IACR Transactions
ISSN XXXX-XXXX, Vol. 0, No. 0, pp. 1–23. DOI:XXXXXXXX

Entropy Estimation of Physically Unclonable Functions

Mitsuru Shiozaki[1], Yohei Hori[2] and Takeshi Fujino[3]

[1] Ritsumeikan University, Kusatsu, Shiga, Japan, mshio@fc.ritsumei.ac.jp
[2] The National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki, Japan, hori.y@aist.go.jp
[3] Ritsumeikan University, Kusatsu, Shiga, Japan, fujino@se.ritsumei.ac.jp

Abstract. Physically unclonable functions (PUFs) are gaining attention as a promising cryptographic technique; the main applications using PUFs include challenge-response authentication and key generation (key storage). When a PUF is applied to these applications, min-entropy estimation is essential. Min-entropy is a measure of the lower bound of the unpredictability of PUF responses. A prominent scheme for estimating min-entropy is the National Institute of Standards and Technology (NIST) specification (SP) 800-90B. It includes several statistical tests and ten kinds of estimators aimed at estimating the min-entropy of random number generators (RNGs). Several studies have estimated the min-entropy of PUFs as well as those of RNGs by using SP 800-90B. In this paper, we point out two problems in this scheme to estimate the min-entropy of PUFs. One is that the estimation results vary widely by the ordering of the PUF responses. The other is that the entropy estimation suite of SP 800-90B can overestimate PUF min-entropy. Both problems are related to the cause of lower entropy due to variations in the manufacturing of circuits and transistors (except for the PUF sources, which are circuits and transistors used to extract intrinsic physical properties and to generate device unique responses), named "multiple sources" [1]. We call these circuits and transistors "entropy-loss sources" in contrast to the PUF sources. We applied three orderings to the PUF responses of our static random-access memory (SRAM) PUF and our complementary metal-oxide-semiconductor (CMOS) image sensor with a PUF (CIS PUF): row-direction ordering, column-direction ordering, and random-shuffle ordering. We demonstrated that the estimated min-entropy varies with the ordering. In particular, we found that arranging the PUF responses in readout order results in the overestimation of the min-entropy. We used numerical simulation to create numerical PUFs with the entropy-loss source. We demonstrated that the entropy estimation suite overestimates their entropy.

Keywords: Physically unclonable function (PUF) · Min-entropy · NIST SP 800-90B · SRAM PUF · CMOS image sensor with a PUF (CIS PUF)

1 Introduction

Physically unclonable functions (PUFs) [2, 3] were developed to improve the security of low-cost applications such as those to be connected to the Internet of Things (IoT) and are gaining attention as a promising cryptographic technique. The main applications using PUFs include challenge-response authentication and key generation (key storage); min-entropy estimation is essential for such applications. Entropy is a measure of the unpredictability of PUF responses; min-entropy is used to establish this lower bound and represents the worst-case scenario. A PUF designer should estimate the min-entropy to prevent PUF cloning and to ensure a given security level. Many studies have discussed

PUF entropy, and the entropy estimation suite of the National Institute of Standards and Technology (NIST) special publication (SP) 800-90B [4] is becoming a primary entropy estimation scheme. It includes several statistical tests and ten kinds of estimators aimed at estimating the min-entropy of a random number generator (RNG). In several studies, the PUF responses were concatenated in the readout order. The min-entropy of the concatenation was estimated using SP 800-90B and used as an indicator of PUF performance. However, more careful estimation of PUF entropy is needed.

In this paper, we discuss the problems of estimating PUF entropy based on NIST SP 800-90B. We also present experimental results demonstrating that the entropy estimation suite of SP 800-90B overestimates the min-entropy of PUFs. Our study is not aimed at overturning the findings of previous studies but rather at alerting PUF designers to the need for extreme caution in estimating entropy.

1.1 Related Work

The entropy of optical PUFs [3] and coating PUFs [5] has been widely studied [6, 7, 8]. Several studies on silicon PUFs [9, 10, 11, 12] estimated the entropy by referring to the secrecy rate [7], which takes into account reliability (i.e., the intra-Hamming distance (intra-HD) [13]). Schaub et al. investigated the relationship between the entropy and reliability of three well-known delay-based PUFs (i.e., ring-oscillator (RO) PUF [14], RO sum PUF [15], and Loop PUF [16]) by using a stochastic model [17]. Since PUFs extract tiny physical properties from random and uncontrollable variations in manufacturing, PUF responses often contain noise. The reliability is a performance metric that indicates the amount of noise in PUF responses. Since noise makes entropy estimation difficult, the effect of noise on min-entropy is an important research topic.

Several studies have also discussed the relationship between the characteristic of the physical mechanisms behind the variation and PUF min-entropy. There have been studies discussing a PUF-response bias, debiasing schemes, and entropy [18, 19, 20]. The PUF-response bias is a research topic related to the mean of inter-HD [13] and is basically due to an unbalanced layout in an LSI chip. Gu et al. also studied the relationship between uniqueness (i.e., inter-HD) and min-entropy [21]. They focused on the mean of inter-HD, but even though the standard deviation is related to the dependence between PUF responses, they did not discuss it. Xu et al. pointed out the importance of understanding the physical mechanisms behind variations in flash-memory PUFs (FPUFs) [22]. They reported that systematic layout variations and manufacturing lots might reduce entropy. When such various causes of lower entropy are intertwined, NIST SP800-90B is a powerful tool for estimating entropy. Therefore, the NIST entropy estimation suite is becoming a primary entropy estimation scheme.

However, there is a case that PUF responses can not be directly inputted into the NIST entropy estimation suite. It is well known that the pairwise comparison used in RO PUFs reduces entropy, and the entropy is based on the frequency ordering of the ROs [14]. Maes et al. reported that encoding the frequency ordering is required to remove the dependencies between PUF responses [23]. A PUF designer thus needs to carefully consider whether the entropy estimation suite can reliably estimate the min-entropy.

1.2 Our Contributions

As mentioned above, the entropy estimation suite of NIST SP 800-90B can overestimate the min-entropy of PUFs. This paper makes the following contributions.

- PUF models with entropy-loss sources

  We present the models of the static random-access memory (SRAM) PUF [24, 25] and of the complementary metal-oxide-semiconductor (CMOS) image sensor with a PUF

(CIS PUF) [26] that have entropy-loss sources. We also present a typical PUF model with an entropy-loss source. In general, circuits and transistors used to extract the inherent physical properties from variations in manufacturing and to generate PUF responses are called "PUF sources." In this paper, we discuss the effect of variations in the manufacturing of circuits and transistors (except for the PUF sources) on PUF entropy. We term these circuits and transistors "entropy-loss sources" in contrast to the PUF sources since they reduce PUF entropy. The entropy-loss source is closely related to the two following overestimations. We use our SRAM PUF, our CIS PUF, and the numerical PUFs with entropy-loss sources to demonstrate that the entropy estimation suite of SP 800-90B overestimates PUF entropy.

- Overestimation due to the ordering of PUF responses

  We discuss the problem of the results of the entropy estimation suite varying widely with the ordering of the PUF responses. Several previous studies collected the PUF responses and concatenated them in readout order and then estimated the min-entropy by using the entropy estimation suite. If the entropy estimation suite correctly estimates the min-entropy of PUFs with entropy-loss sources, the results should be the same regardless of the ordering of the PUF responses. We applied three orderings to the PUF responses of our SRAM PUF and CIS PUF and estimated the min-entropy. We demonstrated that the estimated results vary with the ordering and that the entropy estimation with the PUF responses in readout order results in overestimation.

- Overestimation due to the estimators

  We discuss the estimated min-entropy of PUFs with entropy-loss sources. We point out that the entropy estimation suite overestimates min-entropy compared with an estimate based on theoretical considerations and explain the reason for this. Using numerical simulation, we created numerical PUFs with an entropy-loss source and used them to investigate the degree of overestimation.

The remainder of this paper is organized as follows. In Section 2, we introduce the conventional PUF model and NIST SP 800-90B as background information. In Section 3, we introduce our SRAM PUF and CIS PUF, which are used to investigate the inherent problems in estimating PUF entropy using NIST SP 800-90B. We also present our PUF models and the evaluation results using inter-HD. In Section 4, we present the three orderings we applied to the PUF responses of our SRAM PUF and CIS PUF and demonstrate that the results of the SP 800-90B entropy estimation suite vary with the ordering. In Section 5, we demonstrate that the entropy estimation suite overestimates the min-entropy of PUFs and explain why. We conclude in Section 6 with a summary of the key points.

2 Conventional PUF Model and NIST SP 800-90B

2.1 Conventional PUF model

A PUF extracts inherent physical properties from random and uncontrollable variations in manufacturing. Maes modeled the probabilistic behavior of a PUF and represented the relationship between the PUF responses and manufacturing process variations as follows [27].

$$R_{p,r,m} = \begin{cases} 0 & (v_{p,r} + n_{p,r,m} \leq T) \\ 1 & (\text{otherwise}), \end{cases} \tag{1}$$

where $v_{p,r}$ is a manufacturing process variable of the r-th PUF source in the p-th device, $n_{p,r,m}$ is a noise variable of the r-th PUF source of the m-th measurement from the p-th device, and T is a constant threshold parameter. p ranges from 1 to N_{puf}, where N_{puf} is the number of devices (PUFs). r ranges from 1 to N_{res}, where N_{res} is the number of PUF sources (PUF responses) per device (PUF). In this model, the PUF sources and noise are assumed to be independent. If the PUF sources are independent, as shown in Equation 1, the approach of concatenating PUF responses is not strange, and NIST SP800-90B is suitable for estimating entropy.

When considering PUF entropy, it is necessary to consider both of these effects. As mentioned above, the effect of noise (reliability) on entropy is an important research topic. However, in this paper, we exclude entropy due to reliability to simplify the problem (i.e., $n_{p,r,m} = 0$).

2.2 NIST SP 800-90B

The entropy estimation suite of NIST SP 800-90B is superior to other methods for estimating the quality of an entropy source. Since some sources of entropy may have unknown dependencies, ten kinds of entropy estimators are used in the entropy estimation suite to minimize the probability that the estimation results are significantly overestimated.

NIST SP 800-90B requires a sequential dataset of at least 1,000,000 sample values. After the required number of samples are collected, they are divided into two tracks (i.e., the independent and identically distributed (IID) track and the non-IID track) using statistical tests (i.e., chi-square statistical tests and permutation tests). Entropy is estimated differently for each track.

For the IID track, the result estimated using the most common value (MCV) estimator is used as the value of min-entropy. MCV estimation is the simplest because it is based on the assumption that the sample values in the dataset do not correlate. The maximum probability of the values in the dataset is measured, and the min-entropy of an independent discrete random variable X is defined as

$$H_\infty(X) = -\log_2 \max \mathbf{Pr}(X = x_i) \qquad (i = 1, .., n), \tag{2}$$

where $x_1, x_2, .., x_n$ are possible values, and $\mathbf{Pr}(X = x_i)$ is the probability of random variable $(X = x_i)$. The ideal value per bit is 1.

For the non-IID track, ten kinds of estimators, including the MCV estimator, are used: the collision estimator, the Markov estimator, the compression estimator, the *t*-tuple estimator, the longest repeated substring (LRS) estimator, the multi most common in window (MultiMCW) prediction estimator, the lag prediction estimator, the multiple Markov model with counting (MultiMMC) prediction estimator, and the LZ78Y prediction estimator. The minimum value of the ten estimates is taken as the min-entropy.

The collision estimator, devised by Hagerty and Draper [28], measures the mean number of samples to the first collision. Using this number, it estimates the probability of the most-likely output value. The Markov estimator is based on the assumption that the collected sample values obey a Markov model and measures the dependencies between consecutive values. The compression estimator, also devised by Hagerty and Draper [28], computes the entropy rate of a dataset on the basis of the Maurer Universal Statistic [29]. The *t*-tuple estimator measures the frequency of *t*-tuples (i.e., pairs, triples, etc.) that appear in the dataset and provides an estimate based on the frequency of the most common *t*-tuples. The LRS estimator computes the collision entropy on the basis of the number of repeated tuples. The MultiMCW, lag, MultiMMC, and LZ78Y prediction estimators ("predictors") aim to guess the next sample given the previous one and provide an estimate based on the probability of successfully predicting it. Each predictor consists of several sub-predictors that compete, and the one with the highest prediction rate is

selected. Each sub-predictor of the MultiMCW predictor predicts the next sample on the basis of the MCV in the sliding window for the previous sample. Each sub-predictor of the lag predictor predicts the next sample on the basis of a specified lag. Each MMC sub-predictor of the MultiMMC predictor records the observed frequencies for transitions from one sample to the subsequent one and predicts the next sample on the basis of the most frequently observed transition from the current sample. The LZ78Y predictor is loosely based on LZ78 encoding with Bernstein's Yabba scheme [30] for adding strings to the dictionary.

3 SRAM PUF and CIS PUF

The SRAM PUF and CIS PUF were used to investigate inherent problems in estimating PUF entropy using NIST SP 800-90B. In this section, we introduce our SRAM PUF and CIS PUF, their PUF models, and their performances evaluated using the inter-HD metric [13].

3.1 SRAM PUF

An SRAM PUF is a well-known PUF that uses the initial values of an SRAM as PUF responses [24, 25]. Two cross-coupled inverters in an SRAM cell are symmetrical, and the SRAM cell ideally enters a metastable state during the power-up phase. However, transistors consisting of cross-coupled inverters have mismatches due to manufacturing process variations, and these mismatches are amplified by the positive feedback of the cross-coupled inverters. Each SRAM cell thus boots up with an initial state of either '0' or '1'. PUF sources of the SRAM PUF are assumed to be SRAM cells (two cross-coupled inverters).

3.1.1 Our SRAM PUF

Figure 1 shows a block diagram of our SRAM PUF. We designed a PUF test vehicle chip with a 180-nm CMOS process for benchmarking PUFs [31]. It has four SRAM standard cells with 1K words (Column) of 16 bits (Row). We used the initial values of the single SRAM standard cell as our SRAM PUF. There are thus 16 and 1024 SRAM cells in each row and each column, respectively. The number of PUF-response bits is 16,384 ($=16 \times 2^{10}$) bits.

3.1.2 SRAM-PUF Model

We explain the SRAM-PUF model. In several previous studies, it was assumed that entropy-loss sources (i.e., the variations in the manufacturing of circuits and transistors (except for PUF sources; i.e., SRAM cells)) are negligible. However, this assumption is unrealistic. Manufacturing process variations of other circuits and transistors (entropy-loss sources; e.g., word lines, bit lines, word-line buffers, and sense amplifiers) also affect PUF responses. Thus, we intentionally include these manufacturing process variations in our SRAM-PUF model. In contrast to the PUF model (Equation 1), the relationship between PUF responses and manufacturing process variations is given as

$$R_{p,r_r,r_c} = \begin{cases} 0 & (v_{p,r_r,r_c} + v^{\mathrm{row}}_{p,r_r} + v^{\mathrm{column}}_{p,r_c} \leq 0) \\ 1 & (\text{otherwise}), \end{cases} \tag{3}$$

where v_{p,r_r,r_c} is a manufacturing-process variable of the SRAM cell in the r_r-th row and r_c-th column in the p-th device, v^{row}_{p,r_r} is a manufacturing process variable based on the common circuit (e.g., word line and word-line buffer) in the r_r-th row in the p-th device,

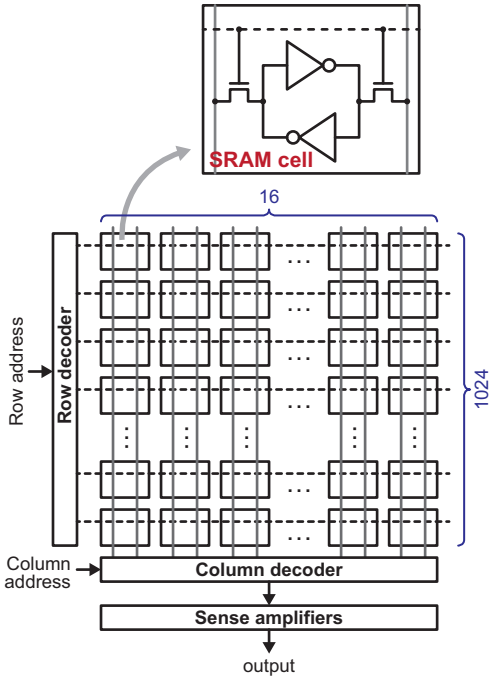

Figure 1: Block diagram of our SRAM PUF

and $v_{p,r_c}^{\text{column}}$ is a manufacturing process variable based on the common circuit (e.g., bit line and sense amplifier) in the r_c-th column in the p-th device (r_r and r_c range from 1 to 16 and 1024, respectively, and v_{p,r_r}^{row} and $v_{p,r_c}^{\text{column}}$ mean adding different offsets to each row's and each column's PUF responses, respectively). When there are multiple manufacturing process variables, as shown in Equation 3, the entropy per PUF is less than the number of PUF-response bits. Shiozaki et al. previously named this cause of lower entropy "multiple sources" [1]. In general, the variation in column circuits, such as sense amplifiers, is higher than that in row circuits and can affect PUF responses (PUF entropy).

3.1.3 Performance Evaluation

A heat map of PUF responses is shown in Figure 2. It depicts the measurement results for a sample SRAM PUF. Red, blue, and white represent stable PUF response '1', stable PUF response '0', and unstable PUF response, respectively. The distribution of responses '1' and '0' seems random.

Next, we present the results of evaluating the performance of our SRAM PUF using the inter-HD metric. We generated all the 16,384-bit PUF responses from 80 ($= 4 \times 20$ chips) SRAM PUFs and calculated the mean and standard deviation of inter-HD. The number of repeated measurements of each PUF-response bit was 100. The mean and standard deviation were 0.4972 and 0.5742, respectively. The mean was approximately 0.5, the ideal value. It means that the number of '0' and '1' response was the same and indicates that our SRAM PUF may maintain high entropy. However, the standard deviation was slightly higher than 0.5.

3.2 CIS PUF

Okura et al. developed a CIS PUF to enhance the security of IoT devices with CMOS image sensors [26]. The reason why we choose the CIS PUF as a case study is that the amount of the variations is measurable. We purposely applied a PUF-response-generation

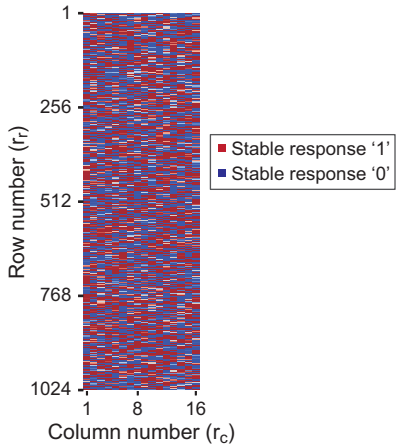

Figure 2: Heat map of PUF responses for a sample SRAM PUF

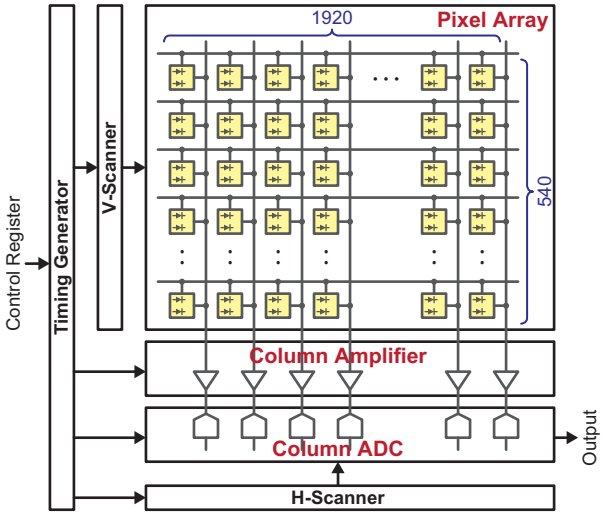

Figure 3: Block diagram of the CIS PUF

scheme that creates the entropy-loss sources to investigate the inherent problems in entropy estimation.

3.2.1 Our CIS PUF

Our CIS PUF has two operation modes: image readout and PUF. In image readout mode, it detects light and converts it into a 2M-pixel image. In PUF mode, all the photodiodes are disabled, and PUF responses are generated from fixed-pattern noise (FPN), which is based on manufacturing process variations. Figure 3 shows a block diagram of our CIS PUF. Our CIS PUF is composed of a 2M ($= 1980 \times 1080$) pixel array with a two-shared pixel structure, a column amplifier, a column analog-to-digital converter (ADC), and digital control blocks. The control-register signals switch between image readout mode and PUF mode. Since CMOS image sensors typically have a column-parallel readout scheme, the components of the CIS PUF are divided into 1920 independent column circuits like the one shown in Figure 4. Each column circuit is composed of 540-pixel cells, an amplifier, and an ADC. There are 540 source-follower (SF) transistors shared by two photodiodes. We hypothesize that the threshold-voltage (Vth) variation of the SF transistors is a PUF

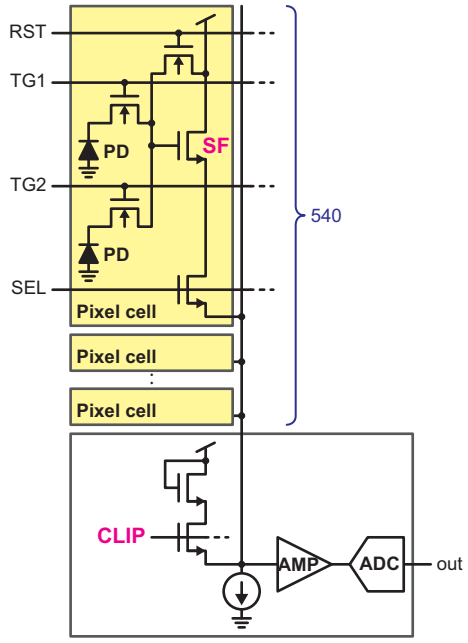

Figure 4: Block diagram of column circuit

source. In PUF mode, the correlated double sampling (CDS) function, which attenuates or removes an undesired offset noise, is disabled and the digitized value of the Vth variation is readout. In image readout mode, the Vth variation is removed by the CDS function. No information about the PUF responses leaks from the 2M-pixel image. A diode-connected clip transistor is used to reduce the supply-voltage/ground bounce during image readout and to derive the Vth of the SF transistor during PUF-mode operation.

CMOS image sensors suffer from column FPN, which appears as stripes in the image and results in significantly degraded image quality. The column FPN is mainly caused by variations in the manufacturing of the amplifier and ADC. The PUF responses of a CIS PUF are also affected by the column FPN, where the clip-transistor variation is dominant. The PUF responses are generated by comparing the digitized values of two selected SF-transistor variations. When two SF transistors are randomly selected, variations of different clip transistors are added to those of the SF transistors. Thus, we compared vertically adjacent SF transistors to generate PUF responses.

We call the PUF-response-generation scheme in which the use of SF transistors does not overlap, and vertically adjacent SF transistors are compared "vertical comparison." Since an identical clip transistor is used in the vertical comparison, the variation of the clip transistor is canceled through the comparison, and the column FPN is removed. The number of PUF responses is 270 ($= \frac{540}{2}$) bits in each column. Since there are 1920 column circuits, the number of PUF responses is 518K bits ($= 1920 \times 270$).

We purposely apply a PUF-response-generation scheme that differs from vertical comparison to study the effect of variations in the manufacturing of clipping transistors on entropy estimation. The PUF-response-generation scheme in which the use of SF transistors does not overlap, and horizontally adjacent SF transistors are compared is termed "horizontal comparison." The number of PUF responses is 540 bits in each pair of column circuits. Since there are 960 ($= \frac{1920}{2}$) pairs of column circuits, the number of PUF responses is 518K ($= 960 \times 540$) bits, the same as for vertical comparison.

3.2.2 CIS-PUF Model

The CIS-PUF model also includes variations in the manufacturing of circuits and transistors (except for PUF sources). Here, the PUF source is an SF transistor in a pixel cells. Each manufacturing process variable of the pixel cell, including other circuits, is expressed as

$$v'_{p,r_r,r_c} = v_{p,r_r,r_c} + v^{\mathrm{row}}_{p,r_r} + v^{\mathrm{column}}_{p,r_c}, \tag{4}$$

where v_{p,r_r,r_c} is a manufacturing process variable of the SF transistor in the r_r-th row and r_c-th column in the p-th device, v^{row}_{p,r_r} is a manufacturing process variable of the common circuit (e.g., v-scanner) in the r_r-th row in the p-th device, and $v^{\mathrm{column}}_{p,r_c}$ is a manufacturing process variable of the common circuit (e.g., clip transistor) in the r_c-th column in the p-th device. PUF responses are generated by comparing the digitized values of two pixel-cell variables. The difference between the pixel-cell variables of the vertical comparison is expressed as

$$\begin{aligned} v'_{p,2\times r_r,r_c} - v'_{p,2\times r_r-1,r_c} &= v_{p,2\times r_r,r_c} - v_{p,2\times r_r-1,r_c} + v^{\mathrm{row}}_{p,2\times r_r} - v^{\mathrm{row}}_{p,2\times r_r-1} \\ &= \Delta v_{p,r_r,r_c} + \Delta v^{\mathrm{row}}_{p,r_r}. \end{aligned} \tag{5}$$

The variation based on the column circuits, such as clip transistors, is canceled through the comparison, and the column FPN is removed (i.e., $\Delta v^{\mathrm{column}}_{p,r_c} = 0$) since identical variables ($v^{\mathrm{column}}_{p,r_c}$) are compared in the vertical comparison. The variation based on the less affected row circuits remains as an entropy-loss source. A PUF-response bit of our CIS PUF using the vertical comparison is modeled as

$$R_{p,r_r,r_c} = \begin{cases} 0 & (\Delta v_{p,r_r,r_c} + \Delta v^{\mathrm{row}}_{p,r_r} \leq 0) \\ 1 & (\text{otherwise}), \end{cases} \tag{6}$$

where $\Delta v_{p,r_r,r_c}$ is a differential variable in the r_r-th row and r_c-th column in the p-th device, and $\Delta v^{\mathrm{row}}_{p,r_r}$ is a differential variable based on the row circuits (e.g., v-scanner) in the r_r-th row in the p-th device (r_r and r_c range from 1 to 1920 and from 1 to 270, respectively).

Similar to the vertical comparison, the difference between the pixel-cell variables of the horizontal comparison is expressed as

$$\begin{aligned} v'_{p,r_r,2\times r_c} - v'_{p,r_r,2\times r_c-1} &= v_{p,r_r,2\times r_c} - v_{p,r_r,2\times r_c-1} + v^{\mathrm{column}}_{p,2\times r_c} - v^{\mathrm{column}}_{p,2\times r_c-1} \\ &= \Delta v_{p,r_r,r_c} + \Delta v^{\mathrm{column}}_{p,r_r}. \end{aligned} \tag{7}$$

In contrast to the vertical comparison, the variation based on the less affected row circuits is canceled through the comparison (i.e., $\Delta v^{\mathrm{row}}_{p,r_r} = 0$), but the column FPN remains as an entropy-loss source. A PUF-response bit of our CIS PUF using the horizontal comparison is modeled as

$$R_{p,r_r,r_c} = \begin{cases} 0 & (\Delta v_{p,r_r,r_c} + \Delta v^{\mathrm{column}}_{p,r_c} \leq 0) \\ 1 & (\text{otherwise}), \end{cases} \tag{8}$$

where $\Delta v_{p,r_r,r_c}$ is a differential variable in the r_r-th row and r_c-th column in the p-th device, and $\Delta v^{\mathrm{column}}_{p,r_c}$ is a differential variable based on the column circuits (e.g., clip transistors) in the r_c-th column in the p-th device (r_r and r_c range from 1 to 960 and from 1 to 540, respectively).

3.2.3 Performance Evaluation

First, we show heat maps of the ADC outputs and PUF responses. Figure 5 depicts the measurement results for a sample CIS-PUF chip. The ADC outputs in Figure 5 (a) show

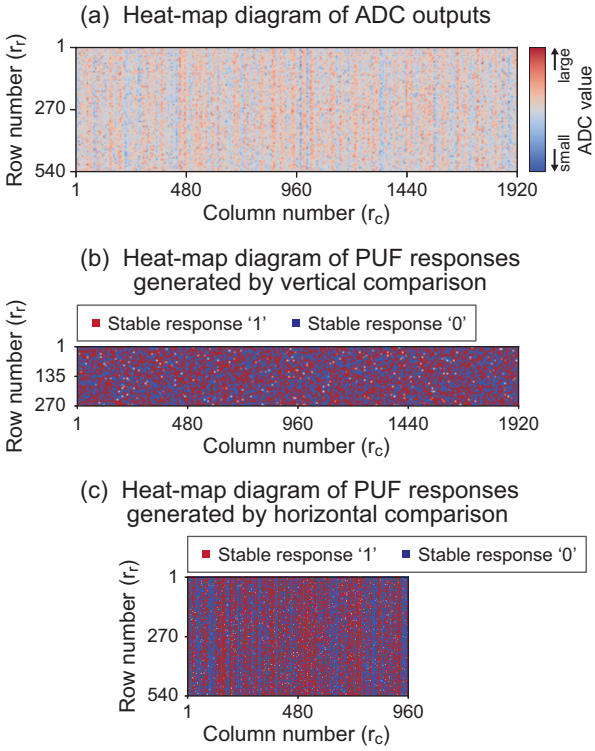

Figure 5: Heat maps of ADC outputs and PUF responses for a sample CIS PUF chip

digitized values of 1920×540 pixel cells. Red and blue mean that the ADC output value is large and small, respectively. The column FPN clearly appears as stripes. Figures 5 (b) and (c) show PUF responses generated by vertical and horizontal comparisons, respectively. Similar to the heat map for our SRAM PUF, red, blue, and white mean stable PUF response '1', stable PUF response '0', and unstable PUF response, respectively. Use of vertical comparison to generate PUF responses resulted in a distribution of responses '1' and '0' that seems random. The effect of the column FPN remains in the PUF-response distribution resulting from the horizontal comparison. The entropy with the horizontal comparison seems much lower than that with the vertical comparison.

Next, we present the results of evaluating the performance of our CIS PUF using the inter-HD metric. We generated all the 518K-bit ($= \frac{1920 \times 540}{2}$) PUF responses from 18 chips and calculated the mean and standard deviation of inter-HD. Similar to our SRAM PUF, the number of repeated measurements of each PUF-response bit was 100. With the vertical comparison, the mean and standard deviation were approximately 0.5. It indicates that the entropy is high. In contrast, the mean with the horizontal comparison was approximately 0.5, but the standard deviation was 2.46. It was 5.5 times larger compared with that with the vertical comparison. The stripes on the heat map in Figure 5 (c) are related to the large standard deviation of inter-HD.

4 Min-Entropy Estimation using NIST SP 800-90B

In this section, we first introduce the two datasets we used to estimate min-entropy. We then present the orderings of the PUF responses to investigate the hypothesis that, if the entropy estimation suite of NIST SP 800-90B correctly estimates PUF entropy, the ordering should not significantly affect the estimation results; otherwise, the estimation

Table 1: Datasets for min-entropy estimation

	SRAM PUF	**CIS PUF**
Number of PUF responses per PUF	16,384 bits	518,400 bits
Number of PUFs per chip	4	1
Number of chips	20	18
Total number of PUF responses	1,310,720 bits	9,331,200 bits

results will vary. Finally, we present the estimated min-entropy for our SRAM PUF and our CIS PUF and demonstrate that the estimation results vary with the ordering.

4.1 Datasets

The datasets we used to estimate min-entropy are summarized in Table 1. We define an SRAM standard cell with 1K words (Column) of 16 bits (Row) as our SRAM PUF and aim to estimate the entropy of a single SRAM PUF. All PUF responses were generated 100 times under typical conditions (1.8 V, 25 °C), and whether each PUF-response bit was '0' or '1' was decided by majority vote to exclude entropy due to reliability. The number of PUF-response bits in our SRAM PUF was 16,384 ($=16 \times 2^{10}$), less than that required for the entropy estimation suite. Although there were four SRAM PUFs on a chip, the number of required samples was not reached even if their responses were concatenated. We thus concatenated the PUF responses for 20 chips, mimicking previous studies. The total number of samples (PUF-response bits) was thus 1,310,720 bits, barely meeting the criteria of NIST SP 800-90B. Similarly, all PUF responses for our CIS PUF were generated 100 times under typical conditions (2.8 V, 25 °C), and whether each PUF-response bit was '0' or '1' was decided by majority vote. The number of PUF response bits in our CIS PUF was 518K ($= \frac{1920 \times 540}{2}$), less than the number required. We thus concatenated the PUF responses for 18 chips to estimate min-entropy. The total number of samples was thus approximately 10M bits, sufficient for NIST SP 800-90B.

4.2 Response Ordering

If we hypothesize that the entropy estimation suite correctly estimates PUF entropy, the estimation results should be the same regardless of response ordering. We thus tested three orderings: row-direction ordering, column-direction ordering, and random-shuffle ordering.

- Row-direction ordering:

  Row-direction ordering is based on the assumption of a simple readout. We arranged the PUF responses assuming incrementation of the access address of an SRAM and general readout scheme of an image sensor. The datasets for row-direction ordering were constructed by concatenating the rows (i.e., $R_{1,1,1}$, $R_{1,1,2}$, $R_{1,1,3}$, ..., $R_{1,1,N_{column}}$, $R_{1,2,1}$, $R_{1,2,2}$, $R_{1,2,3}$, ..., $R_{1,2,N_{column}}$, $R_{1,3,1}$, ..., $R_{1,N_{row},N_{column}}$, $R_{2,1,1}$, ..., $R_{N_{chip},N_{row},1}$, $R_{N_{chip},N_{column},2}$, $R_{N_{chip},N_{column},3}$, ..., $R_{N_{chip},N_{row},N_{column}}$).

- Column-direction ordering:

  As mentioned above, manufacturing process variations based on column circuits (entropy-loss sources) cannot be often ignored. The column-direction ordering was aimed at emphasizing the dependencies (offset effects due to entropy-loss sources) in PUF responses. The datasets for column-direction ordering were constructed by concatenating the columns (i.e., $R_{1,1,1}$, $R_{1,2,1}$, $R_{1,3,1}$, ..., $R_{1,N_{row},1}$, $R_{1,1,2}$, $R_{1,2,2}$, $R_{1,3,2}$, ..., $R_{1,N_{row},2}$, $R_{1,1,3}$, ..., $R_{1,N_{row},N_{column}}$, $R_{2,1,1}$, ..., $R_{N_{chip},1,N_{column}}$, $R_{N_{chip},2,N_{column}}$, $R_{N_{chip},3,N_{column}}$, ..., $R_{N_{chip},N_{row},N_{column}}$).

Table 2: Estimation results for our SRAM PUF

	Ordering		
	Row	**Column**	**Random**
min-entropy	0.875779	0.728256	0.991752
Chi-square	✓		✓
Permutation			✓
MCV	0.991752	0.991752	0.991752
Collision	0.875779	0.946370	–
Markov	0.987185	0.995143	–
Compression	0.884535	0.728256	–
***t*-Tuple**	0.927891	0.888295	–
LRS	0.987152	0.974152	–
MultiMCW	0.990197	0.992418	–
Lag	0.912400	0.912325	–
MultiMMC	0.987595	0.912523	–
LZ78Y	0.988912	0.992865	–

- Random-shuffle ordering:

  Random-shuffle ordering is based on the assumption of random PUF challenges. The PUF responses of each chip were thus arranged using the identical rule. The datasets for random-shuffle ordering were constructed by concatenating randomly shuffled PUF responses (e.g., $R_{1,216,477}$, $R_{1,324,94}$, $R_{1,79,152}$, ..., $R_{1,333,318}$, $R_{2,216,477}$, $R_{2,324,94}$, $R_{2,79,152}$, ..., $R_{2,333,318}$, $R_{3,216,477}$, ..., $R_{N_{chip},216,477}$, $R_{N_{chip},324,94}$, $R_{N_{chip},79,152}$, ..., $R_{N_{chip},333,318}$).

We estimated the min-entropies of our SRAM PUF and CIS PUF using the SP 800-90B C++ code provided by NIST [32]. The symbol size can be set from 1 to 8 bits; we set it to 1 bit due to the limited number of samples for our SRAM PUF.

4.3 Estimation Results for our SRAM PUF

The estimation results for the min-entropy of our SRAM PUF are summarized in Table 2. A checkmark represents that the statistical tests are passed. When we applied the random-shuffle ordering to the PUF responses, the entropy estimation suite determined the PUF responses to be IID, and the result the MCV estimator estimated was 0.991752 bits per bit. It is close to the ideal value, meaning that our SRAM PUF has high performance. However, the hypothesis that PUF responses with row-direction and column-direction orderings are IID was rejected by the statistical tests. In particular, the column-direction ordering failed both the chi-square and permutation tests. The result with row-direction ordering was 0.875779 bits per bit, about 88 % less than that with random-shuffle ordering. With column-direction ordering, the result the compression estimator estimated was minimal, 0.728256 bits per bit. It was about 73 % less than that with random-shuffle ordering. Even though the same PUF responses were used, the estimation results varied with the ordering of PUF responses.

4.4 Estimation Results for our CIS PUF

First, for reference, we estimated the min-entropy of our CIS PUF using the vertical comparison. The results are summarized in Table 3. Similar to the results of our SRAM PUF, a checkmark represents that the statistical tests are passed. The entropy estimation suite determined the PUF responses to be IID regardless of the ordering of

Table 3: Estimation results for our CIS PUF using vertical comparison

	Ordering		
	Row	**Column**	**Random**
min-entropy	0.996587	0.996587	0.996587
Chi-square	✓	✓	✓
Permutation	✓	✓	✓

Table 4: Estimation results for our CIS PUF using horizontal comparison

	Ordering		
	Row	**Column**	**Random**
min-entropy	0.933645	0.035471	0.883727
Chi-square			
Permutation			
MCV	0.989872	0.989872	0.989872
Collision	0.940284	0.423148	0.971080
Markov	0.990410	0.699929	0.991603
Compression	0.934433	0.284987	0.883727
***t*-Tuple**	0.933645	0.035471	0.939267
LRS	0.996167	0.071218	0.962016
MultiMCW	0.983936	0.045914	0.985152
Lag	0.992932	0.046004	0.997805
MultiMMC	0.989258	0.045914	0.990085
LZ78Y	0.989918	0.046004	0.989965

PUF responses. The min-entropy was 0.996587 bits per bit and was close to the ideal value, indicating that our CIS PUF using vertical comparison has high performance.

The estimation results for our CIS PUF using horizontal comparison are summarized in Table 4. The hypothesis that the PUF responses are IID was rejected by the statistical tests regardless of the ordering. The estimation results varied widely with the ordering. With the row-direction and random-shuffle orderings, all the estimators estimated high entropy even though there are clear dependencies between PUF responses in each column, as shown in Figure 5 (c). The estimation results were approximately 0.9 bits per bit. With the column-direction ordering, the estimation results of some estimators were an order of magnitude smaller than those of the other estimators. In particular, the estimation result for the *t*-tuple estimator was only 0.035471 bits per bit, about 4 % those with the row-direction and random-shuffle orderings.

4.5 Discussion

We have demonstrated that the results of the entropy estimation suite vary with the ordering of PUF responses. It is thus inappropriate to judge performance based on estimation results for PUF responses concatenated in readout order without specifying how the PUF responses are used in applications. These findings mean that the entropy estimation obtained by concatenating PUF responses in readout order leads to a vulnerable implementation. The results for our CIS PUF using the horizontal comparison provide a good example. There are stripes in the heat map of the PUF responses, as shown in Figure 5 (c), and there are dependencies between PUF responses in the same column. An attacker can predict PUF responses or narrow down the candidates by assuming that PUF-response bits in the same column are most likely the same. However, the entropy

estimation suite greatly overestimated min-entropy when the PUF responses were arranged in the readout order. If PUF keys for device authentication are generated from the PUF responses on the assumption that the estimation result can be trusted, an attacker can exploit the system by using these PUF keys.

Although the distribution of PUF responses '0' and '1' in the heat map for our SRAM PUF seems random and the inter-HD results were good, the estimation results indicate that there were at least dependencies between the PUF responses in each column. Therefore, a PUF designer needs to be careful when estimating PUF entropy.

In the above results, multiple manufacturing process variations affected the PUF responses, and the manufacturing process variation based on column circuits affected the min-entropy. The estimation results were minimal when we applied the ordering that emphasized the dependencies between PUF responses (i.e., column-direction ordering). There is a simplistic idea that this ordering can be used to estimate min-entropy, but the PUF designer should avoid jumping too quickly to this conclusion. Further discussions are needed.

5 Validity of Estimation Results

A PUF designer estimates min-entropy to prevent PUF cloning. The results of the experiments described in Section 4 cast doubt upon whether the SP800-90B entropy estimation suite correctly estimates min-entropy. One reason is that the *t*-tuple estimator computed a minimum estimate. Before explaining this reason, we first discuss the most common identification (ID) output from a PUF with entropy-loss sources. Next, we explain the reason for doubting the estimation results. Finally, we compare the estimates calculated from the appearance probability of the most common ID with those of the entropy estimation suite.

5.1 Most Common ID of PUFs with Entropy-Loss Sources

Let us consider a simple PUF model with an entropy-loss source and discuss the appearance probability of the most common ID output from it. We assume that the PUF model has N_{res} PUF sources and a digitizer. When the model generates a PUF response, one among the N_{res} PUF sources is selected, and '0' or '1' as the PUF response is output by the digitizer. The relationship between PUF responses and manufacturing process variations is given by

$$R_{p,r_r,r_c} = \begin{cases} 0 & (v_{p,r} + v_p^{\mathrm{digit}} \leq T) \\ 1 & (\text{otherwise}), \end{cases} \tag{9}$$

where $v_{p,r}$ is a manufacturing process variable of the r-th PUF source in the p-th device, v_p^{digit} is a manufacturing process variable of the digitizer (entropy-loss source) in the p-th device, and T is a constant threshold parameter. As in the PUF model in Equation 1, p ranges from 1 to N_{puf}, and r ranges from 1 to N_{res}. We assume normal distributions for $v_{p,r}$ and v_p^{digit}: $v_{p,r} \sim \mathcal{N}(\mu_v, \sigma_v^2)$ and $v_p^{\mathrm{digit}} \sim \mathcal{N}(\mu_{v^{\mathrm{digit}}}, \sigma_{v^{\mathrm{digit}}}^2)$. The manufacturing process variation of the digitizer is added as a threshold offset to each device and causes dependencies between PUF responses. This PUF model is a simpler version of our SRAM-PUF model (Equation 3) and our CIS-PUF models (Equations 6 and 8).

The PUF model with an entropy-loss source is not limited to memory-based PUFs, such as the SRAM PUF and CIS PUF. The previous study reported that the standard deviation of the offset-time distribution on the arbiter circuit (digitizer) in arbiter PUFs (APUFs) was 31.1 % that of delay-time-difference distribution on the 128-stage selector chain [1]. It may also apply to delay-based PUFs.

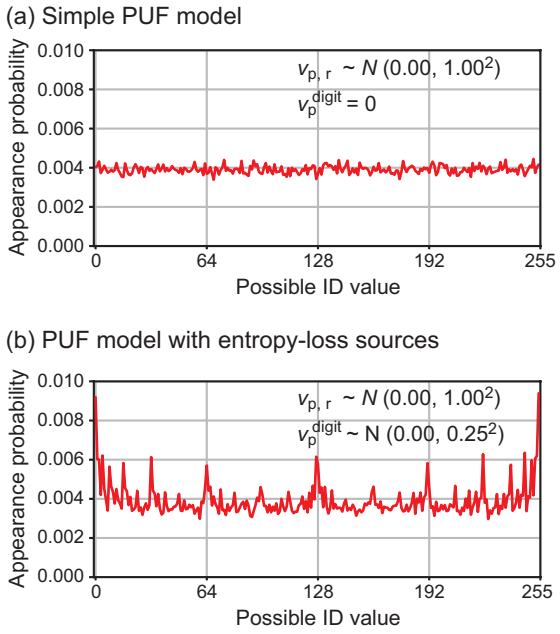

Figure 6: Appearance probabilities of output IDs

We created PUFs using numerical simulation by Equation 9 to determine the appearance probabilities of the output IDs, which are generated using all the PUF responses. As an example, we plot the appearance probabilities of the output IDs of a numerical PUF with eight PUF sources (i.e., $N_{res} = 8$) in Figure 6. Since each output ID is created from the 8-bit PUF responses, the ID ranges from 0 to 255. T was set to 0, and the mean and standard deviation of $v_{p,r}$ were set to 0.00 ($= \mu_v$) and 1.00 ($= \sigma_v$), respectively (i.e., $v_{p,r} \sim \mathcal{N}(0.00, 1.00^2)$). First, both the mean and standard deviation of v_p^{digit} were set to 0.00 (i.e., $v_p^{\text{digit}} = 0$). It means that the variation in the manufacturing of the digitizer was negligible. We generated 1,000,000 numerical PUFs and investigated the appearance probabilities of the output IDs. As shown in Figure 6 (a), all the IDs were output with equal probability, and the appearance probability was approximately 0.004 ($= \frac{1}{256}$). The entropy calculated from this probability was approximately 8 ($= -\log_2(0.004)$) bits. When normalized by the number of PUF-response bits, the entropy is 1 bit per bit.

Next, increasing the standard deviation of v_p^{digit} to 0.25 (i.e., $v_p^{\text{digit}} \sim \mathcal{N}(0.00, 0.25^2)$) maximized the appearance probability of the "0" IDs (all '0' bits) and "255" IDs (all '1' bits), as shown in Figure 6 (b). The appearance probability was approximately 0.0091, and the entropy calculated from it was 0.847 ($= \frac{-\log_2(0.0091)}{8}$) bits per bit. This bias in the appearance of IDs was due to the manufacturing process variable of the digitizer (entropy-loss source). The above appearance probability and entropy can be theoretically calculated as follows. The probability of the PUF-response bit being '0' at any offset (x^{offset}), which is caused by v_p^{digit}, is expressed as a cumulative distribution function (CDF):

$$\mathbf{cdf}(x^{\text{offset}}) = \int_{-\infty}^{x^{\text{offset}}} \frac{1}{\sqrt{2\pi\sigma_v^2}} \mathrm{e}^{-\frac{(x-\mu_v)^2}{2\sigma_v^2}} \, dx. \tag{10}$$

Since the number of PUF sources is N_{res}, the probability of all the PUF-response bits being '0' is the N_{res} power of $\mathbf{cdf}(x^{\text{offset}})$. The appearance probability of the "0" ID can

be calculated by integrating with respect to x^{offset}:

$$\mathbf{Pr}(X=0) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2_{v^{\text{digit}}}}} \mathrm{e}^{-\frac{\left(x^{\text{offset}}-\mu_{v^{\text{digit}}}\right)^2}{2\sigma^2_{v^{\text{digit}}}}} \cdot \mathbf{cdf}(x^{\text{offset}})^{N_{res}}\, dx^{\text{offset}}. \tag{11}$$

The appearance probability of the ID for which the PUF-response bits are all '1' is calculated in the same way. The min-entropy normalized by the number of PUF-response bits is calculated using

$$H_\infty = \frac{-\log_2 \max \mathbf{Pr}(X=x_i)}{N_{res}} = \frac{-\log_2 \mathbf{Pr}(X=0)}{N_{res}}. \tag{12}$$

As mentioned above, the min-entropy represents the worst-case scenario. For PUFs with entropy-loss sources, the worst-case scenario is when an attacker predicts the ID in which the PUF-response bits are all '0' or all '1'. The result of Equation 11 is the maximum probability that the attacker will succeed in predicting the ID.

If the two IDs in which the PUF-response bits are all '1' or all '0' are excluded due to lack of randomness, the next most common IDs are those in which one among the PUF-response bits is reversed. In the example shown in Figure 6 (b), these are the IDs "1", "2", "4", "127", "128", "251", "253", "254", and so on. The appearance probability can be calculated as follows.

$$\begin{aligned}\mathbf{Pr}(X=1) \quad = \quad & \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2_{v^{\text{digit}}}}} \mathrm{e}^{-\frac{\left(x^{\text{offset}}-\mu_{v^{\text{digit}}}\right)^2}{2\sigma^2_{v^{\text{digit}}}}} \\ & \cdot \mathbf{cdf}(x^{\text{offset}})^{N_{res}-1} \cdot (1-\mathbf{cdf}(x^{\text{offset}}))\, dx^{\text{offset}}.\end{aligned} \tag{13}$$

The result is 0.0591, which is approximately equal to the appearance probabilities in Figure 6 (b). The entropy normalized by the number of PUF-response bits is calculated using

$$H \approx \frac{-\log_2 \mathbf{Pr}(X=1)}{N_{res}}. \tag{14}$$

In the same way, the appearance probabilities of the other IDs can be calculated theoretically.

These results indicate that identical PUF-response bits are generated with high probability due to entropy-loss sources (i.e., variations in the manufacturing of circuits and transistors (except for PUF sources)). Specific IDs tend to be output with high probability. These features can be understood from the heat map of PUF responses in Figure 5 (c), in which the stripes mean that identical PUF-response bits appear with high probability in each column. In other words, an attacker can roughly guess the probability of a certain ID appearing even without knowing the ratio of the variations between the PUF sources and entropy-loss sources. The attacker can thus predict an ID, narrow down the candidate IDs, or find a device with a vulnerable ID. The probability of a successful attack is based on theoretical estimates calculated from the above equations. Therefore, the result of the entropy estimation suite should match the theoretical estimate.

5.2 Issue with *t*-tuple estimator

We now explain why the *t*-tuple estimation did not match the theoretical estimate. The entropy estimation suite requires splitting the output ID into multiple sample values. Since we set the symbol size to 1 bit, the output IDs were split into 1-bit symbols. It means

Table 5: Example probabilities of a 2-bit value (ID)

2-bit value	Probability
0	0.4
1	0.1
2	0.1
3	0.4

that the t-tuple estimator measured tuples consisting of sample values across the IDs. As a result, it overestimated min-entropy.

Let us consider a simple example of a 2-bit value (ID) with the probabilities shown in Table 5. We suppose the dataset is

$$S = (0\ \ 3\ \ 0\ \ 0\ \ 1\ \ 3\ \ 3\ \ 0\ \ 2\ \ 3), \tag{15}$$

where the number of samples (L) is 10. The MCV estimator measures the proportion of the MCV (p_{mcv}) in the dataset; $p_{\mathrm{mcv}} = 0.4$ is derived from the number of "0"s and "3"s. The t-tuple estimator is an extension of the MCV estimator and first finds the largest t such that the number of occurrences of the most common t-tuple in the dataset is at least 35. The number of occurrences of the most common i-tuple ($Q[i]$) is measured for $i = 1, 2, ..., t$. Next, it computes an estimate on the maximum individual sample value probability:

$$P_{max}[i] = \left(\frac{Q[i]}{L - i + 1}\right)^{\frac{1}{i}} \qquad (i = 1, 2, ..., t). \tag{16}$$

Next, the maximum probability is selected (i.e., $p_{t-\mathrm{tuple}} = \max(P_{max}[1], ..., P_{max}[t])$). If we assume that the cutoff in this example is 3 instead of 35, the largest t is 1, and $Q[1] = 4$ is derived from the number of "0"s and "3"s. The $p_{t-\mathrm{tuple}}$ is 0.4, which is equal to the MCV-estimator result (i.e., $p_{t-\mathrm{tuple}} = p_{\mathrm{mcv}}$). The entropy estimated from this result is 0.6610 ($= \frac{-\log_2(0.4)}{2}$) bits per bit.

If the 2-bit values are split into 1-bit symbols, the dataset is represented as

$$S = (0\,0\ \ 1\,1\ \ 0\,0\ \ 0\,0\ \ 0\,1\ \ 1\,1\ \ 1\,1\ \ 0\,0\ \ 1\,0\ \ 1\,1), \tag{17}$$

where the number of samples (L) is doubled to 20. $Q[1] = 10$ is the number of '0's and '1's, $Q[2] = 6$ is derived from the number of tuples "00" and "11", and $Q[3] = 3$ is derived from the number of tuples "000", "001", and "011". $P_{max}[1]$, $P_{max}[2]$, and $P_{max}[3]$ are 0.5, 0.5620, and 0.5503, respectively, and $p_{t-\mathrm{tuple}}$ is 0.5620. The entropy estimated from this result is 0.8314 ($= -\log_2(0.5620)$) bits per bit, which is larger than that calculated using 2-bit symbols. The t-tuple estimator thus overestimates entropy.

The reason for this overestimation is counting tuples across 2-bit values. When measuring the number of occurrences of the most common 2-tuples ($Q[2]$), the occurrence ratios of "00", "01", "10", and "11" should be the same as in Table 5. However, tuples extending across the 2-bit values are also counted. Since the ratios of "00"s, "01"s, "10"s, and "11"s across the 2-bit values are the same, their addition changes the occurrence ratio of "00", "01", "10", and "11" to $0.325 : 0.175 : 0.175 : 0.325$, and the maximum appearance probability is reduced. The t-tuple estimator thus overestimates entropy. This mechanism is the same regardless of tuple size.

5.3 Comparison

We investigated how much the theoretical estimates calculated using Equations 12 and 14 differ from those of the entropy estimation suite. We created 12 numerical PUFs

Table 6: Parameter settings and theoretical estimates of numerical PUFs

PUF	$v_{p,r}$		v_p^{digit}		N_{res}	Entropy	
	μ_v	σ_v	$\mu_{v^{\text{digit}}}$	$\sigma_{v^{\text{digit}}}$		(Equ. 12)	(Equ. 14)
A	0.00	1.00	0.00	0.10	128	0.6680	0.6788
B	0.00	1.00	0.00	0.10	256	0.5231	0.5312
C	0.00	1.00	0.00	0.10	512	0.3803	0.3859
D	0.00	1.00	0.00	0.10	1024	0.2590	0.2626
E	0.00	1.00	0.00	0.20	128	0.3844	0.4066
F	0.00	1.00	0.00	0.20	256	0.2616	0.2761
G	0.00	1.00	0.00	0.20	512	0.1688	0.1779
H	0.00	1.00	0.00	0.20	1024	0.1046	0.1100
I	0.00	1.00	0.00	0.30	128	0.2476	0.2777
J	0.00	1.00	0.00	0.30	256	0.1584	0.1770
K	0.00	1.00	0.00	0.30	512	0.0974	0.1086
L	0.00	1.00	0.00	0.30	1024	0.0582	0.0647

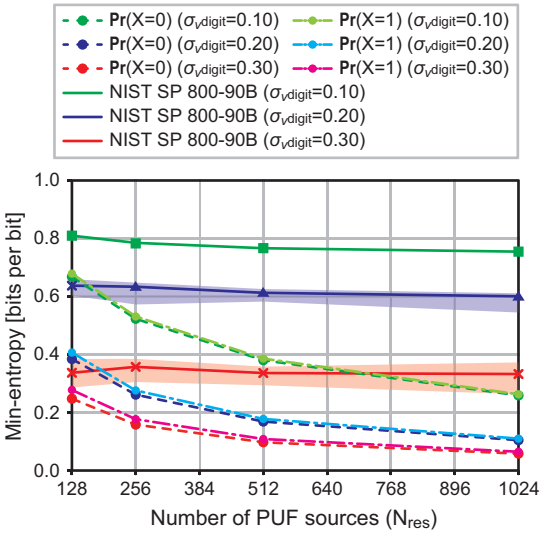

Figure 7: Relationship between estimated min-entropy and number of PUF sources

using Equation 9 for this purpose. Their parameter settings and theoretical estimates are summarized in Table 6. As in the previous section, T was set to 0, and the mean and standard deviation of $v_{p,r}$ were set to 0.00 ($= \mu_v$) and 1.00 ($= \sigma_v$), respectively (i.e., $v_{p,r} \sim \mathcal{N}(0.00, 1.00^2)$). The mean of v_p^{digit} was set to 0.00 ($= \mu_{v^{\text{digit}}}$), and the standard deviation of v_p^{digit} was set to 0.10, 0.20, or 0.30 ($= \sigma_{v^{\text{digit}}}$). The number of PUF sources (N_{res}) was set to 128, 256, 512, or 1024. The number of PUFs (N_{puf}) was set so that the number of samples for the entropy estimation suite is 1G bits. We concatenated the PUF responses and estimated the min-entropy of each 1G-bit dataset using the entropy estimation suite of NIST SP 800-90B. The number of samples required by the entropy estimation suite is 1M, but we increased the number to 1G because of the wide variation in the estimation results. Since the estimation results still varied widely, we estimated min-entropy under each condition ten times.

Figure 7 shows the relationship between estimated min-entropy and the number of PUF sources. The solid, dashed, and dash-dotted lines represent the min-entropy estimated using the entropy estimation suite, Equation 12, and Equation 14, respectively. When

the ratio $\frac{\sigma_{v^{\mathrm{digit}}}}{\sigma_v}$ was 0.20 or 0.30, the min-entropy estimated using the entropy estimation suite varied and was sometimes lower. The reason for the lower estimated entropy is that IDs without randomness, such as those in which the PUF-response bits were all '1' or all '0', are unfortunately contained in the dataset. The mean of the ten times was relatively large. We also calculated the theoretical estimates by excluding the IDs in which the PUF-response bits were all '1' or all '0'. There was no clear difference between the theoretical estimates calculated using Equations 12 and 14. On the other hand, there was a clear difference between the theoretical estimates and the estimates calculated using the entropy estimation suite. The NIST entropy estimation suite greatly overestimated min-entropy. The overestimation became more noticeable as the number of PUF sources was increased. The theoretical estimates decreased with an increase in the number of PUF sources. These results suggest that the IDs that include identical PUF-response bits appear with high probability, so there is not much benefit from increasing PUF responses. In contrast, the min-entropy estimated with the entropy estimation suite did not change significantly with an increase in the number of PUF sources. It indicates that the entropy estimation suite fails to detect occurrence bias in PUF responses.

The t-tuple and compression estimators competed to compute a minimum estimate. Which estimate was minimum was related to the ratio $\frac{\sigma_{v^{\mathrm{digit}}}}{\sigma_v}$. When $\frac{\sigma_{v^{\mathrm{digit}}}}{\sigma_v}$ was 0.2, their estimates were close. Sometimes the t-tuple estimator computed the minimum estimate, and sometimes the compression estimator did. When $\frac{\sigma_{v^{\mathrm{digit}}}}{\sigma_v}$ was greater than 0.2, the t-tuple estimator computed the minimum estimate; otherwise, the compression estimator computed the minimum estimate.

5.4 Discussion

As explained above, an attacker is more likely to predict PUF responses when PUF entropy is lower due to the entropy-loss sources. We demonstrated that the entropy estimation suite of NIST SP 800-90B overestimates PUF min-entropy. The comparison results suggest that the simplistic idea of using an ordering that emphasizes the dependencies between PUF responses (i.e., column-direction ordering) leads to a vulnerable implementation. Since a system design that relies on only the estimation result of the entropy estimation suite has the potential to lead to vulnerability, a PUF designer first needs to understand what causes the decrease in entropy and how much it may decrease. The PUF designer must then consider how to use PUF responses to ensure a given security level or how to remove dependencies between PUF responses as well as consider whether the entropy estimation suite can reliably estimate the min-entropy.

The estimation result with the column-direction ordering of our SRAM PUF was 0.728256, which was close to that of the parameter setting "D" in Table 6. The estimation condition that there were 1024 PUF sources for an entropy-loss source was the same, and the tendency that the compression estimator computed a minimal estimate was also the same. We guess that the ratio of the standard deviation of $v^{\mathrm{column}}_{p,r_c}$ to that of v_{p,r_r,r_c} in Equation 3 is approximately 0.1. Besides, since the estimation result with the row-direction ordering was lower, we guess that there are also dependencies between the PUF responses in each row, and the row and column entropy-loss sources are intertwined. However, since we used SRAM standard cells, there was, unfortunately, no way to investigate the variations of the PUF sources and entropy-loss sources using Monte Carlo simulation, etc. Since the theoretical estimate is about 0.26, lower than the estimation results, we certainly need to consider the effect of the entropy-loss sources more carefully.

Finally, we discuss the validity of the estimation results for our CIS PUF using the horizontal comparison. We measured the digitized values through the ADC and calculated the means and standard deviations of $\Delta v_{p,r_r,r_c}$ and $\Delta v^{\mathrm{column}}_{p,r_c}$ in Equation 8. We calculated the mean and standard deviation of $\Delta v_{p,r_r,r_c}$ from all the digitized differential values. Next, we calculated the mean of the digitized differential values for each column and took

it as the column mean. We calculated the mean and standard deviation of $\Delta v^{\text{column}}_{p,r_c}$ from all the column means. Besides, we calculated the mean of the digitized differential values for each row and took it as the row mean. We calculated the mean and standard deviation of $\Delta v^{\text{row}}_{p,r_r}$ from all the row means for reference.

The mean and standard deviation of $\Delta v_{p,r_r,r_c}$ were -0.995 and 168.088, respectively. The mean and standard deviation of $\Delta v^{\text{column}}_{p,r_c}$ were -0.995 and 99.717, respectively. The mean and standard deviation of $\Delta v^{\text{row}}_{p,r_r}$ were -0.995 and 5.746, respectively. The ratio of $\Delta v^{\text{column}}_{p,r_c}$ to $\Delta v_{p,r_r,r_c}$ in standard deviation was about 0.6, which is larger than 0.2. It met the condition the t-tuple estimator computes the minimum estimate. The standard deviation of $\Delta v^{\text{row}}_{p,r_r}$ was much smaller than that of $\Delta v_{p,r_r,r_c}$, which is negligible. The theoretical estimate calculated using Equation 12 was 0.326, which is close to the estimation result in Table 4. However, we found that the estimation result varied widely and that the reason for this small estimation result was that the PUF responses in some of the columns had insufficient randomness. We created numerical CIS PUFs using the above parameters and the estimation conditions described in Section 4.1 and estimated min-entropy 20 times using the entropy estimation suite. The estimated entropy was computed by the t-tuple estimator, which ranged from 0.0352 to 0.0928. The estimated entropy was low when the dataset contained PUF responses in which a single column consisted of all '1' or all '0'. When the number of devices for the entropy estimation was increased to 2000 (i.e., the total number of PUF-response bits was 1G), the estimated entropy settled to approximately 0.045. There was a difference between the theoretical estimates and the estimates calculated using the entropy estimation suite. We confirmed the effect of the entropy-loss sources (clip transistors) on min-entropy and the validity of the estimated min-entropy through experiments.

6 Conclusion

In previous studies, PUF responses from several chips have been concatenated in readout order to meet the criteria of SP 800-90B. And PUF performance has been based on the entropy estimated using the entropy estimation suite of NIST SP 800-90B. In this paper, we pointed out two problems with this approach. One is that the estimation results vary widely with the ordering PUF responses. The other is that the entropy estimation suite can overestimate PUF min-entropy. Both problems are due to entropy-loss sources (i.e., the variations in the manufacturing of circuits and transistors (except for PUF sources)).

To investigate the variation in the estimated entropy with the ordering of PUF responses, we applied three orderings to the PUF responses of our SRAM PUF and our CIS PUF: row-direction ordering, column-direction ordering, and random-shuffle ordering. We demonstrated that the results estimated using the entropy estimation suite varied with the ordering. The estimated entropy was high for the readout order. Moreover, changing the ordering to one that emphasized the entropy-loss sources resulted in a lower estimated entropy. It is thus inappropriate to judge performance based on a single result estimated using concatenated PUF responses.

To investigate the overestimation problem by using numerical simulation, we demonstrated that the SP 800-90B entropy estimation suite overestimates the min-entropy. Even if an ordering minimizing entropy is found, relying only on the results estimated using the entropy estimation suite leads to a vulnerable implementation. It is thus needed to understand what causes the decrease in entropy and how much it may decrease. A PUF designer should consider the effect of entropy-loss sources on PUF min-entropy theoretically. Also, the PUF designer should consider how to use PUF responses to ensure a given security level or how to remove the dependencies due to the entropy-loss sources.

Acknowledgments

This work is based on results obtained from a project, JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] Mitsuru Shiozaki, Yohei Hori, Tatsuya Oyama, Masayoshi Shirahata, and Takeshi Fujino. Cause analysis method of entropy loss in physically unclonable functions. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2020.

[2] Blaise Gassend, Dwaine E. Clarke, Marten van Dijk, and Srinivas Devadas. Silicon physical random functions. In *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, pages 148–160, 2002.

[3] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[4] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A McKay, Mary L Baish, and Mike Boyle. Recommendation for the entropy sources used for random bit generation. *NIST Special Publication*, 800:90B, 2018.

[5] Pim Tuyls and Boris Škorić. Secret key generation from classical physics: Physical uncloneable functions. In *AmIware Hardware Technology Drivers of Ambient Intelligence*, pages 421–447. Springer, 2006.

[6] Pim Tuyls, Boris Skoric, S. Stallinga, Anton H. M. Akkermans, and W. Ophey. Information-theoretic security analysis of physical uncloneable functions. In *Financial Cryptography and Data Security, 9th International Conference, FC 2005, Roseau, The Commonwealth of Dominica, February 28 - March 3, 2005, Revised Papers*, pages 141–155, 2005.

[7] Tanya Ignatenko, Geert Jan Schrijen, Boris Skoric, Pim Tuyls, and Frans M. J. Willems. Estimating the secrecy-rate of physical unclonable functions with the context-tree weighting method. In *Proceedings 2006 IEEE International Symposium on Information Theory, ISIT 2006, The Westin Seattle, Seattle, Washington, USA, July 9-14, 2006*, pages 499–503, 2006.

[8] Boris Škorić, Stefan Maubach, Tom Kevenaar, and Pim Tuyls. Information-theoretic analysis of capacitive physical unclonable functions. *Journal of Applied physics*, 100(2):024902, 2006.

[9] Jorge Guajardo, Sandeep S. Kumar, Geert Jan Schrijen, and Pim Tuyls. Physical unclonable functions, fpgas and public-key crypto for IP protection. In *FPL 2007, International Conference on Field Programmable Logic and Applications, Amsterdam, The Netherlands, 27-29 August 2007*, pages 189–195, 2007.

[10] Daisuke Suzuki and Koichi Shimizu. The glitch PUF: A new delay-puf architecture exploiting glitch shapes. In *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, pages 366–382, 2010.

[11] Geert Jan Schrijen and Vincent van der Leest. Comparative analysis of SRAM memories used as PUF primitives. In *2012 Design, Automation & Test in Europe Conference & Exhibition, DATE 2012, Dresden, Germany, March 12-16, 2012*, pages 1319–1324, 2012.

[12] Mudit Bhargava and Ken Mai. An efficient reliable puf-based cryptographic key generator in 65nm CMOS. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2014, Dresden, Germany, March 24-28, 2014*, pages 1–6, 2014.

[13] Roel Maes. *Physically Unclonable Functions - Constructions, Properties and Applications*. Springer, 2013.

[14] G. Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th Design Automation Conference, DAC 2007, San Diego, CA, USA, June 4-8, 2007*, pages 9–14, 2007.

[15] Meng-Day Mandel Yu and Srinivas Devadas. Recombination of physical unclonable functions. 2010.

[16] Zouha Cherif, Jean-Luc Danger, Sylvain Guilley, and Lilian Bossuet. An easy-to-design PUF based on a single oscillator: The loop PUF. In *15th Euromicro Conference on Digital System Design, DSD 2012, Cesme, Izmir, Turkey, September 5-8, 2012*, pages 156–162. IEEE Computer Society, 2012.

[17] Alexander Schaub, Jean-Luc Danger, Sylvain Guilley, and Olivier Rioul. An improved analysis of reliability and entropy for delay pufs. In *21st Euromicro Conference on Digital System Design, DSD 2018, Prague, Czech Republic, August 29-31, 2018*, pages 553–560, 2018.

[18] Patrick Koeberl, Jiangtao Li, Anand Rajan, and Wei Wu. Entropy loss in puf-based key generation schemes: The repetition code pitfall. In *2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014, Arlington, VA, USA, May 6-7, 2014*, pages 44–49. IEEE Computer Society, 2014.

[19] Roel Maes, Vincent van der Leest, Erik van der Sluis, and Frans M. J. Willems. Secure key generation from biased pufs. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 517–534. Springer, 2015.

[20] Rei Ueno, Kohei Kazumori, and Naofumi Homma. Rejection sampling schemes for extracting uniform distribution from biased pufs. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):86–128, 2020.

[21] Chongyan Gu, Weiqiang Liu, Neil Hanley, Robert Hesselbarth, and Máire O'Neill. A theoretical model to link uniqueness and min-entropy for PUF evaluations. *IEEE Trans. Computers*, 68(2):287–293, 2019.

[22] Sarah Q Xu, Wing-kei Yu, G Edward Suh, and Edwin C Kan. Understanding sources of variations in flash memory for physical unclonable functions. In *2014 IEEE 6th International Memory Workshop (IMW)*, pages 1–4. IEEE, 2014.

[23] Roel Maes, Anthony Van Herrewege, and Ingrid Verbauwhede. PUFKY: A fully functional puf-based cryptographic key generator. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 302–319. Springer, 2012.

[24] Jorge Guajardo, Sandeep S. Kumar, Geert Jan Schrijen, and Pim Tuyls. FPGA intrinsic pufs and their use for IP protection. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 63–80. Springer, 2007.

[25] Daniel E. Holcomb, Wayne P. Burleson, and Kevin Fu. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Trans. Computers*, 58(9):1198–1210, 2009.

[26] Shunsuke Okura, Yuki Nakura, Masayoshi Shirahata, Mitsuru Shiozaki, Takaya Kubota, Kenichiro Ishikawa, Isao Takayanagi, and Takashi Fujino. A proposal of puf utilizing pixel variations in the cmos image sensor. In *Proceedings of International Image Sensor Workshop IISW 2017*, 2017.

[27] Roel Maes. An accurate probabilistic reliability model for silicon pufs. In *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, pages 73–89, 2013.

[28] Patrick Hagerty and Tom Draper. Entropy bounds and statistical tests. In *Proceedings of the NIST Random Bit Generation Workshop, Gaithersburg, MD, USA*, pages 5–6, 2012.

[29] Ueli M. Maurer. A universal statistical test for random bit generators. *J. Cryptology*, 5(2):89–105, 1992.

[30] David Salomon. *Data compression - The Complete Reference, 4th Edition*. Springer, 2007.

[31] Mitsuru Shiozaki and Takeshi Fujino. Simple electromagnetic analysis attacks based on geometric leak on an ASIC implementation of ring-oscillator PUF. In Chip-Hong Chang, Ulrich Rührmair, Daniel E. Holcomb, and Patrick Schaumont, editors, *Proceedings of the 3rd ACM Workshop on Attacks and Solutions in Hardware Security Workshop, ASHES@CCS 2019, London, UK, November 15, 2019*, pages 13–21. ACM, 2019.

[32] *SP 800-90B Entropy Assessment (v1.0)*, 2019 (22 May 2019). https://github.com/usnistgov/SP800-90B_EntropyAssessment.