J Cryptogr Eng (2015) 5:187–199
DOI 10.1007/s13389-014-0091-9

REGULAR PAPER

A new method for enhancing variety and maintaining reliability of PUF responses and its evaluation on ASICs

Dai Yamamoto · Kazuo Sakiyama · Mitsugu Iwamoto · Kazuo Ohta · Masahiko Takenaka · Kouichi Itoh · Naoya Torii

Received: 2 October 2013 / Accepted: 11 November 2014 / Published online: 25 November 2014

Abstract Physically unclonable functions (PUFs) are expected to provide a breakthrough in anti-counterfeiting devices for secure ID generation and authentication, etc. Factory-manufactured PUFs are generally more secure if the number of outputs (the *variety* of responses) is larger (e.g., a 256-bit full-entropy response is more secure than a 128-bit response). In Yamamoto et al. (J Cryptogr Eng 3(4):197–211, 2013), we presented a latch-based PUF structure, which enhances the variety of responses by utilizing the location information of the RS (Reset-Set) latches outputting random numbers. We confirmed the effectiveness of this method using two kinds of different Xilinx FPGA chips: Spartan-3E and Spartan-6. In this paper, we propose a novel method of further enhancing the variety of responses while maintaining the *reliability* of responses, i.e., consistency over repeated measurements. The core idea in this method is to effectively utilize the information on the proportion of '1's in the random number sequence output by the RS latches. This proportion information is determined during the manufacturing process, making it relatively stable and reliable once PUFs are manufactured. We estimated the variety of responses generated by the PUFs to which the proposed method was applied. According to our experiment with 73 ASIC chips fabricated by a 0.18-μm CMOS process, latch-based PUFs with 256 RS latches can improve the variety of responses to as much as 2^{379}. This is much larger than 2^{220} for conventional methods, and 2^{314} for our previous method presented in Yamamoto et al., J Cryptogr Eng 3(4):197–211, 2013). The average error rate (reliability) of responses is only 0.064 when both temperature and voltage are changed to $-20 \sim 60$°C and 1.80 ± 0.15V, respectively. Our proposed PUF enhances the variety of responses dramatically while maintaining reliability.

Keywords PUF · RS (Reset-Set) Latch · Random number · Entropy · ASIC

D. Yamamoto (✉) · M. Takenaka · K. Itoh · N. Torii
Fujitsu Laboratories Ltd, 4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa 211-8588, Japan
e-mail: yamamoto.dai@jp.fujitsu.com; yamamoto.dai@uec.ac.jp

M. Takenaka
e-mail: ma@jp.fujitsu.com

K. Itoh
e-mail: ito.kouichi@jp.fujitsu.com

N. Torii
e-mail: torii.naoya@jp.fujitsu.com

D. Yamamoto · K. Sakiyama · M. Iwamoto · K. Ohta
The University of Electro-Communications, 1-5-1, Chofugaoka, Chofu, Tokyo 182-8585, Japan

K. Sakiyama
e-mail: sakiyama@uec.ac.jp

M. Iwamoto
e-mail: mitsugu@uec.ac.jp

K. Ohta
e-mail: kazuo.ohta@uec.ac.jp

1 Introduction

Big data solutions are increasingly being adopted in various social systems for including the fields of transportation, agriculture and energy [24]. These systems require embedded devices to collect useful information, such as mobile devices, physical sensors and smart-meters. Counterfeiting of these embedded devices is prevented by storing a secret key into a chip of the devices and using secure cryptographic protocols between the devices and the systems. However, the secret key can be revealed by de-packaging the embedded devices and analyzing digital chip design in the devices [32]. The possibility of counterfeiting the embedded devices

is serious because it greatly reduces the security of the system as a whole. Counterfeit chips are often found in consumer and even in military devices [9].

Recently, physically unclonable functions (PUFs) have come under focus as a solution to this issue [26]. When PUFs are mounted in individual digital chips, the circuit structure remains identical but the PUFs generate unique output IDs (responses) to the same input value (challenge) for each individual digital chip. This uniqueness is provided by random process variations such as wire/gate delay or memory characteristics [7,8], which occur in the manufacturing process of each digital chip. It is expected that PUFs will represent a breakthrough in technology for anti-counterfeiting devices through their application in the key generation being utilized by secure cryptographic protocols, making cloning impossible even when the chip design is revealed.

There are two categories of PUF on digital chips: delay-based PUFs utilizing wire/gate delay variations and memory-based PUFs utilizing memory characteristics [22]. Delay-based PUFs include Arbiter PUFs [18] and ring oscillator (RO) PUFs [31], etc. The strength of these PUFs is to have an exponential number of challenge–response pairs. Arbiter PUFs, however, have a vulnerability in that their responses can be predicted by a machine learning attack [27]. This attack can also be applied to RO PUFs in the same way. However, RO PUFs can be used securely through restriction to O(x) different challenges, where x is the number of blocks in the RO PUF. Memory-based PUFs include SRAM PUFs [10,13], latch-based PUFs (LPUFs) [29,30], Butterfly PUFs [17], Flip-flop PUFs [20] and Buskeeper PUFs [28], etc. What is common to these memory-based PUFs is that each bit of responses is generated based on various kinds of memory cells, i.e., SRAM cell, latch cell, flip-flop, butterfly (cross-coupled latches) and buskeeper cell (cross-coupled inverters). The memory-based PUFs can be regarded as secure storage elements, the stored value of which cannot be specified through their mask pattern images. However, all of the above PUFs are vulnerable to *invasive attacks*, namely, drilling a hole in the chip with a focused ion beam (FIB), etc., and then using a microprobe to read the value of the response outputted from the PUFs. Therefore, PUFs need to be used together with other techniques to prevent invasive attacks, e.g., active shielding techniques [5]. In contrast, Coating PUFs have been proposed to defend against such threats without the use of such techniques [33]. Their responses are based on the randomness in the local capacitance of the protective coating on the chip. FIBs and other invasive attacks change the capacitance, which makes it difficult for the attacker to read the original response.

This paper focuses on memory-based PUFs and especially LPUFs, multiple responses of which can be quickly obtained without switching the power off/on. This is an advantage of LPUFs over SRAM PUFs, which is one of the most feasible and secure memory-based PUFs. Our proposed method in this paper needs multiple responses for repeated measurements, hence we focus on LPUFs. LPUFs generate an N-bit response whose 1-bit value is generated based on the output from a RS (Reset-Set) latch. LPUFs have many RS latches outputting constant numbers (consecutive '1's or '0's), i.e., "fixed latches". The 1-bit value corresponding to a fixed latch is 1 or 0 at any time, so the part of response corresponding to fixed latches is highly reliable. In contrast, the LPUF has several RS latches that generate a mixture of '1's and '0's (random numbers), i.e., "random latches". A problem caused by these random latches is to reduce the reliability of the response, i.e., the consistency of the values of PUF responses for repeated measurements. This is because the values of the response corresponding to the random latches flip every time a response is generated. Hence, the random responses should be eliminated in order to generate responses with high reliability, i.e., "reliable responses". As a result, the response bits become lower as the number of random latches increases, which reduces the *variety* of responses in manufactured PUFs. For example, if LPUF with 256 RS latches has 100 random latches, the maximum variety of responses reduces from 2^{256} to 2^{156}. Hence the increase of random latches results in reducing the variety of reliable responses.

We have proposed an efficient method of using the random outputs to enhance the variety of responses [35]. This method regards the three types of output patterns from the RS latches (0's, 1's, and random numbers) as three values (00/11/10), respectively. The method can ideally increase the variety of responses from $\log_2(2^N)$ to $\log_2(3^N)$, where N is the number of implemented RS latches in an LPUF. We verified the effectiveness of the proposed method by using two types of different Xilinx FPGA chips: Spartan-3E and Spartan-6. However, this method does not make the maximum use of the entropy extracted from random latches in LPUFs. Further, we have evaluated the proposed method just on FPGA chips and not on ASIC chips, which are often used in the embedded devices. An evaluation on ASIC chips is very important because PUF performances are strongly affected by chip properties, which are quite different between FPGA and ASIC chips.

1.1 Our contributions

This paper makes three contributions: (1) we propose a new method of increasing the variety of responses over that in [35], while maintaining their reliability; (2) we manufacture 73 LPUF ASIC chips with 256 RS latches, and evaluate the effectiveness of our method using the chips; (3) we evaluate the reliability of responses against both temperature and voltage fluctuations.

In detail: (1) our proposed method utilizes the proportion of '1's in the random numbers outputted from each random

latch. This method enables us to distinguish each random latch, while the previous method in [35] regards all random latches as the same. Consequently, our proposed method extracts more entropy from random latches than that in [35], and enhances the variety of responses. However, it is not desirable to use the value of the proportion of '1's without any consideration. This is because this information is easily affected by environmental conditions such as temperature and voltage, which causes the problem of reducing the reliability of responses. To avoid this problem, our proposed method categorizes the random latches not according to the value but the range of the proportion of '1's. The proportion of '1's falls within a particular range of values even when temperature and supplied voltage fluctuate. As a result, the number of random latches in each category is expected to be relatively reliable, so can be used to enhance the variety of responses while maintaining reliability. Here, the most important parameter is the number of categories K, because a large K improves the variety but worsens the reliability.

(2) In order to evaluate the effectiveness of the proposed method and determine the appropriate value of K, we fabricate 73 LPUFs with 256 RS latches on Fujitsu 0.18-μm CMOS technology. According to our experiments using the 73 chips, the varieties of responses are 2^{220}, 2^{314} and 2^{379} when $K = 2, 3, 16$, respectively. The proposed method in this paper (i.e., $K = 16$) and the previous method presented in [35] (i.e., $K = 3$) generate 1.72 and 1.42 times a larger variety of responses than the conventional method of eliminating random latches (i.e., $K = 2$), respectively.

(3) We confirm that all LPUFs in the 73 chips provide ideal performance as PUFs even when $K = 16$ and both temperature and voltage change to $-20\,^{\circ}\mathrm{C} \sim 60\,^{\circ}\mathrm{C}$ and 1.80 ± 0.15 V, respectively. The maximum error rate of responses is approximately 0.096, which is less than the 0.15 assumed in [4] for reliable responses based on Error Correcting Code (ECC) with a reasonable size of redundant data. Our proposed PUF dramatically enhances the variety of responses while maintaining reliability, which is very practical and useful.

1.2 Organization of the paper

The rest of the paper is organized as follows. In Sect. 2, we discuss previous related work. Section 3 gives an outline of an RS latch and an LPUF. Section 4 proposes our method of generating a larger variety of responses. Section 5 describes implementing the LPUFs on ASIC chips and shows our experimental evaluation system. In Sect. 6, we evaluate the effectiveness of our proposed method using the chips. In addition, we evaluate various PUF performances such as reliability with respect to changing environmental temperature and power supply voltage. Finally, in Sect. 7 we summarize our work.

2 Related work

SRAM PUFs [10,13] and LPUFs [29,30] have been proposed to extract one-bit entropy from a one-bit memory cell, e.g., a SRAM cell and an RS latch, respectively. In contrast, the motivation of our proposed method is to extract multi-bit entropy from an RS latch. Similar concepts are presented in previous work.

The MECCA PUF [16] is basically based on an SRAM PUF, but has a mechanism for changing the word line duty cycle of SRAM cells (i.e., challenge). The value of SRAM cells (i.e., response) is influenced by the duty cycle duration. The SRAM PUF using a data retention voltage (DRV) [14] also provides a more informative non-binary identifier from each cell. This PUF utilizes the DRVs of SRAM cells by repeatedly lowering the SRAM supply voltage and observing the highest voltage at which each cell fails. The highest voltage resulting in a fail is unique, and so used to generate a multi-bit response. These PUFs need a hardware modification to change the duty cycle duration or the SRAM supply voltage. In [21], a method for error correction using soft-decision information has been proposed to realize a smaller entropy loss in responses. This method can be used as an efficient ECC mechanism for conventional PUFs and even for PUFs based on our proposed method.

The theoretical upper bound of extractable entropy from memory-based PUFs has been evaluated based on mutual information [2]. While the paper [2] discussed a similar topic to this paper in terms of the upper bound of extractable entropy, our paper is significantly different in that we show a concrete method and evaluate the performance of the PUF using a custom-designed ASIC implementation of LPUFs. Moreover, we have evaluated the combined effect of environmental parameters such as supplied voltage and temperature.

In [1], the idea of excluding *dark bits*, corresponding to the responses from random latches, has been proposed to improve reliability of responses. In contrast, the above-mentioned approaches, including our proposed method, utilize the dark bits to extract more entropy for PUF responses.

3 Conventional methods

3.1 RS latch

The behavior of an RS latch consisting of two cross-coupled NAND gates is shown in Fig. 1. The NAND-based RS latch is one component of our LPUF. An RS latch can be configured by NAND, NOR, and other types of gate. This difference, however, does not influence the performance of LPUFs. A general RS latch has two separate input signals connected to an upper NAND gate and a lower one. Note that the RS latch for the LPUF in Fig. 1 has a single combined input A. The RS

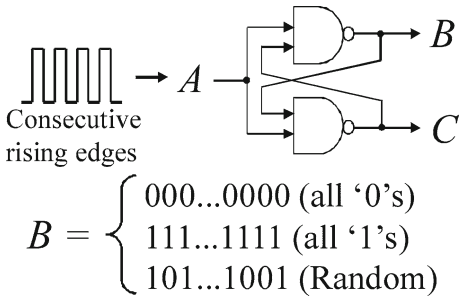

Fig. 1 NAND-based RS latch

latch is in a stable state with outputs $(B, C) = (1, 1)$ when input $A = 0$, while it temporarily enters a metastable state right after input A changes from 0 to 1 (= rising edge). Right after this, it enters into one of two stable states: its outputs are $(B, C) = (1, 0)$ or $(B, C) = (0, 1)$. The probability of transition to either of these states is equal in ideal RS latches. Actually, however, most RS latches have a high probability of entering one specific state. This is caused by a slight difference in the drive capabilities of the two NAND gates or the wire length of the cross-coupled part. When a signal with consecutive rising edges, i.e., a clock signal, is applied to input A, output B sampled in the stable state after the rising edges falls into one of three patterns: all '0's, all '1's, or a mixture of '0's and '1's (= random numbers). The ratio of '1's in the random numbers depends on each RS latch, which is important and useful in this paper.

3.2 Latch-based PUF

This section explains the mechanism of an LPUF, shown in Fig. 2. An LPUF consists of N parallel-implemented RS latches, which generate an N-bit response:

$$RES = R_{N-1} \parallel R_{N-2} \parallel \cdots \parallel R_i \parallel \cdots \parallel R_1 \parallel R_0,$$

where R_i is a unique value outputted from an RS latch i $(0 \leq i \leq N-1)$ and the operation $\parallel$ means a concatenation of two variables. Note that the more significant bits of the response correspond to the outputs of RS latches with larger latch numbers, in order to simplify discussion in this paper. The LPUF in Fig. 2 has some random latches such as LATCH_2 and LATCH_{N-2}. These random latches cause a problem insomuch that the reliability of the response RES is reduced since their outputs are unstable random numbers. There are two widely known conventional approaches to solve this problem.

The first approach does not use random latches for the generation of responses. This approach maintains the reliability of responses, but reduces the bit length of responses, i.e., the variety of responses, as the number of random latches increases. Thus it is necessary to implement extra RS latches in an LPUF in accordance with the number of random latches.

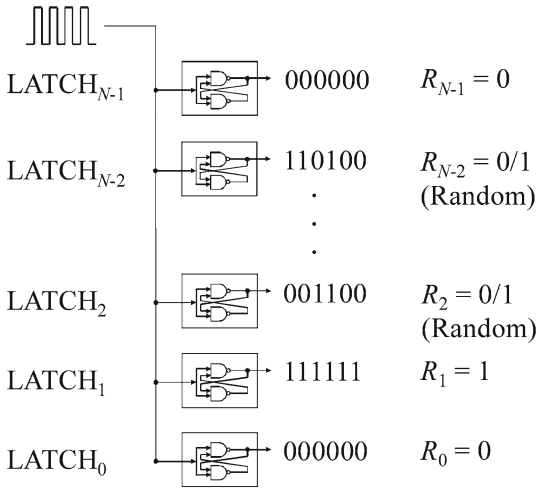

Fig. 2 Latch-based PUF

However, such a solution is not suitable for embedded systems with limited hardware resources. This is because it is necessary to make the area size of RS latches and peripheral circuits as small as possible in LPUFs in embedded devices.

The second approach uses ECCs to correct the non-reliable responses caused by the random latches. This approach requires larger redundant data for response correction as the number of random latches increases. The large redundant data increases the ROM code size, which is not suitable for embedded devices. In addition, an LPUF with n RS latches naturally extracts $k(< n)$ bits of entropy even if $[n, k, d]$-code is used as an ECC.

From the above discussion, the first approach is not desirable when extracting more entropy from PUFs. The second approach is essential for memory-based PUFs used for secure key storage, although it is not sufficient to use this approach alone. In Sect. 4, we propose a method for extracting more entropy from PUFs by utilizing these unwanted random latches. The proposed method dramatically improves the variety of responses and maintains the reliability of responses.

4 Proposed method

The conventional LPUF in Fig. 2 generates responses based on the output values themselves ('0's or '1's). We introduced the method in [35] (i.e., the previous method), which extracts the entropy of locations of random latches, rather than eliminating them. If LPUF with N RS latches has T random latches, the number of locations of random latches equals ${}_N\mathrm{C}_T$. Here, ${}_N\mathrm{C}_M$ is defined as the number of combinations of N elements taken M at a time. The PUF based on the previous method utilizes the entropy derived from the locations of random latches. In this paper, we propose a novel method which extracts more entropy from random latches in order to

increase the variety of responses. Concretely, our proposed method uses the information of the proportion of '1's ('0's) in the random numbers outputted from each RS latch. The reason why we focus on the proportion of '1's is that this information is different for each RS latch, so is expected to include high entropy. LPUFs using this proposed method are expected to generate a larger variety of responses than the LPUFs using the previous method.

However, the value of the proportion of '1's is affected by environmental conditions such as temperature and voltage. Hence we propose a simple and efficient mechanism to keep the responses as reliable as possible and enhance the variety of responses. This mechanism consists of two processes: a *dividing* process and a *labeling* process.

First, the *dividing* process must determine an important factor K: the number of output patterns resulting from RS latches. The previous method distinguishes just three (i.e., $K = 3$) types of output patterns from the RS latches ('0's, '1's and random numbers). In contrast, the LPUFs using the proposed method distinguish $K(> 3)$ types of RS latches ($T_0 \sim T_{K-1}$). Consequently, LATCH_i is defined as belonging to type $T_k(0 \le k \le K-1)$ as follows:

$$\begin{cases} T_{K-1} & \text{if } X_i = 1, \\ T_{K-2} & \text{if } \frac{K-3}{K-2} < X_i < 1, \\ T_k(2 \le k \le K-3) & \text{if } \frac{k-1}{K-2} < X_i \le \frac{k}{K-2}, \\ T_1 & \text{if } 0 < X_i \le \frac{1}{K-2}, \\ T_0 & \text{if } X_i = 0, \end{cases}$$

where X_i is the percentage of '1's within a certain amount of random numbers. Parameter K is very significant for the dividing process because it has a great impact on the variety and reliability of responses. A larger value of K increases the variety of responses, but is anticipated to make reliability worse. This is because the smaller range of X_i (i.e., the larger value of K) we define, the more RS latches are distinguished into different types before/after temperature or voltage changes. This leads to large error bits of responses. Therefore, a large size of redundant data is necessary for ECC. Further, a larger value of K causes a larger bit length of responses, which increases the area size of peripheral circuits (e.g., flip-flops for storing responses). Hence we should determine the appropriate value of K through experiments using LPUF implementations.

Next, the *labeling* process determines unique values L_k for each type of T_k, where $0 \le k \le K-1$. In the previous method corresponding to $K = 3$, the unique values were just simply labeled as 00/11/10 according to the RS latches outputting '0's, '1's and random numbers, respectively. When the proposed method is used (i.e., $K > 3$), the unique values L_k can be labeled in various ways. Figure 3 shows a proposed method of labeling the unique value of L_k for each $T_k(0 \le k \le K-1)$ in various K settings ($3 \le K \le 16$). We will verify if this labeling process is suitable for LPUFs based on

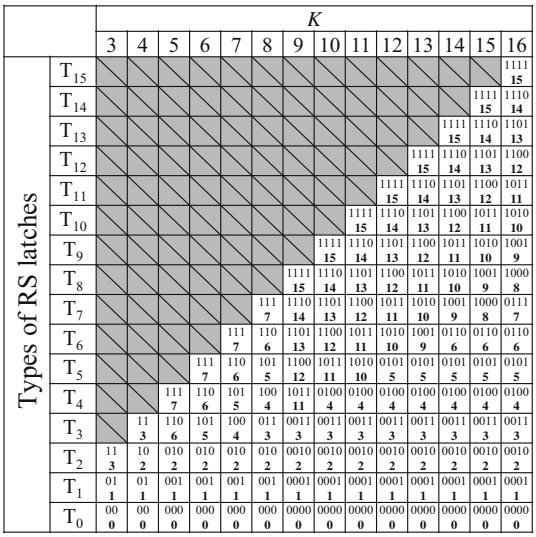

Fig. 3 Labeling method for unique value L_k corresponding to type T_k

the proposed method. This labeling is principally based on the *binary represents*, where the unique value corresponding to T_k is simply labeled as $k(0 \le k \le K-1)$. The reason why we use binary represents is that this simplicity results in almost no additional increase in the design cost to decide the labeling way. The naive binary represents are, however, not suitable for the labeling of unique values because PUF performances such as uniqueness and uniformity are not close to ideal ($\approx$ 0.50).

If we use the naive binary represents, the Hamming weight of a unique value for T_{K-1} is not $\lceil \log_2 K \rceil$ except when $\lceil \log_2 K \rceil = \log_2 K$ (i.e., $K = 4, 8, 16$). When $K = 6$, for example, the unique values L_0 and $L_{K-1}(= L_5)$ for T_0 and $T_{K-1}(= T_5)$ are '0b000' ($k = 0$) and '0b101' ($k = 5$), respectively. As the number of these two types of RS latches (i.e., fixed latches) is almost the same in all implemented RS latches, the proportion of '1's in the response bits RES (i.e., uniformity) is approximately 0.33 ($\approx 2/6$), which is smaller than ideal 0.5. Hence the unique value for T_{K-1} should be $2^{\lceil \log_2 K \rceil} - 1$ (e.g., '0b111' when $K = 6$). The method of labeling described in Fig. 3 satisfies the above-mentioned conditions by simply eliminating the middle range of binary represents.

The reason why we regard the labeling process as important is that, if the labeling method is not appropriate, the entropy derived from PUFs becomes lower or the reliability of PUF responses becomes worse, which increases ECC costs. Further, there are various methods of labeling, and the Gray code seems to be an effective labeling method. The Gray code realizes high tolerance to noise, i.e., high reliability of PUF responses. However, the uniqueness becomes lower due to the same reason as the naive binary represents,

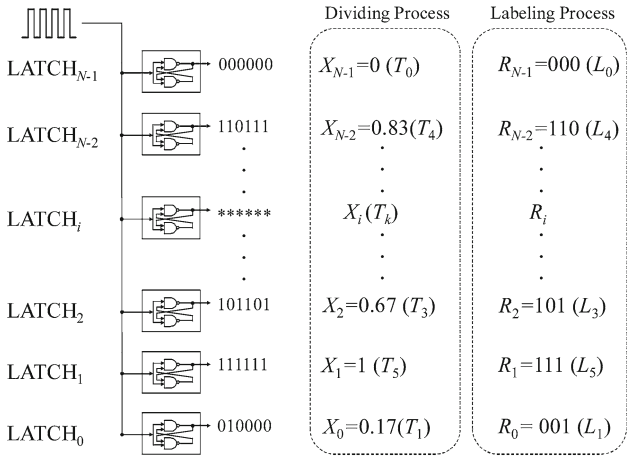

Fig. 4 Example of the proposed LPUF with the dividing and labeling processes ($K = 6$)

as mentioned before. This is why we use the labeling method as shown in Fig. 3.

We explain the dividing and labeling processes for the specific example of when $K = 6$ in Fig. 4, as follows. If LATCH$_0$, LATCH$_1$, LATCH$_2$ and LATCH$_{N-2}$ include 175, 1,024, 686 and 850 '1's in a data stream of 1,024 bits, X_0, X_1, X_2 and X_{N-2} are approximately 0.17, 1, 0.67 and 0.83, respectively. LATCH$_{N-1}$ has no '1's in the data stream, so X_{N-1} is 0. The proposed method for $K = 6$ classifies RS latches into six types according to the range of X_i: (T_0) $X_i = 0$, (T_1) $0 < X_i \leq 0.25$, (T_2) $0.25 < X_i \leq 0.50$, (T_3) $0.50 < X_i \leq 0.75$, (T_4) $0.75 < X_i < 1$ and (T_5) $X_i = 1$, respectively. According to the labeling method in Fig. 3 for $K = 6$, L_0, L_1, L_2, L_3, L_4 and L_5 are '0b000', '0b001', '0b010', '0b101', '0b110' and '0b111', respectively. R_i, the unique value for LATCH$_i$ ($0 \leq i \leq N - 1$), is shown in Fig. 4. Finally, our LPUF generates a $3N$-bit response.

Next we discuss the way of implementing the proposed method, consisting of the dividing and labeling processes. The proposed method has to be implemented on a co-processor alone if our proposed LPUF is to be implemented as a pure-ASIC design. This dedicated circuit on ASICs, however, causes additional overhead in the circuit area for the PUF implementation. Note that embedded systems consist not only of a co-processor with a PUF circuit, but also a microprocessor, ROM, RAM, etc. We therefore assume that a software approach enables us to realize both processes. Output data from RS latches are stored in the RAM and processed by the microprocessor. This approach does not need additional hardware resources, but needs a slight increase in ROM code size. However, this software approach might lead to serious security threats such as response eavesdropping on the microprocessor or the RAM. Even the hardware approach, implementing the proposed method on the co-processor, might face the same threat due to physical analysis, e.g., side channel attacks [12]. Concrete ways of implementing the proposed method and their security evaluation are very important and need to be discussed in detail, and this is included in future work.

Section 5 implements LPUFs with multiple RS latches into ASIC chips. In Sect. 6, we discuss the appropriate value of K through experimental results using the ASIC chips. We also evaluate the performances of LPUFs such as the variety and reliability of responses by actually generating responses according to the proposed method.

5 Implementation and setup

5.1 ASIC implementation

We fabricated LPUFs on 73 ASIC chips using the Fujitsu 0.18-μm CMOS process (CS86 technology [19]) in order to evaluate LPUFs with the proposed method. An RS latch was custom-designed as a hard-macro in the process of designing an IC mask layout. The purpose of this design is to equalize wire lengths between the cross-coupled NAND gates shown in Fig. 1. This enables the RS latch to enter a metastable state more readily and improve the probability of the RS latch outputting random numbers. We implemented an LPUF with 256 RS latches on a chip by an automatic placement of the 256 instances of the hard-macro. RS latches in our chips do not include flip flops (FFs) in front of the two NAND gates [11,35] in order to reduce circuit area size. The 73 chips were embedded in DIP-28 non-sealed packages. Note that in fact we fabricated 80 ASIC chips, of which 73 chips work correctly. The other seven chips have problems concerning the bonding wires, which are disconnected or short-circuited as a result of the non-sealed packages for other studies (e.g., side channel analysis). The rated supply voltage range of the chips is 1.80 ± 0.15 V.

5.2 Measurement setup

We setup the experimental evaluation system shown in Fig. 5. This system consists of two boards: a custom-made expansion board with six sockets for fabricated chips, and a Spartan-3E (SP3E) starter kit board [34] with a Xilinx SP3E FPGA (XC3S500E-4FG320C). The expansion board can evaluate six fabricated chips at the same measurement time. A complex programmable logic device (CPLD) was implemented on the expansion board, allowing us to select one target chip out of the six chips. The core voltage of the chips can be changed by 0.01 V using an external stabilized power supply. The starter kit board possesses peripheral circuits for data acquisition processes such as a Digital Clock Manager (DCM), a block RAM, an RS232C module and a SD write module. A 50-MHz clock signal generated by an oscillator on the SP3E board was applied to the DCM primitive yield-

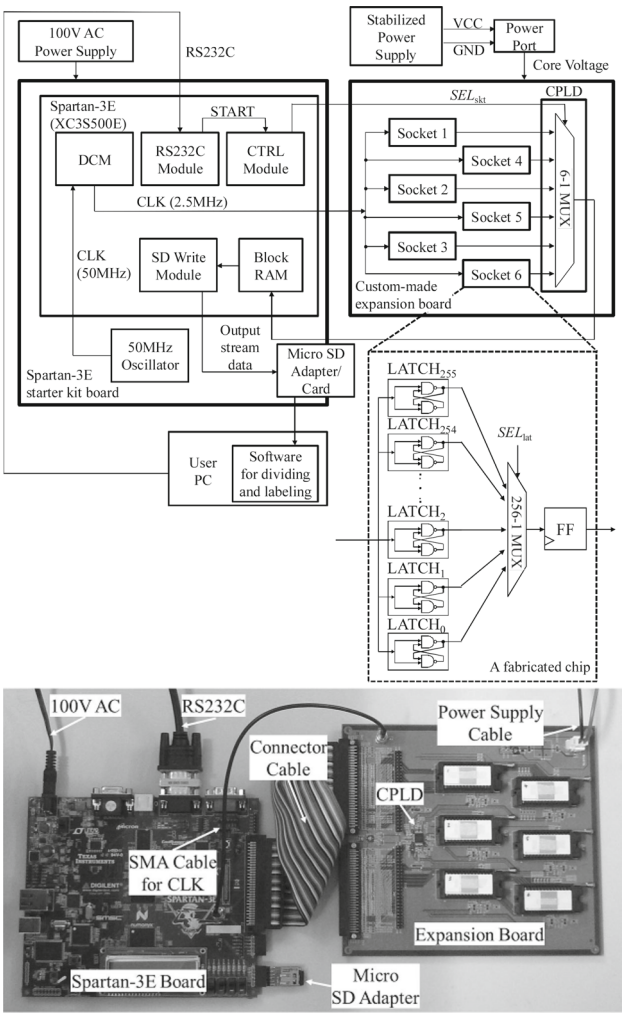

Fig. 5 Experimental evaluation system

ing a 2.5-MHz clock signal that was applied to the ASIC chips. The two boards were connected with user I/O interfaces by a connector cable. The clock signal was provided separately through a SMA cable and port from the SP3E to the expansion board in order to prevent signal degradation. A micro-SD adapter and card were also connected to the SP3E board to store output data from the chips.

The data acquisition process is as follows. When the RS232C module receives a start command from a user PC, the module sends a start signal to the control (CTRL) module. The CTRL module sends a signal $\mathrm{SEL_{skt}}$ to a 6-1 multiplexor (MUX) in order to select one socket. It also sends a signal $\mathrm{SEL_{lat}}$ to a 256-1 MUX in the chips to select a target RS latch. First, $\mathrm{SEL_{skt}}$ and $\mathrm{SEL_{lat}}$ are set to one and zero, respectively. This means that $\mathrm{LATCH_0}$ in the chip on socket 1 is selected for measurement. The CTRL module measures twenty-one 1,024-bit (= 21,504-bit) output streams from $\mathrm{LATCH_0}$ in our evaluation. $\mathrm{SEL_{lat}}$ is incremented by 1 from 0 to 255 in order to obtain output streams from all 256 RS latches. After obtaining all data from the chip, $\mathrm{SEL_{skt}}$ is incremented by 1 from 2 to 6. During this process we evaluate 73 LPUFs implemented on 73 ASIC chips. The output stream data is stored in the block RAM through an FF, sent to the SD write module, and written into the micro-SD card. The PC can obtain the data via the micro-SD card. In our evaluation, software on the PC provides the dividing and labeling processes rather than this being done on the chips. We consider that the technique for the processes does not influence PUF performance because this performance depends just on the output of the RS latches itself.

Next, Sect. 6.1 evaluates the appropriate value of K in the proposed method in consideration of the tradeoff between variety and reliability of LPUF responses. Later, in Sect. 6.2 we evaluate the following performance-related metrics (reliability, uniqueness, uniformity and bit-aliasing) [23] of LPUFs when changing both voltage and temperature. We chose the standard voltage of 1.80 V and the maximum allowed ASIC chip rated voltages of 1.65 and 1.95 V. We also chose the room temperature of 27°C and set the minimum and maximum allowed temperatures of −20° and 60°C in a thermostatic chamber. Only the expansion board was put in the thermostatic chamber, while the other experimental items were outside it throughout the experiment. The temperature change does not impact the peripheral circuits and does not cause data garbling, which enhances the confidence of our experimental results.

6 Performance evaluation

6.1 Evaluation of the proposed method: appropriate value of K

This section evaluates the appropriate value of K according to two metrics: the response error rate (related to reliability) and the variety of responses.

Figure 6 shows the response error rate in each K at each operating condition c as shown in the upper part of Fig. 6. The response error rate is defined as follows with the notation summarized in Table 1. We extract a reference response (RES_i^K) from the ith ASIC chip ($1 \le i \le w$, $w = 73$ in this paper) in normal operating conditions (room temperature of 27°C and a standard supply voltage of 1.80 V) when setting K ($3 \le K \le 16$). Similarly, the response ($\mathrm{RES}_i'^{K,c}$) is extracted at an operating condition c. Then, m samples ($m = 20$ in this paper) of $\mathrm{RES}_i'^{K,c}$ are collected. Here, $\mathrm{RES}_{i,t}'^{K,c}$ is the tth ($1 \le t \le m$) sample of $\mathrm{RES}_i'^{K,c}$. The average of error bits for the parameter K and the operating condition c ($\mathrm{AEB}^{K,c}$) is defined as follows:

$$\mathrm{AEB}^{K,c} = \frac{1}{w \cdot m} \sum_{i=1}^{w} \sum_{t=1}^{m} \mathrm{HD}_{i,t}^{K,c}, \quad (1)$$

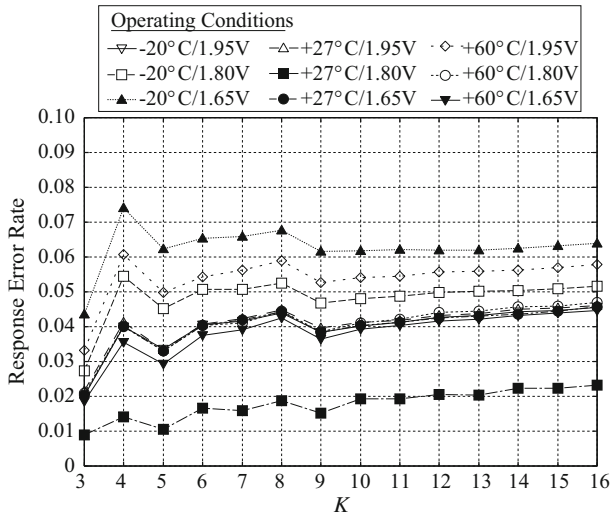

Fig. 6 Response error rate $\mathrm{RER}^{K,c}$ for K and c. The decrease between $K = 4$ and 5 is due to the value of $\mathrm{RES}^K_{\mathrm{bit}}$ increasing from 512 to 768. The decrease between $K = 8$ and 9 is for the same reason

where $\mathrm{HD}^{K,c}_{i,t} = \mathrm{HD}\{RES^K_i, RES'^{K,c}_{i,t}\}$, and $\mathrm{HD}\{x, y\}$ is the Hamming distance between variable x and y. Our next interest is the response error rate ($\mathrm{RER}^{K,c}$), which is defined as follows:

$$\mathrm{RER}^{K,c} = \mathrm{AEB}^{K,c}/\mathrm{RES}^K_{\mathrm{bit}}, \tag{2}$$

where $\mathrm{RES}^K_{\mathrm{bit}}$ is the number of response bits obtained from 256 RS latches for K, this being calculated from $\lceil \log_2 K \rceil \cdot 256$.

Basically, from Fig. 6, a larger value of K gives a slightly larger response error rate (i.e., a lower reliability). An unexpected positive result is that the response error rate does not increase dramatically as the value of K increases. This is because the parameter K only has an impact on random latches ($T_1 \sim T_{K-2}$) and not on fixed latches (T_0 and T_{K-1}). The average number of random latches is just 36 of the total implemented 256 latches in a fabricated chip.

It is desirable that $\mathrm{RER}^{K,c}$ be less than 0.15 assumed in [4] for a reasonable size of redundant data in ECC. $\mathrm{RER}^{K,c}$ is less than 0.15 for all values of K from 3 to 16 according to Fig. 6, which is the reason why the value of 16 is appropriate for K in our LPUF in terms of the reliability of responses. However, some LPUFs implemented on different types of CMOS process might include many random latches. In that case, excessively large values of K should not be used since $\mathrm{RER}^{K,c}$ is anticipated to increase, which leads to large costs for ECC.

Figure 7 shows the entropy of responses with respect to $K = 3, \ldots, 16$, which contains three graphs: (1) the ideal upper bound on Shannon entropy of responses, (2) the experimental Shannon entropy and (3) the entropy based on the mutual information of responses. These graphs are experimentally calculated based on responses derived from the 73 fabricated LPUFs.

Table 1 Notation summary for this paper

Notation	Definition
RES	A response generated from an LPUF
i	Chip number ($1 \le i \le 73$)
K	Number of output patterns from RS latches (determined in the proposed method)
RES^K_i	A response from ith chip for K
c	Operating condition
$\mathrm{RES}'^{K,c}_i$	RES^K_i generated under c
t	Number of measurement ($1 \le t \le m$, $m = 20$ in this work)
$\mathrm{RES}'^{K,c}_{i,t}$	tth measurement of $\mathrm{RES}'^{K,c}_i$
AEB	Average of error bits defined in Eq. (1)
AEB^K_i	AEB for K under c
$\mathrm{RES}^K_{\mathrm{bit}}$	Number of bits in RES for K ($= \lceil \log_2 K \rceil \cdot 256$)
$\mathrm{RER}^{K,c}$	Response error rate ($= \mathrm{AEB}^{K,c}/\mathrm{RES}^K_{\mathrm{bit}}$)

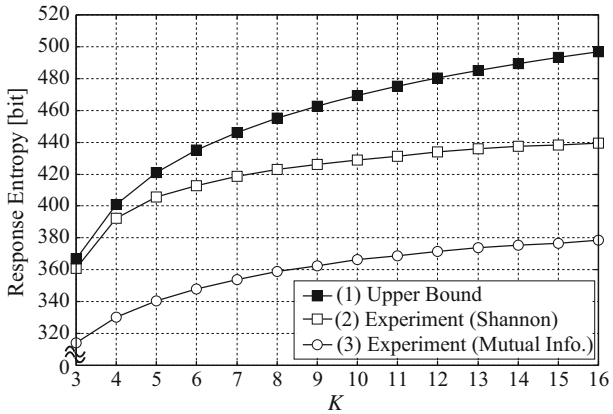

Fig. 7 Estimations of entropy of responses: (*1*) the ideal upper bound on Shannon entropy of responses, (*2*) the experimental Shannon entropy, (*3*) the entropy based on the mutual information of responses

First, we explain how the graphs (1) and (2) are constructed. Let the ratios of the RS latches numbered as LATCH_i and classified as types ($T_0 \sim T_{K-1}$) be $P_i(T_0)$, …, $P_i(T_{K-1})$, respectively. Assuming that each RS latch is independent, the Shannon entropy derived from LATCH_0 to LATCH_{255} are given as

$$\sum_{i=0}^{n-1} E_i, \tag{3}$$

where $n = 256$ and E_i, the Shannon entropy derived from LATCH_i, is defined as

$$E_i = -\sum_{j=0}^{K-1} P_i(T_j) \log_2 P_i(T_j).$$

The graphs (1) and (2) are given by Eq. (3). The graph (1) assumes that the number and ratio of random latches are 36 and 0.14 ($\approx 36/256$) strictly on every chip, respectively. This value of 36 comes from the average number of random latches in our LPUFs on ASIC chips. This ideal upper bound is also based on the following two requirements: (1) the numbers of random latches belonging to all types ($T_1 \sim T_{K-2}$) are equally $36/(K-2)$, so $P_i(T_1) = \ldots = P_i(T_{K-2}) = \{36/(K-2)\}/256$, (2) the numbers of fixed latches belonging to T_0 and T_{K-1} are equally $(256-36)/2 = 110$, so $P_i(T_0) = P_i(T_{K-1}) = 110/256$.

Next, we explain the graphs (3) as follows. The graph (2) assumes that the responses are completely reliable; they are identical in both enrollment and reconstruction phase. Actually, however, the responses have some error bits (i.e., noise) due to environmental fluctuations, therefore some bits have to be sacrificed as redundancy bits for error correction. In order to estimate the entropy bits that actually survive the noise, we calculate $\mathbf{I}(X;Y)$: the mutual information between responses obtained in enrollment (27°C/1.80 V, normal condition), X, and in reconstruction (−20°C/1.65 V, worst condition, see Fig. 8), Y. This estimation is based on the method introduced in [15], whose core idea is presented in [25].

From Fig. 7, K increases with the difference between the experimental results and the upper bound. This means that a larger value of K cannot necessarily result in a much larger variety of responses. For example, the experimental Shannon entropy and the entropy based on the mutual information increase approximately 62 and 45 bits from $K = 3$ to 8, in contrast, it increases only 16 and 20 bits from $K = 8$ to 16, respectively. This is because the aforementioned requirement (1) is not satisfied, that is, there are a lot of random latches outputting random numbers whose proportion of '1's is very low or high, such as T_1 or T_{K-2}. Our LPUFs can generate responses with maximum variety by setting $K = 16$ since the response error bit is relatively small with a larger value of K.

The entropy of responses based on the mutual information is estimated to be 220 bits when LPUFs using 256 RS latches generate responses eliminating 36 random latches. The LPUFs based on the previous method, i.e., the proposed method for $K = 3$, generate responses with approximately 314 bits of entropy, which is about 1.42 times as large as 220 bits. In contrast, the LPUFs using the proposed method for $K = 16$ generate responses with approximately 379 bits of entropy, which is about 1.72 times as large as 220 bits and about 1.21 times as large as 314 bits. Our proposed method therefore dramatically increases the Shannon entropy of responses, i.e., the variety of responses.

Note that appropriate values of K depend on the methods of implementing RS latches and the process technologies of ASIC chips. Hence the values of K should be carefully decided in consideration of the tradeoff between reliability and variety of responses.

6.2 Evaluation of basic PUF performance

This section evaluates our LPUFs in terms of the basic performance defined in [23], i.e., reliability, uniqueness, uniformity and bit-aliasing at $K = 3$, 8, and 16. The LPUF based on the proposed method gives the results for reliability and the other three metrics shown in Figs. 8 and 9, respectively. Our LPUF with 256 RS latches generates a $\lceil \log_2 K \rceil \cdot 256$-bit response.

In the reliability evaluation, the reliability of responses is evaluated under the condition that the supply voltage and environmental temperature are changed within the rated voltage range of the ASIC chips (1.65, 1.80, 1.95 V) and the allowed temperature range of the thermostatic chamber (−20°, 27°, 60°C). Different from Fig. 6, this reliability evaluation focuses not on the average but on the histogram of response error rates when $K = 3$, 8, and 16. In this evaluation, one response (i.e., RES_i^K) is generated as a reference at normal operating conditions (27°C and 1.80 V), and the remaining 20 responses (i.e., $\mathrm{RES}_{i,t}^{\prime K,c}$) are generated for analysis at each condition c at $K = 3$, 8, and 16 from ith ASIC chip. Figure 8 shows histograms of normalized Hamming distances (NHD) between the reference response and each repeated one (i.e., 20 × 73(chips) = 1,460 elements). For chip i and sample t, each data element of the reliability histogram is calculated as follows:

$$\mathrm{NHD}_{i,t}^{K,c} = \frac{\mathrm{HD}\{RES_i^K, \mathrm{RES}_{i,t}^{\prime K,c}\}}{\lceil \log_2 K \rceil \cdot 256}.$$

Our LPUFs are the most susceptible to conditions under the low temperature of −20°C and the low supply voltage of 1.65 V. Even under this condition and $K = 16$, the average and maximum of NHD (i.e., error rate) are approximately 0.064 and 0.096, respectively. These error rates are much less than the 0.15 assumed in [4] for reliable responses based on a Fuzzy Extractor [6] with a reasonable size of redundant data. Hence our result shows that the LPUFs implemented on ASIC chips with our proposed method yields highly reliable responses even under environmental fluctuations.

The uniqueness evaluation generates a total of 73 responses using all 73 ASIC chips (one response per chip). Figure 9(I-a), (I-b) and (I-c) show histograms of NHD between every combination of two responses, i.e., ${}_{73}C_2 = 2,628$ combinations at $K = 3, 8, 16$, respectively. The NHDs between the responses of two arbitrary LPUFs at $K = 3$, 8 and 16 are approximately 0.489, 0.497 and 0.497, respectively. The ideal NHD at $K = 8$, 16 is 0.5, so our LPUF gives responses with a high level of uniqueness. In contrast, the ideal NHD at $K = 3$ is around 0.444 because '10' is not used for a unique value (see Fig. 3). This is why the NHD at $K = 3$ is a little smaller than the others. The NHD is, however, a little larger than the ideal 0.444 because the average number of random latches is only 36 in our LPUFs, which is

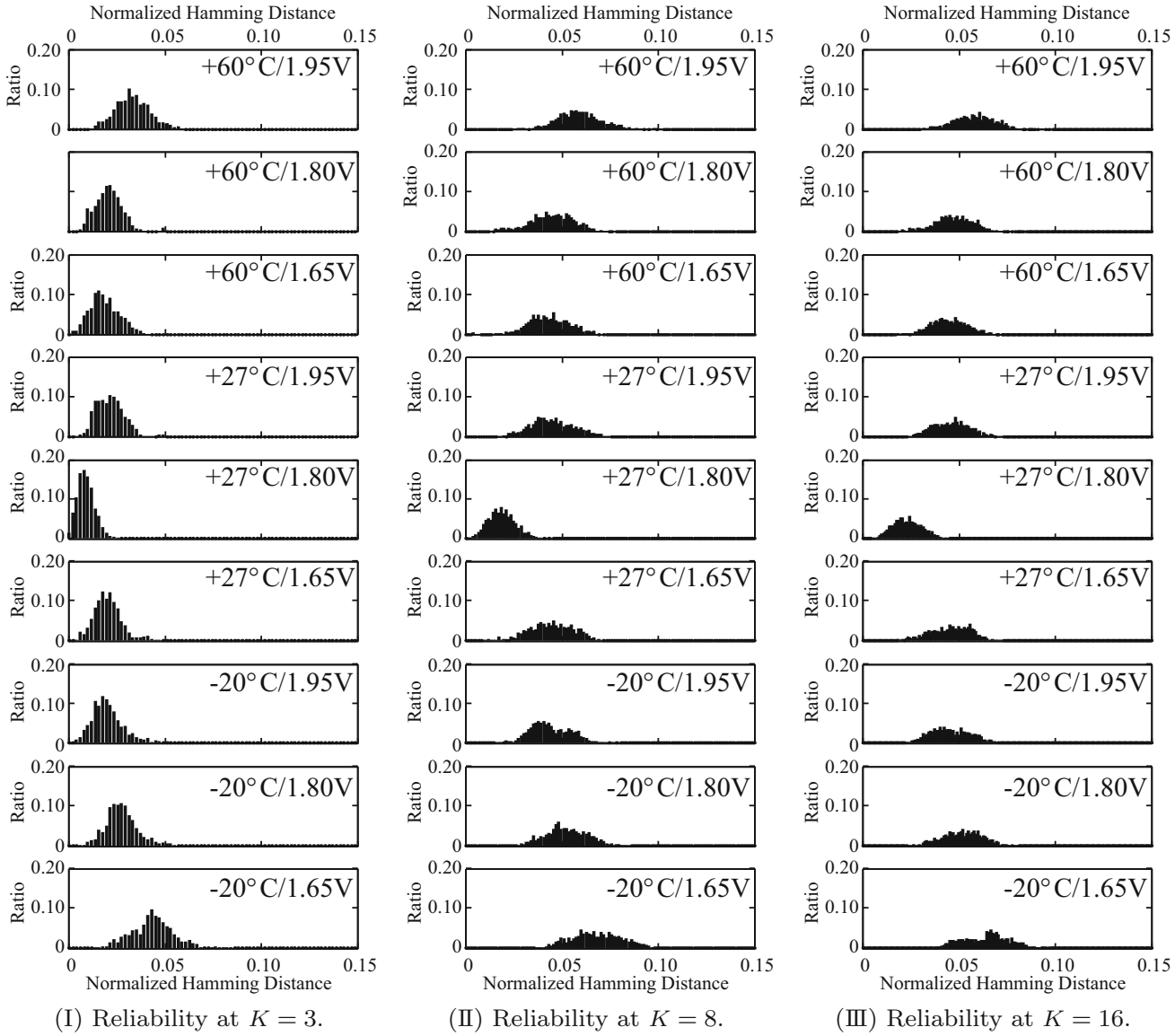

Fig. 8 Reliability at various conditions

smaller than 85 ($\approx 256/3$). Consequently, most of the 2-bit unique values are '00' or '11', so the NHD approaches 0.5.

The uniformity evaluation also generates 73 responses using all 73 ASIC chips, i.e., 73 data elements. Figure 9(II-a), (II-b) and (II-c) show the uniformity: how uniform the proportion of '0's and '1's is in the response bits of an LPUF at $K = 3, 8$, and 16, respectively. For our LPUFs on ASIC chips, the averages of uniformity at $K = 3$, 8 and 16 are approximately 0.486, 0.485 and 0.484, respectively. Since the ideal uniformity is 0.5 for truly random PUF responses [23], our LPUFs satisfy the requirement for uniformity. However, we can see two isolated data elements around 0.34 and 0.67 in three uniformity figures. This is because two particular chips have more one-typed fix latches (T_0 or T_{K-1}) than the other-typed ones (T_{K-1} or T_0).

The bit-aliasing evaluation measures the difference in the proportion of '0's and '1's in the 73 R_i's extracted, respectively from the 73 LPUFs corresponding to $\text{LATCH}_i (0 \leq i \leq 255)$, i.e., 256 data elements. Figure 9(III-a), (III-b) and (III-c) show histograms of the proportion of '1's at $K = 3, 8$, and 16, respectively. The averages of bit-aliasing at $K = 3$, 8 and 16 are approximately 0.486, 0.485 and 0.484, respectively.

The ideal bit-aliasing is also 0.5 because, if the bit-aliasing is close to 0 or 1, it means that different ASIC chips may generate nearly identical PUF responses [23]. Hence our LPUFs satisfy the requirement for bit-aliasing.

If a user needs a secure 128-bit AES key, he can obtain it as an output from a 128-bit hash function (e.g., SPONGENT-128 [3]), the input of which is the response of an LPUF after

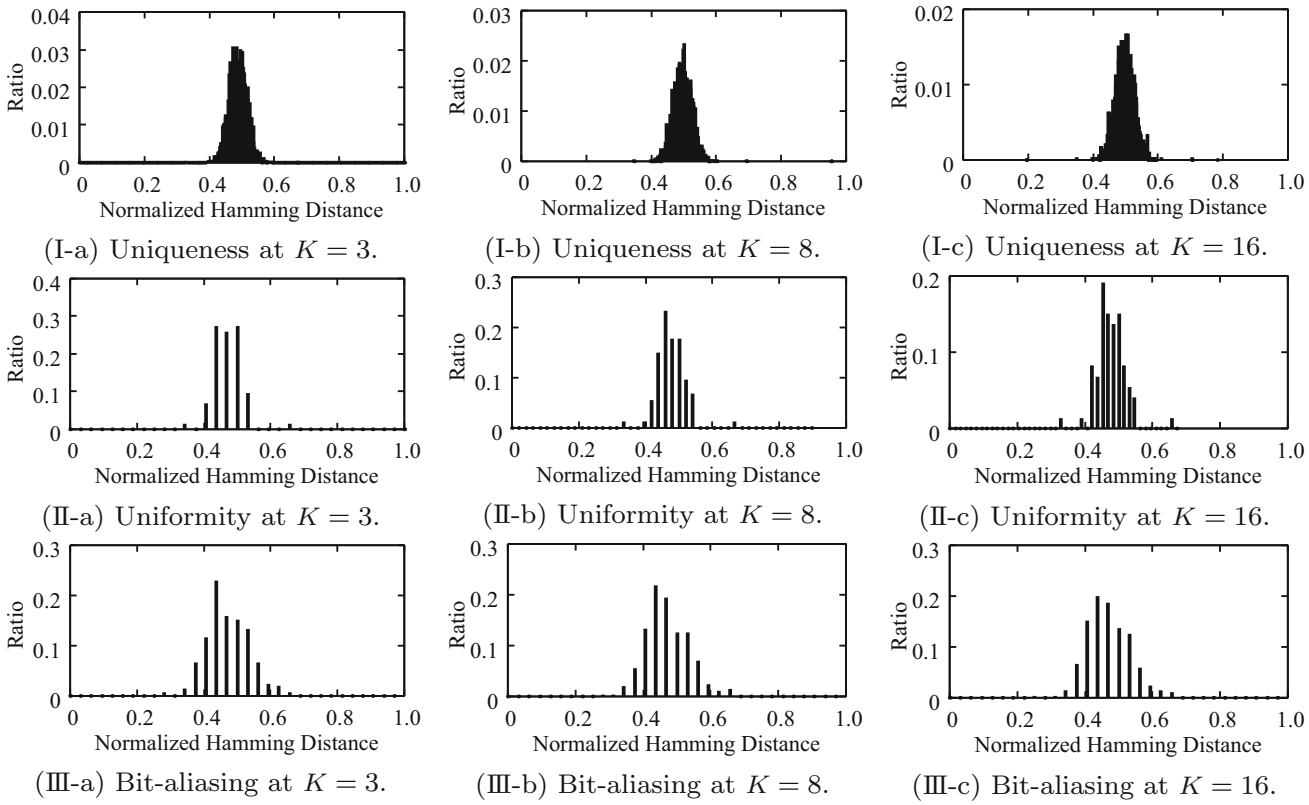

Fig. 9 PUF performances under normal conditions (27°C/1.80 V)

being corrected by ECC. If the uniqueness, uniformity and bit-aliasing of the responses are not close to 0.5 (i.e., close to 0 or 1), an attacker may be able to predict the responses and even the output of the hash function (i.e., 128-bit AES key). This is why we evaluate these metrics of PUF responses themselves.

6.3 Cost

Table 2 indicates the processing time and the gate count of our LPUF on a chip, shown in Fig. 5.

The processing time is estimated around 105 ms, this being the total time taken to extract a 1,024-bit output stream from each RS latch. One way of improving the processing time is to reduce the bitstream length, which was 1,024 bits in our experiment. However, too short a length may result in *misdividing*, an inaccuracy of X_i corresponding to LATCH_i. For example, RS latches outputting a large number of '0's and very few '1's (i.e., T_1) might be detected not as random latches, but as fixed latches (i.e., T_0). This misdividing leads to a decrease in reliability of responses, so our LPUFs make a tradeoff between reliability and processing time.

The gate count is obtained by synthesizing the LPUF on the Fujitsu 0.18-μm CMOS process [19] with the Design Compiler 2003.03. Note that one gate is equivalent to a 2-input/1-output (2-to-1) 1-bit NAND gate. The total gate count of the LPUF is about 1.2K gates. This cost is necessary for extracting constant 379 bits of entropy. We consider, therefore, that our LPUF is sufficiently small to be implemented in embedded systems. Note that our proposed method requires additional costs for multiple enrollment and reconstruction measurements. Here, we do not consider these costs since the concrete way of implementing the proposed method must be careful, as mentioned in Sect. 4.

Table 2 Processing time and gate count of our LPUF

Processing time	105 ms (1,024 cycles @ 2.5 MHz)
Total gate count	1164.3 gates
256 RS latches	512.0 gates
256-to-1 MUX	647.3 gates
1-bit FF	5.0 gates

7 Conclusion

We proposed a method of enhancing the variety and maintaining the reliability of responses from LPUFs. We focused

on the information of the proportion of '1's in the output stream from each random latch. The *dividing* process classifies implemented RS latches into K types according to the proportion of '1's in the output stream. The *labeling* process defines the unique values generated by K-type RS latches. According to our experiment with 73 fabricated ASIC chips, LPUFs with 256 RS latches can generate responses with 379-bit entropy based on the proposed method for $K = 16$, considering their errors caused by environmental fluctuations. This is about 1.72 times as large as the 220-bit entropy achieved by a conventional method of eliminating random latches, and is approximately 1.21 times as large as 314-bit entropy achieved by our previous method in [35], corresponding to the proposed method for $K = 3$. Even in the worst-case condition ($-20\,^{\circ}$C/1.65 V), the error rate of responses is 0.096. This means that our LPUFs have high robustness (reliability) against both temperature and voltage variation. Our LPUFs also satisfy the basic requirements of PUFs such as uniqueness, uniformity and bit-aliasing.

Future work could include a discussion of the concrete ways of implementing the proposed method and their security evaluation.

References

1. Armknecht, F., Maes, R., Sadeghi, A.R., Sunar, B., Tuyls, P.: Memory leakage-resilient encryption based on physically unclonable functions. In: Sadeghi, A.R., Naccache, D. (eds.) Towards Hardware-Intrinsic Security, Information Security and Cryptography, pp. 135–164. Springer, Berlin (2010)
2. van den Berg, R., Skoric, B., van der Leest, V.: Bias-based modeling and entropy analysis of PUFs. In: Proceedings of the 3rd International Workshop on Trustworthy Embedded Devices, TrustED '13, pp. 13–20. ACM (2013)
3. Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., Verbauwhede, I.: Spongent: the design space of lightweight cryptographic hashing. IEEE Trans. Comput. **62**(10), 2041–2053 (2013)
4. Bösch, C., Guajardo, J., Sadeghi, A.R., Shokrollahi, J., Tuyls, P.: Efficient helper data key extractor on FPGAs. In: CHES, pp. 181–197 (2008)
5. Briais, S., Cioranesco, J.M., Danger, J.L., Guilley, S., Naccache, D., Porteboeuf, T.: Random active shield. In: Fault Diagnosis and Tolerance in Cryptography (FDTC), 2012 Workshop on, pp. 103–113 (2012)
6. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**, 97–139 (2008)
7. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Silicon physical random functions. In: Proceedings of CCS pp. 148–160 (2002)
8. Gassend, B., Clarke, D., Lim, D., van Dijk, M., Devadas, S.: Identification and authentication of integrated circuits. In: Concurrency and computation: practice and experiences., pp. 1077–1098 (2004)
9. Gorman, C.: Counterfeit chips on the rise. Spectr. IEEE **49**(6), 16–17 (2012)
10. Guajardo, J., Kumar, S.S., Schrijen, G.J., Tuyls, P.: FPGA intrinsic PUFs and their use for IP protection. CHES **2007**, 63–80 (2007)
11. Hata, H., Ichikawa, S.: FPGA implementation of metastability-based true random number generator. IEICE Trans. **95**(D–2), 426–436 (2012)
12. Helfmeier, C., Boit, C., Nedospasov, D., Seifert, J.P.: Cloning physically unclonable functions. In: HOST (2013)
13. Holcomb, D.E., Burleson, W.P., Fu, K.: Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In: Proceedings of the Conference on RFID Security (2007)
14. Holcomb, D.E., Rahmati, A., Salajegheh, M., Burleson, W.P., Fu, K.: DRV-fingerprinting: using data retention voltage of SRAM cells for chip identification. In: RFIDSec, pp. 165–179 (2012)
15. Ignatenko, T., Schrijen, G.J., Skoric, B., Tuyls, P., Willems, F.: Estimating the secrecy-rate of physical unclonable functions with the context-tree weighting method. In: Information Theory, 2006 IEEE International Symposium on, pp. 499–503 (2006)
16. Krishna, A.R., Narasimhan, S., Wang, X., Bhunia, S.: Mecca: a robust low-overhead PUF using embedded memory array. In: CHES, pp. 407–420 (2011)
17. Kumar, S.S., Guajardo, J., Maes, R., Schrijen, G.J., Tuyls, P.: Extended abstract: the Butterfly PUF: protecting IP on every FPGA. In: HOST, pp. 67–70 (2008)
18. Lee, J.W., Lim, D., Gassend, B., Suh, G.E., van Dijk, M., S.Devadas: A technique to build a secret key in integrated circuits for identification and authentication applications. In: Proceedings of the IEEE VLSI Circuits Symposium, pp. 176–179 (2004)
19. Ltd., F.S.: CS86 technology. http://www.fujitsu.com/downloads/MICRO/fma/pdf/e620209_CS86_ASIC.pdf
20. Maes, R., Tuyls, P., Verbauwhede, I.: Intrinsic PUFs from flip-flops on reconfigurable devices. In: 3rd Benelux Workshop on Information and system security (WISSec 2008), p. 17 (2008)
21. Maes, R., Tuyls, P., Verbauwhede, I.: Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs. CHES pp. 332–347 (2009)
22. Maes, R., Verbauwhede, I.: Physically unclonable functions: a study on the state of the art and future research directions. In: Towards hardware intrinsic security: foundation and practice, pp. 3–37. Springer (2010)
23. Maiti, A., Gunreddy, V., Schaumont, P.: A systematic method to evaluate and compare the performance of physical unclonable functions. In: Embedded systems design with FPGAs, pp. 245–267. Springer, New York (2013)
24. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big data: the next frontier for innovation, competition, and productivity. Tech. rep, McKinsey Global Institute (2011)
25. Maurer, U.M.: Secret key agreement by public discussion from common information. IEEE Trans Inf Theory **39**(3), 733–742 (1993)
26. Pappu, R.S.: Physical one-way functions. Ph. D. thesis, Massachusetts Institute of Technology (2001)
27. Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., Schmidhuber, J.: Modeling attacks on physical unclonable functions. In: Proceedings of the 17th ACM conference on Computer and communications security, CCS 2010, pp. 237–249 (2010)
28. Simons, P., van der Sluis, E., van der Leest, V.: Buskeeper PUFs, a promising alternative to D Flip-Flop PUFs. In: Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on, pp. 7–12 (2012)
29. Su, Y., Holleman, J., Otis, B.: A 1.6pJ/bit 96% stable chip-ID generating circuit using process variations. In: IEEE International Solid-State Circuits Conference—ISSCC 2007. IEEE, pp. 406–611 (2007)
30. Su, Y., Holleman, J., Otis, B.P.: A digital 1.6pJ/bit chip identification circuit using process variations. Solid-State Circuits IEEE J. **43**(1), 69–77 (2008)

31. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: Proceedings of DAC pp. 9–14 (2007)
32. Torrance, R., James, D.: The state-of-the-art in IC reverse engineering. In: CHES, pp. 363–381 (2009)
33. Tuyls, P., Schrijen, G.J., Skoric, B., van Geloven, J., Verhaegh, N., Wolters, R.: Read-proof hardware from protective coatings. In: CHES, pp. 369–383 (2006)
34. Xilinx: Spartan-3E starter kit board. http://www.xilinx.com/products/boards-and-kits/hw-spar3e-sk-us-g.html
35. Yamamoto, D., Sakiyama, K., Iwamoto, M., Ohta, K., Takenaka, M., Itoh, K.: Variety enhancement of PUF responses using the locations of random outputting RS latches. J Cryptogr Eng **3**(4), 197–211 (2013)