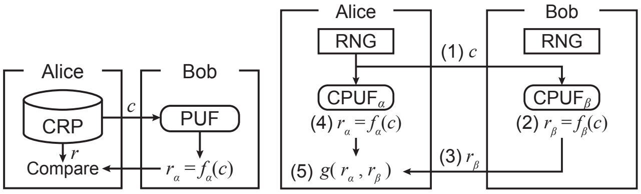
Clonable PUF: On the Design of PUFs That Share Equivalent Responses

Takashi Sato
Graduate School of Informatics
Kyoto University
Kyoto, Japan

Yuki Tanaka
Graduate School of Informatics
Kyoto University
Kyoto, Japan

Song Bian
Graduate School of Informatics
Kyoto University
Kyoto, Japan

***Abstract*—While numerous physically unclonable functions (PUFs) were proposed in recent years, the conventional PUF-based authentication model is centralized by the data of challenge-response pairs (CRPs), particularly when n-party authentication is required. In this work, we propose a novel concept of clonable PUF (CPUF), wherein two or more PUFs having equivalent responses are manufactured to facilitate decentralized authentication. By design, cloning is only possible in the fabrication period and the responses are determined based on the variability induced during the fabrication. We establish the usage model and the circuit design of CPUFs. Numerical experiments using a circuit simulator show an ideal matching rate of responses between the CPUFs.**



(a) Conventional PUF.

(b) Proposed PUF (CPUF).

Fig. 1. Authentication using the conventional PUF and the proposed CPUF.

I. Introduction

Physically unclonable functions (PUFs) [1], [2] are attracting increasing attention in the field of hardware security. The PUFs can be used in various ways: chip authentication, cryptographic protocols [3]–[6], identification of IoT devices [7], etc. The PUF circuit serves as a function, $r = f_\chi(c)$, which returns a response r to a challenge input c. Depending on the physical variation in a hardware instance, χ, the function f_χ becomes unique and unclonable for each PUF instance. The set of challenge-response pairs (CRPs), (c, r), is used as the chip-specific secret keys.

Consider an authentication between Alice and Bob [8] using a conventional PUF. In advance to the authentication, Alice collects a sufficient number of CRPs (CRP data) of a PUF, and physically passes the PUF to Bob. Then, in an authentication, Alice chooses a CRP (c, r) from the stored CRP dataset and sends the challenge c to Bob (Fig. 1a). When Bob receives the challenge, a response r_α obtained from the PUF is returned to Alice. Bob will be verified if the responses r and r_α match.

Although the above PUF-based authentication scheme is widely recognized, there remain several inconveniences and issues. The major limitation is the CRP data size. It is difficult for Alice to store all the CRPs when the CRP space of the PUF is large. Alice can store only a part of the CRPs, which limits the number of authentication processes to that of the CRPs which Alice has. Another issue is the asymmetry of the authentication process. In the case of Fig. 1a, only Alice can select a challenge, for Bob does not know the CRP data that Alice has, resulting in an unidirectional authentication scheme. In addition, PUF only permits two-party (pairwise) authentication. When we want to simultaneously authenticate more than two users, e.g., if Alice wants to authenticate both Bob and Charlie with different PUFs, Alice must have the CRP data of Charlie's PUF as well as that of Bob's PUF. These issues are all associated with the collection of CRP data. Even worse, the existence of CRP data can be a security risk. If the CRP data are stolen or their transmission is intercepted, forged or counterfeit copies of the PUF can be made. The CRP data are thus the most vulnerable part of the authentication using conventional PUFs.

In this paper, we propose a novel security primitive, clonable PUF (CPUF), which resolves all the above issues. The CPUF is a set of two or more PUFs sharing equivalent responses. Although the name sounds contradictory, the cloning of the CPUF is allowed only once during the manufacturing process. After the fabrication, cloning is never possible as in the case of conventional PUFs.

The contribution of this paper is summarized as follows:

- We propose a novel PUF architecture, clonable PUF, with which true *PUF-based* peer-to-peer and/or n-party authentications without storing CRP database become possible.
- We propose an example circuit realization of the CPUF that generates equivalent responses. The proposed CPUF circuit consists of memristors whose values are determined by the inherent variability during fabrication.
- Through circuit simulations, we show excellent performance in the new metrics of *equivalence* for measuring the closeness of the responses of the CPUF, as well as

This work was partially supported by JSPS KAKENHI Grant No. 20K19799, 20H04156, and 20K21793. This work was also supported by JST, PRESTO Grant Number JPMJPR20M7, Japan.

the existing metrics as a conventional PUF.

II. Clonable PUF: A concept

The proposed CPUF is a group of PUFs that share an *equivalent* response. Here, *equivalent* indicates that, for any challenges, the response bits of the CPUF instances are always equal. Obviously, the set of responses that are mutually negated can also be considered as *equivalent*. Because the responses of the PUFs are determined by physical variations, 100% match of the responses may not be expected in the fabricated chip. Even in such cases, we expect the responses of the CPUFs match at a very high probability close to 1. Meanwhile, an instance of the CPUF should satisfy other criteria that the conventional PUFs have to satisfy — randomness, uniqueness, robustness, etc. The CRP of a CPUF is eventually indistinguishable from the CRP of the conventional PUFs. A remarkable difference is that there is a known number of *clones* for a CPUF, whose CRPs are practically equal.

The example authentication process using CPUFs is shown in Fig. 1b. Alice and Bob respectively own paired instances of the CPUF, CPUF$_\alpha$ and CPUF$_\beta$. The CPUF has to be handed physically and securely. In the authentication, first, Alice (or Bob) generates a challenge, c, by a random number generator (RNG) and sends c to Bob (Alice). Then, Alice and Bob input c to CPUF$_\alpha$ and CPUF$_\beta$ to obtain respective responses, r_α and r_β. Bob (Alice) sends r_β (r_α) back to Alice (Bob). Finally, Alice (Bob) verifies the responses for their equivalence. What we would like to stress here is that, in the authentication using CPUFs, no CRP data is recorded or used, and hence either Alice or Bob can equally initiate and carry out authentication. Similarly, n-party authentication can be carried out using CPUFs having n clones.

III. Circuit design of CPUF

A. Memristor

The CPUF circuit in this paper uses memristors as a key component. A memristor, which is predicted theoretically by Chua in 1970's [9] and found by Strukov et al. in 2008 [10], is a passive circuit element. Typical structure of memristive devices includes a metal oxide layer in between two metal layers. The conductance between the metal layers varies depending on the history of the passed current. The I-V curve of a memristor is characterized by the hysteresis. The conductance G of the memristor becomes low when a positive high voltage is applied, whereas G becomes high when a negative high voltage is applied. The low and high conductance states are also called as high resistance state (HRS) and low resistance state (LRS), respectively.

The change of the conductance in the write operation is modeled as [11]

$$\text{write}: \frac{\Delta G}{G} \propto \Delta T \exp(\sigma V_\text{w}), \tag{1}$$

where V_w is a high voltage (write voltage), ΔT is an application period of the write voltage, and σ is a constant that depends on the physical variation. During the read operation, we apply a low read voltage V_r, so as not to change the current memristor state. The read current is simply given by Ohm's law:

$$\text{read}: I = GV_\text{r}. \tag{2}$$

We utilize the above characteristics and inherent variability of the memristors to design CPUF circuits.

B. Coupled-inverter CPUF

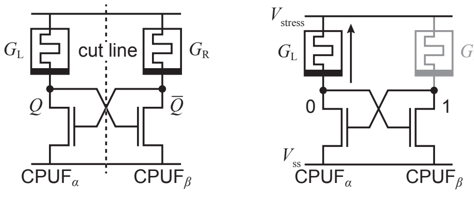
The manufacturing steps of the CPUF is summarized as follows:

1) Instances of the CPUFs, containing memristor array, are designed and placed side-by-side on a wafer having wire connections in between.
2) Once the wafer process is finished, a write voltage is serially applied to the connected CPUFs for determining the memristor values. Here, the written state solely depends on the mutual variation of the connected circuit elements. Hence, the responses of one memristor array are *equivalent* to those of the other array(s) connected, while the responses are unique and cannot be reproduced even by the manufacturer, as in the case of the conventional PUFs.
3) The wafer is diced into individual CPUF instances. The periphery circuits are destroyed in the chip cutting to preclude rewriting.

The circuit structure of the paired CPUF cells is shown in Fig. 2a. For the sake of simplicity, an example of two clones case is explained below but extention for larger number of clones is not difficult. Each CPUF cell is a pseudo inverter consisting of a memristor and a transistor. The on-chip wire connections in between the paired cells are established in a cross coupled manner to form a coupled inverter. In real implementations, the layouts of the two CPUF cells are drawn in the respective CPUF instances that are physically placed side by side, though a pair of CPUF cells is drawn adjacent for ease of understanding the circuit operation.

In the write operation, a write voltage, V_w, is applied to program the CPUF pair. Due to the feedback operation of the two pseudo inverters, one of the memristors (G_L in this example), initially having higher conductance than the other, conducts a large negative current (Fig. 2b). This current turns the memristor into a HRS while the state of the other memristor (G_R) remains in its initial low resistance state. The difference of the initial memristor resistances is amplified and eventually forces memristor states to split into a distinct HRS or LRS. Therefore, in the proposed circuit, the unstable bits [12], in which two resistances are so close that the response becomes fluctuated upon the change of environment, are never generated. All cells generate stable response, which eliminates the use of error correction schemes, such as that in [13].

After the write operation, the CPUFs are split into two CPUFs: CPUF$_\alpha$ and CPUF$_\beta$. The responses of these CPUFs, r_α and r_β, are generated based on the state of the memristors: 1 for HRS and 0 for LRS by a simple comparator with a reference voltage V_ref. In this circuit, r_α and r_β are mutually



(a) A CPUF bit pair.

(b) Write mode.

Fig. 2. Bit pair of the proposed CPUF.

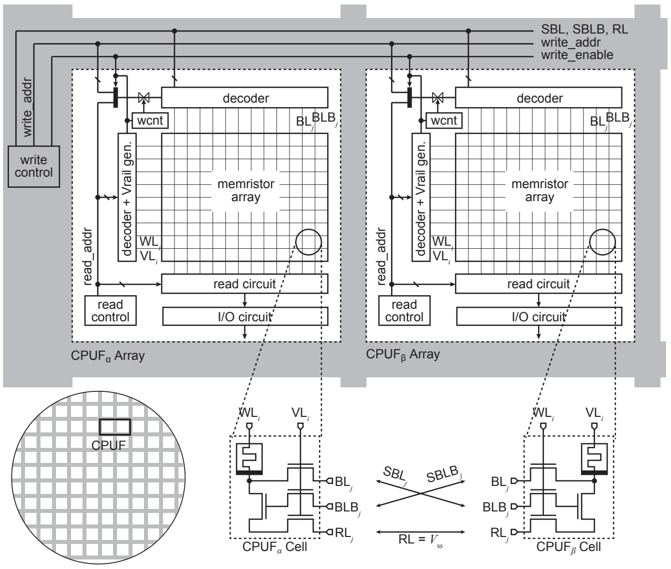


Fig. 3. Schematic diagram of a coupled-inverter CPUF. A CPUF pair on a wafer (bottom left) is shown. Through the shared wires, a CPUF cell forms a cross coupled inverter during the write operation. The write controll circuit is located on the scribe line (shaded), and thus eliminated when each CPUF is separated by die cutting.

in inverted states, i.e., $r_\alpha = \overline{r_\beta}$, hence, these responses can be considered *equivalent*.

C. Coupled-inverter Array

The arrayed implementation of the proposed coupled-inverter CPUF is shown in Fig. 3. The top schematic diagram contains two arrays. The two CPUF chips are fabricated while sharing a write control circuit and shared wires that are in the scribe line. Once the CPUF is diced as individual chips, the read address generator, which selects a memristor cell, serves as a challenge, and the readout of the specified address becomes the response. Although this implementation of CPUF circuit is considered as a weak PUF [8], [13] having a limited number of CRPs, the number of CRPs can be increased exponentially by implementing the CPUF with a pseudo random function circuit, such as [14], [15], when a large CRP space is necessary.

Each CPUF array consists of a memristor array, read control circuit, row decoder, Vrail generator, column decoder, and input/output circuits. A write control circuit is implemented for a pair (or n-copies) of CPUFs that share an equivalent response. The write circuit is only used when setting the values of the CPUF arrays, and is later destroyed with dicing. These circuits are initially fabricated as one chip, indicated in the wafer map at the bottom left in the figure. A write protection circuit, which disconnects wire ends from the internal array, prevents rewriting of the memristors. Additionally, there is a write-counter circuit, `WCNT`, which monitors the number of assertions of the `write_enable` signal. Once `WCNT` detects the rewriting attack through the exposed wires at the edge of the chips, the alarm signal is activated and the operation of the CPUF is disabled.

This schematic diagram also shows the detailed wire connections in the cell circuits formed between the paired memristor array. All cell circuits are designed identical, and the global connection is realized by the shared wires, `SBL` and `SBLB`, forming the pseudo inverter. The difference with those in Fig. 2a is the three switch transistors that determine the write/read modes. The input and output of the cell are connected to the bus lines running vertically, `BL` and `BLB`, which are shared with the cells located in the same column. These bus lines are then connected to `SBL` and `SBLB` that are running across all CPUFs. Here, the cells are cross coupled, as shown in Fig. 2a. The connection of `SBL` and `SBLB` are twisted so that they are respectively connected with `BLB` and `BL`, with which the inverters are connected back-to-back. A supply voltage, V_{dd}, are provided to the pseudo inverters and switch inputs through `VL` and `WL` to enable response determination of all the cells in a row-parallel manner.

In the write mode, the conductance of the memristors are written by a row. First, `SBL` and `SBLB` are reset to V_{ss}. Then, in writing i-th row of CPUF$_\alpha$ and CPUF$_\beta$, the row decoder and Vrail generator activate the cells in the i-th row through $\mathtt{WL}_i$, and V_{stress} is applied through $\mathtt{VL}_i$.

In the read mode, the responses are generated row by row. The select signal $\mathtt{WL}_i$ is asserted to enable i-th row. `RL` is now connected to the resistor (possibly realized by a diode-connection of a transistor) in the read circuit to generate the output voltage V_{out} corresponding to the conductance of the memristor. In this operation, a supply voltage is applied to `BLB` in order to turn on the pull-down transistor of the CPUF cell. V_{out} of each column in the activated row is compared with a reference voltage, V_{ref}, to generate the response.

D. Design considerations

Adopting the helper data has become a common practice to improve the stability of the response of PUFs [13]. However, the hardware overhead required for such function is unignorable and the security may be compromised due to possible information leakage associated with the helper data. In general, this stability issue originates from the fact that the analog output signal, which is given to a comparator to determine a response bit either 0 or 1, tends to concentrate about the comparator threshold. The proposed CPUF is free from this issue by the circuit structure. The distribution of the analog output of the proposed circuit is bi-modal as opposed to the conventional single Gaussian. The simulations in the next section will reveal that the gap between the two distributions

TABLE I
EQUIVALENCE, RANDOMNESS, DIFFUSENESS, UNIQUENESS, CTW COMPRESSION, AND NIST TESTS

Array	Party	E	R	D	U	CTW
		Frequency	Cumulative	Sums	Runs	
8×8	8	1.000	0.974	0.994	0.991	1.049
		-	-	-	-	
16×16	2	1.000	0.998	0.999	0.990	1.017
		1.00	1.00	1.00	0.97	

(associated with HRS and LRS) of the analog output exceeds the half of the supply voltage with no overlap. Hence, the application of ECC circuits can be safely eliminated.

IV. EVALUATION

The performance of the proposed coupled-inverter CPUF is evaluated through simulations using a commercial circuit simulator [16] with a commercial 65 nm process library, and a memristor model based on the actual device-measurement results [17]. A Gaussian distribution is assumed for the variation of the threshold voltages of the MOS transistors. The initial conductance of the memristor is determined by changing the model parameter gap [18], which is the length of high-impedance region, to follow a Gaussian distribution.

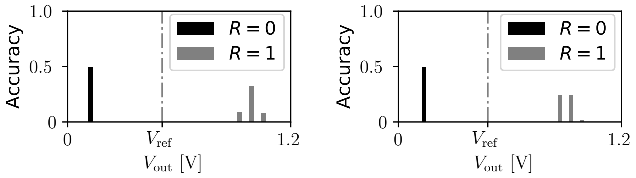
The proposed CPUF is evaluated in two configurations: 8×8 array of 8-parties and 16×16 array of 2-parties. In both configurations, the optional SRAM circuit is implemented and the SRAM-based assist operation is performed in the write mode. A 100 CPUF instances for each configuration were virtually fabricated and evaluated in terms of three criteria: equivalence (E), unpredictability, and reliability.

The equivalence is a novel criterion that we introduce to quantify the performance of the CPUFs, in which the matching rate between the corresponding responses of the CPUFs are evaluated. It is calculated as

$$E = \frac{1}{C} \sum_{c=1}^{C} g(f_\alpha(c), f_\beta(c), f_\gamma(c), ...), \tag{3}$$

where C is the total number of challenges and g is the indicator function that returns 1 (0) when the responses of the corresponding CPUFs are all equivalent (the responses of at least one CPUF are not equivalent).

The unpredictability is evaluated in five criteria: randomness (H), diffuseness (D), uniqueness (U) [19], context-tree weighting (CTW) [20] compression and, NIST tests [21]. The randomness represents the number of appearances of 0's and 1's in the response of a PUF. The diffuseness represents whether a single PUF returns different responses to different challenges. The uniqueness represents whether different PUFs return different responses for the same challenges. Each metric takes a value between 0 and 1, where 0 is the worst and 1 is the best. The CTW compression represents an upper bound of an entropy of the PUF's responses [22]. The NIST tests are the set of randomness tests [21], represent unpredictability of PUF's responses [23]. The result is shown as the rate of PUF instances that passed the tests.



(a) 8×8 array in 8-party.

(b) 16×16 array in 2-party.

Fig. 4. Distributions of the output voltage.

The equivalence and unpredictability are summarized in Table I. The equivalence E has been evaluated as ideal value of 1.0, meaning that the proposed CPUFs always generate equivalent responses. The values of R, D, and U are also very close to the ideal value, and the compressibility of CTW has been higher than 1.0. These results indicate that the proposed CPUF achieves very good performance as a standalone PUF, while there is a perfect match between the responses of CPUFs. The CPUF instance is practically as good as and indistinguishable from the conventional PUFs.

As for the NIST tests, three tests: frequency, runs and cumulative sums, are applied to the 16×16 array as they require more than 100 bits. According to [21], the confidence interval is 0.96015 at the significance level of $\alpha = 0.01$ for the binary sequences, $m = 100$. The pass ratios in Table I of all tests are higher than the threshold confidence level. Thus, the responses of the proposed CPUF are considered sufficiently unpredictable.

Finally, the reliability of the proposed CPUFs was evaluated for the temperature range of -40°C and $+100$°C being the response of the nominal temperature of 25°C as the reference. The reliability is also evaluated for the read supply voltage range of ± 0.24 V (20%) against the supply voltage of 1.2 V. No response has been changed in any of the above environments. Excellent reliability of the proposed circuit has been observed, owing to the large voltage margin for the output $V_{\rm out}$ of the read circuit as shown in Fig. 4.

V. CONCLUSION

In this paper, we proposed a novel concept of CPUF that realizes decentralized authentication without CRP database. A circuit realizations, coupled-inverter CPUF have been presented and evaluated. Through the circuit simulations, it was shown that the responses of the CPUFs are equivalent under process variations of memristors and MOSFETs. The existing metrics for conventional PUFs were also close to ideal values.

ACKNOWLEDGMENT

The authors acknowledge support through the activities of VDEC, the University of Tokyo in collaboration with Synopsys, Inc.

References

[1] B. Gassend *et al.*, "Silicon physical random functions," in *Proc. Computer and Communication Security Conf.*, 2002, pp. 148–160.

[2] G. E. Suh *et al.*, "Physical unclonable functions for device authentication and secret key generation," in *Proc. DAC*, 2007, pp. 9–14.

[3] A. Stanciu *et al.*, "Analysis and evaluation of PUF-based SoC designs for security applications," *IEEE Trans. Industrial Electronics*, vol. 63, no. 9, pp. 5699–5708, Sept 2016.

[4] C. Brzuska *et al.*, "Physically uncloneable functions in the universal composition framework," in *Proc. CRYPTO*, 2011, pp. 51–70.

[5] R. Ostrovsky *et al.*, "Universally composable secure computation with (malicious) physically uncloneable functions," in *Proc. EUROCRYPT*, 2013, pp. 702–718.

[6] U. U. Rührmair *et al.*, "PUFs in security protocols: Attack models and security evaluations," in *Proc. IEEE Symp. Security and Privacy*, 2013, pp. 286–300.

[7] A. P. Johnson *et al.*, "A PUF-enabled secure architecture for FPGA-based IoT applications," *IEEE Trans. Multi-Scale Computing Systems*, vol. 1, no. 2, pp. 110–122, 2015.

[8] U. Rührmair *et al.*, *Security Based on Physical Unclonability and Disorder*. Springer New York, 2011, ch. 4, pp. 65–102.

[9] L. O. Chua, "Memristor—the missing circuit element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[10] D. B. Strukov *et al.*, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[11] F. M. Bayat *et al.*, "Phenomenological modeling of memristive devices," *Applied Physics A*, vol. 118, pp. 779–786, March 2015.

[12] M. Yu *et al.*, "Secure and robust error correction for physical unclonable functions," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 48–65, 2010.

[13] C. Herder *et al.*, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[14] M. Bhargava *et al.*, "An efficient reliable PUF-based cryptographic key generator in 65nm CMOS," in *Proc. DATE*, 2014, pp. 1–6.

[15] R. Liu *et al.*, "Extending 1kb RRAM array from weak PUF to strong PUF by employment of SHA module," in *2017 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, Oct 2017, pp. 67–72.

[16] *HSPICE User Guide: Basic Simulation and Analysis Version L-2011.09*, Synopsys, Inc., 2011.

[17] P. Y. Chen *et al.*, "Compact modeling of RRAM devices and its applications in 1T1R and 1S1R array design," *IEEE Trans. Electron Devices*, vol. 62, no. 12, pp. 4022–4028, 2015.

[18] S. Yu *et al.*, "A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation," *Advanced Materials*, vol. 25, no. 12, pp. 1774–1779, 2013.

[19] Y. Hori *et al.*, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs," in *Proc. Int'l Conf. Reconfigurable Computing*, 2010, pp. 298–303.

[20] F. M. J. Willems *et al.*, "The context-tree weighting method: Basic properties," *IEEE Trans. Information Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[21] L. E. Bassham *et al.*, "SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," Tech. Rep., 2010.

[22] Stefan Katzenbeisser, Ünal Kocabas, Vladimir Rožic, Ahmad-Reza Sadeghi, Ingrid Verbauwhede and Christian Wachsmann, "PUFs: Myth, fact or busted? a security evaluation of physically unclonable functions (PUFs) cast in silicon," in *Prof. CHES*, 2012, pp. 283–301.

[23] D. E. Holcomb *et al.*, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. Computers*, vol. 58, no. 9, pp. 1198–1210, Sept 2009.