Mutual authentication method between PUFs

Yunyeong Choi
dept. of Electronic and Electrical Engineering
Ewha Womans University
Seoul, Korea
https://orcid.org/0000-0002-8094-3951

Jungwon Lee
dept. of Electrical and Computer Engineering
Seoul National University
Seoul, Korea
https://orcid.org/0000-0002-3080-1760

Hyungsoon Shin
Smart Factory Multidisciplinary Program
Ewha Womans University
Seoul, Korea
hsshin@ewha.ac.kr

Wookyung Sun
dept. of Electrical and Computer Engineering
Seoul National University
Seoul, Korea
https://orcid.org/0000-0002-3078-6839

2022 International Conference on Electrical, Computer and Energy Technologies (ICECET) | 978-1-6654-7087-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICECET55527.2022.9873480

***Abstract*—This paper proposes a new authentication method that enables mutual verification between two different PUFs without a server. The output signal (response) of one PUF is hidden in the opponent's input signal (challenge) list by interlinking the challenge and response information of the two PUFs. In this way, when sending and receiving authentication information, a challenge list index is used instead of exposing the challenge or response information to the network directly. The verification information for checking the validity of the received index is protected with encryption using a hash function. Also, two novel PUF systems, key-map PUF and cube PUF, are presented that improve the security of the PUF-based system using previous authentication information. The proposed authentication method and PUF systems can be used for IoT devices.**

Keywords—physical unclonable functions, hardware-based security system, mutual authentication, key-map PUF, Cube PUF

I. INTRODUCTION

The advancements in sensor electronics, wearable technology, and mobile platform have increased communication in the domain of the Internet of Things (IoT). As conventional cryptographic approaches which rely on secret keys stored in non-volatile memories(NVM) are vulnerable to physical and side-channeling attacks, physical hardware roots of trust have attracted significant attention to protect the IoT systems against cyber-attacks. Physical unclonable functions (PUFs) are promising security primitives that do not save the secret keys explicitly but generate the keys using the intrinsic properties of hardware devices. PUFs, as a physical function, map challenges(input data) into unique responses(output data) based on uncontrollable and unpredictable variations of the devices from manufacturing[1-6].

Fig. 1 shows a typical authentication process using a PUF[7]. Before providing the PUF to a user, a server creates a challenge-response pair(CRP) table by mapping the unique responses for various challenges in advance. After initial work, the authentication process consists of three steps. The server picks up a challenge and sends it to the user. The user returns a response using PUF according to the challenge. The server verifies the authenticity of the user by comparing the user's response to the correct answer in the CRP table. This method has three major problems. First, it has a hacking risk inherently because the server stores the CRP table in memory. Second, it is impossible to distinguish whether one who sends a challenge is a hacker or a server. Third, authentication information is exposed to the network in order of challenge-response-result of authentication, making it vulnerable to hacking using machine learning.

In order to solve these problems, mutual verification should be possible without a participant who stores the CRP table. Several existing mutual verification methods have been suggested, but these methods need a server having a CRP table to verify users[8-12]. A method that allows verification between PUFs without going through a server has not been suggested. In this paper, we propose a novel method for two different PUFs to authenticate each other directly without the need of storing a CRP table. And new PUF systems, keymap PUF and cube PUF, which include an encryption process using a keymap or a cube are also suggested. These systems provide stronger security by utilizing authentication history in the encryption process.

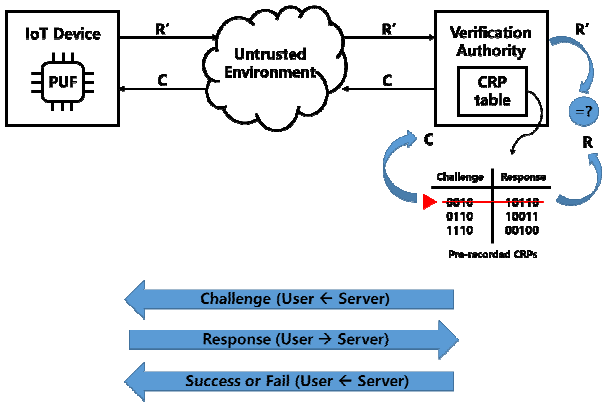

Fig. 1. A typical PUF-based authentication method

The paper was developed with funds from the National Research Foundation of Korea (NRF) (NRF-2020R1I1A1A01065622)

II. Mutual Authentication Method And PUF Systems

A. Information interlinking of PUFs

The main idea of mutual authentication without a server is the interlinking information of PUFs. In order to interlink two PUFs, PUF A utilizes PUF B's output signal as an input signal and vice versa for PUF B. Fig. 2 shows the initial process of authentication between two different PUFs. CA, CB, RA, and RB are abbreviations indicating the challenge and response of PUF A and PUF B, respectively. First, choose a challenge(CA1) in PUF A's challenge list and obtain a response (RA1) by PUF A. Then, use RA1 as a challenge for PUF B(CB5) and generate the corresponding response(RB5). Likewise, RB5 is used as a challenge for PUF A. This interlinking process is repeated until a sufficient amount of CRP is collected.

In fig. 2, the CRPs table is depicted for easy explanation, but the responses are not stored because PUF can generate a response when a challenge is given. Also, the order of the challenge list is randomly shuffled. Therefore, it is not possible to distinguish the response corresponding to a specific challenge, although the challenge list contains response information. Even if a hacker has a list of challenges, he cannot find the right combination of challenge and response without PUF. Two users have their own and each other's challenge lists and exchange authentication information by the list's index not by challenge or response directly.

B. Validation of index

A server in the general authentication method stores the CRP table to verify that the authenticating subject has sent the appropriate information. If the CRP table is hacked, the authentication method using PUF becomes incapacitated. The server requires a high level of security technology to prevent hacking. The proposed PUF system has an encrypted dataset with a hash function instead of a CRP table. The challenge and response are encrypted together with the hash function, and the two pieces of information cannot be restored separately from the hash result. A normal user with PUF can find a correct combination of index information and generate the same encryption key. On the other hand, a hacker with an encrypted dataset cannot decrypt the CRP from the hash result conversely and cannot find an appropriate index.

C. Mutual authentication

In the case that the user cannot verify who requested authentication, the user returns a response even if the hacker sends a challenge. This problem can be solved by sending the request along with authentication information that the sender of the request can prove himself to be. The proposed method simultaneously performs verification and request in one communication step without a three-step process of exchanging challenges and responses and sharing authentication results. The detailed process for users A and B to perform mutual verification is described below(Fig. 3).

- Request from user A: user A selects one challenge(CA1) from his challenge list and generates a response(RA1). Find the challenge index in PUF B's list of challenges that matches PUF A's response. In this case, RA1 is a match with CB5. Encrypt the challenge and response together using the hash function and send the hashing result with index(indexA=1, indexB=5) to user B.
- Validation of user A: user B finds user A's challenge (CA1) and response (RA1=CB5) according to the index from user A and hash them in the same way. Verify that user A is the sender of the request by checking the encryption dataset for hashing data.
- Reply from user B: User B also generates a response using PUF B and sends cryptographic information (hash(CB5+RB5)) with a challenge index (indexB=5) and response index (indexA=9) to user A.
- Validation of user B: User A verifies the authentication information sent by user B.

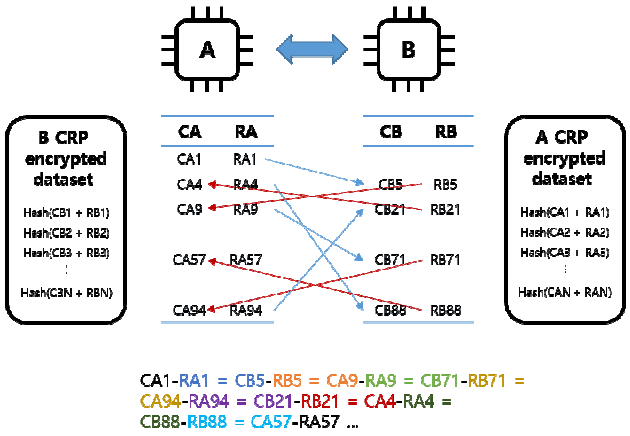

Fig. 2. The process of interlinking two PUF information

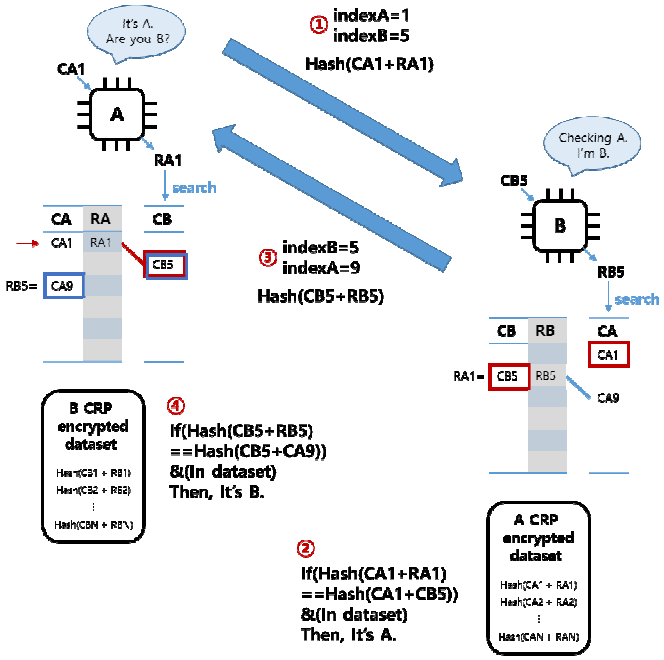

Fig. 3. Detailed process of mutual authentication between two PUFs

D. Encryption with Keymap

The keymap is a matrix of keys, and the table determines a movement in the keymap from an initial address according to the PUF output signal. Tables 1 and 2 are examples of two keymap movement rules. For example, when the PUF output signal is 01 11 11 00 11 00 10 00, the movement path by Table 1 is down, right, right, up, right, up, left, and up. And the keymap PUF collects the key from the keymap according to the path. Using different keymaps with the same path results in different responses. The responses using the keymap in Fig. 4(a) and (b) are "0010 0100" and "0110 1000 0100 0000 1010 0101 0001 1101" respectively. Also, the response depends on the movement rule and the starting coordinate as shown in Fig. 4(c), (d).

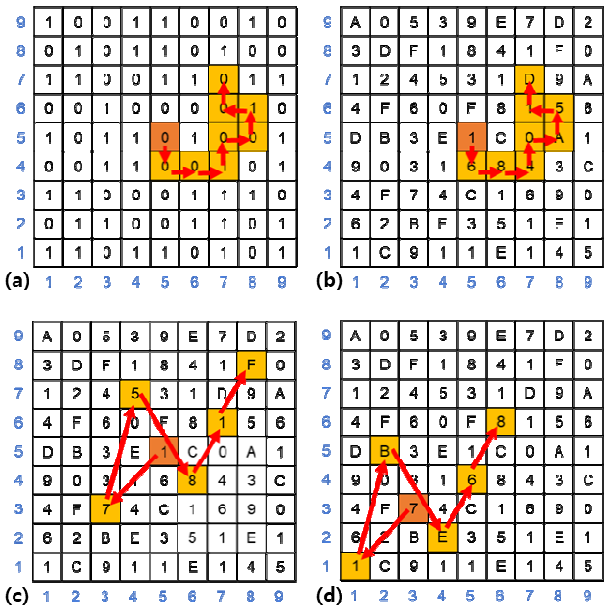

Fig. 4. Key collection using (a) 1-bit keymap, (d) 4-bit keymap according to Table 1, (c) 4-bit keymap (d) 4-bit keymap and changed starting coordinates according to Table 2

TABLE I: KEYMAP MOVEMENT RULE 1

Output signal	*Moving vector (row, column)*	*Output signal*	*Moving vector (row, column)*
00	Up (1,0)	10	Left (0,-1)
01	Down (-1, 0)	11	Right (0, 1)

TABLE II: KEYMAP MOVEMENT RULE 2

Output signal	*Moving vector (row, column)*	*Output signal*	*Moving vector (row, column)*
000	(1,0)	100	(2,1)
001	(-3,2)	101	(-1,0)
010	(0,-1)	110	(0,-1)
011	(-2,-2)	111	(4,1)

E. Encryption with Cube

Cubes can rotate in three directions: row, column, and layer. Table 3 summarizes the rules for mixing 3×3×3 Rubik's cube according to 4-bit unit information. For example, when the PUF output signal is 1001 0001, Table 3 determines how to mix the cube with the given PUF signal and rotate the second column upward and rotate the second row to the left(Fig. 5). After all rotations, the key is extracted from the top surface of the cube. There are about 43×10^{18} combinations that can be made with a general 3×3×3 Rubik's cube puzzle. Responses encrypted through the cube can have various patterns, which can disrupt the learning process for weight updates in hacking techniques using machine learning algorithms.

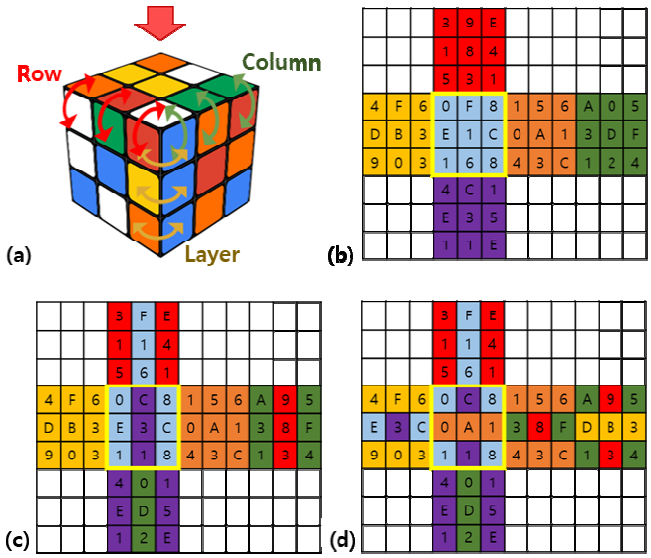

Fig. 5. (a) key cube (b) initial state of a cube (c) cube after rotation of the second column upward (d) cube after rotation of the second row to the left

TABLE III: CUBE ROTATION RULE

Output signal	*Moving vector (row, column)*	*Output signal*	*Moving vector (row, column)*
0000	Row1 left	1000	Column1 up
0001	Row2 left	1001	Column2 up
0010	Row3 left	1010	Column3 up
0011	Row1 right	1011	Column1 down
0100	Row2 right	1100	Column2 down
0101	Row3 right	1101	Column3 down
0110	Top left	1110	Bottom left
0111	Top right	1111	Bottom right

F. Keymap PUF and Cube PUF

The biggest difference between the proposed PUF and the existing PUF is that the PUF output signal is not used as it is, but is encrypted using a keymap or cube. Keymap PUF

consists of a PUF, a keymap, and a movement rule table. Cube PUF consists of a PUF, a cube, and a rotation rule table. It doesn't matter what kinds of PUF are used. The post encryption process can be added to any multi-bit PUF system.

III. Discussion

The proposed authentication method has the following differences from the existing method. First, a server is not required and users who participate in authentication do not store any information that can identify individual CRPs such as CRP tables. Second, CRP information is not exposed to the network directly. Even if authentication information composed of indexes and hashed data is exposed during communication, it is impossible to decrypt a challenge and a response. Therefore, hacking techniques using machine learning are not able to be used. Third, the distribution of authentication information is possible. In a structure in which only the server can determine whether authentication succeeds or not, the server must be directly connected to all users and must have all authentication information. The proposed method allows PUFs can be connected with other PUFs, and there is no need to keep a large amount of authentication information in one place.

TABLE IV: Comparison between the typical authentication method and the proposed method

Classification	Evaluation item			
	Not storing CRP table	*Not exposing CRP on the network*	*Mutual authentication*	*Preventing ML[a] hacking*
Existing	X	X	X	X
Proposed	O	O	O	O

[a.] Machine learning (ML)

Keymap and cube PUF have two advantages: providing stronger security and lifespan extension of PUF. First, the proposed PUF systems improve security by generating information including authentication history. The response is dependent on the last coordinates of the keymap or the mixed states of the cube from the previous authentication in the systems. This means that the system can identify a user and a hacker by checking the response without any additional information. Especially, a cube is continuously mixed in the authentication process and the encryption algorithm is updated every rotation. Thus, even if the cube state is hacked at a certain point, the information will become outdated and useless after a single rotation.

Second, the keymap PUF and the cube PUF prolong the lifespan of the PUF. CRPs used for the previous authentication are generally not reused because the response to a specific challenge is always the same. And the lifespan of a PUF is determined by how many CRPs the server can store. However, the proposed PUFs generate different responses to the same challenge according to initial conditions such as the starting coordinates of the map and the mixing state of the cube. Therefore, it is possible to generate various authentication information with a small number of CRPs.

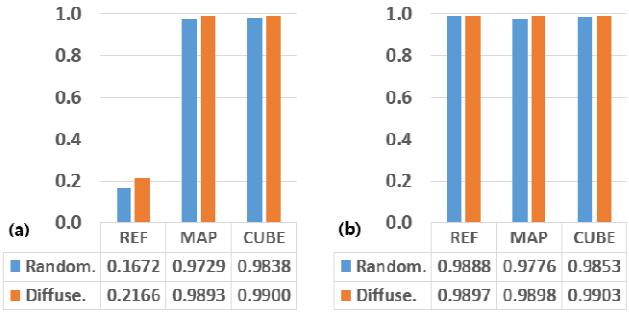

Fig. 6. Comparison of performance between the proposed PUF systems and (a) a poor performance PUF system, (b) a good performance PUF system

We compared the randomness and diffusivity of the responses of PUFs with and without post encryption. Randomness is an indicator of how evenly the bits in all responses are distributed without biasing to either 0 or 1. Device randomness is 1 when the probability of a response bit to be 1 (p_n) is 0.5. And the value of randomness is 0 when all the response bits are 0 or 1. Diffuseness is an indicator of how low the similarity between responses is. Device diffuseness is 1 when the hamming distance between responses with different challenges ($d_{n,l}$) is maximum and is 0 when all the responses are identical. The evaluation elements range from 0 to 1. The detailed equations for randomness and diffuseness are as follows[13].

$$p_n = \frac{1}{K \cdot L} \sum_{k=1}^{K} \sum_{l=1}^{L} b_{n,k,l} \tag{1}$$

$$Randomness = -\log_2(\max(p_n, 1-p_n)) \tag{2}$$

$$d_{n,l} = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} (b_{n,i,l} \oplus b_{n,j,l}) \tag{3}$$

$$Diffuseness = \frac{4}{L \cdot K^2} \sum_{l=1}^{L} d_{n,l} \tag{4}$$

Figure 6 shows PUF performance before and after encryption using a keymap or cube. A resistive random access memory(RRAM) based PUF system suggested in our previous work[14] is used and 100,000 64-bit responses are generated by MATLAB simulation. The performance of the RRAM PUF before encryption is labeled REF. The systems that encrypt PUF's output data using a keymap or cube are labeled as MAP, and CUBE, respectively. The movement rule and rotation rule used for encryption are summarized in table2 and table3. REF represents the existing PUF system and MAP/CUBE represents the proposed PUF system. In the case of poor-performing PUF(Figure 6 (a)), the low randomness(0.1672) and diffuseness(0.2166) are significantly improved to 0.9729 and 0.9893 for keymap PUF and 0.9838 and 0.9900 for cube PUF. The high randomness(0.9888) and diffuseness (0.9897) of PUF with good performance is maintained after encryption using a keymap and cube(Fig. 6(b)).

IV. Conclusion

In this paper, a novel mutual authentication method and PUF systems were proposed. The PUF-to-PUF authentication method uses indexes and hashing datasets to enable secure mutual verification without storing or transmitting CRP information. Key-map and Cube PUF enhance security by generating authentication information including authentication history. The proposed method and PUF systems are expected to be widely used in IoT devices.

Acknowledgment

This research was funded by the National Research Foundation of Korea (NRF) (NRF-2020R1I1A1A01065622).

References

[1] B. Chatterjee, D. Das, S. Maity, S. Sen, "RF-PUF: Enhancing IoT Security Through Authentication of Wireless Nodes Using In-Situ Machine Learning," IEEE Internet of Things Journal, vol. 6, no. 1, 2019.

[2] H. Nili, G. C. Adam, B. Hoskins, M. Prezioso, J. Kim, M. R. Mahmoodi, F. M. Bayat, O. Kavehei, D. B. Strukov, "Hardware-intrinsic security primitives enabled by analogue state and nonlinear conductance variations in integrated memristors," Nature Electronics, vol. 1, no. 3, pp. 197-202, 2018.

[3] A. Al-Omary, A. Othman, H. M. AlSabbagh, H. Al-Rizzo, "Survey of hardware-based security support for IoT/CPS systems," Sustainability and Resilience Conference: Mitigating Risks and Emergency Planning, KnE Engineering, pp. 52-70, 2018.

[4] C. Herder, M. D. Yu, F. Koushanfar, S. Devadas, "Physical unclonable functions and applications: A tutorial," Proceedings of the IEEE, vol. 102, no. 8, pp. 1126-1141, 2014.

[5] G. S. Rose, C. A. Meade, "Performance analysis of a memristive crossbar PUF design," In 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1-6, San Francisco, CA, USA, June 2015.

[6] A. Mazady, M. T. Rahman, D. Forte, M. Anwar, "Memristor PUF—A security primitive: Theory and experiment," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 5, no. 2, pp. 222-229, 2015.

[7] B. Halak, M. Zwolinski, and M. S. Mispan, "Overview of PUF-based hardware security solutions for the Internet of Things," In 2016 IEEE 59th MWSCAS, pp. 1-4, Abu Dhabi, United Arab Emirates, October 2016.

[8] K. Lounis, M. Zulkernine, "T2T-MAP: A PUF-based thing-to-thing mutual authentication protocol for IoT," IEEE Access, vol. 9, pp. 137384-137405, 2021.

[9] W. Liang, S. Xie, D. Zhang, X. Li, K. C. Li, "A mutual security authentication method for RFID-PUF circuit based on deep learning," ACM Transactions on Internet Technology (TOIT), vol. 22, no. 2, pp. 1-20, 2021.

[10] G. Bansal, N. Naren, V. Chamola, "Rama: Real-time automobile mutual authentication protocol using puf," IEEE, pp. 265-270, January 2020 [2020 International Conference on Information Networking (ICOIN), 2020]

[11] S. Li, T. Zhang, B. Yu, K. He, "A provably secure and practical PUF-based end-to-end mutual authentication and key exchange protocol for IoT," IEEE Sensors Journal, vol. 21, no. 4, pp. 5487-5501, 2020.

[12] Y. Zheng, C. H. Chang, "Secure mutual authentication and key-exchange protocol between PUF-embedded IoT endpoints," IEEE, pp. 1-5, May 2021 [2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021]

[13] Y. Hori, T. Yoshida, T. Katashita, A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs," In 2010 International conference on reconfigurable computing and FPGAs, pp. 298-303, Cancun, Mexico, December 2010.

[14] D. Kim, T. H. Kim, Y. Choi, G. H. Lee, J. Lee, W. Sun, B. Park, H. Kim, H. Shin, "Selected Bit-Line Current PUF: Implementation of Hardware Security Primitive Based on a Memristor Crossbar Array," *IEEE Access*, vol. 9, pp. 120901-120910, 2021.