Journal of Electronic Testing (2022) 38:511–525
https://doi.org/10.1007/s10836-022-06025-8

Using both Stable and Unstable SRAM Bits for the Physical Unclonable Function

Zhi-Wei Lai[1] · **Po-Hua Huang**[2] · **Kuen-Jong Lee**[2]

Received: 18 February 2022 / Accepted: 4 September 2022 / Published online: 14 October 2022

Abstract
Recently Physical Unclonable Functions (PUFs) of IC chips have been used in electronic systems for secret key generation and device authentication. Among all available PUFs, SRAM PUF is a popular one because SRAM is a standard component for most electronic devices, and it possesses good randomness during power-on. Previously only strongly stable SRAM bits are selected as PUF bits, which generally requires a large number of SRAM bits. Furthermore, SRAM PUFs may suffer from PUF clone attacks as attackers may use the Photon Emission Analysis (PEA) device to observe the behavior of stable bits and conduct circuit edit via Focused Ion Beam (FIB) to produce identical PUFs. In this paper we propose two methods that employ unstable bits as PUF bits in addition to stable bits to increase the SRAM bit usage rate. These two methods can resist the PUF clone attack as it is very difficult to reproduce unstable bits. Extensive experiments have been conducted, and the results show that though unstable bits are used, high reliability is still achieved.

Keywords Key security · Physical Unclonable Function (PUF) · PUF clone · SRAM · unstable bits

1 Introduction

A physical unclonable function (PUF) of a chip is a unique physical feature of the chip due to the unique physical variations that occur naturally during manufacturing. PUFs are typically used in cryptography applications such as data encryption/decryption, ID identification, device authentication, etc. Many PUF designs have been proposed and implemented in recent years, including Ring-Oscillator PUF [1–3], Arbiter PUF [3, 4], Optical PUF [3], Coating PUF [5], SRAM PUF [3], Butterfly PUF [6], XOR-based reconfigurable PUF [2], One-Time-Programming PUF [7], etc. Among these PUFs, the SRAM PUF is a popular one because SRAM is a standard component in most electronic devices. Also when an SRAM cell is turned on without a reset, it may be biased toward 0 or 1 based on its manufacturing process variation and operating environment. Such randomness makes SRAM PUF suitable for cryptographic applications.

Previous SRAM PUFs rely only on strongly stable bits that always bias in one direction to generate a reliable PUF response. However even if only strongly stable bits are taken, the stability of SRAM PUF still needs to be concerned, and some mechanisms to guarantee the reliability of PUF have been employed. One popular approach is to employ some Error Correction Code (ECC) logic [8, 9]. This requires the extra areas to implement the ECC logic, increasing the cost. Approaches attempting to reduce the ECC cost by modifying the SRAM structure have been proposed [10–12]. These approaches add some latches or buffers to the SRAM to enhance reliability and uniqueness. Other approaches emphasize bit selection methods to enhance reliability without modifying the SRAM structure. Some of them use Temporal Majority Voting (TMV) to improve the stability of PUF response [10, 13]. The basic principle is to repetitively test the PUF using the same challenge and take the majority value of the response as the final

Responsible Editor: B. Ghavami

✉ Zhi-Wei Lai
ab3896423@gmail.com

Po-Hua Huang
phh111@beethoven.ee.ncku.edu.tw

Kuen-Jong Lee
kjlee@mail.ncku.edu.tw

[1] RealTech, Inc, Hsinchu, Taiwan

[2] Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan

output. It usually conducts a large number of tests to get a highly reliable response. In [14], a selection method based on neighborhood analysis to select highly reliable bits to reduce ECC cost is proposed. The work in [15] makes use of the data remanence effect to select the highly stable bits. However the selection methods above generally require a large SRAM area to get enough reliable PUF bits. In [16] a selection method called the Maximum Trip Supply Voltage (MTSV) method is proposed to identify and rank the most reliable cells. In [17] another selection process is proposed, which first identified potentially reliable bits uses a quick and slow power-on ramp rates, and then uses data retention tests to rank the most reliable bits. It is shown that the bit utilization rate can be significantly improved with these two methods.

One problem with most of the current PUFs is that there exist several methods to attack PUF design, including modeling attack [18], side-channel attack [19], and PUF clone attack [20]. These methods attempt to get the challenge-response pairs (CRPs) of PUFs. For an SRAM PUF, it is hard to implement a modeling attack such as a machine learning attack since SRAM PUF has plenty of independent bits of information, and no model to correlate these bits exists. As for the side-channel attack, it might get some SRAM information by the power, timing, photon emission, etc. It usually combines with other approaches to get the correct PUF response. The PUF clone attack attempts to make an SRAM core or chip that has the same CRPs as the SRAM to be cloned. In [20] it is shown to be feasible by first performing Photon Emission Analysis (PEA) to observe the stable states of SRAM cells and then conducting Focused Ion Beam (FIB) circuit edit to produce the same PUF response in a new SRAM.

In this work, we propose two methods that utilize not only the stable bits but also some unstable bits for the SRAM PUF. The bit usage rate can also be significantly increased. We classify a PUF bit as a "stable" bit if it always has the same value among multiple (at least two) readouts. Otherwise, it is classified as an "unstable" bit. We also use the terms "reliable" and "unreliable" to further describe the PUF bits with the strong or weak tendency to have the same behavior. As in our first proposed method (see Sect. 4.1), if for a PUF bit, both logic 0 and 1 appear at least three times in the 16 readouts, the PUF bit is defined as a "reliable" unstable bit. If there is only one or two logic 0 or 1, it is defined as an "unreliable" unstable bit because of its weak tendency to act as an unstable bit. Note that there are only "reliable" stable bits but no "unreliable" stable bits because the stable bit always has a strong tendency to bias toward a certain value.

In the literature, several works are related to unstable bits [21–24]. These works usually employed ternary logic (0, 1, and X) to represent the state of each SRAM cell, including the unstable bits. The work by Cambou and Orlowski [21] and the follow-up works [22, 23] use ternary logic to record those unstable bits in a third state "X" during the enrollment process such that once a bit is recorded as an unstable bit, it will no longer be considered a PUF bit and can be skipped in the following enrollment process. Therefore the unstable bits are not used as PUF bits. In our work we do employ some "reliable" unstable bits as PUF bits, and hence can greatly increase the bit usage rates, resulting in a much smaller SRAM needed for the same required number of PUF bits.

In [24] Yamamoto et al. did propose using both stable and unstable bits in an RS-latch-based PUF. This method uses ternary logic "11", "00", and "10" to represent a stable 1, stable 0, and unstable bit, respectively. Each RS-latch is read out multiple times, and an (RS-latch) bit is classified as unstable if both 1 and 0 appear at least once in its readouts. This may have the following problem. If "1" ("0") appears only once in the multiple reads of a latch, say 1 out of 100 readouts, then it is quite possible that all "0" ("1") will be read out next time since the latch strongly biases to "0" ("1"). Such a bit will be unreliable, and hence the reliability of the PUF may be low. In our work we use a strict enrollment procedure to screen out those "unreliable" unstable bits and ensure that all adopted unstable bits are "truly unstable" in all cases. In other words, we will classify all unstable bits into "reliable" unstable bits and "unreliable" unstable bits in the enrollment procedure and use only the "reliable" ones as the PUF bits. Also, we do not use ternary logic; we have developed a novel method that uses the commonly-used binary logic to represent each PUF bit. Hence the storage to store the PUF information is smaller.

The two methods proposed in this paper can also resist the PUF clone attack since cloning unstable bits is almost impossible using the above-mentioned clone techniques [20]. In fact, to the best of our knowledge, no method can make a "truly" unstable SRAM bit. We will also show that high reliability can be achieved with our method even if unstable bits are used. The primary contributions of this work can be summarized as follows.

1) We conduct extensive experiments to analyze the impact of various voltages, temperatures, and the data remanence effect on the SRAM bits to explore the characteristics of stable and unstable bits of SRAM.
2) We present two methods to extract not only the stable bits but also the "reliable" unstable bits from SRAM ICs under various temperatures and voltages.
3) We utilize not only the unstable bits that are unstable in all conditions but also the unstable bits that are unstable only in some conditions as our PUF bits.

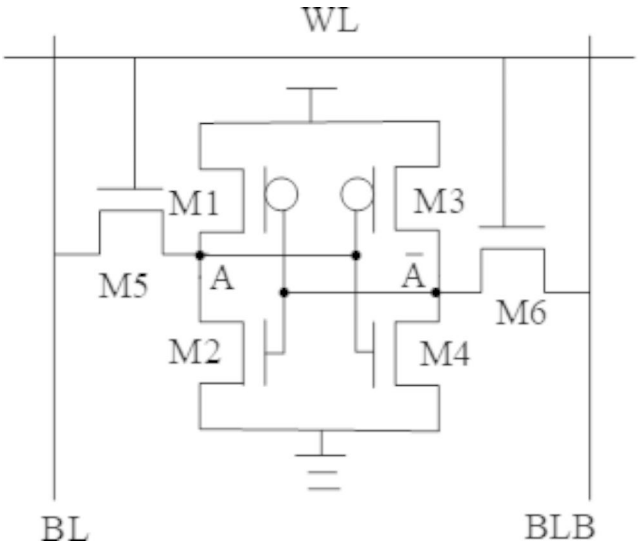

Fig. 1 The structure of an SRAM cell

4) Our experiment results show that both of our methods can maintain the high reliability and uniqueness of PUFs by using much smaller numbers of SRAM bits.
5) Clonability analysis shows that these two methods make the PUF clone attack infeasible as it is very difficult to reproduce unstable bits.

Compared to our preliminary work in [25], in this paper we not only present a new (second) method that has better reliability but also provide much more analysis and experimental results, including the clonability analysis, extensive experiments, and comprehensive comparison with previous work. The remainder of this paper is organized as follows. Section 2 provides some background information on this work. Section 3 analyzes the SRAM power-up behavior based on some experiments. Section 4 presents the two proposed SRAM PUF methods. Experimental results are given in Sect. 5. Section 6 provides the clonability analysis of our proposed methods. The comparisons with previous methods are shown in Sect. 7. Finally, we conclude this paper in Sect. 8.

2 Background

2.1 SRAM and SRAM PUF

In this work, we use the standard 6T SRAM, in which every cell is composed of six transistors, including two access transistors and four transistors in two cross-coupled inverters, as shown in Fig. 1. Each cell stores a single bit of information. The inverters ideally are designed to be symmetric and to match in size. However random variations that occur during manufacturing will cause a mismatch between the inverter pair. The mismatch makes the power-up state of the SRAM cell bias toward zero or one.

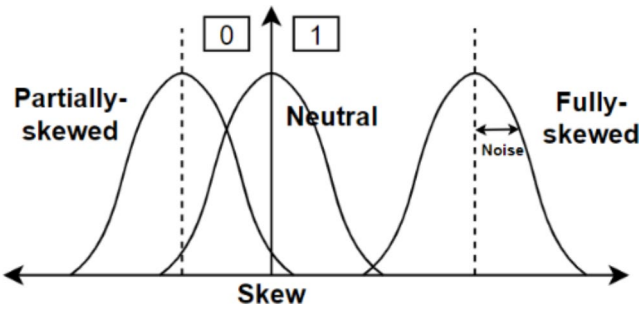

Fig. 2 The influence of process variation and noise on the cell's skew [14]

The threshold voltage mismatch between transistors M1 and M3 determines the power-up state of an SRAM cell. If the variation causes $V_{th,M1}$ slightly larger (smaller) than $V_{th,M3}$, then M1 will start conducting slower (faster) than M3, causing A to logic low (high), and the power-up state of the cell is A = 0 (1). The larger the mismatch between the two transistors is, the more stable the power-up value of a cell will be.

To use the SRAM PUF in the field, the addresses of selected SRAM cells need to be provided as the challenge to the SRAM, and the power-up values of the selected SRAM cells will be read out and taken as the responses. However the response of the SRAM PUF under the same input addresses may not always be the same. It is impacted not only by process variations but also by environmental variations, including voltage supply, operating temperature, aging effects, etc. These factors may cause cells to start up in a different state leading to unexpected PUF responses. Based on the skew values of the SRAM cells, they can be classified as partially-skewed, fully-skewed, and neutral cells, as described by a probability distribution function shown in Fig. 2. The fully-skewed cells are more stable, whereas the partially-skewed and non-skewed cells are unstable. Traditionally only fully-skewed SRAM cells that are stable in all conditions are selected as PUF cells, and hence the percentage of available cells is small.

2.2 PUF Quality

Two primary metrics for PUF quality have been used. One is Uniqueness which indicates whether the responses resulting from applying the same challenge to different PUF instances are different or not. Inter-Hamming distance (Inter-HD) is commonly used to show the degree of Uniqueness, where the Hamming distance of two words with equal length is the number of positions at which the corresponding symbols are different. Let $\boldsymbol{R_i}$ and $\boldsymbol{R_j}$ $(i \neq j)$ be the n-bit responses of

chip i and chip j, respectively, for some challenge C. The average inter-HD among K chips is defined as follows [26]:

$$\text{inter} - \text{HD} = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \frac{HD(R_i, R_j)}{n} \times 100\% \quad (1)$$

Ideally, Inter-HD should be 50%, which means that if different chips are applied with the same challenge, their responses should have a 50% difference.

Another metric is Reliability which indicates whether the responses resulting from applying the same challenge to the same PUF instance are always identical or not. Intra-Hamming distance (Intra-HD) [26] is usually used to show the degree of Reliability. Let R_i be the n-bit response extracted from chip i at the nominal condition and R'_i be the n-bit response of chip i extracted at some specific operating condition. We can collect m samples of R'_i, denoted as $R\prime_{i,t}$,$t = 1, \ldots, m$, by applying the challenge C to the same chip m times at different operating conditions. For chip i, its average intra-HD is calculated as follows:

$$\text{intra} - \text{HD} = \frac{1}{m} \sum_{t=1}^{m} \frac{HD(R_i, R\prime_{i,t})}{n} \times 100\% \quad (2)$$

We then define the Reliability of a PUF as follows [26]:

$$\text{Reliability} = 100\% - \text{intra-HD} \quad (3)$$

Ideally, intra-HD should be 0%. It means that if the same chip is applied with the same challenge, its response should always be the same in any condition.

2.3 Data Remanence in SRAM

Since SRAM is volatile, the data stored in SRAM will lose when it is powered off. However the contents of an unpowered SRAM chip will not disappear immediately; the decay process may take several seconds. Thus, if we power on an SRAM chip after powering it off for a short period, some cells will retain their previously stored data after the chip is powered on. This phenomenon is called "data remanence" [15, 27–29], and it is caused by various physical effects, such as carrier transport processes, ionic contamination, and hot carries. It was found in [15] that some bits are strongly biased to some logic values and thus can resist the data remanence effect. Their power-up values will not be influenced by previously stored data. In [15] these bits are selected as the PUF bits, which can maintain high reliability in various conditions.

2.4 PUF Clone Attack

Recently a PUF clone attack method has been presented and realized [20]. This is a big challenge for PUF designers as the attackers can successfully produce a second IC instance that has the same PUF response as a legal IC. The PUF clone attack uses the Photon Emission Analysis (PEA) to analyze the start-up behaviors of the SRAM and then goes through the Focused Ion Beam (FIB) circuit edit to make a second instance. The process will first remove the package and conduct Near-Infrared photon emission (NIR) in a nominal condition to get the start-up image of the SRAM, which can be used to determine whether the bit is skewed to 0 or 1. When the amplifier reads the SRAM, some carriers must undergo many cycles to characterize the cells to get an acceptable image from the emitted light. The photon image of stable bits will be much clearer than the unstable bits, and hence one can find out the locations of stable bits and perform FIBCE to clone the chip.

There are two ways of FIBCE for modifying the start-up behavior of SRAM. One is to remove individual transistors completely, but this will yield a cell incapable of storing data. The other is to trim the transistors individually to alter their dynamic performance and leakage. It can retain the full functionality of the SRAM. Based on the photon image from the NIR, methods to make the second instance with the same start-up behavior of stable bits as the attacked SRAM have been identified [20]. Obviously, it is possible to clone a stable bit since its power-up value will always be 1 or 0. However it would be very difficult to reproduce the behavior of an unstable bit.

3 SRAM Experiments and Observations

In this section, we will do two sets of experiments to explore the SRAM power-up behavior. These experiments are performed on a Mega 2560 board with a 1 M-bit off-the-shelf SRAM from Microchip Technology.

3.1 16-Times Power-Up Test

We read the power-up values of each SRAM bit 16 times for each test condition. If one SRAM cell's power-up values are all one or all zero, it will be considered stable. The bits that cannot satisfy this constraint are considered unstable bits.

Figure 3 shows the impacts of various temperatures at a nominal voltage. Figure 4 shows the influence of various voltages at a nominal temperature. Although over 30% of bits are stable for all temperatures at nominal voltage and over 30% of bits are stable bits for all voltages at nominal temperature, we found that there are only 8–10% bits that

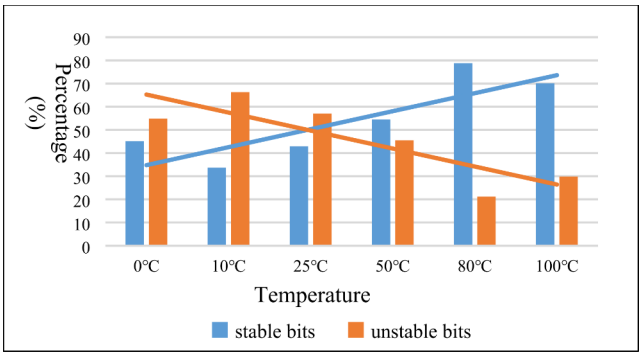

Fig. 3 SRAM bits across temperatures at nominal voltage

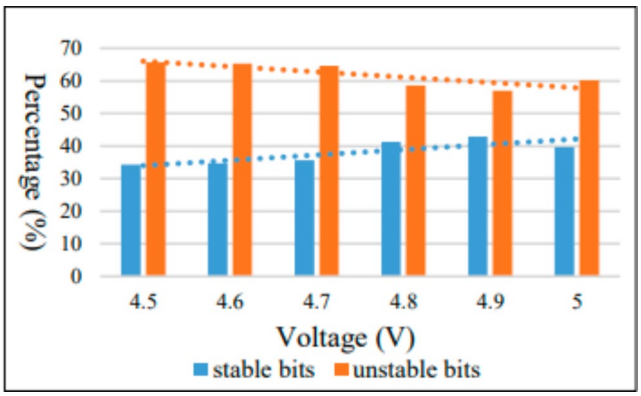

Fig. 4 SRAM bits across voltages at nominal temperature

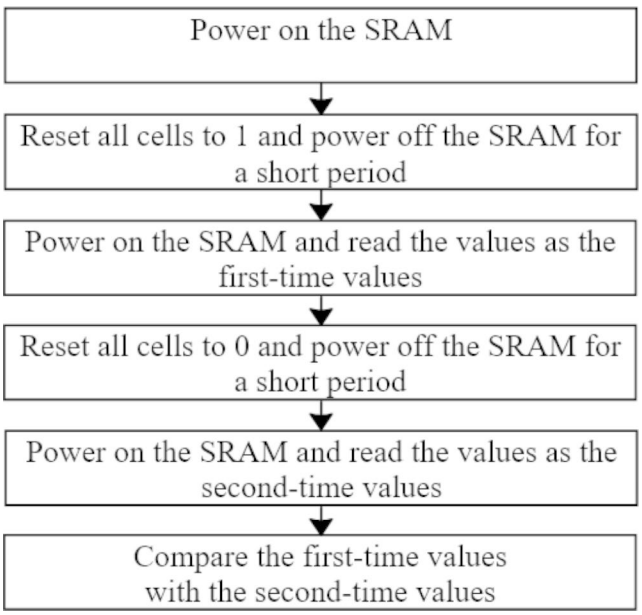

Fig. 5 The preliminary experiment flow [15]

are stable for all conditions, implying that if all application situations need to be considered, only 8–10% of memory bits can be used.

Table 1 Numbers of stable/unstable bits across temperatures with a 200 ms power-off time

T(°C)	Stable bits (%)	Unstable bits (%)
10	0	100
15	9.34×10^{-3}	99.99
20	2.86×10^{-2}	99.97
25	3.17	96.83
50	81.06	18.94
80	92.51	7.49
100	93.1	6.9

Table 2 SRAM bits across various power-off times at 10 °C

Time (ms)	Stable bits (%)	Unstable bits (%)
200	0	100
300	~0	~100
400	4.29×10^{-3}	99.99
500	5.15×10^{-2}	99.95
600	0.5	99.5
700	4.67	95.33
800	6.31	93.69
900	12.43	87.57

3.2 Data Remanence Test

In this test, we aim to analyze the data remanence behavior of SRAM. The test procedure we used is similar to that used in [15]. This experiment is conducted by changing the temperatures from 10 to 100 °C with the power-off time set at 200 ms and power supply voltage set at 5 V.

As Fig. 5 shows, we power on the SRAM, reset all cells to 1, and power off the SRAM. After powering off the SRAM for 200 ms, we power on the SRAM and read out all the power-up values which are marked as the first-time values. Then we will reset all bits of the SRAM to 0 and do the same procedure. We read out all the values again, marked as the second-time values. We compare the two values of each bit. If they are the same, it means that this bit is strongly biased in one direction. We then marked it as a stable bit. Otherwise, it will be marked as an unstable bit, even if its first-time value and second-time value are all opposite to the reset value. The experiment result shows that the data remanence effect will heavily impact the power-up values of the SRAM at low temperatures and less impact at high temperatures, as shown in Table 1. The number of stable bits is much smaller at low temperatures, whereas that of the unstable bits is much smaller at high temperatures.

Next, we analyze the data remanence effect on the SRAM under various power-off times from 100 ms to 900 ms at nominal voltage and temperature 10 °C (see Table 2). When the power-off time is small than 200 ms, there is no stable

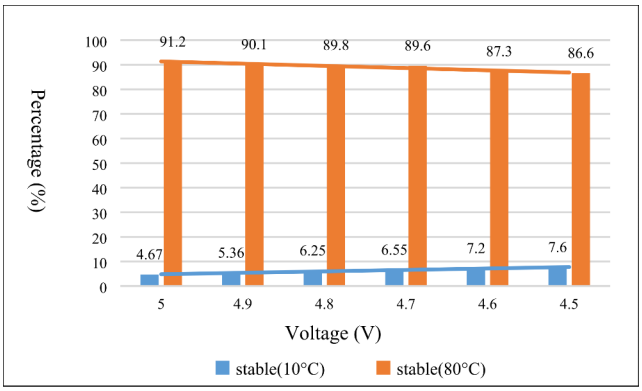

Fig. 6 Stable SRAM bits across voltages at 10 and 80 °C

bit. When the power-off time is 900 ms, we can get nearly 12.43% stable bits from SRAM. Therefore, if we want to get stable bits to produce enough PUF key bits at low temperatures, we need to choose an appropriate power-off time.

Additionally, we observed the data remanence effect on the SRAM under various voltages. The experiments were conducted with various voltages from 4.5 to 5 V, and the results are shown in Fig. 6. We found that the stable bit trend is different between low and high temperatures. Thus we show the variation at temperatures 10 and 80 °C to demonstrate the two different trends. The results reveal that at low temperatures, stable bits increase as voltage decreases, whereas at high temperatures stable bits decrease as voltage decreases.

This section uses two kinds of tests to classify the stable and unstable bits. The accuracy of this classification can be seen from two aspects. The first is the consistency of the classification between the tests and the enrollment (selection) procedures of our two proposed methods. Our experiments show that the bits classified by these two tests have almost perfect consistency with the bits selected by the two proposed methods. The second one is the error rates of the cells selected by the two proposed methods. The average error rates from the experimental data show a high level of accuracy in the classification under various voltage/temperature conditions. Thus the accuracy of the classification here ensures that the numbers of stable and unstable bits under various conditions can be accurately obtained. These numbers can then be used to determine the "worst" cases in our methods, i.e., the minimum numbers of stable and unstable bits that can be obtained. These minimum numbers will be used in the enrollment procedure to ensure high reliability of the stable and unstable bits under all conditions.

4 Proposed Methods

In this work, we propose two methods to use unstable bits in addition to stable bits as SRAM PUF bits. These two methods can increase the SRAM bit usage rate and resist the PUF clone attack. Furthermore, they do not need to modify the SRAM structure, so they can be readily applied to existing systems. Even though our proposed SRAM PUFs use unstable bits as PUF bits, they can achieve high reliability and maintain enough uniqueness.

4.1 16-Power-On Method

The first method "16-Power-on" is described as follows. We power on SRAM 16 times and compare all their power-up values. During the enrollment phase, if the values of a cell are all 1's or all 0's in the 16 readouts for each condition, the cell will be considered a reliable stable bit. On the other hand, if for a cell both logic 0 and 1 appear at least three times in the 16 readouts for each condition, it will be considered a "reliable" unstable bit. In the reconstruction phase, if the values of a cell are all 1's or all 0's, its response will be set to 1. Otherwise, its response will be set to 0. The times of power-on of SRAM in our method are referred to [10] and [11] in which 15 times of power-on-off are employed. We use 16 because it is the power of 2 that is closest to 15, and our experiments also show that this is a number that will give us good reliability without spending too much time.

Note that both the enrollment and reconstruction processes can be implemented by software running on an on-chip processor or by dedicated hardware. If an on-chip processor can be used, then no additional control or processing logic is needed. For the hardware implementation we would need a controller and some logic gates to process the data, as can be seen in enrollment and reconstruction processes to be described in this section later.

Refer to Fig. 7 which shows how we select reliable stable bits and reliable unstable bits in the enrollment phase. From Figs. 3 and 4, we have shown that the number of unstable bits tends to be smaller at high temperature and high voltage, whereas the number of stable bits tends to be smaller at low temperature and low voltage. Therefore, we will first go through the 16-times power-up tests at high temperature and high voltage to select candidate unstable bits. Then we will select candidate stable bits at low temperature and low voltage by going through the 16-times power-up test. By this procedure, we can screen out a large number of unreliable bits (either stable or unstable). Next, we will go through various temperatures at the nominal voltage to further discard those unreliable bits to enhance the reliability of the selected bits. In this step, we will go through the 16-times power-up test just once in each condition. Finally, we store

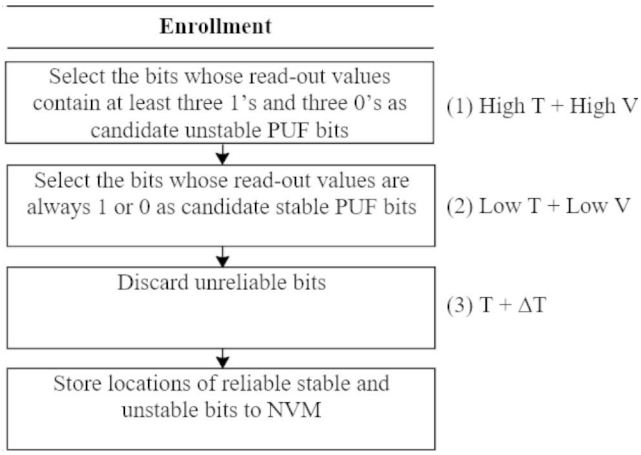

Fig. 7 "16-Power-On" Enrollment flow

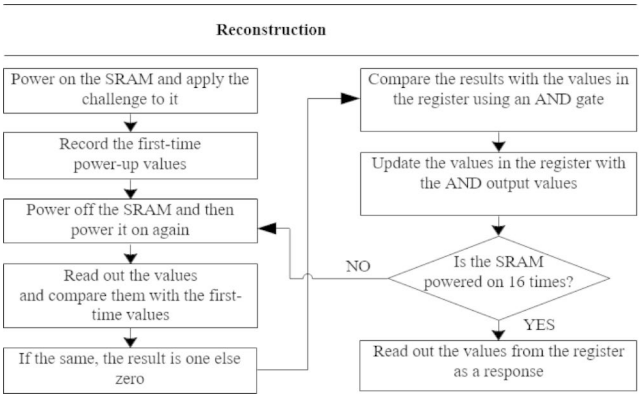

Fig. 8 "16-Power-On" Reconstruction flow

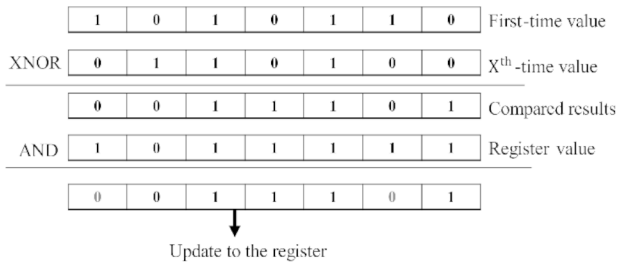

Fig. 9 "16-Power-On" Reconstruction example

Table 3 SRAM Bits across various power-off times at 10 °C

Time (ms)	SS0, SS1(%)	RD0, RD1(%)
200	0	100
300	~0	~100
400	2.29×10^{-3}	99.99
500	2.65×10^{-2}	99.97
600	0.3	99.7
700	3.23	96.77
800	4.01	95.99
900	10.23	89.77

the addresses of all selected (stable and unstable) bits as the

Table 4 SRAM bits across temperatures with a 700 ms power-off time at 5 V

T(°C)	SS0, SS1(%)	RD0, RD1(%)
10	3.23	96.77
25	49.52	50.48
50	52.59	47.41
80	54.3	45.7
100	54.5	45.5

Table 5 SRAM bits across voltages with a 700 ms power-off time at 10 °C

Voltage (V)	SS0, SS1(%)	RD0, RD1(%)
5	3.23	96.77
4.9	5.14	94.86
4.8	6.76	93.23
4.7	9.38	90.62
4.6	13.64	86.36
4.5	15.25	84.75

Table 6 SRAM bits across voltages with a 700 ms power-off time at 80 °C

Voltage(V)	SS0, SS1(%)	RD0, RD1(%)
5	54.3	45.7
4.9	52.14	47.86
4.8	50.1	49.9
4.7	49.43	50.57
4.6	47.92	52.08
4.5	46.02	53.98

PUF challenge into nonvolatile memory (NVM) and complete the enrollment procedure.

In the reconstruction phase, each PUF bit determined in the enrollment phase will be read out 16 times. If all 1's or all 0's appear for a PUF bit, then the bit is a stable one; otherwise it is an unstable one. This procedure is demonstrated in Figs. 8 and 9 and described below. The SRAM PUF is first powered on, and the controller will use the stored challenge to read out the first-time power-up values of the selected SRAM bits. The first-time power-up values will be saved into another SRAM (functional SRAM). Then the SRAM PUF will be powered on again, and the second-time power-up values will be read out. The second-time power-up values will be compared to the first-time power-up values stored in functional SRAM bit-by-bit using XNOR gates. If two bits are the same, the compared result will be one; else the result will be zero. The results will be compared with the values in a register using AND gates. The values of the register are initially set to all 1's in the first-time power-up. If the AND output value is 0, it means that the cell's first-time power-up value and second-time power-up value are different. If the output value is 1, it means these two values are the same. The AND output values will then be used to update the register. After that, the SRAM PUF is powered on again, and the same process is executed. We will do the

comparison 15 times, i.e., read out the PUF bits 16 times, to obtain the final responses in the register. If the final value of a bit in the register is one, it means that the cell is a stable bit. Otherwise it is an unstable bit.

This method requires 16 times of power-up processes to generate the response, uses extra SRAM to record the power-up pattern, and needs some logic gates for the bit comparison. We can use functional SRAM in the system as the extra SRAM. Thus, our proposed method only needs some extra bit comparison logic, and hence the area overhead is small.

Note that for various PUF ICs, we sometimes cannot find enough reliable unstable SRAM cells under the parameter values adopted in the enrollment procedures here (i.e., the *threshold* of at least "3" times of 1's and 0's among the "16" *readout* values make the cell classified into a "reliable" unstable bit). It can be expected that if we decrease the threshold value under a fixed number of readouts, the number of selected unstable bits will increase but the reliability of these bits will decrease. On the other hand, if we increase the number of readouts with a fixed threshold, the reliability of the stable bits will increase with about the same number of stable bits. The number of selected unstable bits will increase but the reliability of these bits will decrease. Therefore it is a tradeoff between the reliability and the number of selected bits. Thus if more bits are needed, a more complex error-correcting mechanism may be necessary to ensure enough overall reliability.

4.2 Remanence-Based Method

The "16-Power-On" method chooses the bits which are stable or unstable in all conditions. It screens out some unstable bits that may be unstable in some conditions and stable in other conditions. In the second method "Remanence-Based", we will also use some of these unstable bits as PUF bits. We use the data remanence effect to help us choose our PUF bits. Note again that in [15], the data remanence effect is used to select stable bits only, whereas, in our method, we use the effect to select not only stable bits but also unstable bits.

From Tables 1 and 2, the unstable bits at high temperatures are relatively rare because of the weak data remanence effect. They are less than 10% at high temperatures but higher than 90% at low temperatures. If we only use the "reliable" unstable bits that are unstable in all conditions, the number of available "reliable" unstable bits might be less than 10%. To verify this, we do the same procedure as the data remanence tests in Sect. 3.2. The result shows that only 128 bits from 1 M bits are "reliable" unstable bits in all conditions on average. From this result, we can see that the "reliable" unstable bits are very rare in the SRAM under the data remanence effect, and lots of unstable bits are stable in some conditions but unstable in other conditions. So, we propose a new method that reuses some of these unstable bits as PUF bits. We will first describe the basic operation of our PUF, from which we can classify those "reliable" SRAM bits into two categories. The enrollment and reconstruction procedures will then be described.

The basic operation of our PUF is that the controller will firstly power on the SRAM and reset half of the SRAM bits to 1 and the others to 0. Then SRAM will be powered off for a short period and then powered on. After powering on the SRAM again, we will read the power-up value of each cell to determine whether it will be used as a PUF bit or not, as described next.

Based on the results of the above operation, the SRAM bits can be classified into two categories depending on whether their power-up values after reset are the same as their reset values or not. We call the bits that have the same values as their reset values the Reset Dominated bits (RDs). The other bits are called Strong stable bits (SSs) since they can resist the data remanence effect and are strongly biased in one direction. If one bit is reset to 1 and its power-up value is also 1, it will be called a Reset-Dominated-at-1 bit (RD1). Similarly RD0, SS1, and SS0 can be defined. Notice that RDs contain not only unstable bits but also stable bits. Unstable bits of RDs are originally weak biased to 1 or 0, but they become stable at 1 or 0 under the data remanence effect. Stable bits of RDs might originally strong bias to the reset value, so their power-up values are the same as the reset value. As for the SSs, they only contain the stable bits because they can resist the data remanence effect and retain their original power-up value. We will use "Reliable" RDs and "Reliable" SSs as PUF bits. "Reliable" means these bits act with the same behavior in all conditions. Our PUF response indeed includes both stable bits and unstable bits. To reduce the cost of finding "Reliable" SSs and "Reliable" RDs, we did some experiments to observe the variation of SSs and RDs and found out some specific conditions to help us quickly select our PUF bits.

Table 3 shows the impact of various power-off times at 10 °C. It infers that the SSs increase when the power-off time increases. Table 4 displays the temperature influence on RDs and SSs at a fixed power-off time. It demonstrates that the data remanence effect plays a more important role in SRAM at low temperatures. Tables 5 and 6 exhibit the impact of various voltages under the low temperature and the high temperature, respectively. From these experimental results it can be concluded that the number of RDs decreases when the temperature and power-off time increase. The SSs contrast with the RDs. The number of SSs increases with the decreasing voltage at low temperatures, whereas

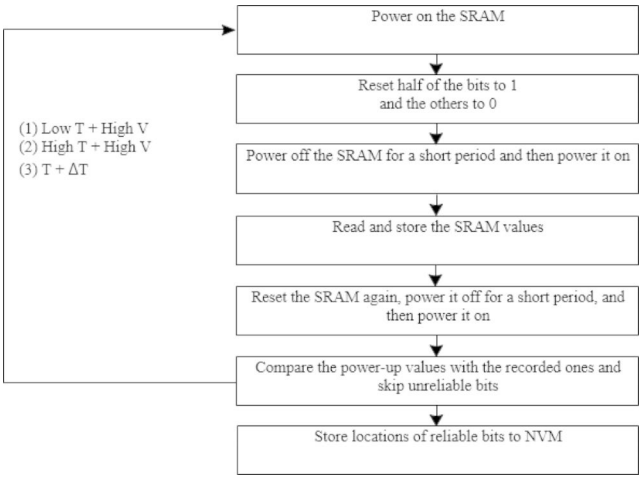

Fig. 10 "Remanence-Based" Enrollment flow

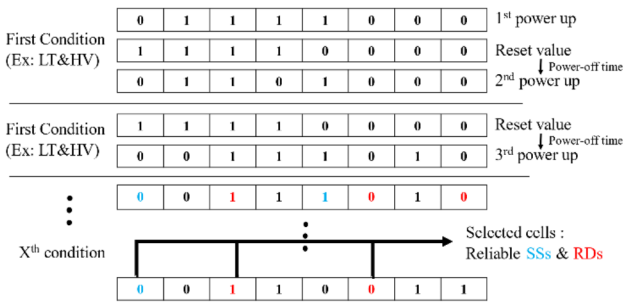

Fig. 11 "Remanence-Based" Enrollment example

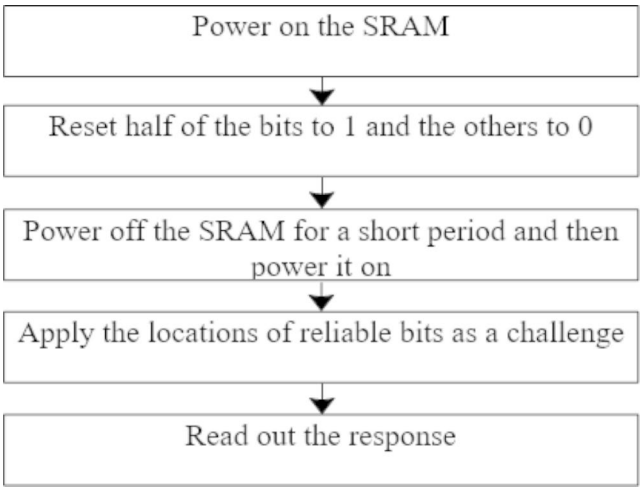

Fig. 12 "Remanence-Based" Reconstruction flow

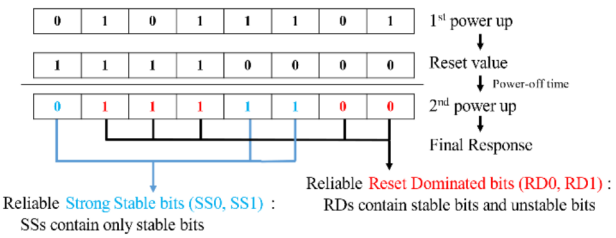

Fig. 13 "Remanence-Based" Reconstruction example

the number of RDs increases with the decreasing voltage at high temperatures.

Based on our experimental results and observations, the process of enrollment phase will firstly undergo low temperature, high voltage, and an appropriate power-off time to choose SSs. Then it will go through high temperature, high voltage to choose RDs with the same power-off time as choosing SSs. After that, it will go through various temperatures at nominal voltage to further screen out the unreliable bits. Each bit whose power-up values are all the same under those conditions will be selected as a PUF bit. Finally, the addresses of these selected reliable SSs and RDs will be our PUF challenges.

Refer to Figs. 10 and 11. In the enrollment phase, after the controller powers on the SRAM, it will reset half of the SRAM cells to 1 and the others to 0. Then the SRAM will be quickly powered on after powering down for a short period. The power-up values of SRAM bits will be recorded. After that, we reset the SRAM, power it off for a short period, and power it on again. The power-up values will be compared with the previously recorded power-up values. If the power-up value of one cell is different from the recorded value, it will be screened out. It will go through different conditions, and finally the locations of those reliable bits will be stored in NVM.

Refer to Figs. 12 and 13, which explain the reconstruction method of our SRAM PUF. In the first step, the SRAM PUF is first powered on, and the controller will reset half of the SRAM bits to 1 and the other half to 0. It then powers off the SRAM for a short period. Then the stored challenge will be applied to read out the power-up values of the previously determined PUF bits.

In this proposed method, we use the data remanence effect to employ the unstable bits as PUF bits. The RDs contain both stable bits and unstable bits, whereas SSs only contain stable bits. It can confuse the attacker who owns the skill of the PUF clone attack. Notice that "16-Power-On" uses the unstable bits that are always unstable, whereas "Remanence-Based" uses the unstable bits that are not always unstable but may occasionally be unstable in some conditions. In the next section, we will show the experimental results of our two proposed methods.

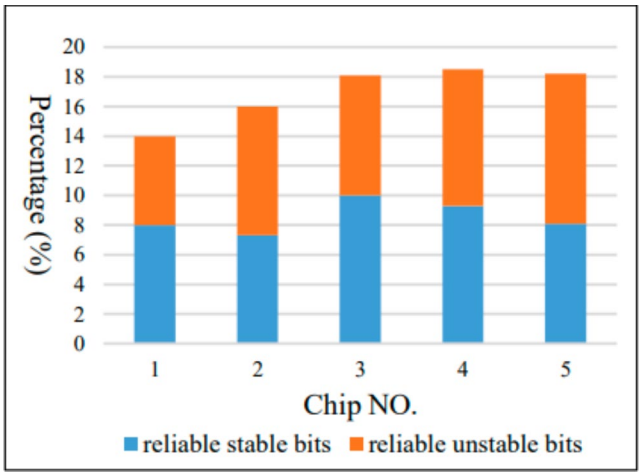

Fig. 14 "Reliable" PUF bits percentage in each SRAM chip

5 Measurement Results

This section shows measurement results under various environments and aging conditions. Experiments are performed on a Mega 2560 board with a 1 M bits off-the-shelf SRAM chip from Microchip Technology inside a temperature-controlled chamber. We tested each proposed method with 5 off-the-shelf SRAM ICs.

For the first proposed method "16-Power-On", we divided 1 M bits SRAM into multiple 1 K-bit components. We first evaluate the percentages of reliable stable bits and reliable unstable bits after we did our enrollment procedure. In Fig. 14, the result shows that the reliable stable bits in SRAM are 8–10% and that reliable unstable bits are 6–10%, totally in the range of 14–20%. Therefore, we can expect that each 1 K-bit PUF can produce a 128-bit reliable PUF response.

For the second proposed method, "Remanence-Based", we divided 1 M bits SRAM into multiple PUFs. Each PUF has 6 pages of SRAM bits (1536 bits) to generate a 128-bit key because after we went through our enrollment procedure, there are 2–4% reliable SSs with a 700 ms power-off time in the whole SRAM, whereas the percentage of reliable RDs in the whole SRAM is 35%. Due to security issues, we expect that SSs and RDs occupy respectively one-half. The selected PUF bits contain all the SSs and some RDs to generate a 128-bit key. In that case, attackers are difficult to figure out the correct key-value based on the addresses of PUF bits, even if they know which part is reset to 1 and which to 0.

Then we evaluate the reliability of the PUF key generated by the "16-Power-On" method. We test SRAM PUFs under various temperatures from 0 to 100 °C. In each temperature degree, the experiment had been done through various voltages from 4.5 to 5 V, and each condition will be tested 10 times. Table 7 shows the experimental results of the error rates of selected stable bits and unstable bits across various temperatures without the aging effect. We can see that the

Table 7 PUF bits across various temperatures without aging for "16-times Power-on" method

T(°C)	Error rate of selected stable bits (%)	Error rate of selected Unstable bits (%)
0	2.1	1.8
10	0.9	1.9
25	1.1	3
50	0.9	3.1
80	1.07	3.3
100	2.1	2
Avg.	1.36	2.52

Table 8 PUF bits across various temperatures without aging for "Remanence-Based" method

T(°C)	Error rate of selected SS0, SS1(%)	Error rate of selected RD0, RD1(%)
10	0.14	0.013
25	0.05	0.1
50	0.04	0.4
80	0.05	0.5
100	0.005	0.8
Avg.	0.057	0.3626

Table 9 PUF bits across various temperatures with aging for "16-times Power-on" method

T(°C)	Error rate of selected stable bits (%)	Error rate of selected Unstable bits (%)
0	2.3	1.5
10	1.0	1.7
25	1.13	2.8
50	0.9	3.1
80	1.1	3.2
100	2.6	1.7
Avg.	1.51	2.33

Table 10 PUF bits across various temperatures with aging for "Remanence-Based" method

T(°C)	Error rate of selected SS0, SS1(%)	Error rate of selected RD0, RD1(%)
10	0.14	0.02
25	0.06	0.22
50	0.07	0.42
80	0.056	0.48
100	0.005	0.7
Avg.	0.0662	0.368

error rates of stable (unstable) bits range from 0.9 to 2.1% (1.8 to 3.3%) with an average of 1.36% (2.52%). These experimental results show that on average our proposed method can reach about 98.06% reliability (100% minus error rates as defined in Sect. 2.2) even though unstable bits are used as PUF bits.

As for the reliability of the "Remanence-Based" SRAM PUF, we test SRAM PUFs under various temperatures from 10 to 100 °C with a 700 ms power-off time. In each temperature degree, the experiment had been done through various voltages from 4.5 to 5 V, and each condition will be tested 10 times. The reason why we did not go through 0 °C is that it is difficult to find enough reliable SSs at 0 °C with a 700 ms power-off time under the data remanence effect. So, the "Remanence-Based" method can only go from 10 to 100 °C with a 700 ms power-off time. Table 8 shows the experimental results of the error rates of the selected RDs and selected SSs across various temperatures without the aging effect. We can see that the error rates of SSs (RDs) bits range from 0.005 to 0.14% (0.013–0.8%) with an average of 0.057% (0.3626%). These experimental results show that on average our SRAM PUF can reach about 99.79% reliability.

We also simulate the aging effect by applying high temperature and high voltage to the ICs. The aging process is the same as that in [14] which is conducted at high voltage (5.5 V) and high temperature (100 °C) for 5 h. After aging the SRAM, we conducted experiments at various temperatures with various voltages from 4.5 to 5 V. Tables 9 and 10 show the aging results. For the "16-Power-On" method, the error rate of selected stable bits slightly increased, whereas the error rate of selected unstable bits slightly decreased. It is because both stable and unstable bits are affected by the aging effect and they tend to become more unstable with aging. This happens to make the unstable bits "more reliable". Therefore the reliability gain of selected unstable bits may somehow make up the reliability loss of selected stable bits due to the aging effect, and the "16-Power-On" method is likely to keep similar reliability (98.08%) when aging occurs. This also implies that if the enrollment process can be re-done after a device is used for some time, more reliable unstable bits may be available to select when aging does occur. This feature does not exist in most other PUF designs where when aging occurs, the number of usable bits will become less. As for the "Remanence-Based" method, the error rate of the selected SSs and RDs increases slightly but remains at high reliability (99.78%).

From the measurement of the error rates, we can observe the accuracy of the classification of the reliable and unreliable bits in our proposed methods. We can also see whether this enrollment classification (or selection) is realistic. As the experimental results indicate, the average error rates of the two proposed methods are comparable with those

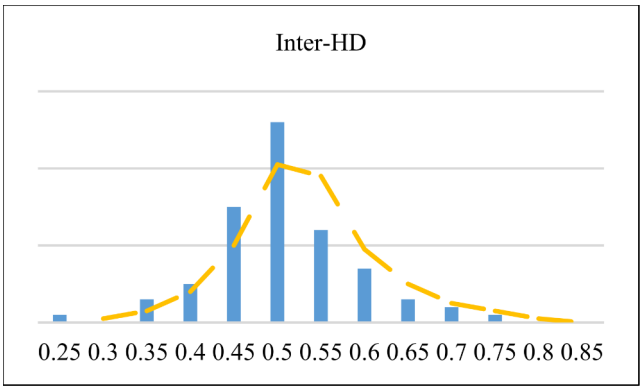

Fig. 15 "16-Power-On" Inter-HD

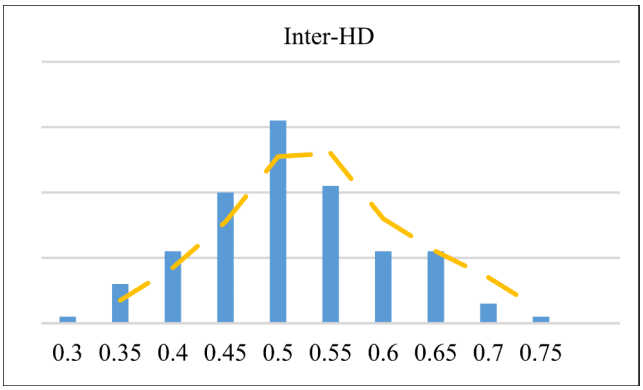

Fig. 16 "Remanence-Based" Inter-HD

obtained in most SRAM PUFs that used stable bits only, hence these methods should be feasible for generating various keys if some error-correcting technique is also used. Also since our average error rate is low, the selected bits should be reliable, which means our classification dividing the cells into reliable and unreliable is accurate. Due to the usage of both the unstable bits, the number of reliable bits is greatly increased, which ensures that the classification can always find enough reliable bits for the key generation with a relatively small PUF size. Therefore the enrollment classification (or selection) of our proposed methods is suitable for generating the keys.

Next, we consider the uniqueness of the PUF key. We randomly take 100 PUFs from each SRAM IC and calculate the inter-Hamming-distance (HD). Refer to Figs. 15 and 16. On average the inter-HD for the "16-Power-On" method is 49.33%, and the "Remanence-Based" method is 49.5%, which confirms the sufficient randomness of our PUF keys in both proposed methods.

In the previous description, the RDs contain not only unstable bits but also stable bits. So finally, we do some tests to find out the percentage of the unstable bits selected in the "Remanence-Based" SRAM PUF. The SRAM PUF will go through 10 times power-up without using the data

Table 11 Comparison with other works

	HOST 2014 [14]	ISLPED 2017 [15]	Microelectron Reliab 2021 [16]	JETTA 2022 [17]	16-Power-On	Remanence-Based
128-bit key	1 M bits	256k bits	NR.	NR.	1k bits	1.5k bits
SRAM bit usage rate (%)	0.0125	0.05	> 6	< 5	> 12.5	8
Reliability (%)	Fresh: 99.89 Aging: 99.70	Fresh: ~ 100 Aging: ~ 100	Fresh: ~ 100 Aging: ~ 100	Fresh: 100 Aging: 100	Fresh: 98.06 Aging: 98.08	Fresh: 99.79 Aging: 99.78
Inter-HD (%)	49.8	49.35	49.94	NR.	49.33	49.5
Clonability	No defense	No defense	No defense	No defense	Infeasible	Infeasible

NR.: Not Reported

remanence effect. If the power-up value of the cell is flipped during the 10 times powering up, it will be considered an unstable cell. Otherwise, it will be considered a stable cell. The result shows that over 30% of the selected PUF cells will be considered unstable cells at room temperature and that 20% of the selected RDs bias to values opposite to their reset values.

6 Clonability Analysis

As described in [20], cloning a PUF requires two main procedures. One is a characterization procedure that extracts the complete challenge/response behavior of a PUF, and the other is an emulation procedure that models or reproduces a PUF with identical challenge/response behavior. Characterization of SRAM PUF may be done by observing the power-up behavior identified by the NIR photon image or by keeping SRAM at a low temperature at which the data in SRAM can be kept for a long period after power-down [28] such that an attacker can move the SRAM to an environment where the power-up value can be extracted. As for emulation, the attacker can modify the NMOS or PMOS drain connection to make the start-up value strong in 1 or 0 to produce an IC with the same response for the same challenge if only stable bits are used.

Next, we show that both of our proposed methods can protect the SRAM PUF because the emulation procedure cannot be carried out when unstable bits are utilized as PUF bits. The unstable bits are basically due to less mismatch of the two cross-coupled inverters of an SRAM cell, and it is almost impossible to predict the bias values of the unstable bits. The cloning method described in [20] may be able to use PEA (Photonic Emission Analysis) to find the locations of stable bits and then conduct FIBCE (Focused Ion Beam circuit edit) to modify the thickness of the drain diffusion of the cross-coupled inverters and hence change the threshold voltage mismatch condition. However though it is possible to make an SRAM bit bias in one direction by trimming a certain drain diffusion using FIBCE to create a stable bit (either at logic 1 or 0), it is very hard to make two transistors have almost the same threshold voltage to create an unstable bit since it would be very hard to trim the drain diffusion regions of two transistors to the same thickness. Furthermore, even this can be done, the process variation during manufacturing can still lead to a mismatch of the two transistors, resulting in a stable or an "unreliable" unstable bit. Thus, when one wants to make a cell unstable using FIBCE, he/she may get a partially-skewed one, a neutral one, or even a fully-skewed one. Clearly one cannot clone the neutral cells used in the "16-Power-On" method and the partially-skewed cells used in the "Remanence-Based" method. Therefore, even if the attackers know the characteristics of each SRAM cell, it is still very hard to clone all cells if some of these cells are unstable.

7 Comparison

We compare our methods with those in [14–17] because these approaches also focus on PUF bit selection without modifying the SRAM structure. Table 11 shows the comparison results. The first row shows the various methods. The second row starts with "128-bit key" and shows how many SRAM bits are needed for each method to produce the 128-bit PUF key. The row "SRAM bit usage rate" means the percentage of SRAM bits that can be used as PUF bits in the SRAM. The Reliability and Inter-HD were calculated based on the equations in Sect. 2.2 under both fresh and aged SRAM conditions (see Sect. 5). The row "Clonability" is the feasibility that the attacker who owns the PUF clone technique [20] can successfully clone the attacked SRAM. From the comparison table we can see that our first and second method can use a much smaller area, 1k bits and 1.5k bits respectively, to produce the 128-bit key, whereas the first two methods [14, 15] use at least 256k bits and the bit

usage rates is lower than 0.1%. As for the SRAM usage rate in [16], the results of the 50 strongest SRAM cells out of 832 cells (about 6%) are given, which show that the reliability is nearly ~ 100%. The work in [17] does not give the exact bit usage rate. However from the data provided, the SRAM usage rate is about 5% if 100% reliability is targeted. The bit usage rates of our methods are about 12.5% and 8% for the 16-power-on and the Remanence-based methods, respectively, but the reliability is lower.

Our inter-HD is close to these methods but the reliability of the "16-Power-On" method is not as high as other methods. This problem can be overcome by using some error-correcting technique [8].

The important thing here is that only our proposed methods can successfully defend the clone attack. Finally, we compare the "16-Power-On" method with the "Remanence-Based" method. For the SRAM bit usage rate, "16-Power-On" is better than "Remanence-Based". In the latter method there are ~ 4% reliable SSs and ~ 35% reliable RDs based on the experiments. Due to security issues, SSs and RDs each are expected to occupy one-half. Thus 128-bit key contains all reliable SSs, and the rest are filled with reliable RDs. Even though the number of reliable RDs is large, the bit usage rate is reduced to 8% which is less than that of the "16-Power-On" method. On the other hand, the "Remanence-Based" method has better reliability and Inter-HD. Hence although the SRAM bit usage rate of the "16-Power-On" method is high, the average error rate of the selected bits is also high. This is because there are few "truly unstable" cells that are unstable in all conditions. These "truly unstable" cells are perfectly balanced in the electrical effort required to charge the outputs of the cross-coupled inverters in an SRAM cell. Even if there is a small imbalance introduced by the temperature/voltage variations, the cell can be biased in one direction, implying most of the cells tend to be biased in some conditions. However the remanence-based method exploits not only the stable bits but also the bits that may be stable in some conditions and unstable in other conditions. Therefore the remanence-based method is less influenced by the lack of "truly unstable" bits.

Since we do not change the way to store the PUF information in NVM or any other storage devices, our proposed methods need the same storage space to store the PUF information as the works in [14–17]. We just try to make use of the truly unstable bits such that the size of SRAM required to generate the needed number of PUF bits can be significantly reduced. However we do need more temporary storage to store the PUF data during the enrollment process since some results of XOR and AND operations need to be stored. Also, our proposed methods do have different PUF enrollment and reconstruction procedures that may consume more power and take more time than other methods.

One practical problem with our methods is that we must be able to control the supply power of SRAM for each reconstruction of PUF responses. This capability should exist for any SRAM that is to be used for PUF because during the enrollment process this capability is needed to identify those usable bits under different environments (variable temperatures and voltages). When such SRAM is retrofitted into existing hardware designs, this capability may be disabled in other PUF designs, but in our methods this capability must be kept since our methods also require powering on and off the SRAM during the reconstruction process. This can be done by keeping an independent power control pin for the SRAM. If the SRAM to be used is large, then it may have its own power supply anyway. Otherwise our methods may introduce more resource overhead to control the power supply than other methods. Another way is to separate the PUF SRAM from the general computation SRAM module. This would make the SRAM PUF a non-intrinsic PUF. However since SRAM PUF is a secure primitive, using dedicated SRAM as PUF can reduce the risk of being attacked and hence may be desired in some critical cases.

8 Conclusion and Future Work

In this work, we propose two methods that use both stable and unstable bits to increase the numbers of usable bits for PUF applications. We also show that the proposed methods can defend the current PUF clone technique for SRAM PUF. Experimental results show that the "16-Power-On" method can achieve 98.06% reliability and 49.33% inter-HD and that the "Remanence-Based" method can reach 99.79% reliability and 49.5% inter-HD. If even higher reliability is required, some error-correcting techniques can be applied to our proposed methods. With our methods, it is almost impossible to clone an SRAM chip since it is difficult to clone all unstable bits. Additionally it is worth pointing out that the proposed methods can be applied to more sophisticated SRAM designs such as 7T, dual-port 8T, and dual-port 9T SRAM, as long as the mismatched feature of transistors in the cross-coupled inverters are utilized for the PUF design just like that in a 6T SRAM design. Finally, in the "16-Power-On" method, since the reliability of the PUF bits depends on the number of power-ups and the threshold we define to classify the "reliable" unstable bits, the reliability can still be improved by the selection of the number of power-ups and the threshold. In the future, we plan to conduct more experiments with a wider range of the selected parameters to find the best reliability with suitable performance.

Acknowledgments We would like to thank the anonymous reviewers for their constructive feedback. This work was partially supported

by the Ministry of Science and Technology of Taiwan under Contract 107-2218-E-006-025.

Data Availability All data generated or analyzed during this study are included in this published article.

Declarations

Conflict of Interest The co-author, Kuen-Jong Lee, is one of the Editorial Board Members of Journal of Electronic Testing: Theory and Applications.

References

1. Maiti A, Casarona J, McHale L, Schaumont P (2010) "A large scale characterization of RO-PUF," in *Proc. IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 94–99, https://doi.org/10.1109/HST.2010.5513108
2. Liu W, Zhang L, Zhang Z, Gu C, Wang C, O'Neill M, Lombardi F (2019) XOR-based low-cost reconfigurable PUFs for IoT security. ACM Trans Embedded Comput Syst (TECS) 18(3):1–21. https://doi.org/10.1145/3274666
3. Herder C, Yu M, Koushanfar F, Devadas S (2014) "Physical unclonable functions and applications: A Tutorial," *Proc. of the IEEE*, vol. 102, no. 8, pp. 1126–1141, https://doi.org/10.1109/JPROC.2014.2320516
4. Gu C, Liu W, Cui Y, Hanley N, O'Neill M, Lombardi F (2019) A flip-flop based arbiter physical unclonable function (APUF) design with high entropy and uniqueness for FPGA implementation. IEEE Trans Emerg Top Comput 9(4):1853–1866. https://doi.org/10.1109/TETC.2019.2935465
5. Tuyls P, Schrijen G-J, Škorić B, Van Geloven J, Verhaegh N, Wolters R (2006) "Read-proof hardware from protective coatings," in *Proc. International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 4249, pp. 369–383. https://doi.org/10.1007/11894063_29
6. Kumar SS, Guajardo J, Maes R, Schrijen G, Tuyls P (2008) "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *Proc. IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 67–70. https://doi.org/10.1109/HST.2008.4559053
7. Wu M, Yang T, Chen L, Lin C, Hu H, Su F, Wang C, Huang JP, Chen H, Lu CC, Yang EC, Shen RS (2018) "A PUF scheme using competing oxide rupture with bit error rate approaching zero," in *Proc. IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 130–132. https://doi.org/10.1109/ISSCC.2018.8310218
8. Handschuh H "Hardware-anchored security based on SRAM PUFs, Part 1,"IEEE Security & Privacy, vol. 10, no. 3, pp.80–83, May-June 2012. https://doi.org/10.1109/MSP.2012.68
9. Kazumori K, Ueno R, Homma N (2019) "A ternary fuzzy extractor for efficient cryptographic key generation," in *Proc. IEEE International Symposium on Multiple-Valued Logic (ISMVL)*, pp. 49–54. https://doi.org/10.1109/ISMVL.2019.00017
10. Mathew SK, Satpathy SK, Anders MA, Kaul H, Hsu SK, Agarwal A, Chen GK, Parker RJ, Krishnamurthy RK, De V (2014) "A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm CMOS," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 278–279. https://doi.org/10.1109/ISSCC.2014.6757433
11. Satpathy S, Mathew SK, Suresh V, Anders MA, Kaul H, Agarwal A, Hsu SK, Chen G, Krishnamurthy RK, De VK (April 2017) A 4-fJ/b delay-hardened physically unclonable function circuit with selective bit destabilization in 14-nm trigate CMOS. IEEE J Solid-State Circuits 52(4):940–949. https://doi.org/10.1109/JSSC.2016.2636859
12. Jang J, Ghosh S (2015) "Design and analysis of novel SRAM PUFs with embedded latch for robustness," in *Proc. International Symposium on Quality Electronic Design*, pp. 298–302. https://doi.org/10.1109/ISQED.2015.7085443
13. Zhou C, Satapathy S, Lao Y, Parhi KK, Kim CH (2016) "Soft response generation and thresholding strategies for linear and feed-forward MUX PUFs," in *Proc. International Symposium on Low Power Electronics and Design*, pp. 124–129. https://doi.org/10.1145/2934583.2934613
14. Xiao K, Rahman MT, Forte D, Huang Y, Su M, Tehranipoor M (2014) "Bit selection algorithm suitable for high-volume production of SRAM-PUF," in *Proc. IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 101–106. https://doi.org/10.1109/HST.2014.6855578
15. Liu M, Zhou C, Tang Q, Parhi KK, Kim CH (2017) "A data remanence based approach to generate 100% stable keys from an SRAM physical unclonable function," in *Proc. IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 1–6. https://doi.org/10.1109/ISLPED.2017.8009192
16. Saraza-Canflanca P, Carrasco-Lopez H, Santana-Andreo A, Brox P, Castro-Lopez R, Roca E, Fernandez FV (2021) Improving the reliability of SRAM-based PUF under varying operation conditions and aging degradation. Microelectron Reliab 118(114049). https://doi.org/10.1016/j.microrel.2021.114049
17. Wang W, Singh AD, Guin U (2022) A systematic bit selection method toward robust and unique SRAM PUFs. J Electron Testing: Theory Appl (JETTA) 38(3):235-246. https://doi.org/10.1007/s10836-022-06006-x
18. Rührmair U, Sölter J (2014) "PUF modeling attacks: An introduction and overview," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6. https://doi.org/10.7873/DATE. 2014.361
19. Wei S, Wendt JB, Nahapetian A, Potkonjak M (2014) "Reverse engineering and prevention techniques for physical unclonable functions using side channels," in *Proc. Design Automation Conference*, pp. 1–6. https://doi.org/10.1145/2593069.2593204
20. Helfmeier C, Boit C, Nedospasov D, Seifert J (2013) "Cloning physically unclonable functions," in *Proc. IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 1–6. https://doi.org/10.1109/HST.2013.6581556
21. Cambou B, Orlowski M (2016) "PUF designed with resistive RAM and ternary states," in *Proc. Cyber and Information Security Research Conference*, pp. 1–8. https://doi.org/10.1145/2897795.2897808
22. Korenda AR, Afghah F, Cambou B (2018) "A secret key generation scheme for Internet of Things using ternary-states ReRAM-based physical unclonable functions," in *Proc. International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 1261–1266. https://doi.org/10.1109/IWCMC.2018.8450341
23. Mohammadinodoushan M, Cambou B, Philabaum C, Hely D, Booher DD (2019) "Implementation of password management system using ternary addressable PUF generator," in *Proc. IEEE International Conference on Sensing, Communication, and* Networking *(SECON)*, pp. 1–8. https://doi.org/10.1109/SAHCN.2019.8824792
24. Yamamoto D, Sakiyama K, Iwamoto M, Ohta K, Ochiai T, Takenaka M, Itoh K (2011) "Uniqueness enhancement of PUF responses based on the locations of random outputting RS latches," in *Proc. International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 390–406. https://doi.org/10.1007/978-3-642-23951-9_26

25. Lai Z, Lee K (2019) "Using unstable SRAM bits for physical unclonable function applications on off-the-shelf SRAM," in *Proc. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 41–44. https://doi.org/10.1109/APCCAS47518.2019.8953143
26. Maiti A, Schaumont P (2014) The impact of aging on a physical unclonable function. IEEE Trans Very Large Scale Integr VLSI Syst 22(9):1854–1864. https://doi.org/10.1109/TVLSI.2013.2279875
27. Saxena N, Voris J (2011) Data remanence effects on memory-based entropy collection for RFID systems. Int J Inf Secur 10(4):213–222. https://doi.org/10.1007/s10207-011-0139-0
28. Anagnostopoulos NA, Arul T, Rosenstihl M, Schaller A, Gabmeyer S, Katzenbeisser S (2018) "Low-temperature data remanence attacks against intrinsic SRAM PUFs," in *Proc. Euromicro Conference on Digital System Design (DSD)*, pp. 581–585. https://doi.org/10.1109/DSD.2018.00102
29. Tuan T, Strader T, Trimberger S (2007) "Analysis of data remanence in a 90nm FPGA," in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 93–96. https://doi.org/10.1109/CICC.2007.4405689

Zhi-Wei Lai received a B.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2016, and an M.S. degree from the National Cheng Kung University, Tainan, Taiwan, in 2020. He is currently working in Realtek Semiconductor Corporation. His research interests include hardware security and PUF design.

Po-Hua Huang received a B.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2020. He is currently pursuing his M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan. His current research interests include hardware security and PUF design.

Kuen-Jong Lee received his B.S., M.S., and Ph.D. degrees from National Taiwan University in Taiwan, University of Iowa in the USA, and University of Southern California in the USA, respectively. He is currently a Distinguished Professor in the National Cheng Kung University (NCKU) of Taiwan. His current research interests include hardware security, test compression, silicon debug, fault diagnosis, and in-field testing. Dr. Lee has received numerous awards, including the Outstanding Technology Contribution Award of National Applied Research Laboratories, the Outstanding Technical Achievement Award of IEEE Tainan Section, and several best paper awards from international/national conferences. He has served as the general/program chair/co-chair in the IEEE Asian Test Symposium, the VLSI Design, Automation and Test Symposium, and the International Test Conference in Asia. He also served as the Chair of the Taiwan IC Design Association and the Director of the SOC Research Technology Center of NCKU. He is an IEEE Fellow.