Received: 20 February 2019 | Revised: 15 July 2019 | Accepted: 28 August 2019

DOI: 10.4218/etrij.2019-0081

ORIGINAL ARTICLE

ETRI Journal WILEY

Design and implementation of an improved MA-APUF with higher uniqueness and security

Bing Li | **Shuai Chen** | **Fukui Dan**

School of Microelectronics, School of Cyber Science and Engineering, SEU-FiberHome Joint Research Center, Southeast University, Nanjing, China

Correspondence
Shuai Chen, Schoool of Microelectronics, School of Cyber Science and Engineering, SEU-FiberHome Joint Research Center, Southeast University, Nanjing, China.
Email: chenshuai_ic@seu.edu.cn

Funding information
Shenzhen Science, Technology and Innovation Commission (SZSTI), Grant/Award Number: JCYJ20170817115500476; National Natural Science Foundation of China, Grant/Award Number: 61571116; Scientific Research Foundation of the Graduate School of Southeast University, Grant/Award Number: YBPY1869

An arbiter physical unclonable function (APUF) has exponential challenge-response pairs and is easy to implement on field-programmable gate arrays (FPGAs). However, modeling attacks based on machine learning have become a serious threat to APUFs. Although the modeling-attack resistance of an MA-APUF has been improved considerably by architecture modifications, the response generation method of an MA-APUF results in low uniqueness. In this study, we demonstrate three design problems regarding the low uniqueness that APUF-based strong PUFs may exhibit, and we present several foundational principles to improve the uniqueness of APUF-based strong PUFs. In particular, an improved MA-APUF design is implemented in an FPGA and evaluated using a well-established experimental setup. Two types of evaluation metrics are used for evaluation and comparison. Furthermore, evolution strategies, logistic regression, and *K*-junta functions are used to evaluate the security of our design. The experiment results reveal that the uniqueness of our improved MA-APUF is 81.29% (compared with that of the MA-APUF, 13.12%), and the prediction rate is approximately 56% (compared with that of the MA-APUF (60%-80%).

KEYWORDS
MA-APUF, ML attack, physical unclonable functions, uniqueness

1 | INTRODUCTION

Currently, low-cost electronic devices are applied pervasively to the Internet of Things. The security of these devices is heavily dependent on the security of digital secret keys stored in nonvolatile memory (NVM). However, NVM-based key storage mechanisms are vulnerable to physical attacks (eg, invasive, semi-invasive, and side-channel attacks), resulting in several security and privacy issues. As a substitute for the traditional key storage scheme, physical unclonable functions (PUFs) [1] offer an effective countermeasure to address the security issues aforementioned.

A PUF can be regarded as a physical random function that reflects the relationship between a set of challenges (inputs) to a set of particular responses (outputs); it is based on unpredictable and uncontrollable manufacturing variations. This unique physical feature can be regarded as the "fingerprints" of integrated circuits and has been used as a promising lightweight security primitive in many security-related fields, such as authentication [2], random number generator [3], key storage [4], and other basic applications [5].

The inputs and outputs of a PUF are called challenge-response pairs (CRPs). According to the number of CRPs, a PUF is categorized as a weak PUF and a strong PUF in [6]. Weak PUFs, such as the SRAM PUF [7], butterfly PUF [8], and ring oscillator PUF (ROPUF) [9], have few CRPs; even in the extreme case, only one CRP is used and is primarily applied to key storage and generation schemes. Strong PUFs, on the contrary, can generate a large number of CRPs,

for example, the standard arbiter PUF (APUF) [10], XOR arbiter PUF (XOR APUF) [11], feed-forward arbiter PUF (FF-APUF) [12] and lightweight secure PUF (LSPUF) [13]. Typically, strong PUFs can be used in authentication systems.

PUFs are regarded as promising low-cost security primitives, but their response unreliability limits their practical applications. A number of methods have been proposed to mitigate errors in PUF response, including the BCH code [11] and soft-decision decoding [14]. However, these methods incur area and power overhead. Vijayakumar and others [15] constructively applied machine learning (ML) modeling and used models to predict and then discard CRPs that would be unreliable with respect to noise and aging. This method is not suited to PUFs that cannot be modeled by (ML).

With further research on PUF techniques, some attacks toward PUFs have been proposed, such as cryptanalysis [16], ML-based modeling attacks [6,17], and side-channel attacks combined with modeling attacks [18–20]. ML-based modeling attack is regarded as powerful attack for strong PUFs. After trained by sufficient CRPs, a numerical delay model of a strong PUF can be built, which can be applied to predict the corresponding output for any new input. Ulrich and others [6] were the first to build an APUF model by ML attack with extremely high accuracy.

Logistic regression (LR) and evolution strategies (ES) are two ML techniques used in ML-based modeling attacks. In a previous study [17,19,20,22], these methods were utilized to evaluate the resistances of APUF, ROPUF, XOR-PUF, FF-APUF, and LSPUF in operating modeling attacks. Nevertheless, ML-based attacks perform poorly with increasing number of XORs and bit-length of PUFs. Side-channel information can be used to simplify XOR APUFs and LSPUFs to individual APUFs; thus, attacking complexity can be reduced significantly. This approach has been proven efficient in [19–21].

As shown in Table 1, numerous countermeasures against ML-based attacks can be implemented—changing the structure of strong PUFs, adding nonlinearity blocks, and obfuscating the mapping relationship between challenge and response by some protocols. These countermeasures exhibit excellent resistance to ML attacks. However, a few aspects require improvement. For example, the results of [15,23] are based on the simulations for application-specific integrated circuits (ASICs). A double arbiter PUF (DAPUF) reduces the steadiness, as the structural complexity increases [24]. Furthermore, for the obfuscating method, a random number generator (RNG) is used to randomize the mapping relationship between challenge and response, which results in a high-area overhead [25–27]. Herein, we focus on strong PUFs with improved structures.

An excellent structure-improved APUF is the multiplexing aware arbiter physical unclonable function (MA-APUF) proposed in [28]. The MA-APUF adopts a new method that combines sectors and delay detectors, thus resulting in high steadiness, randomness, and correctness but poor uniqueness. To improve the uniqueness and ML attack resistance, we herein propose an improved MA-APUF. More specifically, our contributions are as follows:

- We analyze the causes of low uniqueness of MA-APUFs and propose foundational principles for uniqueness improvement.
- Based on the aforementioned analysis, we propose an improved MA-APUF design. The changes include (a) inserting inverters; (b) changing the delay detection and

TABLE 1 Summary of countermeasures against ML attack

Types of PUFs	Types of noninvasive attacks	Strengths	Weaknesses
PUF using nonlinear tables [22]	SVM, LR, ES, Bagging, Gradient Boost	Provide better resistance against all attacks, especially LR	Relatively higher area overhead
Nonlinear current PUF [23]	SVM	Show excellent security properties measured in terms of its resistance to machine learning attacks	Based on simulation for application-specific integrated circuits
Nonlinear VTC PUF [15]	SVM	Improve ML attack resistance. The proposed circuit is simple, exhibits high uniqueness and randomness. Enhance PUF reliability	Based on simulation for application-specific integrated circuits
Double Arbiter PUF [24]	SVM	Enhance the unpredictability of responses. Improve the uniqueness and randomness.	Reduces the steadiness
RPUF [25]	Heuristic algorithm	Effectively resist modeling attacks, with negligible effects on uniformity, uniqueness, and reliability.	Require additional hardware resources because of adopting RNG components
OB-PUF [26]	LR	Enhance security of the proposed OB-PUF. Reduce hardware complexity.	Require additional hardware resources because of adopting RNG components
Slender PUF [27]	ES	Provide a much higher level of resiliency against all known ML attacks	Need TRNGs and micro-controllers. Require additional hardware resources.

response generation model; and (c) adding a tuning block, which improves the uniqueness and increases resistance to modeling attacks.
- The improved MA-APUF design is implemented efficiently in field-programmable gate arrays (FPGAs) and evaluated using a well-established experimental setup. The test results show that our design has a better uniqueness of 81.29% (43.6%) compared with the MA-APUF. In addition, a better ML prediction rate of 56% is obtained compared with the MA-APUF.

The remainder of this article is organized as follows. We briefly present the background and related studies in Section 2, including the APUF, modeling attacks and MA-APUF structure. Section 3 details the proposed R-XOR APUF. Experiments on FPGAs are described in Section 4. Finally, we conclude the article in Section 5.

2 | BACKGROUND AND RELATED STUDIES

2.1 | Arbiter PUF

As shown in Figure 1, an n-stage APUF comprises n paralleled multiplexer pairs and one flip-flop. A rising edge signal is simultaneously provided to the first paralleled multiplexer pairs. The path where two signals are propagated can be configured by an external n-bit challenge $\mathbf{C}(C_1 \cdots C_n)$. For each stage, when the challenge bit is "0," the signals travel straight; when the challenge bit is "1," the signals are crossed. At the final stage, the arbiter generates a 1-bit response by determining which signal arrives first. The response depends on the differential delay of two paths, which is based on the manufacturing process variations of the transistors and wires. An APUF with n stages has 2^n CRPs.

However, two primary shortcomings exist in APUFs: vulnerability to modeling attacks and low uniqueness. The circuit of an APUF is simple, and the delays caused by the individual components of the signal paths are linear additive. Hence, attackers can build a module and learn the delay parameters of each stage to emulate the challenge-response behavior of APUFs from many CRPs using software. Furthermore, the APUF always suffers from the biasing in FPGA implementation, which is another factor for the low uniqueness and security.

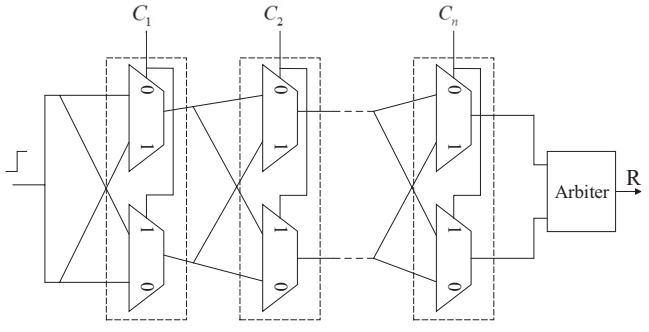

FIGURE 1 Basic APUF design

2.2 | APUF model

An APUF can be emulated or modeled by exploiting a linear additive model because a 1-bit response is generated by comparing the summation of each delay segment for each stage (two 2-1 multiplexers) depending on the challenge **C** [6,17]. The model allows us to predict the response for a new challenge. We can use the following notation to model an APUF.

$$\vec{\omega} = (\omega^1, \omega^2, \ldots, \omega^n, \omega^{n+1}), \tag{1}$$

$$\vec{\Phi}(\vec{C}) = (\Phi^1(\vec{C}), \ldots, \Phi^n(\vec{C}), 1)^{\mathrm{T}}, \tag{2}$$

$$r = \mathrm{sgn}(\vec{\omega} \times \vec{\Phi}), \tag{3}$$

where $\omega^1 = (\delta_1^0 - \delta_1^1)/2$, $\omega^i = (\delta_{i-1}^0 + \delta_{i-1}^1 + \delta_i^0 - \delta_i^1)/2$ for $\omega^{n+1} = (\delta_n^0 + \delta_n^1)/2$, $i = 2, \ldots, n$. In this model, $\vec{\omega} \times \vec{\Phi}$ expresses the final delay difference between the upper and lower paths, where $\vec{\omega}$ and $\vec{\Phi}$ are vectors of dimension $n + 1$. sgn is the sign function. We denote $\vec{\omega}$ as the delay for the subcomponents in the APUF stages. Vector $\vec{\Phi}$, that is, the challenge feature, is a function of the applied n-bit challenge $\mathbf{C}(C_1 \cdots C_n)$. $\delta_i^{0/1}$ are the notations of the delay in the i-th stage for the uncrossed ($C_i = 0$) and crossed ($C_i = 1$) signal paths through the multiplexers, respectively. The goal of an attacker wass to acquire the parameter $\vec{\omega}$, according to the vector $\vec{\Phi}$ calculated by collected CRPs. Consequently, the attacker can predict unknown responses using the model, which could result in a successful cloning of the target APUF.

2.3 | Employed ML methods

2.3.1 | Logic regression

LR is a well-investigated supervised ML framework that has been adapted to build the APUF model in [6] and [17]. In this framework, each challenge of the APUF has a corresponding probability $p(C,t|\vec{\omega})$, which expresses the possibility of the response $t \in \{-1, 1\}$. $\{-1, 1\}$ is used instead of $\{0, 1\}$ to ease calculations. The vector $\vec{\omega}$ thereby encodes the relevant internal parameters, for example, the particular delays of individual PUFs into the coefficient of the model. The probability is calculated by the logistic sigmoid acting on a function $f(\vec{\omega})$ parameterized by the vector $\vec{\omega}$ as

$p(C,t|\vec{\omega}) = \sigma(tf) = (1+e^{-tf})^{-1}$. Therefor, the decision boundary of equal output probabilities is $f = 0$. For a given training set M of the CRPs, the delay time vector can be obtained by the following formula:

$$\begin{aligned}\hat{\vec{\omega}} &= \text{argmin}_{\vec{\omega}} l(M,\vec{\omega}) \\ &= \text{argmin}_{\vec{\omega}} \sum_{(C,t)\in M} -\ln(\sigma(tf(\vec{\omega},C))).\end{aligned} \tag{4}$$

An analytical solution cannot be used to acquire the optimal parameter vector $\vec{\omega}$ directly because an iterative optimization is required. Therefore, the gradient information is used to determine $\vec{\omega}$.

$$\nabla l(M,\vec{\omega}) = \sum_{(C,t)\in M} t(\sigma(tf(\vec{\omega},C))-1)\nabla f(\vec{\omega},C). \tag{5}$$

The RProp gradient descent was used in this study to confirm the optimal parameter vector, which reference to [6].

2.3.2 | Evolution strategy

ES [6] were inspired by the evolutionary adaptation of a population of individuals to certain environmental conditions. Using the ES-based modeling attacks, an attacker imitates the delay model of an APUF implementation by initializing a randomly generated $\vec{\omega}$ (an $n + 1$ dimension delay time vector). The initial $\vec{\omega}$ is treated as a parent to generate many offspring delay vectors based on random mutations. According to a fitness metric based on known CRPs, the best child is selected and used as the parent for the next generation. This process is repeated until a delay time vector of sufficient accuracy is formed.

2.4 | MA-APUFs

Paper [28] proposed a novel MA-APUF based on the APUF. The MA-APUF adopts a new structure and method to generate responses, as shown in Figure 2. In the MA-APUF, each unit is composed of four 4-1 multiplexers and requires a 2-bit challenge to configure four pathways, where rising-edge signals are propagated. The delay detection block is composed of six DFFs (DFF1 ··· DFF6) that are used to detect the delay difference between every pair of signal output. According to the output of each arbiter, the order of arrival of the four signals (a, b, c, d) can be specified. When the outputs {$r1$, $r2$, $r3$, $r4$, $r5$, $r6$} are {1, 1, 1, 1, 1, 1}, the order of signal arrival should be $a > b$, $a > c$, $a > d$, $b > c$, $b > d$, and $c > d$. Therefore, the order is (a), (b), (c), and (d). Each of the four signals can be encoded into a 2-bit identification number, for example, signals (a), (b), (c), and (d) encoded into 00, 01, 10, and 11, respectively. The earliest and latest arriving signals are coupled to generate a 4-bit response. When the order of signal arrival is (a), (b), (c), and (d), the response is 0011. This response generation method improves the randomness, steadiness, and correctness of the MA-APUF structure.

The MA-APUF achieved substantially improved randomness, steadiness, and correctness [28]. Furthermore, this structure could afford an improved resistance to the ML attack, which was not shown in [28]. In our study, two ML

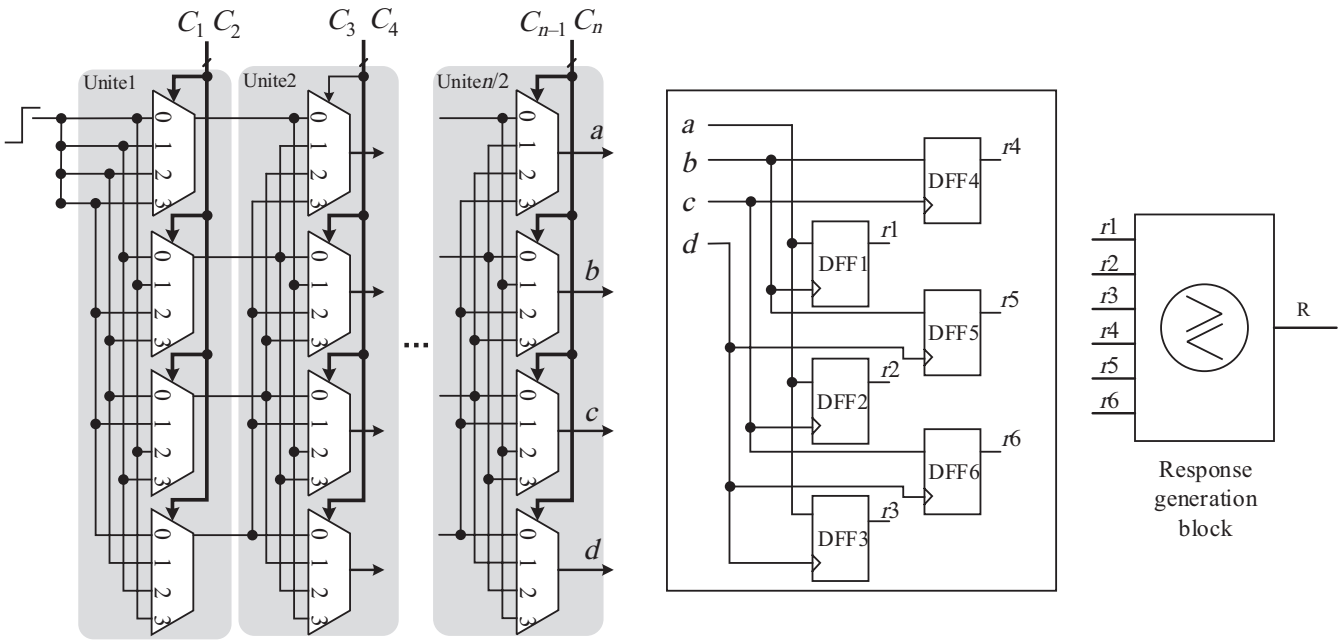

FIGURE 2 Architecture of MA-APUF

algorithms, LR and ES, were used to evaluate the security of the MA-APUF.

3 | FOUNDATIONAL PRINCIPLES FOR UNIQUENESS IMPROVEMENT

The modeling-attack resistance of strong PUFs is attributed to the complexity of the challenge to response mapping in each chip and the uniqueness of the mapping across chips. Although the modeling-attack resistance of the MA-APUF was improved significantly by architecture modifications, the response generation method of the MA-APUF results in poor uniqueness. This issue exists in many APUF-based designs. In this section, we demonstrate several foundational principles to improve the uniqueness of APUF-based strong PUFs. In particular, we present the MA-APUF as an example.

3.1 | Cascading block architecture

Circuit blocks that are cascaded exhibit process variations: a unique feature that is central to the uniqueness of PUFs. However, an asymmetrical delay pathway can hide manufacturing variations, thus resulting in low uniqueness.

For the MA-APUF, four types of signal pathways exist in each block, as shown in Figure 3A. When a challenge is 00, 01, or 10, any two of the four pathways are approximately equal. The responses of these challenges are determined by the manufacturing variations. However, for challenge 11, two long pathways (*a* and *d*) and two short pathways (*b* and *c*) exist, both of which have the same transmission path. Therefore, as shown in Table 2, the outputs of DFF1, DFF2, DFF5, and DFF6 will be 0, 0, 1, and 1, respectively, which are determined by programmable routing instead of manufacturing variations. For challenges 00, 01, and 10, where any two of the four pathways are approximately equal, the number of response combinations for the MA-APUF is 12 (A_4^2). However, for challenge 11, the number of combinations is reduced to eight ($C_2^1 \times C_2^1 \times 2$), thus deceasing the uniqueness and security significantly. Only when *n* "11" and *n*/2 "01" exist in the challenge and their arrangements resemble those in Figure 3B, the four pathways will be symmetrical, and the response will be determined by manufacturing variations. Furthermore, paper [28] indicates that the signal arrivals method of the MA-APUF contains logical contradictions. The uniform disposes of logical contradictions will partially reduce the uniqueness.

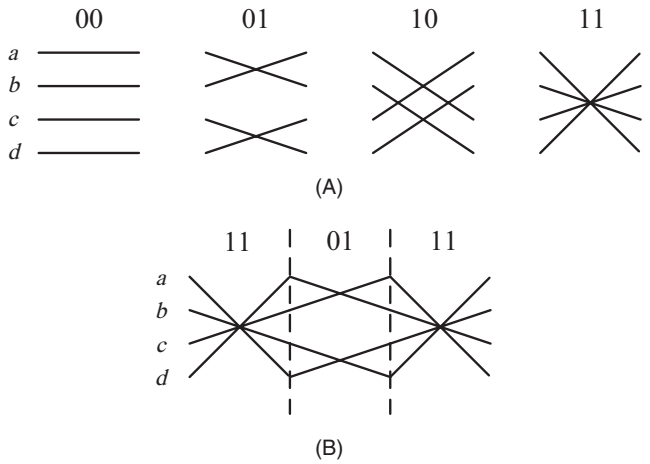

FIGURE 3 Signal pathway for different challenges: (A) four types of signal pathways, (B) symmetrical pathway.

TABLE 2 Response of MA-APUF

Response	Value
r1	0
r2	0
r3	0
r4	x
r5	0
r6	0

3.2 | Routing problems on FPGAs

Figure 4 shows the placement and routing of the basic APUF on FPGAs. Owing to the fixed FPGA units, the routing at the first stage and the arbiter of the APUF are asymmetric, although the others exhibit a symmetrical pattern. For some challenges, the asymmetricity is so large that the manufacturing variations cannot be reflected in the responses. For the APUFs, Majzoobi and others [29] proposed using a tuning block to alleviate the inequality of the wire length and cancel

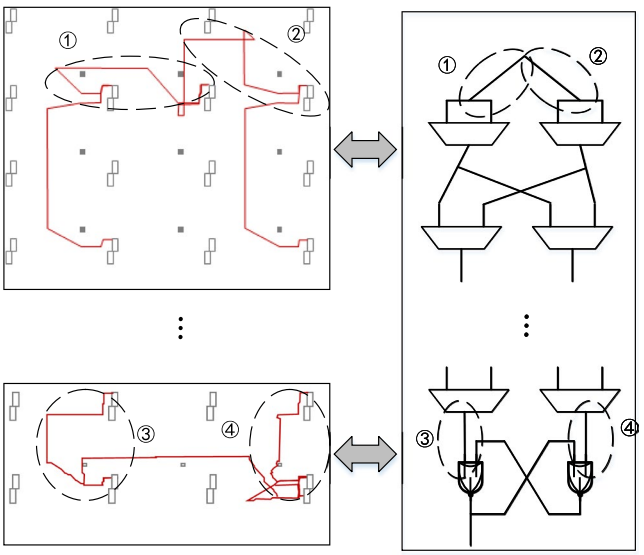

FIGURE 4 Routing problem in APUF implementation on FPGA

out the delay bias caused by routing asymmetry. First, the bias of the APUF can be evaluated using a large number of CRPs. Subsequently, the asymmetry delay line is tuned by adding some delay blocks in one of the delay lines. Finally, the bias of the new circuit is tested. If the test results are satisfactory, restart from step 1 for the next set of parameters.

3.3 | Cascading block architecture

The circuit blocks, such as the inverters, have cascaded harness process variations, which is central to the uniqueness of PUFs [22]. Paper [22] established the foundational principles that guide ML attack resistance in strong PUFs that are based on cascaded structures. The switches perform selection, while the cascaded blocks provide a transfer function that introduces process variations. Many of the previously proposed strong PUFs such as the nonlinear current PUF and nonlinear VTC PUF have a similar structure and exhibit excellent uniqueness [15,23]. Hence, the uniqueness of the MA-APUF can be further improved using nonlinear blocks.

4 | THE IMPROVED MA-APUF

In this section, we use the MA-APUF as an example to demonstrate how the uniqueness of delay-based strong PUFs can be improved based on the strategy aforementioned. Furthermore, the improved uniqueness will increase the resistance of ML attacks.

For the problem described in Section 3.1, one promising solution is to modify the arbiter block that generates the response using the symmetric signal pathway. For example, for challenge 11 in Figure 5, the two long pathways are selected to generate response $r1$ and two short pathways are selected for $r2$. Therefore, for any challenge, this response generation method always compares the symmetrical pathways ((*a*) and (*d*); (*b*) and (*c*)), which adequately reflects the process variations and increases the uniqueness. Furthermore, the response does not exhibit logical contradictions. For routing problems on FPGAs, the tuning block can be used to alleviate the inequality of the wire length. Compared with the MA-APUF, only two responses ($r1$ and $r2$) are to be tuned in our proposal, which decreases the difficulty of equalizing the wire length. Furthermore, the uniqueness can be further improved by nonlinear blocks. In our proposal, some inverters can be inserted between every two-stage multiplexer.

The architecture of the improved MA-APUF is illustrated in Figure 5. For an n-stage IMA-APUF, $n/2$ units exist. Each unit consists of four 4-1 multiplexers and four inverters. The intra-stage network between inverters and multiplexers is shown in Figure 5B. The output of inv0 is linked to the first input of mux0, the second input of mux1, the third input of mux2, and the fourth input of mux3. Similarly, the output of inv1 is linked to the second input of mux0, the first input of mux1, the fourth input of mux2, and the third input

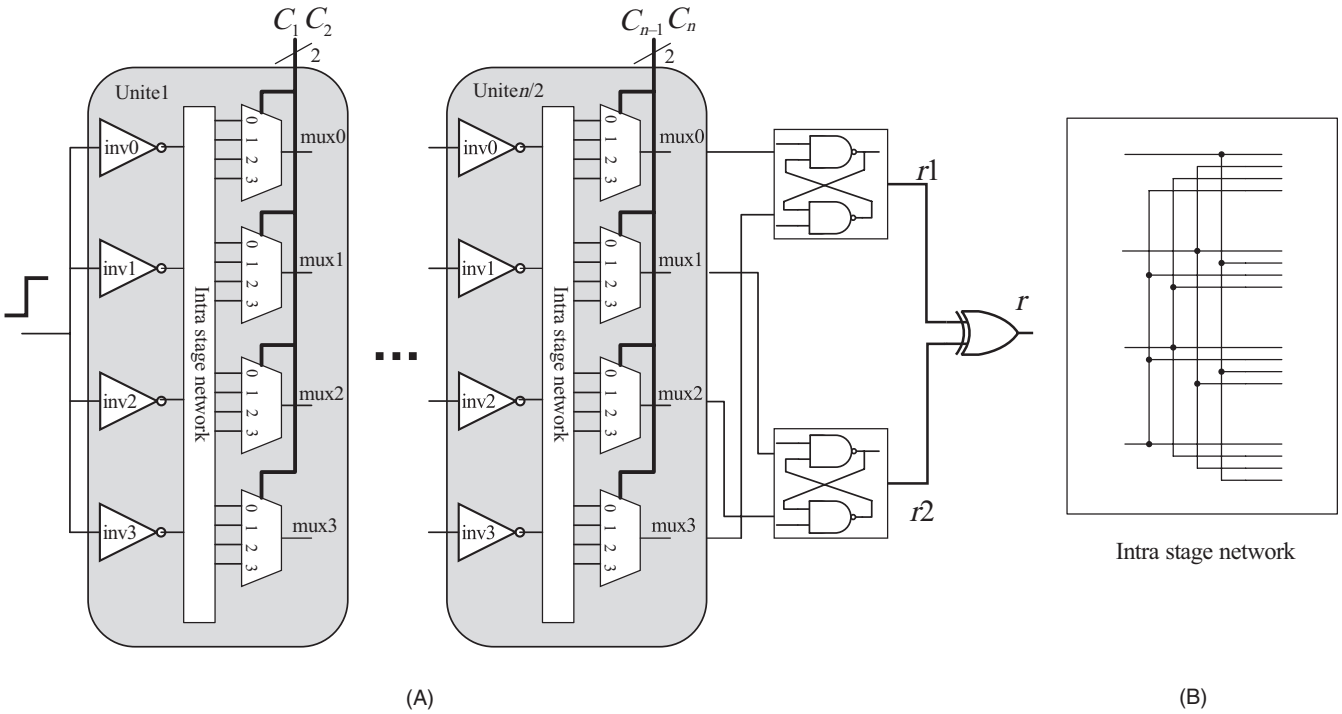

FIGURE 5 Architecture of the improved MA-APUF: (A) schematic of the improved MA-APUF, (B) intra stage network of the improved MA-APUF.

of mux3; the output of inv2 is input to the third input of mux0, the fourth input of mux1, the first input of mux2, and the second input of mux3; the output of inv3 is input to the fourth input of mux0, the third input of mux1, the second input of mux2, and the first input of mux3. Two cross-coupled NAND gates are used as the arbiter. The outputs of mux0 and mux3 in the last unit are compared by the first arbiter for generating $r1$, while mux1 and mux2 are arbitrated by the second arbiter for $r2$. The response of the IMA-APUF is generated by XORing the result of the arbiters. The overall architecture comprises four paths where the rising-edge signal is propagated.

The improved uniqueness also contributes to the ML attack resistance. The difficulty in ML attack is in building the right model to simulate the delay behavior. The model of the basic APUF is shown in Figure 6. The model parameter is regarded as the difference of two MUXs in each stage. The structure of the improved MA-APUF is determined by the challenge; thus, it can effectively resist modeling attacks. Figure 7 shows the propagation paths of the APUF, MA-APUF, and our proposal, when the challenge is changed. We use two challenges $\mathbf{C0} = \{0000 \cdots 0000\}$ and $\mathbf{C1} = \{1000 \cdots 0010\}$ as examples. When the challenge is **C0**,

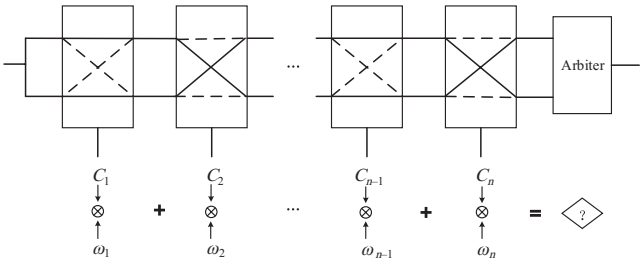

FIGURE 6 Basic APUF design

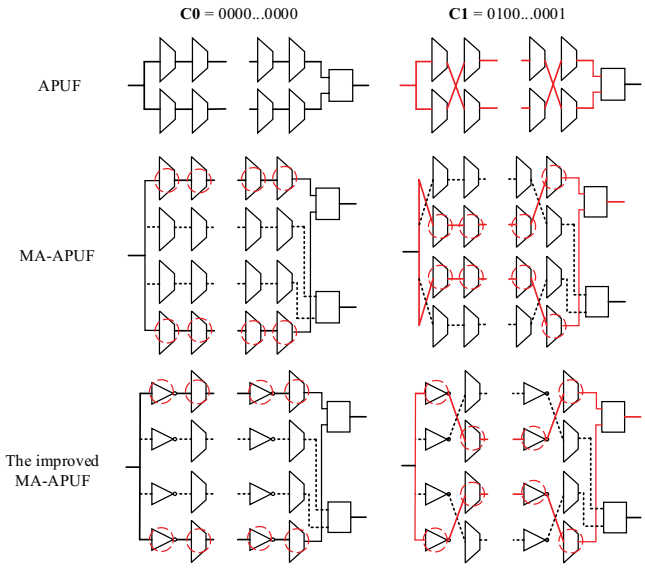

FIGURE 7 Signal propagation path of $\mathbf{C0} = \{0000 \cdots 0000$ and $\mathbf{C1} = \{1000 \cdots 0010\}$

the wires of the structures are expressed using solid black lines. When the challenge is **C1**, the wires are expressed using solid red lines. The components generating the response is marked using red circles. For the traditional APUF, the wire is changed by the challenge. However, for the MA-APUF and IMA-APUF, when the challenge is changed, both the component and the wire are changed.

In the ML attack, a certain number of CRPs should be collected to build the PUF model. For an APUF, two MUXs are fixed at each stage. Therefore, the model parameter is fixed and a unique numerical model corresponding to it exists. However, for the MA-APUF and IMA-APUF, both the wire and the component are changing. Therefore, the model parameter will be unstable. Therefore, it is difficult to build a model based on the universal delay model. Furthermore, the decreased bias and increased nonlinearity of the IMA-APUF can further improve the resistance of ML attacks [22].

5 | EVALUATION SETUP

In our experiments, we implemented an APUF, 2-XOR APUF, MA-APUF, and IMA-APUF (where n is 64) on four Xilinx Virtex-5 FPGA boards. Figure 8 shows the experimental setup. From the Virtex-5 board to the workstation (CPU), the challenges and responses are sent through an Ethernet communication interface at 1 GB/s using the SIRC API based on Mehrdad Majzoobi's public PUF program [29]. The MATLAB scripts on the workstation were used to evaluate the performance of the PUF. Xilinx ISE 14.7 and Xilinx PlanAhead 14.7 were used for logic synthesis and floor planning.

The attacking process is shown in Figure 9. The challenges are generated in a PC and transmitted to the FPGA board. Subsequently, the corresponding responses are collected in the PC. It is assumed that an attacker can obtain some of these CRPs as a training group to model the PUF by legitimate access.

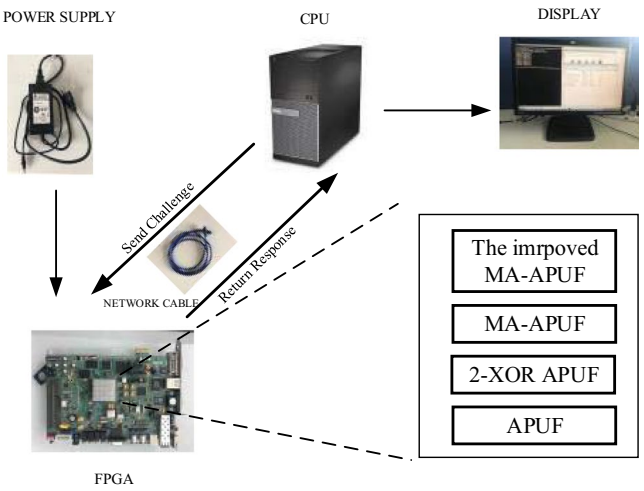

FIGURE 8 Experimental setup

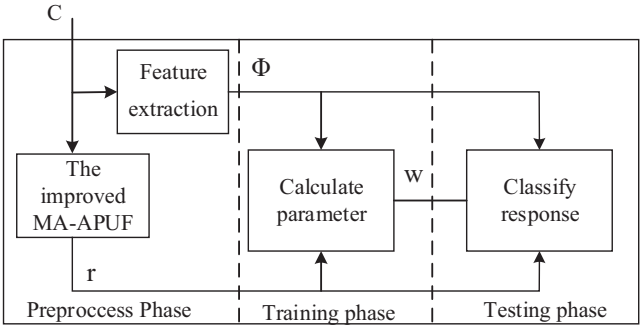

FIGURE 9 Architecture of ML attacks

During the training phase, challenges are transformed into a feature vector in order to simplify the ML process. After feature extraction, a model parameter $\vec{\omega}$ is constructed during the training phase. The accuracy of this model parameter must be tested by comparing the response extracted from the FPGA board and that generated by the PUF model. Finally, our MATLAB code generates a prediction rate.

6 | EXPERIMENTAL RESULTS

In this section, we depict some basic properties of the PUF implementations based on the aforementioned evaluation setup. However, no standards have been established to enable the quantitative performance evaluation of PUFs. In this study, we used two test indicators that are widely accepted and used in prior studies. Maiti and others [30] proposed the following four parameters: uniqueness, reliability, bit-aliasing, and uniformity. The five parameters proposed by Hori and others [32] are uniqueness, randomness, correctness, steadiness, and diffuseness. Figure 10 shows the relation between these parameters. Three parameters from each group have similar definitions.

For the quantitative security evaluation, two widely used ML algorithms, LR and ES, were employed to evaluate the securities of the APUF, 2-XOR APUF, MA-APUF, and the IMA-APUF. To further study the resistance for ML attack, a more powerful evaluation scheme suite was used. This test can be regarded as a provable framework for ML attacks against a PUF family, whose underlying mathematical model is unknown [31].

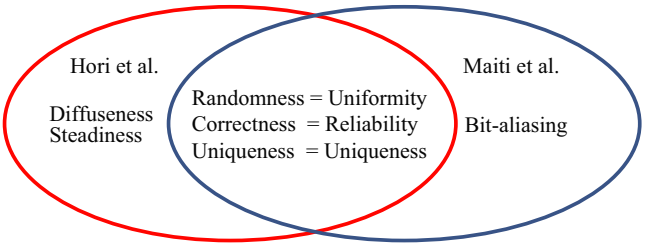

FIGURE 10 Relation between parameters defined by Hori and others [32] and Maiti and others [30]

6.1 | Quantitative performance evaluation

In our experiments, we tested the uniqueness of the improved MA-APUF with 16 implementations on four Xilinx Vertex-5 FPGAs and extracted 8 192 000 responses from each IMA-APUF. As shown in Figure 11, the uniqueness is approximately 43.6%.

The reliability test results of the improved MA-APUF for temperatures from 0°C to 60°C with a step of 5°C are shown in Figure 12. We set the response at 25°C as the reference. The reliability declines when the temperature is extremely high or low. The maximum reliability is >96%, while the minimum reliability is <92.5%. As shown, the average reliability is approximately 94.34%.

Table 3 shows the test results based on the indicator reported in [32]. The uniqueness of the improved MA-APUF is more than six times that of the MA-APUF. An improved uniqueness can significantly improve the authentication accuracy and security of the authentication system based on strong PUFs. The diffuseness is highly similar to the uniqueness in [30]. Therefore, it is also better than MA-APUF to some extent. It is useful when a PUF has a large challenge-response pair space as that of the APUF, where several identifiers can

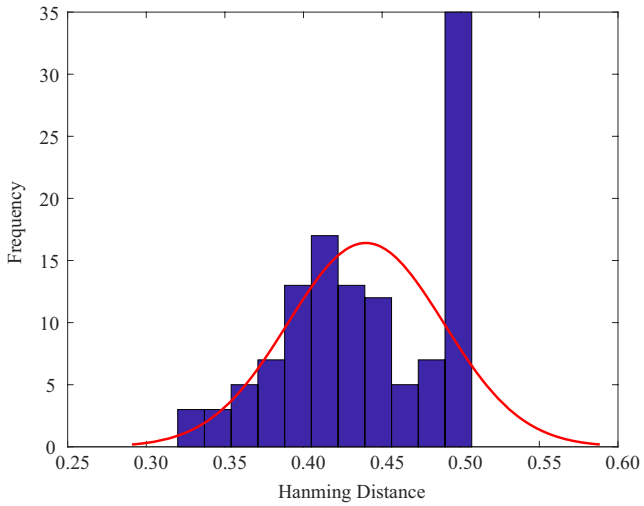

FIGURE 11 Uniqueness evaluation

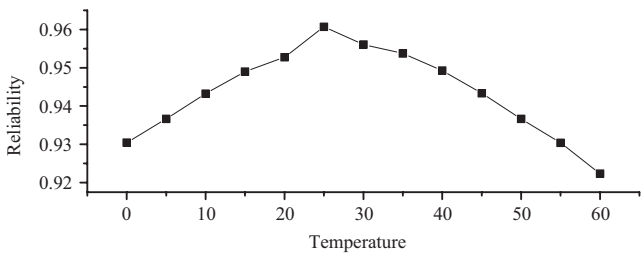

FIGURE 12 Reliability evaluation

TABLE 3 Performance comparison

	Randomness	Steadiness	Correctness	Diffuseness	Uniqueness
APUF	0.8469	0.9848	0.9828	0.9839	0.3675
MA-APUF	0.9912	0.9891	0.9988	0.9689	0.1312
The improved MA-APUF	0.9251	0.9340	0.9240	0.9881	0.8129

be produced from a single chip. Although the uniqueness and security have improved significantly for our proposal, the other characteristics of our proposal degraded slightly (approximately 6%) compared with the MA-APUF. The improved MA-APUF adopts the appropriate response generation method. Symmetrical signal pathways are arbitrated, thus resulting in a small delay difference between the two paths. The small delay difference can be hidden easily by noise, which results in low reliability. Furthermore, the IMA-APUF is less random than the MA-APUF because tuning block cannot completely eliminate the pathway inequality.

Table 4 compares the performance of the IMA-APUF with those of the APUF, DAPUF, RPUF, and OB-PUF based on the method shown in [30]. The improved MA-APUF shows better resistance for ML attacks; 69% of the responses from the 2-1 DAPUF could be predicted [24]. When the author adopted the 3-1 double arbiter PUF, the prediction rate was only 56.75%. However, the steadiness of the 3-1 double arbiter PUF declined to <15%. For the RPUF implemented in FPGAs, the average prediction rate was 64.45% [25]. When the randomization level was 2, the prediction accuracy of the RPUF declined to 57.3%. An OB-PUF was proposed as a primitive for a secure and lightweight PUF-based authentication protocol in [26]. This protocol sends obfuscated challenges to an OB-PUF where the subsequent recovery of the obfuscated challenges by a server (verifier) is guaranteed. This authentication mechanism can prevent an adversary from exploiting model building attacks to break the authentication mechanism and yet allows the server (or verifier) to successfully authenticate OB-PUF-based pervasive devices (or prover). The OB-PUF reported in [33] showed an improved ML resistance, in which the prediction rate was 56.87%. However, the RPUF and OB-PUF require the RNG module to randomize the relationship between challenge and response, which necessitates additional resources. The uniqueness of our proposal is improved compared with those of the APUF and MA-APUF but less than that of the DAPUF. It is significantly difficult to balance the place and route of the APUF. Hence, the DAPUF is used as it can cancel the difference of the wiring bias by competing signals that are propagating on the same route. Therefore, the DAPUF has better uniqueness. Furthermore, the bit-aliasing feature of the IMA-APUF is better compared with that of the APUF, indicating that different chips may not produce nearly identical PUF responses.

6.2 | ML attack resistance

In our tests, LR and ES were executed based on different numbers of training samples and testing samples from 1000 to 50 000 CRPs. Half of these data were used as training samples and the other half as test samples. For each sample, many iterative operations were performed in the attack. The optimal prediction rate of iterative operations was selected as the final accuracy rate.

For the MA-APUF, the response feature was based on the delay comparison for every two in four paths. Two 2-bit identifications were used to form a 4-bit response to the earliest and the latest delay paths. An attacker can collect many CRPs and classify CRPs according to the 4-bit response. For instance, if a response obtained by the adversary is 0011, they can clearly know that the signal (a) is the earliest and (d) is the latest. Therefore, he will know that the outputs of DFF1, DFF2, DFF3, DFF5, and DFF6 are 1. Other responses are inferred similarly.

TABLE 4 Comparison of metrics of different PUF designs

Metrics	APUF [34]	DAPUF [24]	RPUF [25]	OB-PUF [26]	The improved MA-APUF	Ideal
Prediction rate (%)	95.00[a]	69.00	64.45	71.99	56.00	50
Uniqueness (%)	23.00	48.06	52.20	N/A	43.60	50
Reliability (%)	95.20	89.95[b]	94.80	86.24	92.84	100
Bit-aliasing (%)	19.57	N/A	N/A	N/A	51.58	50

[a]These values are obtained from our experiments.

[b]The result of [24] is represented as steadiness.

Therefore, we can collect enough outputs for each arbiter and evaluate the resistance of the MA-APUF using LR and ES.

Figures 13 and 14 depict the results of the ML attacks based on ES and LR for the APUF, 2-XOR APUF, and IMA-APUF. The APUF exhibits the maximum accuracy rate of over 95% when the number of CRPs are more than 10 000. The accuracy rate of the 2-XOR APUF declines to approximately 80% because the XOR operation increases the nonlinearity. For the IMA-APUF, the prediction rate is always <61% even for samples with as many as 50 000 CRPs. A similar tendency is shown between the ES and LR. From the experimental results, it is clear that our proposal has a stronger resistance for ML attacks than the APUF, 2-XOR APUF, and MA-APUF.

Furthermore, Ruhrmair and others [19] and Mahmoud and others [21] discussed the first power and timing side channels on strong PUFs; additionally, an efficient exploitation via adapted ML techniques were described, which was illustrated by an XOR APUF and a lightweight PUF. This method reduces the attack complexity. In greater detail, the power and timing side channels provide information on the single outputs of the paralleled APUF in the XOR APUF or lightweight PUF. They indicate how many of these single outputs are "1" or "0" before the final XOR gate. According to this side-channel information, we could not determine which of the single outputs were 0 or 1. However, if combined with suitably adapted ML techniques, the side-channel information can be efficiently utilized to infer the secret information from the PUF. However, as shown in Figures 13 and 14, responses $r1$ and $r2$ before the XOR operation in the IMA-APUF are extremely difficult to predict with high accuracy. Therefore, the improved MA-APUF can be used for such a case.

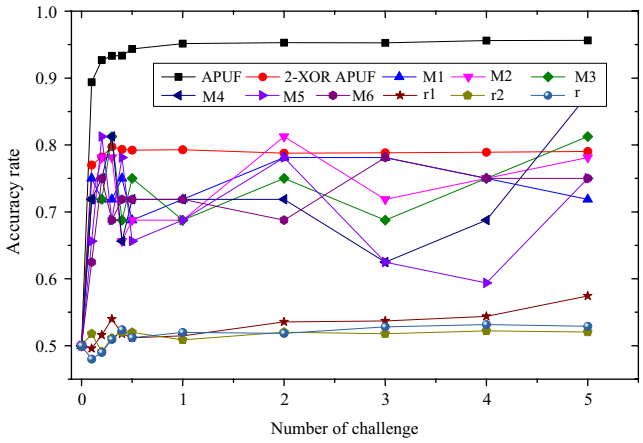

FIGURE 13 Evaluation of ML attacks based on ES

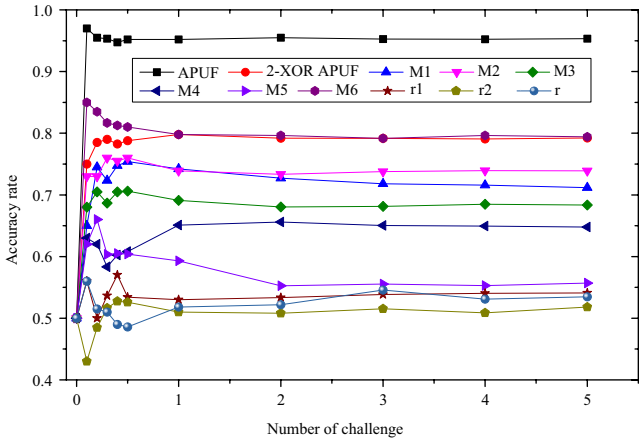

FIGURE 14 Evaluation of ML attacks based on LR

Numerous ML attacks, including LR and ES, rely on the assumption that a mathematical model of the PUF functionality is known in advance. Paper [31] provided a provable framework for ML attacks against a PUF family, whose underlying mathematical model is unknown. The author suggested that the number of Boolean variables k affecting the Boolean function should be computed. To prevent a potential adversary from modeling the PUF, the function representing our PUF should satisfy the following condition: $k \gg \ln(n)$ or $k = 0$, where n is the PUF stage. Table 5 shows the Boolean variables k of our PUF. The results further corroborate that the improved MA-APUF could resist some potential attacks without using a mathematical model.

6.3 | Hardware resource consumption

Table 6 shows the hardware resource consumptions of the MA-APUF and IMA-APUF. As shown, the improved MA-APUF consumes twice as much hardware resources as the MA-APUF. This is primarily attributed to the inverters and the tuning block.

TABLE 5 Results examining whether improved MA-APUF belongs to the class of K-junta functions

Response	k
$r1$	0
$r2$	62
r	0

TABLE 6 Results of MA-APUF for ML attacks based on ES and LR

		Selector	Detection	Response	Others	Total
MA-APUF	FF	0	6	0	0	6
	LUT	128	0	8	6	142
The improved MA-APUF	FF	0	0	0	0	0
	LUT	256	4	1	56	317

The inverters are inserted at each selector block to improve the nonlinearity, and the tuning block is added to tune the asymmetry delay path caused by the routing feature in the FPGA. Therefore, in our design, the tradeoff is that as the uniqueness and security improve, the hardware consumption increases.

7 | CONCLUSION

In this study, we demonstrated several foundational principles to improve the uniqueness of APUF-based strong PUFs. In particular, we used the MA-APUF as an example to demonstrate that symmetrical delay lines and nonlinear blocks affected the uniqueness of delay-based strong PUFs. Subsequently, we proposed an improved MA-APUF. The improved MA-APUF demonstrated improved uniqueness and ML attack resistance. However, our modifications resulted in decreased reliability. Our future work will focus on this issue.

ACKNOWLEDGMENTS

This study is supported by the Shenzhen Science, Technology and Innovation Commission (SZSTI): JCYJ20170817115500476, the Scientific Research Foundation of the Graduate School of Southeast University: YBPY1869, and the National Natural Science Foundation of China (NSFC): 61571116.

AUTHOR CONTRIBUTIONS

Bing Li and Shuai Chen contributed equally to this study.

ORCID

Shuai Chen https://orcid.org/0000-0003-2012-0424

REFERENCES

1. R. Pappu et al., *Physical one-way functions*, Sci. **297** (2002), no. 5589, 2026–2030.
2. L. Bing and C. Shuai, *A dynamic PUF anti-aging authentication system based on restrict race code*, Sci. China Inf. Sci. **59** (2016), no. 1, 12108–012108.
3. S. Chen and B. Li, *A dynamic reseeding DRBG based on SRAM PUFs*, in Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov., Chengdu, China, Feb. 2017, pp. 50–53.
4. J. Delvaux et al., *Helper data algorithms for puf-based key generation: Overview and analysis*, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **34** (2015), no. 6, 889–902.
5. R. Maes and I. Verbauwhede, *Physically unclonable functions: A study on the state of the art and future research directions*, in Towards Hardware-intrinsic Security, Springer, Heidelberg, New York, 2010, pp. 3–37.
6. R. Ulrich et al., *Modeling attacks on physical unclonable functions*, in Proc. ACM Conf. Comput. Commun. Security, Chicago, IL, USA, 2010, pp. 237–249.
7. J. Guajardo et al., *FPGA intrinsic PUFs and their use for IP protection*, in Int. Workshop Cryptograph. Hardw. Embed. Syst., Vienna, Austria, 2007, pp. 63–80.
8. S. S. Kumar et al., Extended abstract: The butterfly PUF protecting IP on every FPGA, in Proc. IEEE Int, Workshop hardw.-Oriented Security Trust Anaheim, CA, 2008, pp. 67–70.
9. A. Maiti and P. Schaumont, *Improving the quality of a physical unclonable function using configurable ring oscillators*, in Proc. Int. Conf. Field Prog. Logic Appl., Prague, Czech Republic, 2009, pp. 703–707.
10. B. Gassend et al., *Silicon physical random functions*, in Proc. ACM Conf. Comput. Commun. Security, Washington, D.C., USA, 2002, pp. 148–160.
11. G. E. Suh and S. Devadas, *Physical unclonable functions for device authentication and secret key generation*, in Proc. ACM/IEEE Des. Autom. Conf., San Diego, CA, USA, 2007, pp. 9–14.
12. L. Yingjie and K. K. Parhi, *Reconfigurable architectures for silicon physical unclonable functions*, in Proc. IEEE Int. Conf. Electro/Inform. Technol., Mankato, MN, USA, 2011, pp. 1–7.
13. M. Mehrdad, F. Koushanfar, and M. Potkonjak, *Techniques for design and implementation of secure reconfigurable PUFs*, ACM Trans. Reconfig. Tech. Syst. **2** (2009), no. 1, 1–33.
14. R. Maes, P. Tuyls, and I. Verbauwhede, *A soft decision helper data algorithm for SRAM PUFs*, in Proc. IEEE Int. Conf. Symp. Inf. Theory, Seoul, Rep. of Korea, 2009, pp. 2101–2105.
15. A. Vijayakumar and S. Kundu, *A novel modeling attack resistant PUF design based on non-linear voltage transfer characteristics*, in Proc. Des. Autom. Test Europe Conf. Exhibit., Grenoble, France, 2015, pp. 653–658.
16. D. P. Sahoo et al., *A case of lightweight PUF constructions: Cryptanalysis and machine learning attacks*, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **34** (2015), no. 8, 1334–1343.
17. U. Ruhrmair et al., *PUF modeling attacks on simulated and silicon data*, IEEE Trans. Inf. Forensics Secur. **8** (2013), no. 11, 1876–1891.
18. P. H. Nguyen and D. P. Sahoo, *Lightweight and secure PUFs: A survey (invited paper)*, in Proc. Int. Conf. Sec., Pune, India, 2014, pp. 1–13.
19. U. Ruhrmair, *Power and timing side channels for PUFs and their efficient exploitation*, in Proc. Cryptographic Hardw. Embedded Syst., Busan, Rep. of Korea, 2013, pp. 476–492.
20. R. Kumar and W. Burleson, *Side-channel assisted modeling attacks on feed-forward arbiter PUFs using silicon data*, in Proc. Int. Workshop Radio Freq. Ident. Sec. Privacy Issues, New York, NY, USA, 2015, pp. 53–67.
21. A. Mahmoud et al., Combined modeling and side channel attacks on strong PUFs, Cryptology ePrintArchive, Report2013/632, 2013.
22. A. Vijayakumar et al., *Machine learning resistant strong PUF: Possible or a pipe dream?* in Proc. IEEE Int. Symp. Hardw. Orient. Sec. Trust, McLean, VA, USA, 2016, pp. 19–24.
23. R. Kumar and W. Burleson, *On design of a highly secure PUF based on non-linear current mirrors*, in Proc. IEEE Int. Symp. Hardw. Orient. Sec. Trust, Arlington, VA, USA, 2014, pp. 38–43.
24. M. Takanori et al., *A new arbiter PUF for enhancing unpredictability on FPGA*, Sci. World J. **2015** (2015), 1–13.
25. J. Ye, Y. Hu, and X. Li, *RPUF: Physical unclonable function with randomized challenge to resist modeling attack*, in Proc. IEEE Asian Hardw. Orient. Sec. Trust, Wilan, Taiwan, 2016, pp. 1–6.
26. G. Yansong et al., *Obfuscated challenge-response: A secure lightweight authentication mechanism for PUF-based pervasive devices*, in Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops, Sydney, Australia, 2016, pp. 1–6.

27. M. Rostami et al., *Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching*, IEEE Trans. Emerg. Topics Comput. **2** (2014), no. 1, 37–49.
28. M. Yoshikawa and A. Naruse, *Multiplexing aware arbiter physical unclonable function*, in Proc. IEEE Int. Conf. Inf. Reuse Integr., Las Vegas, NV, USA, 2012, pp. 639–644.
29. M. Majzoobi et al., *Automated design, implementation, and evaluation of arbiter-based PUF on FPGA using programmable delay lines*, Cryptology ePrint Archive, Report 2014/639, 2014, https://eprint.iacr.org/2014/639.
30. A. Maiti, V. Gunreddy, P. Schaumont, A systematic method to evaluate and compare the performance of physical unclonable functions, Springer, New York, 2013.
31. F. Ganji et al., *Having no mathematical model may not secure PUFs*, J. Cryptogr. Eng. **7** (2017), no. 4, 1–16.
32. Y. Hori et al., *Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs*, in Proc. Int. Conf. Reconf. Comput. FPGAs, Quintana Roo, Mexico, 2011, pp. 298–303.
33. G. Yansong, Modeling attack resilient reconfigurable latent obfuscation technique for PUF based lightweight authentication (2017).
34. D. Lim et al., *Extracting secret keys from integrated circuits*, IEEE Trans. Very Large Scale Integ. Syst. **13** (2005), no. 10, 1200–1205.

AUTHOR BIOGRAPHIES

Bing Li received his PhD degree from the Southeast University, Nanjing, China, in 2004. He is a Professor at the School of Microelectronics and the School of Cyber Science and Engineering, Southeast University, where he leads the advanced cloud system joint research and SEU-FiberHome Joint Research Center team. His current research interests include the design of safety information exchange systems and circuit systems.

Shuai Chen received his MS degree in integrated circuit engineering from the Southeast University, Nanjing, China, in 2016. He is currently pursuing PhD degrees from the Southeast University, Nanjing, China. Additionally, he is a joint PhD student of Yale University (Computer Architecture and Security Laboratory, CAS Laboratory), USA and Southeast University, China. His current research interests include hardware security (physical unclonable functions, in particular) and secure processor architecture.

Fukui Dan is currently pursuing his MS degree in integrated circuit engineering from the Southeast University, Nanjing, China. His current research interests include information security and digital circuit design.