Selection Metrics for Robust SRAM PUF

While Section 3.1 tells us that cells should be most influenced by their neighbors, we do not necessarily know the physical locations of all the cells (proprietary information) in memory because SRAM vendors tend not to supply the physical information [38]. Hence, we shall assume 1D analysis for simplicity. 1D analysis means that the SRAM cells are assumed to be ordered in one long array based on their logical address. One benefit of our metrics is that they may end up exposing the physical locations based on the correlations between cells they capture.

We begin by defining some notation and important variables. Let C represent the conditional probability of a bit not flipping; W is the window size used to assess the neighborhood sum, and T is the threshold weight of a neighborhood sum (Fig. 4). "1" and "0" represent "stable" and "unstable" cells, mentioned in Fig. 2, respectively.

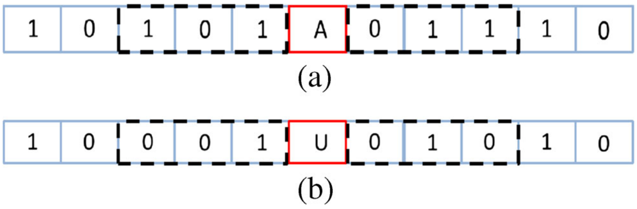


Fig. 4 Window includes (**a**) 4 stable cells (if T = 4, target cell is accepted (A)) and (**b**) 2 stable cells (if T = 4, target cell is not accepted (U))

In determining the "most stable" cells, the number of stable neighbors must be greater than or equal to T. Consider window size $W = 6$. If $T = 4$, then we only consider the target cell to be acceptably stable if at least four of its six neighbors analyzed are stable. In Fig. 4a, this condition is true, and therefore the target cell would be accepted by our bit selection algorithm. In Fig. 4b, the target cell is not accepted because it does not have enough stable neighbors.

Based upon these variables, conditional probability can be used to concisely describe the dependence of a SRAM cell upon the stability of its neighbors. In essence, conditional probability, $P(A|B)$, can be written as

$$P(A|B) = \frac{P(A,B)}{P(B)} \tag{3}$$

where $P(A, B)$ denotes the probability of events A and B occurring while $P(B)$ represents the probability of only B occurring. $P(A)$ is the probability of a target cell not flipping and $P(B)$ is the probability that the target cell's neighbors do not flip.

We investigate three metrics based on conditional probability: (i) *total neighborhood analysis* that captures the impact of all stable neighbors in a window; (ii) *neighborhood pairs analysis* which examines the influence of particular cells in the window; (iii) *environmental analysis* that captures the effect of isolated environmental conditions on the relationships of all cells in a window.

3.2.1 Total Neighborhood Analysis

This form of analysis calculates the stability of the target cell based upon all of its neighbors within a specified

Table 1 The impact of noisy cells on neighbor cells

Neighbor (aggressor) cell's fluctuation	Target (victim) cell's value ($V_{Q_{VR}}$)	Effect on neighbor cell
0 → 1	0	Might flip
0 → 1	1	Might be stronger
1 → 0	0	Might be stronger
1 → 0	1	Might flip

window size W. We define the neighborhood-based probability as

$$C_N = \frac{P(A, B; W, T)}{P(B; W, T)} \tag{4}$$

The notation $P(X; y)$ means that the probability of a random event X depends on the variable y. In this case, y is not random but chosen by the user. In Eq. 4, event A is the probability of the target cell not flipping while event B is the probability of T or more neighbors within the window size, W, (i.e., if T or more cells within a distance, $d = W/2$ from the target cell) do not flip. This metric focuses on the changes in conditional probability for different window sizes and threshold values. In our prior work [8], we assumed (based on intuition) that the larger the threshold, the better the result. However, the number of stable bits selected is lesser as T increases. The benefit of this proposed metric is that the ideal combination of W and T for optimal stability results can be determined for a SRAM and provide trends which can be extended to the examination of all SRAMs of the same type. If there exists some $T < W$ that is better or just as good as $T = W$, then we will be able to identify more stable bits for the PUF key than in [8].

3.2.2 Neighborhood Pairs Analysis

The intention behind this approach is to determine the level of influence neighbors at specific intervals from the target cell have. We define the neighborhood pair based probability as

$$C_{NP} = \frac{P(A, B; W)}{P(B; W)} \tag{5}$$

Here, event A is the same as before. Assuming the target cell is located at position i, we define event B as the probability that the cells located at $i - W/2$ and $i + W/2$ do not flip. The data of event B combines the data for the neighborhood at two isolated conditions. In this case, $T = 2$ (both of the two cells being examined must be stable for the target cell to be considered acceptable). The benefit of this metric is that it helps to find the probable true neighbor cells (i.e., which cells are physically true neighbor). Figure 5 demonstrates that whether a target cell X is influenced by the cells $X - 1$ and $X + 1$ or $X - r$ and $X + r$ and can be determined my total neighborhood analysis metric. $X - r$ and $X + r$ are considered as probable true neighbor cells of target cell X if the conditional probability due to $X - r$ and $X + r$ is higher than conditional probability due to $X - 1$ and $X + 1$. The value of r is determined from the peak value of total neighborhood analysis metric. Similar way, the total neighborhood analysis metric helps to find the $2nd$, $3rd$, kth (i.e, $X + k.r$) right-sided neighbors. The value of k is adjusted based upon the desired window size.

3.2.3 Environmental Analysis

To assess the effectiveness of environmental conditions in finding the most stable cells, this method considers the stability of a target cell using neighborhood data when only a pair of isolated fresh SRAM conditions are used for enrollment. We define the environmental total neighborhood-based probability as

$$C_{EN} = (P(A, B))/(P(B; c_1, c_2)) \tag{6}$$

Here, event A remains unchanged with all conditions (for fresh or aged data) being used to determine target cell stability. Conversely, event B examines all of the neighbors of a window, W, using only a pair of two different operating conditions (label as c_1 and c_2). For example, c_1 could be the high-temperature, high-voltage (HTHV) corner. While the definition of C_{EN} could be expanded to include any number of conditions, we shall only focus on choosing two conditions at a time to keep enrollment costs low. The time and cost of enrollment are directly related to the number of measurements taken from the PUF. The analysis of all of the possible pairs of condition combinations will demonstrate which condition pairs have the most inter-cell dependency and imply which enrollment tests can be most effectively extrapolated to determine the SRAM cells that are "most stable" over time.

4 Improving Uniqueness by Finding Uncorrelated Cells

In this section, we explore the reasons of having similar output by different SRAM PUF. We study the SRAM structure to find the most uncorrelated cells that can make the SRAM PUF unique.

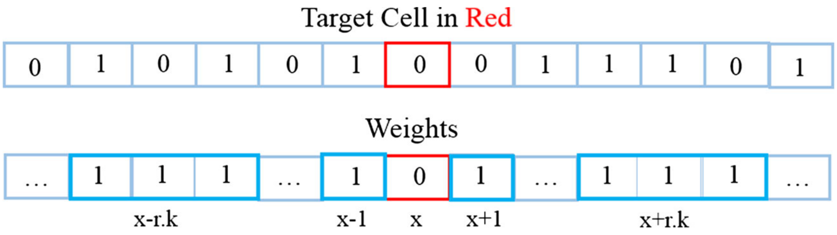


Fig. 5 Neighborhood pairs analysis to find the true probable neighbor cells

4.1 Systematic Correlation

The effect of parameter variation—the deviation of the process, temperature, and voltage (PVT) impacts the quality of secret key extracted from SRAM. Process variation in minute technologies is caused by the inability of controlling chip fabrication precisely. Lithographic lens aberrations, dopant density fluctuation, etc. are the main causes of uncontrollable process variation. Process variations are typically divided into two components: inter-die and intra-die [28]. Die-to-die or inter-die variations affect the performances of all transistors in a chip in the same direction. Intra-die/within-die variations account for variations among different devices within the same chip. Intra-die variation has two parts: random and systematic components. Systematic components are due to masking errors from inaccuracies in the process model, lithographic off-axis focusing errors, and reticle stepper alignment errors. On the other hand, random components generating from dopant fluctuations and line-edge roughness are suitable for the uniqueness of PUFs. Systematic components result in distance dependent correlation [24]. The parameter values are highly correlated to adjacent neighbors than the distant. Die-to-die or inter-die and systematic variations together can make the output from PUF to PUF similar for a given challenge.

4.2 Finding Uncorrelated Cells

Systematic correlation can cause PUFs to generate almost similar outputs for a given challenge. We follow two steps to find the characteristic of correlation among different SRAM PUFs output. In the first step, we find the correlation among SRAM boards. In the second step, we determine the characteristic of correlation found in step 1.

- **Finding correlation:** We XOR two SRAM boards' start-up values (considered as 1D) to get the location of cells that give the different values for two different SRAM boards. Suppose, B_{SRAM_i} and B_{SRAM_j} are the power-up values for SRAM board i and SRAM board j, respectively. B_{SRAM_i} and B_{SRAM_j} have both equal length of 1MB (total size of the ISSI SRAM).

$$O_{SRAM_{ij}} = B_{SRAM_i} \oplus B_{SRAM_j} \tag{7}$$

  where $i = 1, 2, ..., M$ and $j = 1, 2, ..., M$ and $i \neq j$. The total number of SRAM boards are M.
- **Characteristic of correlation:** In second step, our target is to find the periodicity of $O_{SRAM_{ij}}$; if any. This periodicity helps us to find the potential candidates (i.e., SRAM cells) for SRAM PUF. Auto-correlation is commonly used to find the periodicity of a function. Autocorrelation function (ACF) of a function $O_{SRAM_{ij}}$ of length N can be presented as:

$$R_{xx}(l) = \frac{1}{N-l} \sum_{n=0}^{N-l-1} O_{SRAM_{ij}}(n) O_{SRAM_{ij}}(n+l) \tag{8}$$

  where $l = 0, 1, 2, ..., N-1$. The periodicity helps us to determine uncorrelated cells that can be used for unique PUFs.

Algorithm 1 NICSA: Neighbor influenced cell selection algorithm

```
Input: Start-up values of SRAM at all environmental conditions (i.e, corner cases) and corresponding logical locations (1D)
Output: Locations of selected SRAM cells using NICSA
1. Finding optimal window size and threshold using Eq. 4 (TNA). Output: Fig. 6

2. Finding probable true neighbors of a target cell using Eq. 5. Output: Fig. 7

3. Revisit Step 1 with all possible corner cases using Eq. 6. Target: finding optimal enrollment pair, Output: Table 3

4. Revisiting TNA in order to check the conditional probability based metric for optimal window size, with the true probable neighbors found from TNP with rank 1 enrollment pair from Table 3. Output: Fig. 8

5. Low-cost NICSA (L-NICSA): Repeating Step 4 with low-cost enrollment pair (excluding temperature variation) from Table 3.
```

5 Neighbor Influenced Cell Selection Algorithm (NICSA)

Neighborhood-based metrics, described in Section 3.2, help us to choose the "most stable" cells by targeting the conditional probability of target cells not flipping provided that the other neighbor cells don't flip under a wide range of operating conditions. At the same time, finding uncorrelated cells, discussed in Section 4.2, helps us to obtain unique SRAM PUFs. Commonly uncorrelated cells and most stable cells are the best candidates for unique and most reliable key generation from SRAM PUFs. Neighbor influenced cell selection algorithm (NICSA) is used to find the optimal

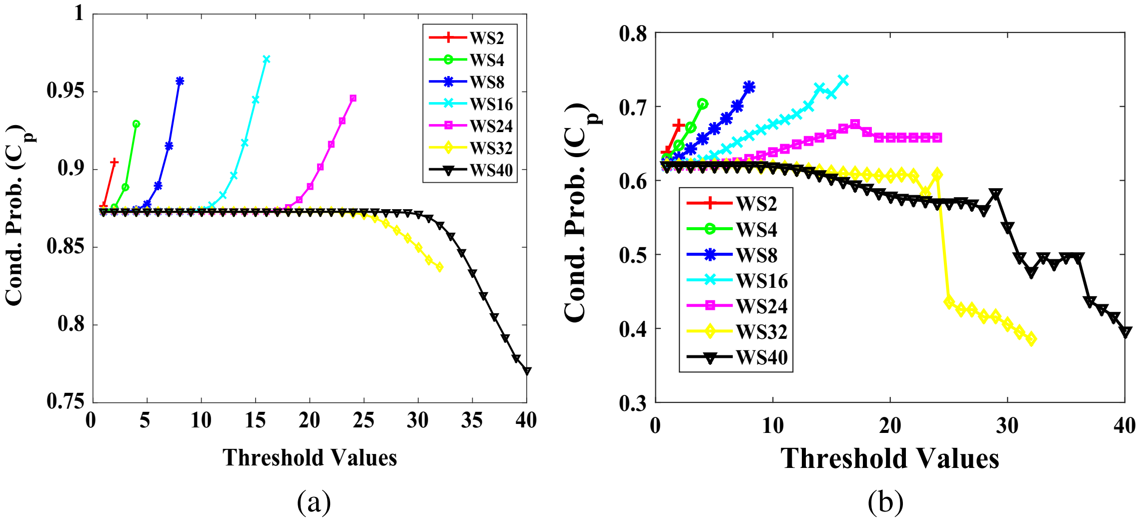
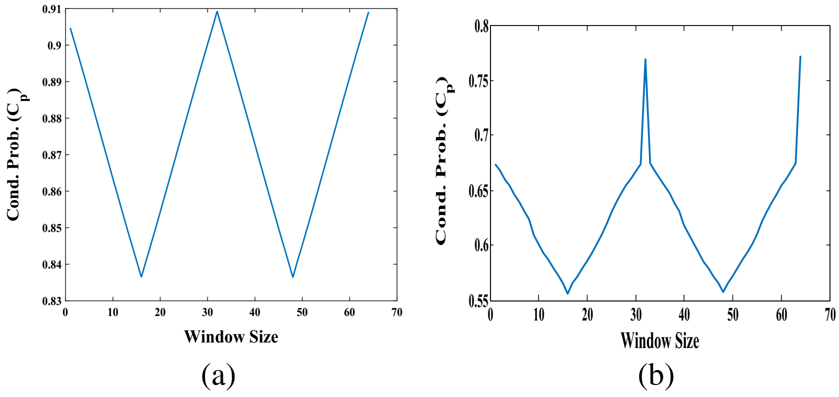


Fig. 6 Total neighborhood analysis (TNA) of **a** one fresh board and **b** average of five boards. The target of TNA is to achieve the highest conditional probability="1"

PUF candidates for low-cost robust SRAM-PUF. The proposed NICSA is based on the total neighborhood analysis, neighborhood pairs analysis, and environmental analysis. The proposed NICSA works in four steps (L-NICSA is excluded at this point) and presented in Algorithm 1.

- Step 1: Optimal window size (W_{opt}) and threshold (T_{opt}) are picked from the highest conditional probability (i.e., dependency of the target upon its neighborhood) using Eq. 4 (*total neighborhood analysis (TNA)).* Figure 6 shows that the optimal threshold value does not necessarily equal to the window size. The results show that threshold 16 is an optimal choice for target cells for ISSI SRAM.
- Step 2: The impact of each cell in the neighborhood on the reliability of the target cell is an important issue for bit selection. The peaks of the plot, Fig. 7, show the locations (regarding window sizes) which have the most influence on the target. Figure 7 shows that X, $X+32$, and $X+64$ are the 1st, 2nd, and 3rd true probable cells (previously those were X, $X+1$, and $X+2$ for neighborhood pair analysis) and can be found using *neighborhood pair analysis (NPA).*
- Step 3: Exhaustive testing for enrollment at all environmental conditions/corners and after aging is costly, time-consuming, and ill-suited for mass-produced devices. Our target is to choose the environment condition that offers the highest conditional probability for minimal enrollment testing. In order to reduce the cost and time, we compare the highest conditional probability at different operating conditions using *best enrollment pair* to select the appropriate enrollment pair for low-cost robust SRAM PUFs. We find that high-temperature high voltage (HTHV) and low temperature (LT) are the best pairs of enrollment corners.
- Step 4: In the final step, we revisit the total neighborhood analysis (TNA) to check the conditional probability based metric for optimal window size, W_{opt} and

Fig. 7 Neighborhood pairs analysis for window sizes $2-128$ for **a** a fresh board and **b** average of five boards. Our Target is to achieve the highest conditional probability="1" (ISSI SRAM (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf))



threshold T_{opt}, with the true probable neighbors found from TNA. The neighborhood influenced cell selection offers improved neighborhood dependency for probable true neighbors, their dependency, optimal window size, and threshold.

– Step 5 (applicable for L-NICSA): In this step, step 4 is repeated with low-cost enrollment pair (excluding temperature variation).

6 Results and Analysis

6.1 Experimental Setup

Our results are based upon experiments conducted using the 1MB on-board SRAM of the Xilinx Spartan-3 FPGA board. The total number of boards, M, is 5. The on-board SRAM provided a proper environment for the implementation of SRAM PUF, with the FPGA reading the SRAM and sending the data to a computer to be recorded. The various temperature and voltage conditions were achieved using a Thermostream system and power supply, respectively. The SRAM used in Spartan-3 is from ISSI. Later, we expand the evaluation of our proposed algorithm with silicon data from two other off-the-shelf SRAMs from IDT (http://www.idt.com/) and Cypress (http://www.cypress.com/file/43006/download) to test the effectiveness of our proposed algorithm.

Table 2 shows a list of the environmental conditions next to the abbreviations which will be used throughout the rest of this paper. Extensive burn-in was performed on the SRAM to create an accelerated aging process. We recorded results after 7.5 h of accelerated aging (@3.6 V and 80 °C) to capture the cell stability at different times. No frequency degradation of ring-oscillator is seen after 7.5 h of accelerated aging (similar observation to [39]). The details of experimental setup can be found in [8]. SRAM memory is aged for a longer period in [12] but enough amount of errors is observed to evaluate our proposed algorithm with 7.5 h of accelerated aging.

Table 2 Temperature/voltage conditions versus abbreviations used

Operating condition	Abbreviation	Value
High temperature	HT	100 °C
High-temperature high voltage	HTHV	100 °C and 3.4 V
High-temperature low voltage	HTLV	100 °C and 3.0 V
High voltage	HV	3.4 V
Low temperature	LT	0 °C
Low-temperature high voltage	LTHV	0 °C and 3.4 V
Low-temperature low voltage	LTLV	0 °C and 3.0 V
Low voltage	LV	3.0 V
Nominal conditions	NC	25 °C and 3.2 V

Ten trials were performed at each of the nine temperature/voltage combinations for each age (fresh and aged 7.5 h). We need these measurements for every single chip because the location of erroneous bits is independent of a chip. For all the trials, conditions, etc. for which we took measurements, this resulted in over 1 billion bits for computing conditional probabilities. Such an extensive data set allows for a full exploration of the entire SRAM and overall dependability in the results of the analysis.

Using the dependency relationship demonstrated by conditional probability, we were able to analyze the interactions of SRAM cells in 1D under the wide variety of conditions outlined earlier (voltages, temperatures, and aging). We performed the following experiments:

1. Initial results used *total neighborhood analysis* (C_N) to determine the dependency of the target cell on its neighbors at various window sizes around a target cell for $1 \leq T \leq w$ (integer values only). Note, that the threshold can never exceed the window size. All temperature and voltage conditions were used for the fresh and aged 7.5 h SRAM data sets to note the reliability of window size and threshold combinations as a SRAM ages.
2. Next, we wanted to highlight cells with the most influence on the target cell by using the more targeted approach of *neighborhood pairs analysis* (C_{NP}). In this scenario, $T = 2$ for all window sizes was tested. The range of tested window sizes was $1 \leq w \leq 4096$ (even values only). Conditional probability C_{NP} was plotted against window size, and only fresh data was examined using this approach.
3. Thirdly, *environmental analysis* C_{EN} was used to determine the effects of specific pairs of conditions on conditional probability in fresh data only. There were ten available temperature/voltage environments, which would combine to 45 possible pairs, each with 6 evaluated window sizes $W = 2, 4, 8, 16, 24, 32$.
4. We find the uncorrelated cells as discussed in Section 4.2 and then repeat the above three steps for uncorrelated cells to get unique and the most stable cells.

Total Neighborhood Analysis Figure 6(a) demonstrates the conditional probability for different window sizes and threshold values of a SRAM board. The results show that the highest conditional probability (i.e., dependency of the target upon its neighborhood) varies for different combinations of window sizes and threshold values. For smaller window sizes, such as 2 through 24, the optimal threshold is equal to the window size. As the window size gets larger, this relationship alters. For these conditions, there is a peak conditional probability located at $T = W$. For larger window sizes (32 and 40), the conditional probability is mainly fixed until around $T = 25$ and $T = 30$, respectively. One possible explanation for this peculiar behavior at larger window sizes could be due to the number of samples. Although we examined a very large data set of over 15 million data points for each window size, we may not have enough samples where target cells are surrounded by 32 or 40 stable cells to draw a significant conclusion. Another aspect missing from total neighborhood analysis is the impact of each cell in the neighborhood on the reliability of the target cell. This is covered by neighborhood pairs analysis below.

Neighborhood Pairs Analysis The plot in Fig. 7 demonstrates the dependency of each neighbor on the target cell's reliability. The peaks of the plot show the locations (regarding window sizes) which have the most influence on the target. Surprisingly, Fig. 7 shows that these peaks occur at regular intervals of 32 which is the value of r (Fig. 5). We believe that this is a direct indication of the physical location of the individual cells about one another. Although the cells may be read out in an order where they appear adjacent (assuming 1D), in actuality they may be located in a completely different arrangement. For instance, the peaks in conditional probability at intervals of 32 cells could demonstrate that cells at such a distance in the SRAM read-out are located closer to each other physically. This provides further motivation for multi-dimensional analysis, as some cells which appear close in the 1D SRAM data (i.e., 16 cells away from the target cell) may not be located close to the target cell and therefore do not have a profound impact on its power-up state and stability. This each 32nd cells might be neighbors and call as probable true neighbors. Figure 6(b) shows the average conditional probability for five SRAM boards have periodic peaks with a period of 32, W_{TN}, that means each 32nd cells might be probable true neighbor cells since they impact each other significantly than other neighbors.

Environmental Analysis Table 3 contains a list of each of the pairs of tested enrollment conditions, arranged in descending order of average conditional probability of five SRAM boards. The last two columns (rows) indicate the average and maximum conditional probability for each row (column). The maximum conditional probability can be seen for the combination HTHV and LT (located at the top of the list). In general, high temperature as condition 1 and either high temperature, nominal temperature, or low temperature at condition 2 work very well for enrollment. At the bottom of the table, we see that use of low temperature as condition 1 and nominal or low temperature for condition 2 works poorly. This makes intuitive sense because SRAMs are known to be affected by thermal noise which increases with temperature. Hence, using high temperature as part of enrollment ensures that we will be able to identify the most stable bits since less stable ones should flip at high temperatures.

Table 3 Total neighborhood analysis at different operating conditions (ISSI SRAM (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf))

Rank	Condition pairs		Max. cond. prob. at each window size						Avg.
	Cond. 1	Cond. 2	2	4	8	16	24	32	
1	HTHV	LT	0.89	0.91	0.94	0.96	0.93	0.86	0.92
2	HTHV	LTHV	0.89	0.92	0.94	0.96	0.93	0.86	0.92
3	HTHV	LTLV	0.9	0.92	0.95	0.96	0.93	0.86	0.91
4	HTHV	NC	0.9	0.92	0.95	0.96	0.93	0.86	0.91
5	HTHV	LV	0.89	0.92	0.95	0.95	0.93	0.86	0.91
6	HTHV	NC	0.89	0.92	0.95	0.95	0.93	0.86	0.91
7	LV	NC	0.89	0.909	0.93	0.94	0.92	0.86	0.91
8	HV	NC	0.89	0.909	0.93	0.94	0.92	0.86	0.91
9	LTLV	LV	0.89	0.909	0.93	0.94	0.92	0.86	0.91
10	HV	LTLV	0.89	0.909	0.93	0.94	0.92	0.86	0.9
11	LT	LTHV	0.89	0.909	0.93	0.94	0.92	0.86	0.9
12	HV	NC	0.89	0.90	0.93	0.94	0.92	0.87	0.91
13	HV	LV	0.88	0.90	0.92	0.94	0.91	0.86	0.9
14	LTHV	LTLV	0.88	0.90	0.93	0.93	0.92	0.86	0.9
	Average		0.89	0.91	0.94	0.95	0.92	0.86	0.90
	Maximum		0.90	0.92	0.95	0.96	0.94	0.86	0.914

Table 3 also illustrates the impact of different window sizes at different enrollment conditions. The conditional probability appears much more dependent upon window size W as opposed to the specific environmental conditions used in the isolated pair. This is an interesting and unexpected result. From the last two rows, it can be observed that the “best” window size is $W = 16$ (similar to Fig. 6).

NICSA-based Cells Selection The proposed NICSA is used to find the optimal candidates and enrollment conditions for low-cost robust key generation from SRAM. Figure 8 shows that the proposed NICSA helps to achieve consistent and higher neighborhood dependency than TNA alone (see Fig. 6). The improved conditional probability of the proposed NICSA is due to the selection of the best candidates from probable true neighbors. When performing neighborhood pairs analysis in 1D, neighboring cells located at specific intervals from the target cell have more influence on its stability than other cells in its neighborhood. We can take advantage of this correlation by considering only the most influential neighbors in our analysis. Neighbor influenced cell selection algorithm, NICSA, creates a window of cells which includes only pairs of cells located at the interval with the highest correlation to probable target cell stability besides choosing optimal window size, threshold, and enrollment conditions. By emphasizing only the most influential neighbors (true probable neighbors), fewer neighboring cells are required to accurately gauge the probability of continued target cell stability. Our analysis shows a high correlation between cells located at intervals of 32 in 1D (see Fig. 7). It should be noted that this correlation of the neighbors in 1D is most likely a reflection of the true probable neighbors. Therefore, influential cells located at a specific interval in 1D are most likely adjacent physically in SRAM.

Uncorrelated Cells Selection The plot in Fig. 9 demonstrates the correlation between two SRAMs’ power-up values. We represent only correlation between two SRAM boards (there are a total of ${}^5C_2 = 10$ combinations possible) in Fig. 9. We XOR two SRAM boards’ power-up values in order to find the correlation among cells. The result shows that uncorrelated/correlated cells, the XOR between two SRAMs ($O_{SRAM_{ij}}$), follow a periodic pattern. Figure 9 shows the $O_{SRAM_{ij}}$ for only the first 200 cells. We explore the auto correlation of $O_{SRAM_{ij}}$ in order to check the consistency of the observation of periodic pattern of correlated/uncorrelated cells to whole SRAM boards. The correlation observed in Fig. 9 supports the observed total neighborhood analysis of Fig. 6.

The selection of uncorrelated cells reduces the number of cells by half. But, the number of stable cells remains enough to generate a large set of keys from a SRAM board. A set of sample chips is enough to find uncorrelated cells. The total number of samples is limited, but bit-aliasing characteristic of Fig. 10 shows that the method of selecting uncorrelated cells works for a lot of samples.

Advantage of Choosing Uncorrelated Cells Ideally, a PUF has to be unique which means different PUFs have to give different responses to a given challenge. Systematic and spatial variation can produce the PUF response almost similar (i.e., less unique). The bit-aliasing is a very useful metric to estimate the bias of a particular SRAM cell across several SRAM boards. It helps us to guess/predict systematic and spatial effect across devices. Because of

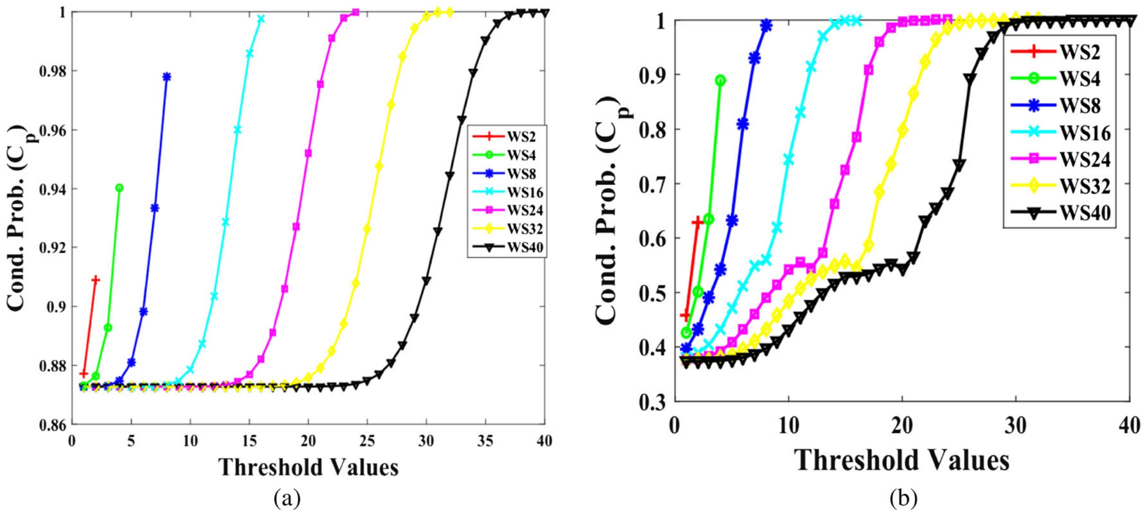


Fig. 8 Revisiting conditional probability C_p for **a** one fresh board and **b** average of five boards (ISSI SRAM (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf))

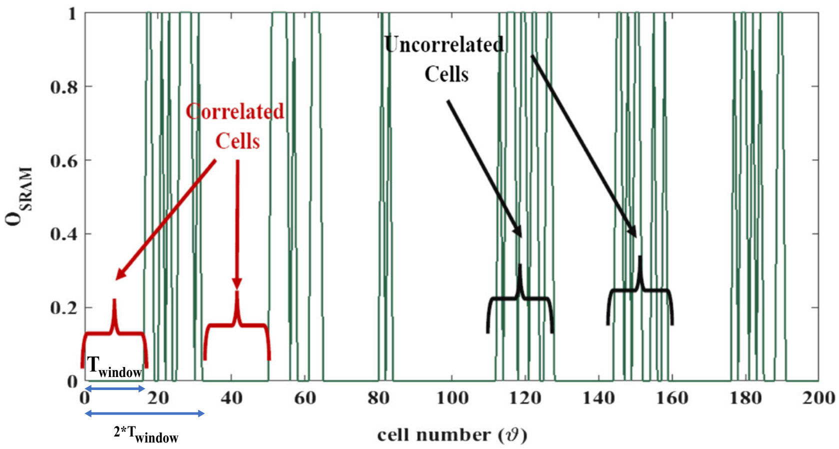


Fig. 9 Uncorrelated cells between two SRAMs follow a periodic pattern (ISSI SRAM (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf))

systematic correlation, a particular SRAM cell might give the same power-up value across different chips. The correlation among cells makes the life of an attacker easier because they are then able to predict the key generated from several SRAM PUFs by knowing one. To make the PUF unique, we need to reduce the effect of systematic correlation among several chips. The selection of uncorrelated cells, as discussed in the previous paragraph, can improve the uniqueness significantly and make the PUF more practical. Figure 10 shows that the selection of uncorrelated cells can reduce the effect of systematic correlation significantly. Figure 10 shows that, in most of the cases, most of the cells produce either all-0 or all-1 for all five SRAM boards. On the other hand, uncorrelated cells produce "0" among "1" with almost equal probability and the bit-aliasing is 0.5 for most of the cells. Table 4 justifies the claim about choosing uncorrelated cells in spite of reducing the cell number by half. The result shows that uncorrelated cells make the PUF unique (inter-HD is 49.72%, ideal value is 50%) where keys generated from PUFs with all cells achieve 44.77% HD. On the other hand, the NICSA-based approach is restrictive in choosing cells, which reduces the uniqueness somewhat, but still, presents a value very close to ideal. However, choosing uncorrelated cells costs half of the total cells (e.g., we can use 1MB of uncorrelated cells from a total of 2MB SRAM cells).

Reliability Comparison Table 5 shows the reliability comparison among NICSA-based bit selection for all cells, NICSA-based bit selection for uncorrelated cells, weighted neighborhood-based approach [8] for all cells and uncorrelated cells, random selection for all cells, and random selection among stable cells. We divided the whole $1MB$ SRAM into eight PUFs. The reliability of SRAM PUFs is compared between the proposed NICSA-based bit selection approach (threshold equal 16) and random bit selection approach in this section for both uncorrelated cells and all cells. Bit error rate (BER), along with reliability, of equivalent 128−bit keys from 80 measurements are reported in Table 6. NICSA-based bit selection offers much better reliability than the weighted bit-selection algorithm [8] because the proposed NICSA chooses the most stable cells

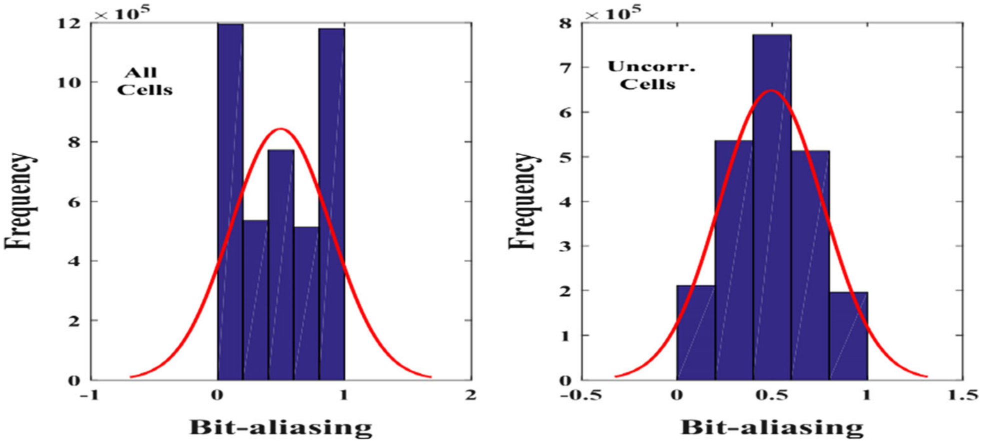


Fig. 10 Uncorrelated cells can produce more unique PUFs (ISSI SRAM (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf))

Table 4 Uniqueness comparison for all cells and uncorrelated cells (ISSI SRAM (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf))

Bit selection method	Inter Hamming distance (HD)			
	All cells		Uncorrelated cells	
	Avg.	std	Avg.	std
NICSA	42.12%	9.11%	47.98%	4.92%
Random	44.77%	8.48%	49.72%	4.37%

based on probable true neighborhood analysis, and best enrollment conditions. On the other hand, random selection has a significant amount of errors and thus requires ECC scheme to handle such amount of errors. The reliability and bit error rate are very close for uncorrelated and all cells because uncorrelated cells are each consecutive 16 cells after other 16. The NICSA algorithm improves the BER from $3.1e^{-2}$ to $7.9e^{-6}$ when compared to random selection. Thus, the NICSA-based algorithm helps us to ease off the ECC scheme. The proposed NICSA offers better BER and reliability for an aged device than fresh because aged SRAM cells might become more dependent on their neighborhood [9]. Besides, the proposed NICSA-based approach can find the optimized threshold unlike [8].

Low-cost Enrollment SRAM PUF is more sensitive to temperature variation. To identify stable cells, one can apply high/low temperature to discard less stable cells (i.e., more noisy cells). However, applying high/low temperature is neither cost effective nor time effective. To minimize the enrollment cost and time, we measure the SRAM PUF output applying a high voltage, low voltage, and nominal voltage at room temperature because voltage fluctuations also help to separate stable cells from unstable ones. The comparison between O-NICSA (NICSA with optimal enrollment) and L-NICSA (NICSA with low-cost enrollment) in Table 6. We use HTHV and LT as enrollment pair for O-NICSA and HV, NV, and LV as enrollment for L-NICSA. The results show that BER for L-NICSA, ($\sim 6.0e-5$), is not as good as O-NICSA ($7.9e-6$) but sufficient for the reliable key generation. Good BER is observed for L-NICSA due to the significant amount of voltage fluctuation, ($+/-10\%$) of V_{DD}, helps to discard noisy cells almost as much as with temperature fluctuation. The results show that NICSA with low-cost enrollment, depending on the key size and available unused SRAM cells, offers much better BER than random selection, because NICSA helps to select the most stable cells from enrollment.

NICSA and L-NICSA with Different SRAM Memories Our proposed NICSA and L-NICSA have to be technology and vendor independent. We evaluate the proposed NICSA (and L-NICSA) with Cypress, IDT, and ISSI SRAM memories. Tables 7 and 8 show that the parameters (window size, threshold, etc.) obtained from NICSA are different for different SRAM vendors. The result implies that different SRAM vendors use different logical/physical structures. The probable true n eighbor SRAM cells for ISSI, cypress, and IDT are each 32nd, 2nd, and 2nd, respectively. The periodicity for uncorrelated cells is 32 and 2 for ISSI, IDT. Uncorrelated cells are not found in cypress (because of design). The low-cost enrollment conditions for all SRAM memories are LV, NV, and HV.

Table 7 shows the comparison of BER among random, NICSA and L-NICSA for different SRAM memories from different vendors. The result shows that both NICSA and L-NICSA offer a significant amount of improvement regardless of SRAM vendors. The proposed NICSA offers at least x1000 improved BER compared to random selection. The result also shows that the PUF responses are more sensitive to temperature variation than the voltage variation. Temperature variation includes thermal noise in addition to changing static noise margin. The result also shows that the proposed NICSA is more efficient than L-NICSA but requires temperature measurement during enrollment which is expensive and time-consuming.

Table 5 Effectiveness of NICSA-based bit-selection approach for both uncorrelated cells and all cells

Bit selection method	Cell type	Bit error rate	
		Fresh	Aged
NICSA	All cells	$7.9e-6$	$6.1e-6$
	Uncorr. cells	$8.35e-6$	$7.61e-6$
K. Xiao et al. [8]	All cells	$8.02e-4$	$8.11e-4$
	Uncorrelated cells	$8.59e-4$	$9.98e-4$
Random (stable cells)	All stable cells	$2.9e-3$	$2.1e-3$
	Uncorr. stable cells	$2.7e-3$	$2.1e-3$
Random (all cells)	All cells	$3.9e-2$	$1.9e-2$
	Uncorr. cells	$3.1e-2$	$2.9e-2$

Table 6 Low-cost enrollment for NICSA and random selection (ISSI SRAM (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf))

Bit sel.	Enrollment conditions							
	O-NICSA (HTHV,LT)				L-NICSA(HV, NV, LV)			
	Reliability(%)		BER		Reliability(%)		BER	
	Fresh	Aged	Fresh	Aged	Fresh	Aged	Fresh	Aged
NICSA	99.99	99.99	7.9e-6	6.1e-6	99.93	99.93	6.0e-5	6.1e-5
Random	97.32	96.99	2.9e-3	2.7e-2	96.73	96.32	4.6e-2	5.5e-2

Besides, the experiential results show that the PUF values do not show any significant correlation between physically adjacent cells. The correlation coefficient is less than 0.1 for all three SRAM vendors.

Randomness NIST test suite [23, 29] is used to evaluate the randomness of bit streams obtained from ISSI SRAM (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf). There is a total of 15 NIST tests, and different tests require a different minimum length of bitstreams. For example, rank test, linear complexity test, and overlapping template matching test need at least 38912, 10^6, and 10^6 bits long bitstreams, respectively. On the other hand, frequency test, block frequency test, and runs test require a minimum of 100-bit long bitstream. Total two SRAMs are used to satisfy the minimum required bits [29]. The result shows, in Table 9, that SRAM PUF considering uncorrelated cells alone passes all NIST tests. On the other hand, SRAM PUF considering all cells does not pass all NIST tests. The result concludes that uncorrelated cells are good candidates for unique and random key generation. The better randomness of uncorrelated cells is due to the effect of reduction in systematic variation's effects since the systematic variations are seen in both intra-die and inter-die. SRAM-based PUFs from Cypress and IDT also show similar randomness.

Table 7 Bit error rate (BER) for three different SRAM vendors

			ISSI IS61LV25616AL (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf)	IDT 71V416S12PHG8 (http://www.idt.com/)	CY62146EV30LL-45ZSXI (http://www.cypress.com/file/43006/download)
			BER	BER	BER
7.5-h accelerated aging	Random	All cells	3.90e-2	2.85e-2	3.12e-2
		Unorrelated cell	3.11e-2	2.89e-2	–
	NICSA	All cells	7.91e-6	6.83e-6	7.91e-6
		Unorrelated cell	8.35e-6	5.93e-6	–
	L-NICSA	All cells	1.52e-6	5.32e-6	4.63e-6
		Unorrelated cell	1.93e-6	1.12e-6	–
Vdd variation	Random	All cells	7.38e-2	8.77e-2	0.44e-2
		Unorrelated cell	7.02e-2	6.12e-2	–
	NICSA	All cells	1.52e-6	1.52e-6	1.52e-6
		Unorrelated cell	2.64e-6	0.92e-6	–
	L-NICSA	All cells	3.26e-6	1.52e-6	1.52e-6
		Unorrelated cell	1.86e-5	3.21e-6	–
Temp. variation	Random	All cells	1.47e-2	1.04e-2	1.52e-2
		Unorrelated cell	1.12e-6	1.11e-2	–
	NICSA	All cells	4.55e-6	6.12e-6	1.52e-6
		Unorrelated cell	1.31e-6	2.98e-6	–
	L-NICSA	All cells	7.17e-5	7.04e-5	8.32e-5
		Unorrelated cell	3.61e-5	1.68e-5	–

Table 8 Parameters from NICSA

SRAM	ISSI (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf)	IDT (http://www.idt.com/)	Cypress (http://www.cypress.com/file/43006/download)
Technology node	–	130 nm	90 nm
# of samples	5	3	3
Adjacent probable true numbers (W_{TN})	16	2	2
Periodicity of uncorr. cells (T_{window})	16	2	–
$Threshold$	16	16	16

Area Overhead Analysis The locations of selected bits can be stored in an off-chip or an on-chip non-volatile memory (NVM) similar to helper data, 1-out-of-k masking, sequential pairing algorithm, bit selection proposed in [4], etc. [31]. One of the major advantages of our proposed method and bit selection proposed in [4] is that the stored locations (on-chip or off-chip) do not reveal any key information because physical locations do not reveal the contents. Unfortunately, all platforms do not support on-chip non-volatile memory, for example, FPGA [31]. Thermal chamber or thermostream is required for temperature variation. However, our proposed L-NICSA is used to avoid taking measurements from temperature variations to make our bit-selection approach more effective for large volume production. ECC is one of the popular choices to generate a robust key from PUFs. The total footprint for SRAM-based PUF consists of the encoder, quantizer/repetition coder, decoder and SRAM [11]. A helper data algorithm (HDA) is used to recover the correct value of a full entropy secret key [11]. Soft-decision based algorithm uses the probability of flipping each bit to reduce the area overhead. This method requires multiple enrollment measurements to find the error probability of an individual bit. The major disadvantage of this soft-decision based approach is that it requires extra NVM to store the information of each bit which increases with measurement. For example, the sum of each bit requires additional two bits of storage per bit. Later, a single measurement based soft-decision algorithm was proposed in [11]. The proposed NICSA requires multiple measurements. The proposed NICSA does not need to store the multi-bit information of each bit in an NVM. However, the locations of selected SRAM cells need to be stored in an NVM. For a 16-bit wide wordline, four bits are required to store the location of a SRAM cell for a one-bit key. Note that ECC scheme requires the NVM to store the helper data. The required memory space for a strong PUF with ECC scheme increases linearly with the CRPs.

Table 10 shows the area overhead comparison between our proposed NICSA, bit selection proposed in [4], HDA-based scheme [10, 11, 17]. The experimental results show that the worst case bit error probability for ISSI SRAM and Cypress SRAM is 13.12 and 16.07%, respectively. The total number of required SRAM cells depends on several factors such as physical design and organization, process variation, inter noise, etc. To make a fair comparison, the average bit error probability, the amount of min-entropy in the SRAM-PUF responses are 14 and 75%, respectively. A key of 128-bit with full entropy requires $128/0.76 = 171$ secret bits because the secrecy rate for SRAMPUF is 0.76 [10, 11, 17]. Gate equivalent (GE) is used to compare the existing work and our proposed NICSA. One GE is the area of an NAND2 (standard drive strength). A standard SRAM cell can be considered as one GE with read and write circuitry [11]. Table 10 shows the effectiveness of our proposed NICSA. The average BER varies from chip to chip and from vendor to vendor. The Cypress SRAM requires more SRAM cells than ISSI SRAM because they produce more erroneous responses. One of the major

Table 9 Randomness comparison between NICSA and random selection for ISSI SRAM (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf)

NIST algorithm [29]	Random		NICSA	
	Proportion of cells			
	All	Uncorr.	All	Uncorr.
Frequency	57/100	99/100	54/100	99/100
Block freq.	64/100	99/100	61/100	99/100
Cumulative sums	62/100	99/100	57/100	98/100
Runs	48/100	97/100	44/100	96/100
Longest runs	47/100	96/100	44/100	96/100
Rank	68/100	100/100	67/100	100/100
FFT	64/100	97/100	57/100	95/100
Non-overlapping template	43/100	92/100	48/100	91/100
Overlapping template	44/100	94/100	41/100	94/100
Serial	67/100	96/100	62/100	97/100
Linear complexity	74/100	100/100	70/100	99/100

Table 10 Area overhead comparison for deriving 171 secret bits with false rejection rate $< 10^{-6}$

Post-processing	Required SRAM cells	
	(ISSI (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf)	Cypress (http://www.cypress.com/file/43006/download)
NICSA-all	~ 720 GE	~ 952 GE
NICSA-uncorrelated	~ 1400 GE	–
L-NICSA-all	~ 800 GE	~ 1286 GE
L-NICSA-uncorrelated	~ 1640GE	–
Bit-selection in [4]+ ECC	~ 2.4 kGE	~ 2.6 kGE
Hard RM [16, 5, 8] [11]	~ 8.3 kGE	
Soft RM [16, 5, 8] [11]	~ 5.2 kGE	
Golay RM [24, 12, 8] [11]	~ 6.1 kGE	
Golay RM [24, 12, 8] [11]	~ 6.3 kGE	

challenges is the enrollment at different operating conditions. The testing of PUFs has been proposed in several works ([47], for example). The objective of these work is to have a specific robustness and uniqueness for a PUF. Besides, it is very common practice to test the robustness of a digital IC for large manufacturing. In many cases, we need multiple measurements and different operating corner cases [46–48]. We need at least a single measurement to register CRP (both for weak and strong PUF) in a database. Furthermore, the NICSA requires less amount of SRAM cells than the L-NICSA because the temperature variation helps us to select the more stable cells. The bit-selection of [4] suffers 3.51% error in average when we age the device, and hence it requires ECC scheme. The proposed NICSA is also 11X efficient than the bit-selection of [4] because of cell-by-cell analysis and also because of our findings (i.e., noisy cell is surrounded by noisy cells).

Time Overhead Analysis Usually, multiple measurements are collected from the nominal condition and some other operating conditions (such as temperature variations, voltage variations, or both) during registration. Our motivation is to use those data for bit-selection algorithm in order to obtain robust SRAM-PUF. Usually, the temperature measurement takes time to reach a target temperature (it takes 5–7 minutes to reach 80 °C from room temperature). Measurement time for voltage variation is insignificant compared to measurement time for temperature variation. Table 3 shows that high temperature is the best for enrollment in order to obtain robust keys. On the other hand, low-voltage condition is also significant for enrollment because read static noise margin decreases with supply voltage, V_{dd}. A low V_{dd} can help to find out the more noise tolerant cells which might be ideal for the robust key generation. Only voltage variations are considered in L-NICSA and still, a target BER can be achieved (see Table 6. On the other hand, the enrollment tool is designed using MATLAB R2015a (http://www.mathworks.com/) on a computer with a 3.6 GHz Core $i7$ Intel processor with 16GB of RAM. The results show that our proposed NICSA requires 63.12 s to obtain 5184K-bit key. The required time can be divided into two parts: (i) time required to determine the physical locations and (ii) time required to find the most stable cells. Only 4–5 samples can be used to determine the physically adjacent cells in order to reduce the enrollment time. The time required to find the most stable cells is much less compared to finding the physical structure of a SRAM. The average time required to find the most robust cells is ~ 11 s to obtain to obtain 5184K-bit key given that the physical location of each SRAM cell is known. The major limitation of our work is the required time to collect the data from the real chip at different corner cases which depend on the system frequency, the size of the memory array, access time, etc. The access time of ISSI, Cypress, and IDT is 12 ns, 45 ns, and 10 ns, respectively (https://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf) (http://www.cypress.com/file/43006/download, http://www.idt.com/).

The major challenges/limitations of the proposed challenges are:

- The proposed NICSA requires ~ 2.2 SRAM cells than the conventional key generation scheme. However, the conventional approach requires ECC scheme and extra SRAM cells (i.e., parity bits) to correct the errors.
- The proposed NICSA requires high-temperature and high-voltage measurements during the enrollment which requires additional time. Our proposed low-cost NICSA helps to avoid registration from temperature variations. We can use the data from the electrical test (e.g., burn-in test) to avoid extra measurements.

- We need to store the locations of selected SRAM cells. The locations vary from chip to chip and do not leak any information. In traditional key generation method, we need to store the syndrome bits to correct the errors. The NICSA does not require to store any syndrome bits.

6.2 Summary of Results

Overall, we draw the following main conclusions from these results for our SRAM:

- The optimal combination of window size and the threshold value is not necessarily at the point where $W = T$ (for logical location). This observation leads us to find the true physical location of a SRAM.
- For ISSI SRAM, the most influential neighborhood cells appear to be the cells at distances every 32 (W_{TN}) bits from the target cell (in 1D). These are likely to be the ones physically located near the target cells in the actual layout. Different SRAM vendors have different physical structures. W_{TN} is two for both IDT and Cypress SRAM memories.
- Certain pairs of environmental conditions demonstrate higher dependency of the target cell upon its neighbors. Generally, it is better to use high temperature as one of the enrollment conditions because robust SRAM cells can tolerate more amount of noise added by temperature variation.
- The choice of window size W is more important for bit selection than enrollment conditions.
- The NICSA-based approach helps us to find probable true neighbors, optimal window size, threshold, and operating conditions for enrollment. The proposed NICSA helps to ease off the ECC scheme by achieving target BER ($\sim 10^{-6}$). Most of the systems have enough unused SRAM cells that can be used in the proposed NICSA which makes the NICSA-based SRM PUFs superior to SRAM PUFs with large ECC scheme.
- We also observe that the proposed NICSA can help us to determine the low-cost enrollment conditions (exclusion of temperature variation) for low-cost robust SRAM PUF.
- Correlation is seen in both inter-die and intra-die and thus the selection of uncorrelated cells improves both the uniqueness and the NIST randomness test. Not all SRAM vendors show a systematic correlation because of layout and/or design techniques.
- Results show that our proposed algorithm is able to find most stable SRAM cells regardless vendors and technology nodes.

7 Conclusion

In this paper, we proposed NICSA-based approach by investigating candidate cells and enrollment conditions to analyze the outputs of SRAM with the help of proposed three metrics. Our result showed that the optimal threshold value does not necessarily equal the window size. We observed that certain cells set at specific distances from the target cell are more heavily influential than other neighbors within a window. We found that NICSA-based approach make the SRAM PUF almost error free by choosing the optimal enrollment conditions, probable true neighbors and their dependency. We observed that the proposed metrics can help us to reduce the enrollment cost further by excluding temperature variations during enrollment. We also noticed that uncorrelated cells make the PUF very unique and random by reducing the effects of systematic correlation but does not follow a certain periodic pattern for all SRAM cells. We hope to expand our work to embedded SRAM in SoC with secured communication protocol. Additionally, we plan to apply the proposed NICSA-based approach for TRNG and counterfeit IC detection.

Acknowledgments This work was supported in part by the National Science Foundation (NSF) under grant CNS-1561023 and by the Semiconductor Research Corporation (SRC) under contracts 2572 and 2648.

References

1. Tauhidur Rahman M et al (2014) CSST: preventing distribution of unlicensed and rejected ICs by untrusted foundry and assembly. In: IEEE Int. symposium on defect and fault tolerance symposium (DFTS)
2. Herder C et al (2014) Physical unclonable functions and applications: a tutorial. Proc IEEE 102:1126–1141
3. Maes R et al (2010) Physically unclonable functions: a study on the state of the art and future research directions, section 1. towards hardware-intrinsic security. Springer
4. Eiroa S et al (2012) Reducing bit flipping problems in SRAM physical unclonable functions for chip identification. In: 19th IEEE int. conf. on electronics, circuits and systems (ICECS), pp 392–395
5. Holcomb D et al (2009) Power-up SRAM state as an identifying fingerprint and source of true random numbers. IEEE Trans Comput
6. Su Y et al (2008) A digital 1.6 pJ/Bit chip identification circuit using process variations. IEEE J Solid-State Circ 43(1):69–77
7. Yu M, Devadas S (2010) Secure and robust error correction for physical unclonable functions. IEEE Des Test Comput 27(1): 48–65
8. Xiao K et al (2014) Bit selection algorithm suitable for high-volume production of SRAM-PUF. IEEE Int Symp Hardware-Oriented Secur Trust, pp 101

9. Hosey A et al (2014) Advanced analysis of cell stability for reliable SRAM PUFs. In: 2014 IEEE 23rd Asian test symposium (ATS), pp 348–353
10. Maes R et al (2009) A soft decision helper data algorithm for SRAM PUFs. In: ISIT 2009 IEEE International symposium on information theory, pp 2101–2105
11. van der Leest V et al (2012) Soft decision error correction for compact memory-based PUFs using a single enrollment. In: Cryptographic hardware and embedded systems, CHES 2012, volume 7428 of LNCS. Springer, Berlin, pp 268–282
12. Maes R, van der Leest V (2014) Countering the effects of silicon aging on SRAM PUFs. In: 2014 IEEE International symposium on hardware-oriented security and trust (HOST), pp 148–153
13. Hofer M et al. (2010) An alternative to error correction for sram-like pufs. Cryptograph Hardware Embedded Syst 335–350
14. Garg A, Kim TT (2014) Design of SRAM PUF with improved uniformity and reliability utilizing device aging effect. In: IEEE International symposium on circuits and systems (ISCAS), pp 1941–1944
15. Zheng Y et al (2013) RESP: a robust physical unclonable function retrofitted into embedded SRAM array. In: 50th ACM/IEEE design automation conference (DAC)
16. Bhargava M et al (2012) Reliability enhancement of bi-stable PUFs in 65nm bulk CMOS. In: IEEE Intl symposium on hardware-oriented security trust (HOST)
17. Cortez M et al (2013) Adapting voltage ramp-up time for temperature noise reduction on memory-based PUFs. In: IEEE Intl symposium on hardwareoriented security trust (HOST)
18. Tauhidur Rahman M et al (2015) A pair selection algorithm for robust RO-PUF against environmental variations and aging. In: IEEE International conference on computer design (ICCD)
19. Tauhidur Rahman M et al (2014) TI-TRNG: technology independent true random number generator. In: Proceedings of the the 51st annual design automation conference on design automation conference (DAC), pp 179:1–179:6
20. Merl D et al (2011) Side-channel analysis of PUFs and fuzzy extractors. In: McCune JM, Balacheff B, Perrig A, Sadeghi A-R, Sasse A, Beres Y (eds) Trust and trustworthy computing (TRUST), ser. LNCS, vol 6740. Springer, pp 33–47
21. Xu SQ et al (2014) Understanding sources of variations in flash memory for physical unclonable functions. In: IEEE 6th international memory workshop, pp 1–4
22. Mazady A et al (2015) Memristor PUF–a security primitive: theory and experiment. IEEE J Emerging Select Topics Circ Syst 5(2):222–229
23. Tauhidur Rahman M et al (2016) An aging-resistant RO-PUF for reliable key generation. IEEE Trans Emerg Topics Comput PP(99):1
24. Sarangi SR et al (2008) VARIUS: a model of process variation and resulting timing errors for microarchitects. IEEE Trans Semicond Manuf 21(1):3–13
25. Onabajo M, Silva-Martinez J (2012) Process variation challenges and solutions approaches. In: Analog circuit design for process variation-resilient systems-on-a-chip, p 930
26. Aktouf C (2002) A complete strategy for testing an on-chip multiprocessor architecture. Des Test Comput 19:18
27. Bae J et al (2012) Characterizing the capacitive crosstalk in SRAM cells using negative bit-line voltage stress. IEEE Trans Instrum Measur 61:3259–3272
28. Stine BE et al (1997) Analysis and decomposition of spatial variation in integrated circuit processes and devices. IEEE Trans Semicond Manuf 10(1):24–41
29. Rukhin A et al (2010) A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22 Rev1a
30. Maiti A et al (2013) A systematic method to evaluate and compare the performance of physical unclonable functions. In: Athanas P, Pnevmatikatos D, Sklavos N (eds) Embedded systems design with FPGAs. Springer, New York, pp 245–267
31. Delvaux J, Verbauwhede I (2014) Key-recovery attacks on various RO PUF constructions via helper data manipulation. In: Design, automation & test in europe conference & exhibition, DATE 2014. Dresden, p 16
32. Merli D et al (2013) Protecting PUF error correction by codeword masking. In: Proc. IACR cryptology eprint archive. Buenos Aires, p 334
33. Hiller M et al (2013) Breaking through fixed PUF block limitations with differential sequence coding and convolutional codes. In: Proceedings of the 3rd international workshop on trustworthy embedded devices, pp 04–04
34. Armknecht F et al (2009) Memory leakage-resilient encryption based on physically unclonable functions. In: Advances in cryptology (ASIACRYPT), ser. LNCS, vol 5912, pp 685–702
35. Dodis Y et al (2004) Fuzzy extractors: how to generate strong keys from biom etrics and other noisy data. In: Proc. Eurocrypt, pp 523–540
36. Maes R et al Secure key generation from biased PUFs. In: Proc. Cryptographic hardware and embedded systems, CHES 2015, vol 9293 of LNCS, pp 517–534
37. Van Herrewege A et al (2013) DEMO: inherent PUFs and secure PRNGs on commercial off-the-shelf microcontrollers. In: Proceedings of the 2013 ACM SIGSAC conference on computer communications security. Hangzhou, pp 1333–1336
38. Schrijen GJ, van der Leest V Comparative analysis of sram memories used as puf primitives. In: Proceedings of the conference on design, automation and test in Europe, pp 1319–1324
39. Dogan H et al Aging analysis for recycled fpga detection. In: 2014 IEEE international symposium on defect and fault tolerance in VLSI and nanotechnology systems (DFT), pp 171–176
40. Koeberl P et al Entropy loss in PUF-based key generation schemes: the repetition code pitfall. In: Proc. Int. symp. hardw.-orient. security trust (HOST), pp 44–49
41. Herder C et al Trapdoor computational fuzzy extractors, [Online]. Cryptology ePrint Archive, Rep. 2014/938. Available: http://eprint.iacr.org/
42. Maes R et al (2012) PUFKY: a fully functional PUF-based cryptographic key generator. In: Proceedings of the 14th international conference on cryptographic hardware and embedded systems, pp 302–319
43. Suzuki D, Shimizu K (2010) The glitch PUF: a new delay-PUF architecture exploiting glitch shapes. In: Cryptographic hardware and embedded systems, pp 366–382
44. Chuang C-T et al (2007) High-performance SRAM in nanoscale CMOS: design challenges and techniques. In: 2007 IEEE international workshop on memory technology, design and testing. Taipei, pp 4–12
45. Xu X (2015) Reliable physical unclonable functions using data retention voltage of SRAM cells. IEEE Trans Comput-Aided Des Integr Circ Syst 34(6):903–914
46. Kinseher J et al (2016) Improving testability and reliability of advanced SRAM architectures. IEEE Trans Emerg Topics Comput PP(99):1–1
47. Vijayakumar A et al (2016) On testing physically unclonable functions for uniqueness. In: 2016 17th International symposium on quality electronic design (ISQED), pp 368–373
48. Sumikawa N et al (2012) An experiment of burn-in time reduction based on parametric test analysis. Proc Intl Test Conf 1–1