Novel intrinsic physical unclonable function design for post-quantum cryptography

Novel Intrinsic Physical Unclonable Function Design for Post-quantum Cryptography

Baosheng Wang*,Yijun Cui*, Chongyan Gu†, Chenghua Wang* and Weiqiang Liu*
*College of Electronics and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China
†Centre for Secure Information Technologies, Queen's University Belfast, Belfast, U.K.

***Abstract*—The hardware implementations of post-quantum cryptography (PQC) algorithms are vulnerable to fault injection attacks. As a hardware security primitive, the intrinsic physical unclonable function (PUF) is a possible countermeasure for these attacks with low resource overheads. In this work, a novel intrinsic PUF, frequency adjustable software PUF (FAS-PUF), is proposed to provide a device identification for PQC chips. The FAS-PUF is based on an inherent timing logic in the ring-learning with error (R-LWE) decryption circuit of PQC chips. The FAS-PUF uses a 256*13*3-bit input ciphertext of the decryption circuit as a challenge, and uses a 256-bit decryption output as a response with an adjustable overclocking. Since the entropy of the FAS-PUF utilises the manifested timing errors caused by the overclocking, the FAS-PUF does not need to modify the existing hardware circuits, *i.e.* preserves the original circuit functions, which significantly reduces hardware resource consumption and power overhead. Meanwhile, to mitigate the affection of circuits' metastablities to PUF's stability under overclocking, a dynamic clock frequency selection method is used to determine the optimal frequency point for generating PUF responses. The proposed FAS-PUF is also a Strong PUF design with a significant number of Challenge/Response Pairs (CRPs) provided. The proposed design is implemented on Xilinx Basys3 FPGAs. The experimental results show that the FAS-PUF has a good uniqueness, uniformity and stability compared with other intrinsic PUFs.**

***Index Terms*—Intrinsic PUF; Strong PUF; Post-quantum cryptography; Decryption circuit; Frequency adjustable method**

I. Introduction

The transmission and storage of information in the Internet of Things (IoT) devices require a secure and reliable cryptographic system. Post-quantum cryptography (PQC) is currently the most promising cryptographic system against quantum attacks. Among the existing PQC schemes, the Ring-Learning with Error (R-LWE) encryption scheme based on a lattice puzzle on a specific ring is widely developed.

While PQC algorithms follow their respective mathematical models, their implementations on hardware are vulnerable to attacks, such as fault injection attacks [12]. Prior research [14] [15] presented fault injection attacks on R-LWE, which include zeroing during the Number Theoretic Transform (NTT) and skip-addition fault for decryption. These attacks disrupt the decryption function and always require that the targeted device decapsulates any attacker-generated ciphertexts for thousands times. Traditional countermeasures for fault injection attacks rely on the introduction of redundancy in the execution, either in the form of error detecting codes (information redundancy) or through duplicated execution (time or hardware redundancy), which are heavyweight, not suitable for IoT devices.

For resource-constrained applications, *e.g.* IoT, there is a need for a secure and efficient way to protect PQC devices. A physical unclonable function (PUF) [1], a hardware security primitive, uses the differences from the circuit manufacturing process to provide a unique identification for chips. A majority of PUF designs [2] [3] [4] [5] proposed so far consume additional hardware resources while intrinsic PUFs are based on existing resources, which use the pre-existed circuits without requiring extra resources. However, the existing intrinsic PUFs still have challenges. For example, the PUF in [4] presented poor uniqueness and less CRP generation efficiency. DTA-PUF [6] has high CRP generation efficiency but also having poor uniqueness.

In this paper, a novel intrinsic PUF design, FAS-PUF, based on timing errors, is proposed to protect PQC chips from fault injection attacks. The hardware implementation of FAS-PUF is the decryption circuit for R-LWE, which has a large number of butterfly arithmetic units for the Number NTT algorithm. These butterfly arithmetic units are very sensitive to timing changes. A small clock change in an overclocking case may cause a huge change for its arithmetic results, which can be utilised as a source of entropy for FAS-PUF. FAS-PUF also has a good efficiency of response generation since a 1-bit response requires only a 39-bit challenge. FAS-PUF can detect fault injection attacks by providing a device authentication security mechanism. A PUF-based device can not pass the authentication when the hardware structure of the decryption circuit is attacked. Compared with traditional countermeasures for fault injection attacks, *e.g.* error detecting codes and duplicated execution, FAS-PUF significantly reduces hardware and time resource overheads. The main contributions of this paper are as follows:

- A novel intrinsic PUF, FAS-PUF, is proposed to secure PQC chips. FAS-PUF provides a device authentication security mechanism for PQC chips using overclocking settings, which can detect fault injection attacks.
- A dynamic system clock frequency selection method is proposed to mitigate the affection of metastability in FPGAs using overclocking settings. This determines the optimal frequency point for response generations.

This work is supported by grants from the Natural Science Foundation of Jiangsu Province (BK20210287), National Natural Science Foundation of China (62104107, 62022041, 62134002), EPSRC New Investigator Award (EP/X009602/1), and Royal Society (IEC\NSFC\211024).

- FAS-PUF is implemented on Xilinx Basys3 FPGAs. The experimental results show that the FAS-PUF has good statistical characteristics with a uniqueness of 0.4954, uniformity of 0.4957 and stability of 0.0467.

II. Related Work

The concept of intrinsic PUF has been proposed for several years, but few intrinsic PUF designs have been developed. Holcomb [7] proposed an intrinsic PUF based on the power-on values of cells in SRAM memory modules, in which the power-on values are random and unique for different instances of logically identical circuits. Maiti *et al.* [4] proposed the first fully processor-based intrinsic PUF, which adjusts the clock frequency of the processor to determine the failure rate of different instructions at different frequencies. Yu Zheng proposed ScanPUF [7], a novel PUF implementation which exploits random delay variations in timing paths between two scan flip-flops. J. Kong proposed ALU PUF [3] based on the delay difference in two different ALUs. Tsiokanos proposed DTA-PUF [6], an intrinsic PUF design that exploits the instruction- and data-dependent dynamic timing behavior of pipelined cores.

Some strong PUFs based on lattice problems were also proposed to secure IoT devices, since PQC is the most promising cryptographic primitive against quantum attacks. Lattice PUF [9], a strong PUF which utilizes lattice-based decryption function, was proposed to against machine learning (ML) attacks. Lattice PUF is constructed using a physically obfuscated key (POK), an LWE decryption function block, and a linear-feedback shift register (LFSR). The security of Lattice PUF is derived from cryptographic hardness of learning decryption functions of semantically secure publickey cryptosystems within the probably approximately correct framework. Stateless PUF [10] presented a stateless construction of a cryptographically-secure PUF by constructing a fuzzy extractor, and the security of Stateless PUF can be reduced to the hardness of Learning Parity with Noise (LPN).

III. Proposed FAS-PUF

The proposed FAS-PUF extracts responses through timing errors generated by the decryption circuit in the case of overclocking. The FAS-PUF could flexibly adjust the clock frequency of circuits to solve the problem of poor stability of the circuit outputs caused by the metastability of the flip-flop, thus improving the stability of the PUF based on overclocking.

A. Timing errors caused by overclocking

Due to the affection of temperature, humidity, voltage and layout wiring in the integrated circuit (IC) manufacturing process, even if chips are manufactured by the same manufacturer in the same batch, there will be variations from chip to chip. In the timing analysis, these differences are mainly reflected in the different lengths of critical paths.

When a circuit operates at a normal frequency, the circuit has correct functionalities, and the timing differences are hardly visible. When the clock frequency increases slowly, known as overclocking, the timing differences are reflected in the point of first failure and result in errors, as the timing constraints of the circuit are not met and the circuit has an incorrect function. The timing differences are unique to each circuit. Therefore, it is clear from the above analysis that the timing differences within the hardware circuits can be used as a source of entropy for PUFs.

B. Frequency regulation strategy

In digital circuits, a flip-flop has a certain probability of being in a metastable state if the system clock frequency is too high to satisfy the setup time or hold time of the flip-flop. While the flip-flop is in a metastable state, its output will be in a relatively long time uncertain state after the effective clock edge. In this time, the output will be in an oscillation state and will randomly be 0 or 1. The time is called resolution time. After the resolution time the output will stabilize to 0 or 1, and the state 0 or 1 is random with no relationship to the input.

The metastable state of the flip-flop results in a poor stability of the circuit output in overclocking. The probability of the metatability becomes larger when the clock frequency increases. Additionally, the location of the timing errors in the data path and the location that the metatable state of the flip-flop occurs in the circuit will also have an impact on the stability of the outputs of the circuit. It is found that the stability of the circuit's outputs, the number of bits and the position of errors are changed with the clock frequency are in the overclocking. The incorrect bits in the output are called as entropy source bits (ESBs), and these bits can be used as an entropy source of the PUF. The ideal value of the proportion of the entropy source bits in an output is 50%, and the experimental results of the proportion of ESBs and stability of the PUF response on one of the Xilinx Basys3 FPGAs are shown in Fig.1.

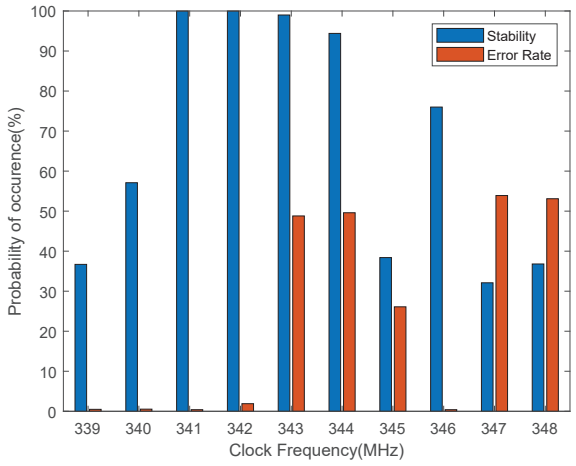

Fig. 1. Variations in stability and error rate with the clock frequency changing.

When the circuit is overclocked and thus in the metastable state, the output of the circuit will become unstable. The errors that the circuit generates due to the timing constraints

will become unstable, and the stability of the PUF response will get worse. In order to address the problem of the poor stability of PUF responses, a dynamic clock regulation strategy is proposed to select an optimal frequency point for the PUF response genera. The stability of the PUF responses and entropy source bits will be taken into account in the selection of the optimal frequency point. The frequency point that affects the stability and error rate is investigated. The implementation process is shown in section III-C.

C. Configuration steps of FAS-PUF

The hardware implementation carrier of FAS-PUF is the decryption circuit, as shown in Fig.2, which accepts a 256*13*3-bit challenge from the control unit as the system input and finally outputs the decrypted 256-bit plaintext. The received 256*13-bit ciphertext C_1 performs a point-wise multiplication with a 256*13-bit private key R_1, and the result executes a point-wise addition with a 256*13-bit ciphertext C_2, which is followed by a modulus reduction. The operation result is converted back to the time domain through the Inverse-NTT (INTT) module, and finally the 256-bit plaintext M is generated through the Decode module. The implementation of FAS-PUF is mainly divided into three steps, as shown in Fig.3. The first step is to determine the first error frequency (FEF) of the circuit, and in this step the system clock frequency of the decryption circuit is gradually increased until bit errors occur at the first time in the output of the circuit. When this happens in the 256-bit output of the decryption circuit, the system clock frequency is recorded as the FEF.

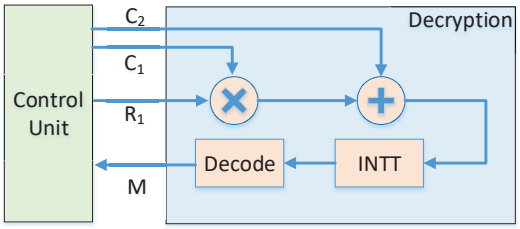

Fig. 2. Decryption circuit function.

After determining the FEF, the following shows the identification of the optimal frequency point. As described in section III-B, the error bits of the circuit output become unstable in the overclocking due to the occurrence of circuits' metastability, which affects the stability and entropy source bits of the PUF. To solve this problem, a dynamic system clock frequency selection method is proposed. The frequency interval is firstly set for generating the PUF response according to the system clock frequency, denoted as Δf, and the clock frequency is gradually increased from the FEF, each time increasing Δf. The stability and entropy source bits of the generated response are calculated at each sampled frequency point until the frequency point where the stability and entropy source bits are close to the ideal value. The frequency point is denoted as the optimal frequency point (OFP).

Finally, in a hardware design, the decryption circuit will be divided into two function modes: a normal mode and a PUF mode. When the decryption circuit is in the normal mode, the system clock frequency is fixed, and the circuit works normally. When the decryption circuit is switched to the PUF mode, it will performs the PUF function. In this mode, the circuit outputs will be used to generate PUF responses and will not affect the normal function of the circuit. The system clock frequency will be adjusted to the optimal frequency point as mentioned above, and the 256*13*3-bit ciphertext input will be used as a PUF challenge while the 256-bit decryption result of the decryption circuit will be used as a PUF response.

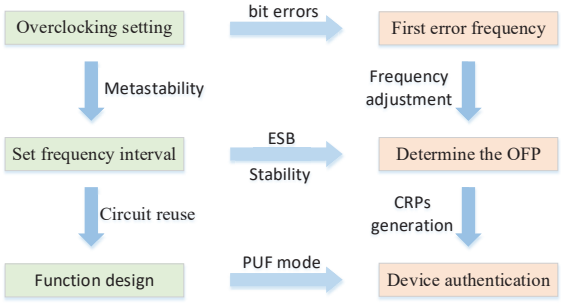

Fig. 3. Configuration steps of FAS-PUF.

IV. EXPERIMENTAL RESULTS

In this section, we validate the FAS-PUF on Xilinx Basys3 FPGAs (Xilinx Artix®-7 FPGA XC7A35T-1CPG236C) and analyze the uniqueness, uniformity and stability. The employed decryption circuits [13] are based on a multi-path delay commutator (MDC) pipeline structure, and the INTT cores are based on a Radix-2 circuit, which is used in R-LWE structure. Meanwhile, the frequency of the decryption circuits is controlled by the method used in [4].

A. Uniqueness

Uniqueness, also known as the variability of PUF responses between different chips. It is calculated by counting the average of the Hamming distances (HD) between the responses of different PUF chips under the same challenge as shown below:

$$Uniqueness = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{HD(R_i, R_j)}{l} \times 100\%$$

where N denotes the number of PUF chips in the population; R_i and R_j denote the responses of PUF chip numbered i and PUF chip numbered j. The total bits number of responses are l; HD denotes the Hamming distance between the two responses. The uniqueness results of FAS-PUF are shown in Fig.4, with the minimum value of 0.4336, the maximum value of 0.5508, and the average value is 0.4954, which is close to the ideal property.

TABLE I
COMPARISON WITH OTHER PUFS

PUF	Type	Hardware Additions	Uniqueness	Stability	Hardware Implementation
ALU PUF [3]	Partially Intrinsic	Arbiters	0.359	0.113	Simulation
PUF in [4]	Fully Intrinsic	None	0.375	0.021	FPGA
DTA-PUF [6]	Fully Intrinsic	None	0.314	—	Simulation
ScanPUF [8]	Partially Intrinsic	Scan chains and signal generating circuits	0.471	0.031	FPGA
Lattice PUF [9]	Non-intrinsic	POK, counter, controller and LFSR	0.500	0.013	Simulation
Proposed FAS-PUF	Fully Intrinsic	None	0.495	0.046	FPGA

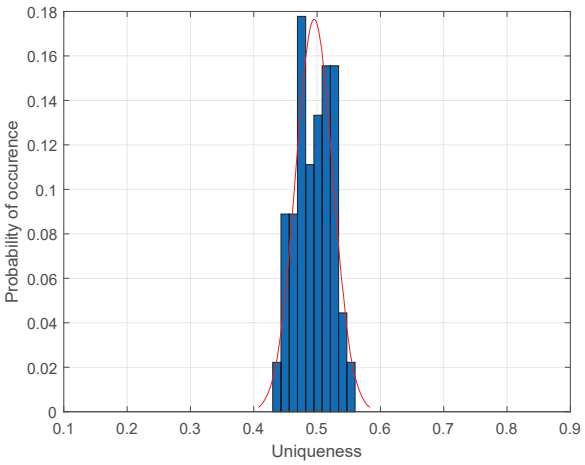

Fig. 4. Uniqueness distribution of FAS-PUF.

B. Uniformity

Uniformity refers to the uniformity of the proportion of 0 and 1 in the PUF responses and its ideal value is 0.5. For a given chip *i* and challenge *k*, it is expressed as:

$$Uniformity = \frac{1}{n}\sum_{j=1}^{n} r_{ijk} \times 100\%$$

where *n* is the number of bits in the PUF response, r_{ijk} is bit *j* in the PUF response of chip *i* for challenge *k*. The uniformity of FAS-PUF is 0.4957.

C. Stability

Stability refers to the probability of bit-hopping in the response generated repeatedly by the same PUF chip under the same challenge and test conditions. The ideal value of stability is 0, and the stability equation is shown as below:

$$Stability = \frac{2}{W(W-1)}\sum_{p=1}^{W}\sum_{q=p+1}^{W}\frac{HD(R_p, R_q)}{l} \times 100\%$$

where R_p and R_q denote the responses generated by a particular PUF chip at the *p*th and *q*th times under a certain ambient temperature and supply voltage conditions under the same challenge; *W* is the total number of response; and *l* is the number of bits in each response. The average stability of FAS-PUF is 0.0467.

D. Comparison with other PUFs

We compare FAS-PUF with lattice PUF and several other intrinsic PUFs as shown in Table I in terms of uniqueness, stability, and hardware resource consumption. As a fully intrinsic PUF, FAS-PUF does not consume additional hardware resources compared with lattice PUF and other not fully intrinsic PUFs, such as ALU PUF [3] and ScanPUF [8]. The FAS-PUF does not require changes to the original hardware circuitry. Therefore, it is more suitable for resource-constrained platforms. At the same time, compared to ALU PUF [3], there is a huge improvement in uniqueness and stability.

Compared to other intrinsic PUFs, the FAS-PUF takes into account both uniqueness and stability metrics by using a dynamic clock adjustment strategy in generating the CRPs. Compared to the PUF in [4], FAS-PUF has excellent uniqueness and comparable stability. At the same time, FAS-PUF has a high CRP generation efficiency. The PUF in [4] requires a processor to execute hundreds or even thousands of instructions for every two-bit response, while FAS-PUF can obtain 256-bit response by input 256*13*3-bit challenge. Compared to DTA-PUF, FAS-PUF also greatly improves the uniqueness. DTA-PUF was only validated via simulation. The proposed PUF achieves the verification on FPGAs.

V. CONCLUSION

In this paper, the FAS-PUF is proposed to protect PQC chips. The FAS-PUF extracts the entropy by exploiting the inherent timing differences between the logically identical decryption circuits, thus eliminating the need to make changes to the existing hardware circuits. Meanwhile, a dynamic clock frequency adjustment strategy is proposed to solve the problem of unstable decryption results under overclocking, which greatly improves the stability of the proposed FAS-PUF. Finally, the proposed FAS-PUF is implemented on Xilinx

Basys3 FPGAs, and the experimental results show that FAS-PUF has a good uniqueness, uniformity and stability.

REFERENCES

[1] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in Proc. 44th ACM/IEEE Design Automation Conference, 2007, pp. 9-14.

[2] J. Kong and F. Koushanfar, "Processor-Based Strong Physical Unclonable Functions With Aging-Based Response Tuning," IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 1, pp. 16-29, March 2014.

[3] J. Kong, F. Koushanfar, P. K. Pendyala, A. -R. Sadeghi and C. Wachsmann, "PUFatt: Embedded platform attestation based on novel processor-based PUFs," in Proc. 51st ACM/EDAC/IEEE Design Automation Conference (DAC), 2014, pp. 1-6.

[4] A. Maiti and P. Schaumont, "A novel microprocessor-intrinsic Physical Unclonable Function," in Proc. 22nd International Conference on Field Programmable Logic and Applications (FPL), 2012, pp. 380-387.

[5] M. Sauer, P. Raiola, L. Feiten, B. Becker, U. Rührmair and I. Polian, "Sensitized path PUF: A lightweight embedded physical unclonable function," in Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, pp. 680-685.

[6] I. Tsiokanos, J. Miskelly, C. Gu, M. O'neill and G. Karakonstantis, "DTA-PUF: Dynamic Timing-aware Physical Unclonable Function for Resource-constrained Devices," ACM Journal on Emerging Technologies in Computing Systems (JETC), pp. 1-24, 2021.

[7] D. E. Holcomb, W. P. Burleson and K. Fu, "Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers," IEEE Transactions on Computers, vol. 58, no. 9, pp. 1198-1210, Sept. 2009.

[8] Yu Zheng, A. R. Krishna and S. Bhunia, "ScanPUF: Robust ultralow-overhead PUF using scan chain," in Proc. 18th Asia and South Pacific Design Automation Conference (ASP-DAC), 2013, pp. 626-631.

[9] Y. Wang, X. Xi and M. Orshansky, "Lattice PUF: A Strong Physical Unclonable Function Provably Secure against Machine Learning Attacks," in Proc. IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2020, pp. 273-283.

[10] C. Herder, L. Ren, M. van Dijk, M. -D. Yu and S. Devadas, "Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions," IEEE Transactions on Dependable and Secure Computing, vol. 14, no. 1, pp. 65-82, 1 Jan.-Feb. 2017.

[11] Y. Zhang, C. Wang, D. E. S. Kundi, A. Khalid, M. O'Neill and W. Liu, "An Efficient and Parallel R-LWE Cryptoprocessor," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 5, pp. 886-890, May 2020.

[12] A. Barenghi, L. Breveglieri, I. Koren and D. Naccache, "Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures," Proceedings of the IEEE, vol. 100, no. 11, pp. 3056-3076, Nov. 2012.

[13] D. -e. -S. Kundi, Y. Zhang, C. Wang, A. Khalid, M. O'Neill and W. Liu, "Ultra High-Speed Polynomial Multiplications for Lattice-based Cryptography on FPGAs," IEEE Transactions on Emerging Topics in Computing, doi: 10.1109/TETC.2022.3144101.

[14] Pessl, Peter, and Lukas Prokop, "Fault attacks on CCA-secure lattice KEMs," IACR Transactions on Cryptographic Hardware and Embedded Systems: pp. 37-60, 2021.

[15] Valencia, F., Oder, T., Güneysu, T., and Regazzoni, F., "Exploring the vulnerability of R-LWE encryption to fault attacks," in Proc. Fifth Workshop on Cryptography and Security in Computing Systems, 2018, pp. 7-12.