PERFORMANCE EVALUATION OF PRINCE BASED GLITCH PUF WITH SEVERAL SELECTION PARTS

Yusuke Nozaki and Masaya Yoshikawa

Department of Information Engineering, Meijo University, Nagoya, Japan

ABSTRACT

To enhance the internet of things (IoT) security, lightweight ciphers and physically unclonable functions (PUFs) have attracted attention. Unlike standard encryption AES, lightweight ciphers can be implemented on embedded devices with strict constraints used in IoT. The PUF is a technology extracting manufacturing variations in LSI as device's unique ID. Since manufacturing variations cannot be cloned physically, the generated ID using PUF can be used for device's authentication. Actually, a method combining lightweight cipher (PRINCE) and PUF (glitch PUF) called PRINCE based glitch PUF has been proposed in recent years. However, PRINCE based glitch PUF was not optimized for PUF performances. Therefore, this study evaluates the detailed PUF performance of PRINCE based glitch PUF with changing the parameters. Experimental results using FPGAs clarified that PRINCE based glitch PUF had the relationship of trade-off between steadiness and uniqueness depending on the selected part as glitch generator.

KEYWORDS

Hardware Security, Physically Unclonable Function, Glitch PUF, PRINCE, Lightweight Cipher

1. INTRODUCTION

In internet of things (IoT), ensuring the security of connected IoT devices is the important issue.To enhance the IoT security, two types of security technologies have been studied for the concealment of communicated data and the device's authentication. One is lightweight ciphers [1]–[4], and the other is physically unclonable functions (PUFs) [5]–[10]. Unlike advanced encryption standard (AES) which is widely used, lightweight ciphers can be implemented on embedded devices with strict constraints (e. g. circuit area, power consumption, latency, and soon) used in IoT. Several lightweight ciphers, including a small-area cipher SIMECK [2], a low power cipher Midori [3], a low-latency cipher PRINCE [4], and so on, have been proposed. For the PUF, it is a technology extracting manufacturing variations in large scale integration (LSI) as IoT device's unique ID. Since manufacturing variations cannot be cloned physically, the generated ID using PUF circuit can be used for the device's authentication. Several PUFs, including SRAM PUF [5], arbiter PUF [6], ring oscillator PUF [7], glitch PUF [8], and so on, have been proposed. In particular, the glitch PUF has a good performance and the high security against modelling attacks [8][10]. Actually, for the IoT security, a method combining lightweight cipher PRINCE [4] and the glitch PUF [9] called PRINCE based glitch PUF has been proposed in recent years [11]. However, in previous study [11], the PRINCE based glitch PUF was not optimized for PUF performances. Therefore, this study evaluates the detailed PUF performance of PRINCE based glitch PUF with changing the parameters. Experiments using field

Dhinaharan Nagamalai et al. (Eds): AIAP, ICCIoT, CNSA, SIGML, IT, ICBB, DMDB - 2020
pp. 29-39, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100303

programmable gate arrays (FPGAs) verify the PUF performance of the PRINCE based glitch PUF with several parameters.

Our contributions are summarized as follows:

- This study evaluates PUF performances of PRINCE based glitch PUF by changing a region used as a glitch generator.

- Experiments using FPGAs showed that PRINCE based glitch PUF had the relationship of trade-off between steadiness and uniqueness depending on the selected part as glitch generator.

The remainder of this paper is organized as follows. Section 2 introduces the outline of PUF and PRINCE cipher based glitch PUF used in this study. The methodology of evaluation for PRINCE based glitch PUF with changing parameters is presented in section 3. Section 4 describes experimental results using FPGAs. Section 5 concludes this paper.

2. PRELIMINARIES

2.1. Physically Unclonable Function

PUF is used for the device's authentication. A challenge and response authentication is a typical authentication method. This method uses many challenge and response pairs (CRPs). In advance, CRPs are registered with the database. In authentication, an unused challenge is sent to the target device (PUF circuit), an obtained response is sent to the database, and CRPs are compared each other.

Several PUFs, including SRAM PUF [5], arbiter PUF [6], ring oscillator PUF [7], glitch PUF [8], and so on, have been proposed. The SRAM PUF has a constraint of operating timing because it uses the initial state of SRAM cell as a PUF response. Unlike SRAM PUF, the delay based PUF such as arbiter PUF, ring oscillator PUF, and glitch PUF have no such constraint. However, it has been reported that delay PUFs are vulnerable to modelling attacks in [12]–[14]. The modelling attacks construct the mathematical clone to predict PUF response. Hence, ensuring the resistance against modelling attacks is important issues in the field of PUF. In fact, the glitch PUF has the strong resistance against modelling attacks [8][10]; therefore, this study targets glitch PUF.

2.2. Glitch Physically Unclonable Function

The glitch PUF uses the glitch variation due to manufacturing variations. At this time, the glitch is waveforms in unstable term of output signals due to the difference of signal propagation delays in a combinational circuit. Since the difference of signal propagation delays depends on the wiring length and so on, the glitch is different in each device. Figure 1 shows the outline of glitch PUF. As shown in Figure 1, a challenge is provided to a delay combinational circuit called glitch generator. At this time, AES's S-box circuit is typically used as glitch generator. Next, output signals are obtained by a sampling circuit. Finally, obtained signals are converted to response bits by glitch convertor. In this traditional glitch PUF, circuit area of a sampling circuit is large; hence, a simplified glitch PUF called second glitch PUF has been also proposed. The second glitch PUF simplifies the sampling circuit and glitch convertor. Specifically, in the second glitch PUF, the sampling circuit is eliminated, and the output of glitch generator is connected to toggle flip-flops (TFFs) directly. The TFF reverses an input signal at rising signal. Thus, response bits are generated by the number of even-odd number of the output from glitch generator. In addition, for IoT system, an improved glitch PUF combining a lightweight cipher PRICNE and glitch PUF called PRINCE based glitch PUF has been proposed [11].

The PRINCE based glitch PUF uses a partial part of unrolled PRINCE cipher as a glitch generator. At this time, PRINCE is a very low-latency cipher, and it can be implemented smallarea 100 times and low-latency 2 times more than AES circuit in unrolled architecture [15]. The unrolled architecture is a method of implementation realizing all circuits by combinational circuits. In the unrolled cipher, the glitch waveforms generally increase; therefore, this architecture is suitable for extracting glitch variations for PUF circuit. Figure 2 shows the outline of PRINCE based glitch PUF. Here PRINCE cipher consists of processing of 12 rounds [4]. Each processing performs a constant value addition (0th round and 11th round), processing by round function R (from 1st to 5th round), a middle processing, and processing by inverse round function R-1 (from 6th to 11th round). In round function (R and R^{-1}), a non-linear processing S by S-box tables, a linear processing M' by matrix operation, and SR by permutation are performed. In the previous study [11], the PRICNE based glitch PUF uses a part from input to 1st round's output as a glitch generator, as shown in Figure 2. However, other cases using different parts as a glitch generator have not been evaluated.

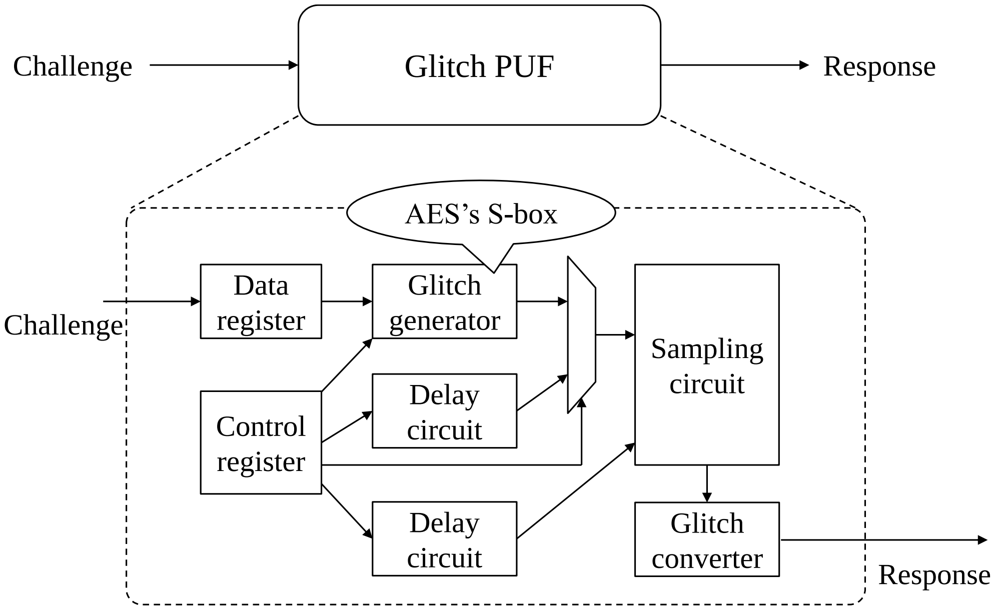

Figure 1. Glitch PUF [8]

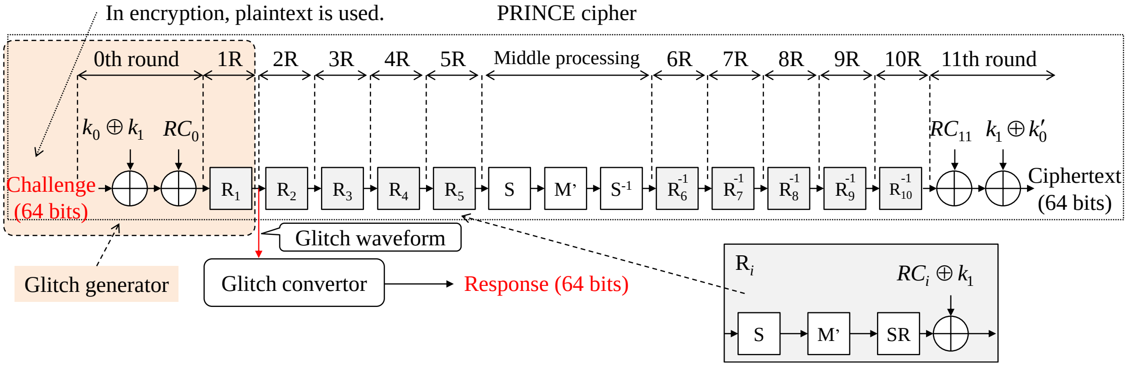

Figure 2. PRINCE based Glitch PUF [11]

3. PROPOSED METHOD

In the previous study [11], PRINCE based glitch PUF was only evaluated when it used 1st round's output as glitch generator. Hence, this study evaluates the other cases for PRINCE based glitch PUF. Figure 3 shows the outline of the evaluation method. As shown in Figure 3, this study uses 4 types of regions (a), (b), (c), and (d) as a glitch generator. For each region, first, selection part (a) uses a region from input to S-box's output in 1st round as a glitch generator. Next, selection part (b), which is similar to previous study [11], utilizes a region from input to 1st round as a glitch generator. Then, selection part (c) uses a region from input to S-box's output in 2nd round as a glitch generator. Finally, selection part (d) uses a region from input to M function's output in 2nd round as a glitch generator.

Each output of selection part is connected to a glitch converter. At this time, a TFF is used as a glitch converter. The TFF converts the number of rising signals in glitch waveforms, which is generated from each selection part of PRINCE cipher, to a 0/1 PUF response. Specifically, when the number of rising signals is the even-number, a response is zero; otherwise, the response is one. Since an output of glitch generator is 64 bits, 64 TFFs are used as the glitch generator. Hence, by 64 TFFs, a 64-bit PUF response is generated, and the PUF performance of generated response is evaluated in experiments (section 4).

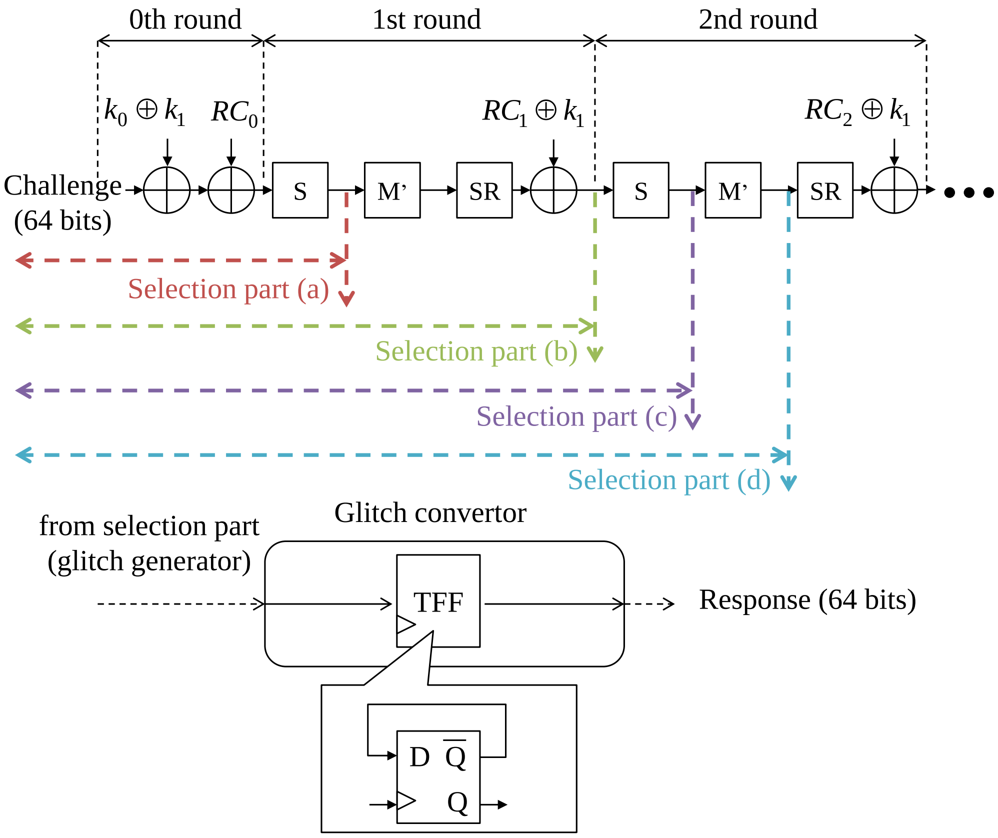

Figure 3. Proposed evaluation method

4. EXPERIMENTS

4.1. Experimental Environment

In experiments, 4 types of PRINCE based glitch PUFs were implemented into a Xilinx FPGA Virtex-5 XC5VLX30 on SASEBO-GII board. For the implementation, PRINCE based glitch PUF was designed by using Verilog hardware description language (HDL) and Xilinx ISE Design Suite 14.7 as an implementation tool. Figure 4 shows the evaluation system. Challenges were randomly generated and they were set to SASEBO-GII board, PRINCE based glitch PUF was operated, and the outputs (responses) of PUF were returned to the laptop PC, as shown in Figure 4.

For the PUF performance evaluation, experiments used typical PUF performance indicators: randomness, steadiness, diffuseness, and uniqueness [16]. To calculate those indices, ID's Hamming weight (HW), same challenge intra-Hamming distance (SC Intra-HD), different challenge intra-HD (DC Intra-HD), and same challenge inter-HD (SC Inter-HD) were used.

Figure 5 shows the evaluation method using SC Intra-HD, DC Intra-HD, and SC Inter-HD. First, for the evaluation of randomness, when ID's HW approaches half of ID length L, it means that response bit of 0/1 is generated uniformly; therefore, randomness is high. Then, SC Intra-HD is HD between PUF IDs generated T times in same device against same challenge (see Figure 5 (i)). When SC Intra-HD approaches 0, it means that same PUF IDs are generated against same input; therefore, steadiness is high. Next, DC Intra-HD is HD between K types of IDs in same device against different challenges (see Figure 5 (ii)). When DC Intra-HD approaches L/2, it means that different IDs are generated against different inputs; therefore, diffuseness is high. Finally, SC Inter-HD is HD between IDs in N types of different devices against same challenge (see Figure 5 (iii)). When SC Inter-HD approaches L/2, it means that ID is different between devices; therefore, uniqueness is high. In experiments, parameters L, K, T, and N were set to 128, 128, 100, and 3, that is, 128 × 128 × 100 × 3 = 4,915,200 bits of response were acquired in one type of PUF.

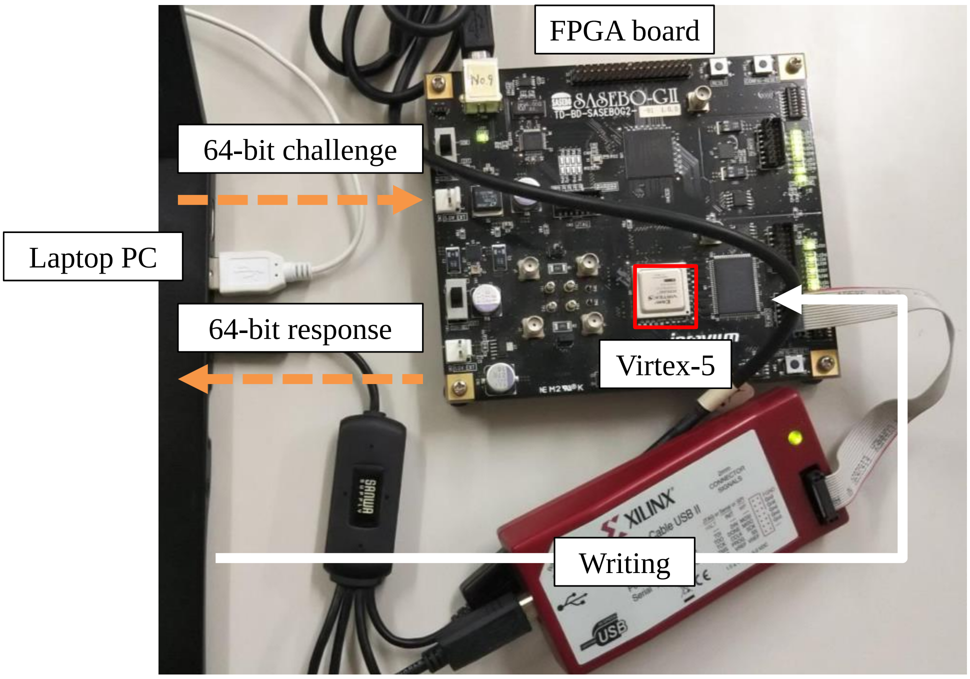

Figure 4. Evaluation system

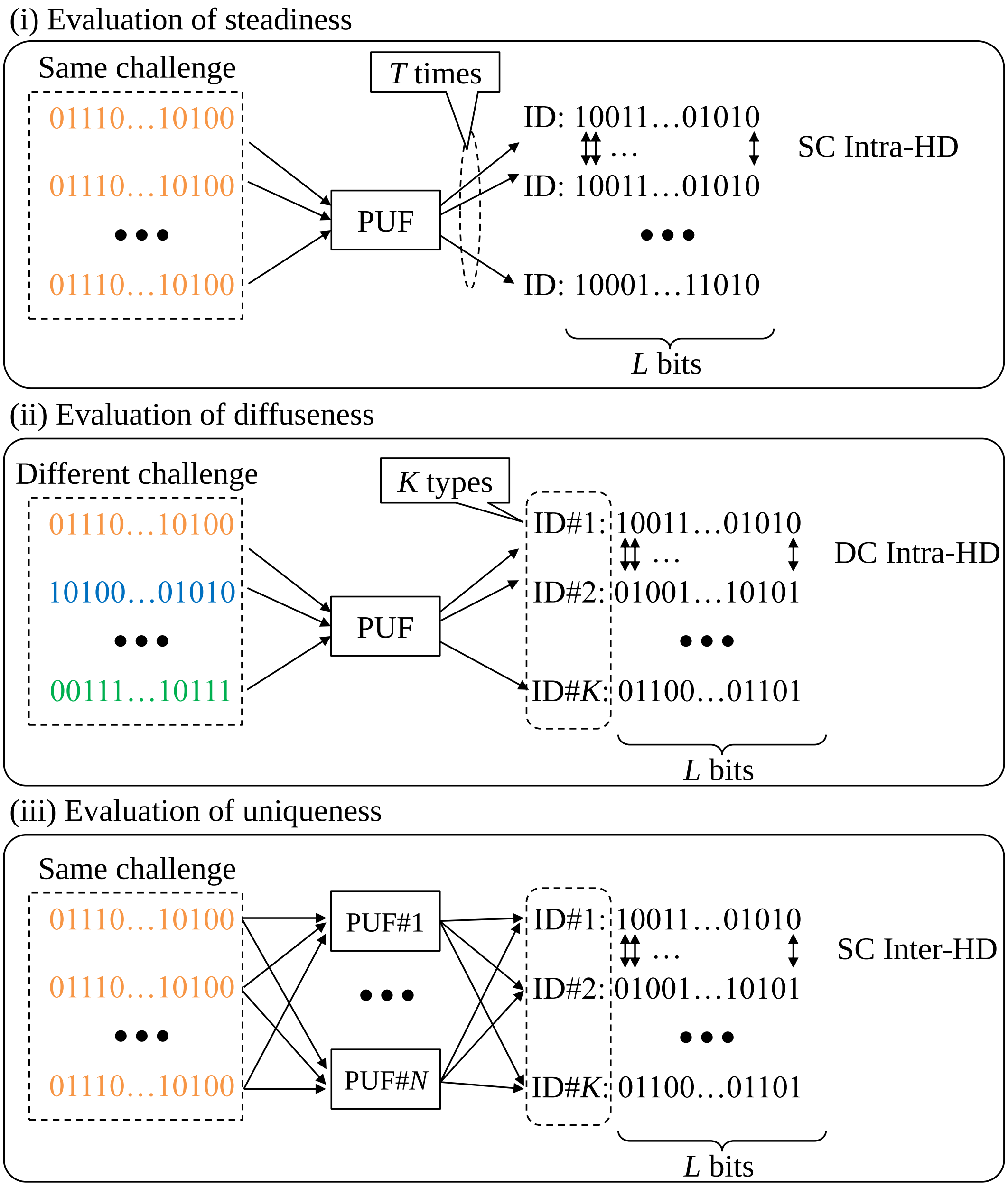

Figure 5. Evaluation using PUF performance indicators

4.2. Experimental Result

Figures 6, 7, 8, and 9 show the experimental results of randomness, diffuseness, steadiness, and uniqueness. In those figures, the vertical axis represents the frequency of ID's HW or HD between IDs and the horizontal axis represents ID's HW or HD between IDs (SC Intra-HD, DC Intra-HD, and SC Inter-HD), respectively. First, as shown in figures 6 and 7, high randomness and high diffuseness were confirmed in all results since each metrics (ID's HW and DC Intra HD) approached 64 which was half of ID length. In addition, the difference was not observed with different selection parts ((a), (b), (c), and (d)).

On the other hand, the differences of steadiness and uniqueness with different selection parts ((a), (b), (c), and (d)) were observed in figures 8 and 9. For the steadiness, selection part (a) had the highest steadiness since the mean of SC Intra-HD was the lowest among those of other parts (see

Figure 8). Thus, selection part (a) is suitable for application of secret key generation in cryptographic circuit requires high steadiness. In contrast, in selection part (d), steadiness became worse drastically. This is presumably because glitch noise becomes large due to combinational circuit complexity.

Next, for the uniqueness, selection part (d) had the highest uniqueness among those of other selection parts since the mean of SC Inter-HD was closer to the half of ID length, as shown in Figure 9. At this time, selection part (a) had the lowest uniqueness. Hence, even if the selection part (a) had the highest steadiness (see Figure 8), it had the worst uniqueness (see Figure 9). This is presumably because glitch variation cannot be extract due to combinational circuit simplicity in selection part (a). Thus, PRINCE based glitch PUF has the relationship of trade-off between steadiness and uniqueness depending on the selected part as glitch generator. Finally, from figures 8 and 9, there was no difference between selection parts (b) and (c) in steadiness and uniqueness. Therefore, S-box circuit of PRINCE does not affect PUF performances. This is presumably because S-box is implemented by a very simple table method in this study unlike AES's S-box such as composite field [8].

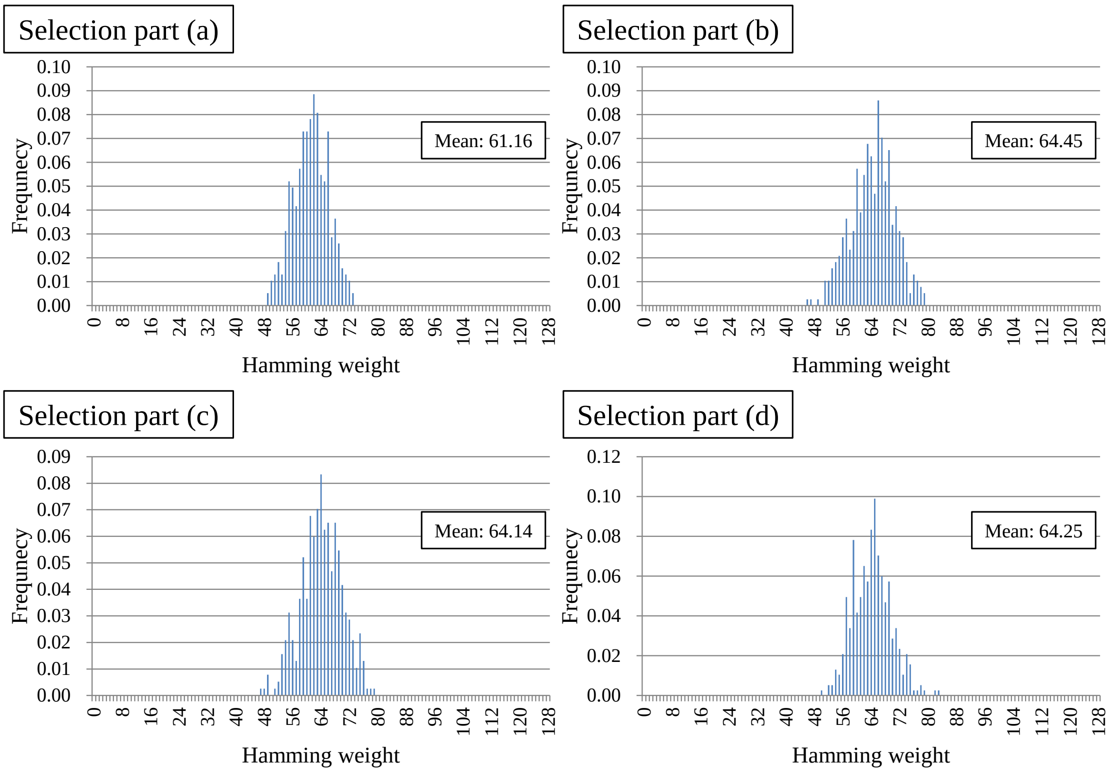

Figure 6. Experimental results of randomness

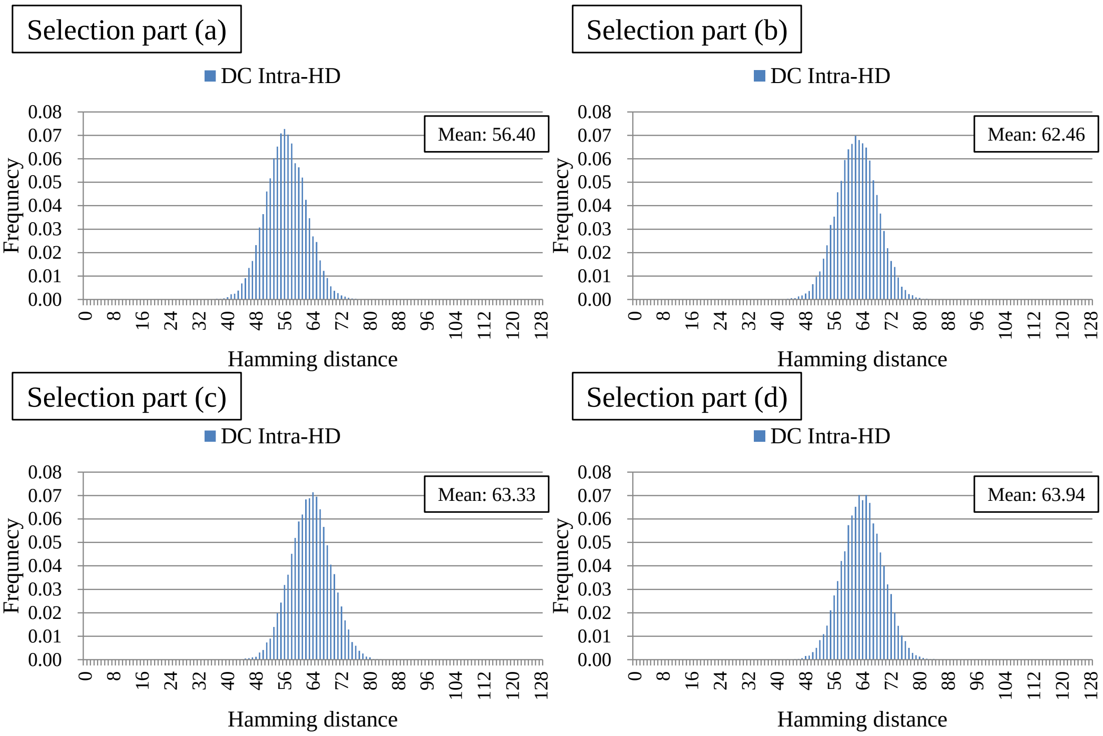

Figure 7. Experimental results of diffuseness

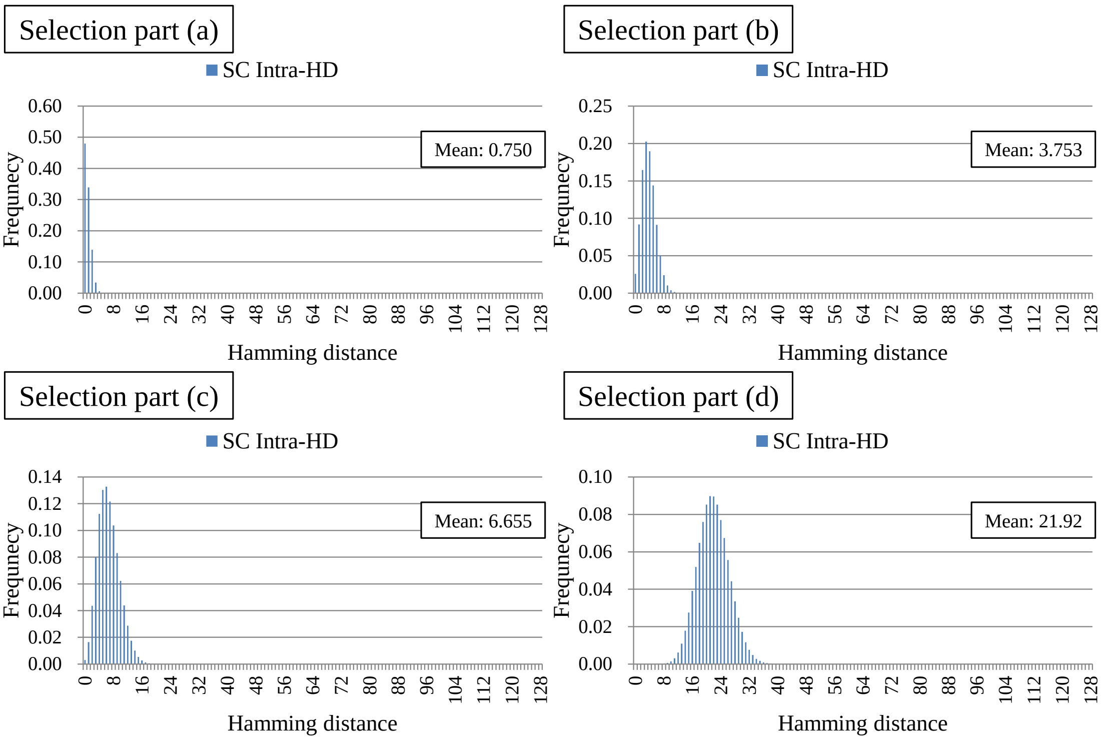

Figure 8. Experimental results of steadiness

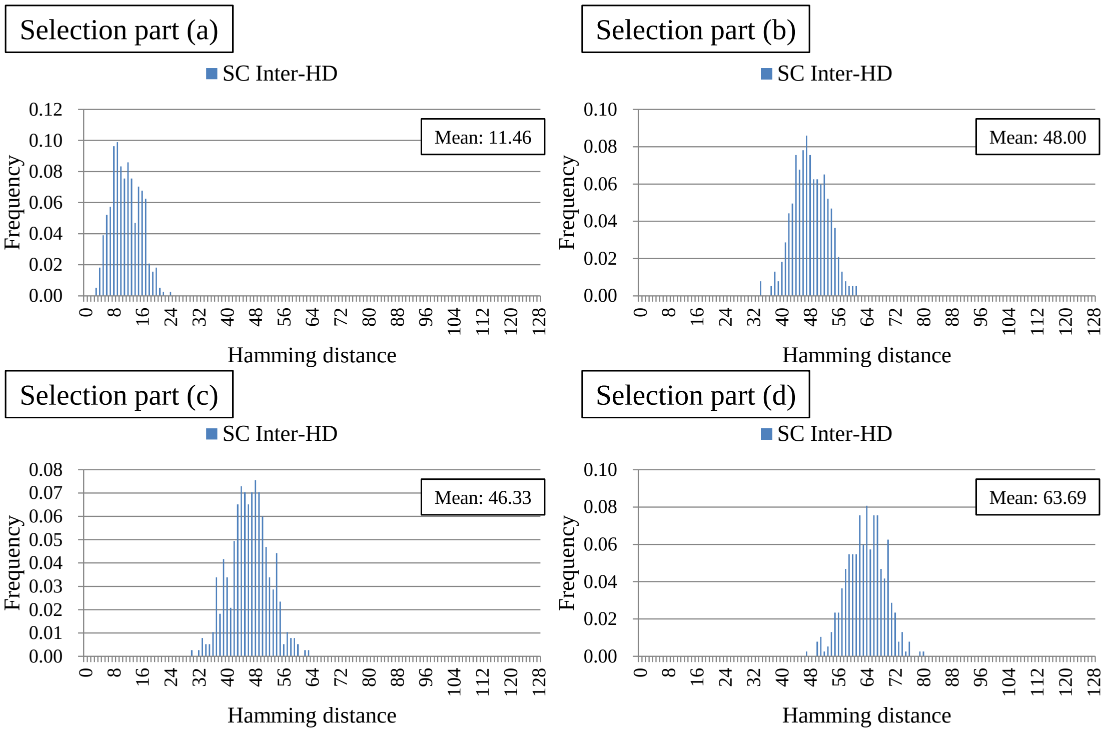

Figure 9. Experimental results of uniqueness

5. CONCLUSIONS

This study evaluated the detailed PUF performance of PRINCE based glitch PUF. In particular, this study implemented 4 types of PRINCE based glitch PUF with different glitch generators. In experiments using FPGAs, typical PUF performance indicators randomness, steadiness, diffuseness, and uniqueness were used. Experimental results using FPGAs showed that PRINCE based glitch PUF had the relationship of trade-off between steadiness and uniqueness depending on the selected part as glitch generator. Specifically, for applications that require high steadiness, such as key generation for cryptographic circuits, PRINCE based glitch PUF with selected part (a) is recommended. Furthermore, PRINCE based glitch PUF with selected part (d) is recommended in applications require high uniqueness.

In the future, we will evaluate the influence of PRINCE based glitch PUF by environmental variations such as supplied voltage or temperature. We will also develop a new glitch PUF structure which can extract glitch variations efficiently.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] A. Bogdanav, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, “PRESENT: An Ultra-Lightweight Block Cipher,” Proc. of 9th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2007), LNCS vol. 4727, pp. 450–466, Springer-Verlag, Sep. 2007.

[2] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, “The Simeck Family of Lightweight Block Ciphers,” Proc. of 17th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015), LNCS vol. 9293, pp. 307–329, Springer, Sep. 2015.

[3] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, “Midori: A Block Cipher for Low Energy,” Proc. of ASIACRYPT 2015, LNCS vol. 9453, pp. 411–436, Springer, Dec. 2015.

[4] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavum, M. Knežević, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin, “PRINCE – A Low-latency Block Cipher for Pervasive Computing Applications,” Proc. of ASIACRYPT 2012, LNCS vol. 7658, pp. 208–225, Springer, Dec. 2012.

[5] J. Guajardo, S. S. Kumar, G. J. S̃chrijen, and P. Tuyls, “FPGA Intrinsic PUFs and Their Use for IP Protection,” Proc. of 9th Int. Conf. on Cryptographic Hardware and Embedded Systems (CHES 2007), LNCS vol. 4727, pp. 63–80. Springer, Sep. 2007.

[6] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. V. Dijk, and S. Debadas, “A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications,” Proc. of IEEE VLSI Circuits Symposium, pp.176–179, Jun. 2004.

[7] G. E. Suh and S. Devadas, “Physical Unclonable Functions for Device Authentication and Secret Key Generation,” Proc. of 44th ACM/IEEE Design Automation Conference (DAC 2007), pp. 9–14, Jun. 2007.

[8] D. Suzuki and K. Shimizu, “The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes,” Proc. of the 12th Int. Conf. on Cryptographic Hardware and Embedded Systems (CHES’10), LNCS vol. 6225, pp. 366–382, Springer-Verlag, Aug. 2010.

[9] Y. Nozaki and M. Yoshikawa, “Energy Harvesting PUF oriented ID Generation Method and its Evaluation System,” Proc. of Int. Conf. on Information Technology and Computer Communications (ITCC 2019), pp. 119–124, Aug. 2019.

[10] K. Shimizu, D. Suzuki, and T. Kasuya, “Glitch PUF: Extracting Information from Usually Unwanted Glitches,” IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, vol. E95–A, no. 1, pp. 223–233, Jan. 2012.

[11] Y. Nozaki and M. Yoshikawa, “Unrolled PRINCE Cipher based Glitch Physically UnclonableFunction,” Proc. of 3rd Int. Conf. on Information Science and Systems (ICISS 2020), (to appear)

[12] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, “PUF Modeling Attacks on Simulated and Silicon Data,” IEEETrans. on Information Forensics and Security, vol. 8, no. 11, pp. 1876–1891, Nov. 2013.

[13] Y. Nozaki and M. Yoshikawa, “Security Evaluation of Ring Oscillator PUF against Genetic Algorithm Based Modeling Attack,” Proc. of 13th Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2019), Advances in Intelligent Systems and Computing (AISC) vol. 994, pp. 338–347, Springer, July 2019.

[14] Y. Nozaki and M. Yoshikawa, "Power Consumption Aware Machine Learning Attack for FeedForward Arbiter PUF," In: Lee R. (eds) Computer and Information Science, Studies in Computational Intelligence (SCI), vol. 791, pp. 49–62, Springer, Sep. 2018.

[15] CRYPTREC Lightweight Cryptography Working Group, "CRYPTREC Cryptographic Technology Guideline (Lightweight Cryptography)," Mar. 2017. https://www.cryptrec.go.jp/report/cryptrec-gl-2003-2016en.pdf

[16] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs," Proc. of Int. Conf. on Reconfigurable Computing and FPGAs (ReConFig 2010), pp. 298–303, Dec. 2010.