A Dynamically Configurable PUF and Dynamic Matching Authentication Protocol

A Dynamically Configurable PUF and Dynamic Matching Authentication Protocol

Yale Wang, Chenghua Wang, Chongyan Gu, *Member, IEEE,* Yijun Cui, Máire O'Neill, *Senior Member, IEEE,* and Weiqiang Liu, *Senior Member, IEEE*

Abstract—A physical unclonable function (PUF) is a hardware security primitive, which can be used secure various hardware-based applications. As a type of PUFs, strong PUFs have a large number of challenge-response pairs (CRPs), which can be used for authentication. At present, most strong PUF structures follow a one-to-one input/output relationship, *i.e.* linear function. As such, strong PUF designs are vulnerable to machine learning (ML) based modeling attacks. To address the issue, a dynamically configurable PUF structure is proposed in this paper. A mathematical model of the proposed dynamic PUF is presented and the design is proposed against the effective ML based attacks, such as deep neural network (DNN), logistic regression (LR) and reliability-based covariance matrix adaptation evolution strategies (CMA-ES). Experimental results on field programmable gate arrays (FPGAs) show that the proposed dynamic structure has achived good uniqueness and reliability. It is also shown that the dynamic PUF has a strong resistance to the CMA-ES attack. Due to the dynamic nature of the proposed PUF structure, an authentication protocol is also designed to generate recognizable authentication bits string. The protocol shows strong resistance to classical machine learning attacks including the new variant of CMA-ES.

Index Terms—Physical unclonable function, dynamically configurable, machine learning, authentication protocol.

1 Introduction

The Internet of Things (IoT) has been widely deployed and will impact many applications in the near future. IoT which comprise huge heterogeneous networks can connect billions of devices. Unfortunately, this heterogeneity and connectivity of devices introduce significant security and privacy risks. The security of IoT will determine its widespread deployment and application [1].

Authentication mechanisms are used to prevent the unauthorized access to IoT devices. However, using traditional authentication approaches leads to a significant overhead for many lightweight IoT devices. Physical unclonable function (PUF), a lightweight hardware security primitive, can provide low-cost identity authentications for IoT devices [2], [3]. PUFs use the natural process variations of integrated circuits to create digital sequences, which are unique to each device. Depending on the quantity of CRPs, PUFs can be divided into weak PUFs and strong PUFs [4], [5]. Weak PUFs have a limited number of CRPs, which make them suitable for generating random sequences. PUFs do not need to store secrets in non-volatile memory (NVM) as other conventional cryptographic approaches. Strong PUFs having an exponential number of CRPs, are suitable for securing authentication protocols [6], [7]. However, most of the strong PUFs are vulnerable to ML based modeling attacks. In order to improve the security of strong PUFs, two approaches have been shown, one is introducing non-linearity into the linear model of strong PUFs, such as the XOR arbiter PUF (XOR APUF) design. The other one is using reconfigurability to increase modeling attack resistance [9], [10]. Since the principle of ML modeling attacks is to model a PUF design from the static relationship between its challenges and responses, reconfigurable mechanisms can greatly improve the resistance of ML based modeling attacks [11]. However, in conventional reconfigurable designs, the reconfigurable signals are provided by inputs from outside, which achieve a limited improvement against ML attacks. Hence, changing the configurations of a PUF design dynamically and autonomously could be a promising way to improve the security of strong PUFs.

The dynamic mechanism provided in [12] reconfigured the PUF structure by loading different bitstream files, where the bitstream files contain different logic and routing constraints. However, recent studies have shown a successful attack on the Xilinx 7-series bitstream file [13]. In addition, the time cost of reloading a bitstream file on an FPGA is significant. It takes about 25 seconds to configure the PUF remotely, and in contrast, it only takes less than a millisecond to receive the PUF responses [12].

In this paper, we propose a dynamically configurable PUF structure, in which the configuration signal is independent of the inputs (challenges). A dynamic matching authentication protocol based on the dynamic PUF is also proposed. The main contributions of the paper can be summarized as follows:

- A dynamically configurable hybrid (DCH) PUF structure is proposed. Each response bit is generated once with one configuration. The configuration signals are generated from an internal linear feedback

- *Y. Wang, C. Wang, Y. Cui and W. Liu are with College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), China, 211106.*
  E-mail: {yalewang, chwang, yijun.cui, liuweiqiang}@nuaa.edu.cn
- *C. Gu and M. O'Neill are with the Centre for Secure Information Technologies (CSIT), Institute of Electronics, Communications & Information Technology (ECIT), Queen's University Belfast (QUB), U.K., BT3 9DT.*
  E-mail: c.gu@qub.ac.uk and m.oneill@ecit.qub.ac.uk

*Manuscript received ** **, 2020; revised ,**, **, 2020.*

shift register (LFSR) and are independent from the inputs. The longer the bit-length of the LFSR, the larger the number of configurations, and the greater the computational cost of a modeling attack.
- DNN, LR and reliability-based CMA-ES algorithms, the most effective modeling attacks against *n*-XOR APUF designs, are utilised to analyze the resistance capability of the proposed DCH PUF. The analysis shows that the proposed DCH PUF structure can resist these ML attacks due to its dynamic configuration features.
- An authentication protocol based on the proposed dynamic PUF is presented. The protocol uses pattern matching and has a strong error tolerance. The authenticating process without using the complete response shows great resistance to classical ML attacks and the latest variant of CMA-ES against pattern matching.

The rest of this paper is organized as follows. Section 2 introduces a background research related to this paper. In Section 3, configurable PUF structures and their mathematical models are introduced. The proposed dynamically configurable PUF structure is also introduced. Section 4 evaluates the response performance of the DCH PUF. Its security is comprehensively evaluated, and its performance against modeling attacks along with their cost is provided. Section 5 introduces the proposed authentication protocol based on the DCH PUF structure. Section 6 evaluates the security performance of the proposed authentication protocol. A discussion and conclusion are given in Section 7.

2 Related Work

2.1 PUF Designs and Attacks

Various PUF designs and attacks have been studied over the past decade. Arbiter PUF (APUF) designs generate responses via a race between the signals in two delay chains. However, due to the linear characteristics of its model, the original APUF structure can be easily modeled [8]. One proposal to improve its resistance to modeling attacks is to introduce nonlinearity. In feed-forward PUFs (FF-PUFs) [14], the challenges of some stages are determined by feed-forward branches. The cost of modeling FF PUF designs increases with the number of FF loops. XOR APUF designs [15] generate responses by XORing the responses of multiple individual APUFs. The cost (in terms of the number of CRPs required) of using a classic ML algorithm to attack this design increases exponentially with the number of individual APUFs used.

Building on the XOR APUF structure, the lightweight secure PUF [16] changed the sequence of individual APUF challenge bits, to further increase the resistance to ML attacks. Based on the lightweight secure PUF, TSMA PUF [17] uses multiplexers to chose different path combinations before the arbiters of two individual APUFs. The more recently proposed interpose PUF structure [18] comprises two XOR APUFs, where the response of the upper XOR APUF structure is used as one bit of the lower XOR APUF challenge and the response of the lower XOR APUF is taken as the final response. This design has stronger ML resistance than XOR APUFs of the same scale.

In terms of attacks against PUF designs, DNN, LR, and CMA-ES are the main methods used in recent studies. DNN-based attack [19] is a form of black box attacks and do not need a specific mathematical model. It can predict the PUF response by training the inner multilayer neurons. In [20], the authors successfully predicted the responses of a double arbiter PUF using DNNs. LR is a gradient descent ML algorithm, which can efficiently model XOR APUFs based on a special transformation of their mathematical model.

Although the CMA-ES algorithm [22] has a large search space and needs a long training time, its fitness function is not limited to a specific form. Therefore it has a wide range of applications and can be used to model all known strong PUF structures. The CRP based CMA-ES algorithm is less efficient than LR when attacking XOR APUFs [18], [23], but Becker [24] proposed a new reliability-based CMA-ES algorithm, which is more efficient than LR when attacking XOR APUFs with large numbers of individual APUFs. The amount of CRPs needed only increases linearly with the number of individual APUFs. Before that, the classical ML algorithms could only model XOR APUFs or lightweight secure PUF composed of a small number of individual APUFs. With the reliability-based CMA-ES algorithm, both the structures with more than 9 individual APUFs have been broken [24]. The interpose PUF has also been broken by a new divide-and-conquer approach [25].

2.2 PUF-based Authentication Protocols

The majority of authentication protocols using PUFs are based on the static response behaviour of PUF structures [17], [26], [27], and therefore, are not suitable for the proposed DCH PUF structure. In the proposed structure the time-varying configuration ensures that the challenges and responses do not have a one-to-one relationship. Therefore, on the basis of the fixed order of different configurations, an approach based on pattern matching is used to authenticate the structure.

Another protocol based on pattern matching is the Slender protocol [28]. In this protocol, only a random substring of the real response is included in the authentication bits string, and the remainder is made up of random padding. This method greatly expands the attacker's search space and makes the traditional CRP-based ML attacks practically invalid. However, Becker proposed a CMA-ES algorithm based on a Hamming weight (HW) [29]. With this method, the two random indexes used in the Slender protocol cannot provide resistance and the protocol was broken by this method at a relatively small cost.

2.3 Model of an Arbiter PUF

The APUF, a basic strong PUF structure, consists of two parallel *n*-stage multiplexer (MUX) chains and an arbiter. During operation, an enable signal activates the circuit. Two parallel *n*-stage MUX chains feed into an arbiter (D flip-flop) to produce one response bit. An *n*-bit challenge $\{c_0, c_1, c_2, \cdots, c_{n+1}\}$ determines whether the signal passes through parallel or cross paths at the *n*-th stage. When the signals from two chains arrive at the arbiter, a decision will be given depending on the arriving time. Finally, the arbiter outputs '0' or '1' according to which signal arrives first.

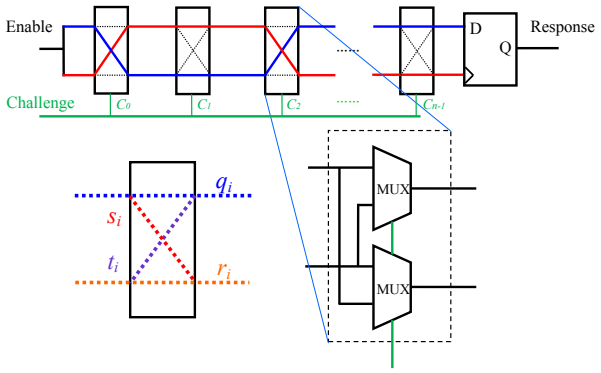

Fig. 1: The basic APUF desin [24].

According to the working principle of an APUF, a model of its response can be considered as an additive delay model [24]. Therefore, the response R_{apuf} of an n-stage APUF can be represented as:

$$R_{apuf} = \begin{cases} 0, \Delta_{apuf}(n) > 0 \\ 1, \Delta_{apuf}(n) \leq 0 \end{cases}, \quad (1)$$

where Δ_{apuf} is the decision delay of an APUF, and

$$\Delta_{apuf}(n) = \vec{P} \cdot \vec{W}^T, \quad (2)$$

where $\vec{P}$ is the challenge parity vector, $\vec{P} = \{p_1, p_2, \cdots, p_n\}$. Due to the APUF characteristics, each bit of the challenge not only determines whether to choose the parallel or opposite paths, but also affects the number of times that two competing paths exchange signals before the two signals arrive at the arbiter. Therefore, the challenge parity vector is calculated by the value and position of each challenge bit. The challenge parity $p_k \in \{-1, 1\}$ is defined by

$$p_k = \prod_{i=k+1}^{n} (1 - 2c_i), \quad (3)$$

and $\vec{W}$ is the delay vector, $\vec{W} = \{\omega_1, \omega_2, \cdots, \omega_{n+1}\}$. The elements of $\vec{W}$ are dependent on the multiplexer switching and routing delays, which are susceptible to manufacturing process variations even if the switching of each APUF stage is identically designed. The elements of the weight vector $\vec{W}$ are given by:

$$\begin{aligned} &\omega_1 = \alpha_1, \\ &\omega_i = \alpha_i + \beta_{i-1}, for\ 2 \leq i \leq n, \\ &\omega_{n+1} = \beta_{n-1}, \end{aligned} \quad (4)$$

where α_i and β_i can be calculated by:

$$\begin{aligned} \alpha_i &= (q_i - r_i - s_i + t_i)/2, \\ \beta_i &= (q_i - r_i + s_i - t_i)/2, \end{aligned} \quad (5)$$

where $q_i, r_i, s_i$and t_i represent the four path delays possible in the i-th stage of the two-MUX chain, as shown in Fig. 1. Using this model, by collecting a certain number of CRPs to train a ML algorithm, the unknown response of an APUF can be effectively predicted [8].

2.4 PUF Response Performance Metrics

Uniqueness. Ideally, no two chips should generate the same responses. A Hamming distance (HD) is utilized to calculate the difference between two responses. The ideal value of uniqueness is 50% when the same challenge C is input. When the same PUF circuit is implemented in the device i and device j, n-bit responses R_i and R_j will be generated. The uniqueness can be expressed as the average inter-chip HD over k devices, as follows:

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{N}. \quad (6)$$

Reliability. The reliability of a PUF design reflects the ability to reproduce a response. The higher the reliability, the lower the response error rate and the more stable the structure. In this work, the bit error rate (BER) is calculated as follows:

$$BER = \frac{1}{s} \sum_{t=1}^{s} \frac{HD(R_i, R'_{i,t})}{N}, \quad (7)$$

where s is the number of measurements, n is the bit-length of the response, R_i is the reference response, and $R'_{i,t}$ is the response of the t-th measurement. The metric for reliability is calculated as follows:

$$Reliability = 1 - BER. \quad (8)$$

3 The Proposed Dynamically Configurable PUF Design

In this section, we first introduce two configurable PUF structures, and then show the proposed DCH PUF structure which is composed of these two structures and a dynamic configuration mechanism .

3.1 Configurable Self-XOR Structure

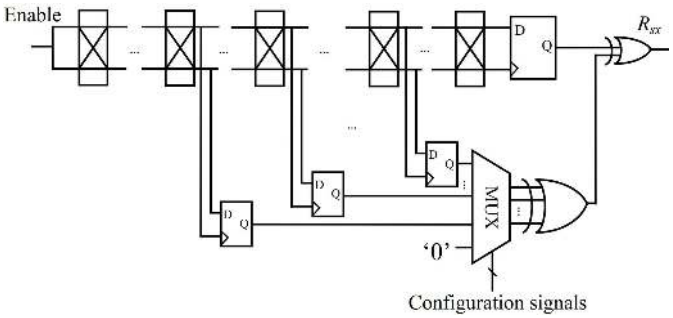

Fig. 2: The proposed configurable self-XOR structure.

A configurable self-XOR (SX) structure is shown in Fig. 2. Different from the traditional XOR APUF structure, the proposed self-XOR structure truncates the original APUF to generate multiple responses and then XORs them to derive a 1-bit final response. With this PUF structure there are many possible ways to choose the branch signal. A group of (m+1:k) MUXs is utilised to select different combinations,

and the selected signals of the MUXs are called configuration signals. In addition, a '0' signal is added to the signal options of MUX, which means that under a specific configuration signal, the structure is equivalent to a traditional APUF structure. It is worth noting that adding one branch requires only a small amount of additional resources (an arbiter, a MUX and an XOR gate). Compared with the traditional *n*-XOR APUF structure, this design can save a significant hardware resource.

Let R_{sx} represent the output of the configurable self-XOR structure. The response can be expressed as follows:

$$R_{sx} = \begin{cases} 0, \Delta_{sx}(n) > 0 \\ 1, \Delta_{sx}(n) \leq 0 \end{cases}, \tag{9}$$

where Δ_{sx} represents the delay difference of two signals fed into the self-XOR structure, which can be calculated as the product of the delay difference between the main circuit (APUF) and the branch circuit, as follows:

$$\Delta_{sx}(n) = \Delta_{apuf}(n) \cdot \Delta_{bran}. \tag{10}$$

The branch delay difference Δ_{bran} represents the delay difference of the selected branch, which is calculated by the configuration vector $\vec{S}_{conf}$ and the delay difference vector $\vec{\delta}_{bran}$ of multiple branches, as follows:

$$\begin{aligned} \Delta_{bran} &= \vec{\delta}_{bran} \cdot \vec{S}_{conf}^{T} \\ &= \prod_{i=1}^{k} \Delta_{apuf}(s_i) \end{aligned}, \tag{11}$$

where $\vec{\delta}_{bran} = \{\Delta_{apuf}(s_1), \Delta_{apuf}(s_2), \cdots, \Delta_{apuf}(s_m)\}$ is the set of delay differences of all configurable branches, k is the number of selected branches, $\Delta_{apuf}(s)$ represents the delay difference of an APUF with s stages, which can be calculated by Eq. (1). Note here that there is an extra '0' signal in the design and when it is selected, the corresponding delay is $\Delta_0 = 1$. The element $s_i \in \{0, 1\}$ in the configuration vector $\vec{S}_{conf} = \{s_1, s_2, \cdots s_m\}$ is defined by the configuration signals.

3.2 Configurable Modified Feed-forward Structure

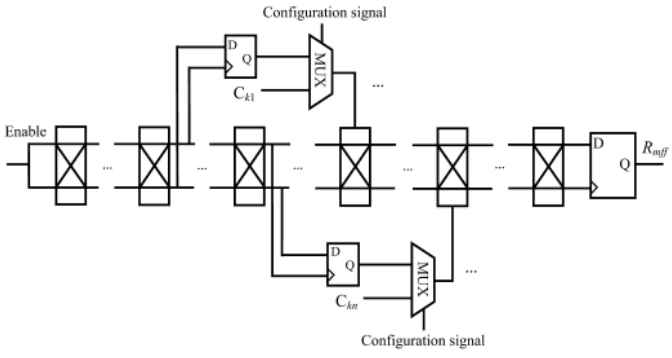

Fig. 3: The proposed configurable feed-forward structure.

A configurable modified feed-forward (MFF) structure is proposed and shown in Fig. 3, which differs from the original feed-forward APUF structure. In this structure, a MUX is added between the feed-forward signal and the target stage. The original challenge bit of the target stage provides the second input to the MUX. The selected signals are called configuration signals. Also, by using a certain configuration signal, the structure is equivalent to a traditional APUF structure. Let R_{mff} represent the output of the modified feed-forward structure, then

$$R_{mff} = \begin{cases} 0, \Delta_{mff}(n) > 0 \\ 1, \Delta_{mff}(n) \leq 0 \end{cases}, \tag{12}$$

where Δ_{mff} represents the delay difference of the modified feed-forward structure, which can be calculated from the challenge parity vector $\vec{P}_{mff}$ and delay vector $\vec{W}$,

$$\Delta_{mff}(n) = \vec{P}_{mff} \cdot \vec{W}^{T}, \tag{13}$$

where $\vec{P}_{mff} = \vec{P} \cdot \vec{S}_{conf}^{T} + \vec{P}_{T} \cdot \left(\vec{1} - \vec{S}_{conf}\right)^{T}$, and $\vec{P} = \{p_1, p_2, \cdots, p_n\}$ is the parity vector of the underlying APUF, and its elements $p_k = \prod_{i=k+1}^{n}(1 - 2c_i)$. $\vec{P}_T = \{p_{T_1}, p_{T_2}, \cdots, p_{T_j}\}$ is the set of all configurable FF loops, where the elements are:

$$\begin{aligned} p_{T_1} &= (1 - 2R_{apuf}(s_1)) \prod_{i=T_1+2}^{n}(1 - 2c_i), \\ p_{T_2} &= (1 - 2R_{apuf}(s_2)) \prod_{i=T_2+2}^{n}(1 - 2c_i), \\ &\vdots \\ p_{T_j} &= (1 - 2R_{apuf}(s_j)) \prod_{i=T_j+2}^{n}(1 - 2c_i). \end{aligned} \tag{14}$$

The elements $s_i \in \{0, 1\}$ in the configuration vector $\vec{S}_{bran} = \{s_1, s_2, \cdots s_m\}$ can be transformed from the configuration bits, and $R_{apuf}(s)$ represents the response of an APUF with s stages, which can be calculated from Eq. (1).

3.3 Dynamically Configurable Structure Design

Different structures can be formed from the above two structures with different configuration signals. Similar as existing work in [10], the configuration signals typically come from the challenge. Different from the way that the configuration signals of conventional configurable PUFs are provided by external inputs, in the proposed design, an internal LFSR provides the configuration signals. The LFSR does not require any seed from external inputs, and is driven by internal clock signal of the device. Its dynamic configuration signals are completely independent from external inputs. During the operation, the LFSR will jump to the next state when a response is generated. The configuration signal will change accordingly. This leads to different responses even if the same challenge is provided multiple times. Therefore, a dynamically configurable hybrid PUF structure which incorporates the dynamically configurable SX-PUF (DC SX-PUF) and the dynamically configurable FF-PUF (DC FF-PUF) is proposed.

Design Constraints. From the two configurable structures proposed above, there are many different configuration options for a multi-stage PUF structure. Previous work shows that the reliability of the XOR APUF and FF-PUF designs were reduced with an increased number of XORs and FF loops, respectively [9]. To mitigate the deterioration of reliability, the following constraints are proposed: 1) the

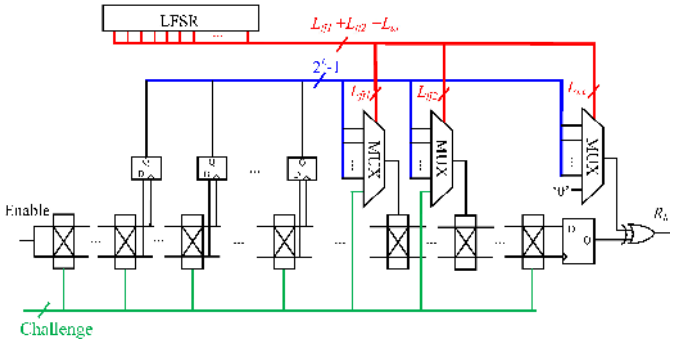

Fig. 4: The proposed APUF-based DCH structure.

number of branches that participate in an XOR operation in the configurable self-XOR structure is 1; 2) In the configurable FF structure, the number of FF loops is 2, and the two FF loops are overlapped.

The response of the proposed DCH PUF is:

$$R_h = \begin{cases} 0, \Delta_h(n) > 0 \\ 1, \Delta_h(n) \leq 0 \end{cases}, \tag{15}$$

where

$$\Delta_h(n) = \Delta_{sx}(n) \cdot \Delta_{mff}(n) \ , \tag{16}$$

where $\Delta_{sx}(n)$ and $\Delta_{mff}(n)$ can be calculated from Eq. (10) and Eq. (13), respectively. Note, due to the constraints of the proposed design, the self-XOR structure and the different FF loops in the hybrid structure will not affect each other and can be calculated separately. In addition, according to these constraints, a 64-stage APUF can provide up to 15 dynamic configuration bits. Fig. 4 shows the APUF based DCH structure.

When the proposed DCH structure is applied to an XOR APUF or lightweight secure PUF structure, it is slightly different: only one of the SX branches containing the different PUFs is selected for input into the XOR operation by the configuration signal. In addition, in a multiple-XOR APUF based DCH PUF structure, only one LFSR is used to provide all the configuration signals. Fig. 5 shows a 2-XOR APUF based DCH structure.

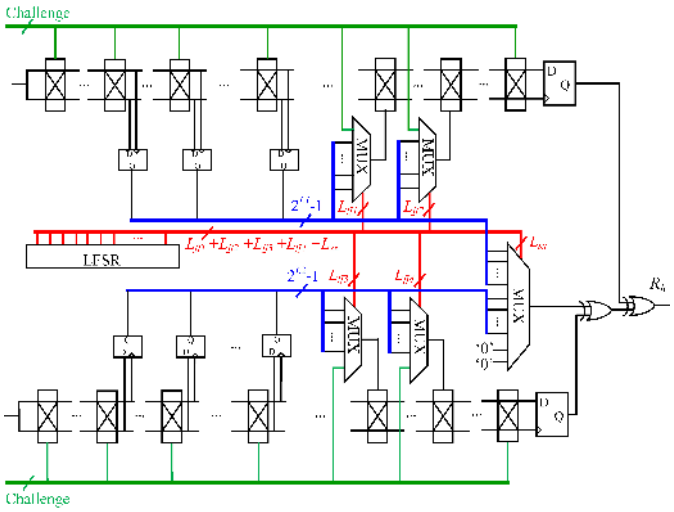

Fig. 5: The proposed 2-XOR-based DCH structure.

4 Analysis of DCH PUF Structure

4.1 Performance Analysis

Uniqueness. Firstly, a 7-bit APUF-based DCH structure was implemented on five Xilinx Artix-7 FPGA (28nm Technology) Nexys4 boards. Fig. 6 shows the uniqueness results (average HD) of both the proposed DCH sructure and its underlying APUF design. The uniqueness of the underlying APUF is only 11.8%, and the uniqueness of the 7-bit APUF-based DCH PUF is 41.0%. The improvement can be explained as follows. The proposed DCH PUF includes XOR and FF branches, and these branches with coarse grain components provide an improvement uniqueness over the underlying APUF [5], [30]. Moreover, the LFSRs in different implementations will provide different configurations as they have a high chance of being in different states. Hence, the DCH design is more likely to produce different responses. A comparison of the HD values for 128 different configurations of a 7-bit DCH design ($(2^7 - 1)$ LFSR configurations) is shown in Fig. 7.

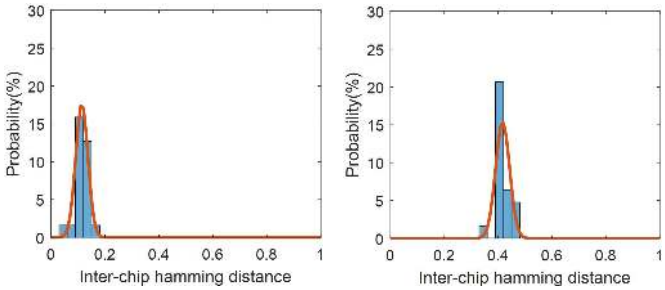

Fig. 6: Uniqueness of DCH PUF and its underlying APUF: (a) Uniqueness of underlying APUF; (b) Uniqueness of DCH PUF.

There are 128×128 pixels in Fig. 7. The horizontal and vertical axes represent the 128 configuration values of a 7-bit DCH structure (including the traditional APUF structure corresponding to the configuration signal '0000000'). Pixel (x, y) represents a comparison between the x-th configuration and the y-th configuration, and the colour shows Hamming distance values. Excluding the diagonal, it can be observed that 85% of the pixels are yellow, which means that about 85% of the Hamming distance values are greater than 0.4, and about 70% of them are in the range of (0.45, 0.5). It shows that different configurations produce good uniqueness for the same DCH PUF design.

In the proposed n-bit DCH structure, there are a total of $2^n - 1$ configurations (structures) that are recycled and without knowing the real-time state of the LFSR, the configuration could be any of them, such that,

$$\begin{aligned} BER(DCH) &= \sum_{i=1}^{2^n-1} \Pr(S_{conf} = S_i) BER(CONF_i) \\ &= \frac{1}{2^n - 1} \sum_{i=1}^{2^n-1} BER(CONF_i) \ , \end{aligned} \tag{17}$$

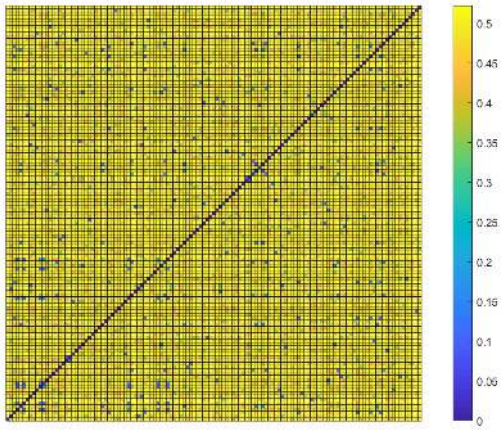

Fig. 7: Hamming distance of 128 configurations of a 7-bit DCH structure.

TABLE 1: BER Comparison (room temperature, normal supply voltage).

PUF structures	BER	PUF structures	BER
APUF	0.69%	3-bit DCH APUF	0.81%
2-XOR APUF	1.40%	4-bit DCH APUF	1.00%
3-XOR APUF	2.17%	5-bit DCH APUF	0.96%
4-XOR APUF	2.91%	6-bit DCH APUF	1.08%
1-SX-APUF	1.06%	7-bit DCH APUF	1.32%
2-FF-APUF	1.47%	8-bit DCH APUF	1.42%
1-SX-2-FF APUF	1.51%	9-bit DCH APUF	1.45%
4-bit DC SX-APUF	0.99%	10-bit DCH APUF	1.40%
6-bit DC FF-APUF	1.12%	11-bit DCH APUF	1.36%

where $CONF_i$ represents one of the configurations provided by the LFSR.

Reliability. We tested the BERs of instances implemented on the FPGA boards, and compared them with n-XOR APUFs and n-stage FF APUFs, as shown in Table 1. For each structure, CRPs were collected 10 times using the same challenge set (contains 10,000 challenges). The APUF designs were implemented on the FPGA with fixed placement: only one LUT is selected in each slice to implement one MUX of the PUF delay chains, and all the slices are from one column. In Table 1, 1-SX APUF refers to the structure with one self-XOR branch. 2-FF APUF refers to the structure with two FF loops. 1-SX-2-FF APUF refers to the structure with one SX branch and two FF loops.

As n increases, the BER of DCH structure increases. However, when n is greater than 7, the BER begins to maintain at a certain level. More specifically, when n is greater than 7, the BER of the APUF-based DCH structure is equivalent to that of the 2-XOR APUF. In the traditional XOR structure, as the number of XOR gates increases, its reliability decreases sharply. With an increase in the bit-length of the configuration signal, the reliability of the DCH structure does not deteriorate significantly. The reason is that the proposed DCH structure is limited to at most one SX branch and two FF loops, and its BER is equivalent to the average BER of all these combination configurations.

TABLE 2: Hardware Consumption (number of slices used on FPGA).

Structures	N_{slice} with n=1	N_{slice} with n=2	N_{slice} with n=3
Underlying n-XOR APUF	129	259	388
5-bit DCH	137	268	400
6-bit DCH	137	268	400
7-bit DCH	141	272	400
8-bit DCH	141	272	400
9-bit DCH	142	273	401
10-bit DCH	150	273	407
11-bit DCH	150	273	407

Hardware Consumption. The original 64-stage APUF, implemented on FPGA with fixed placement, needs 129 slices, 128 for the 128 MUXs with 64 stages, and the last one for the arbiter. Compared with the APUF structure, the DCH PUF structure contains additional branches and a LFSR. Table 2 shows the numbers of slices required for the underlying n-XOR APUF the DCH structures. In fact, these branches and a small-scale LFSR can be implemented with few hardware resources. The implementation of the DCH structure only consumes a little more hardware than its underlying PUF. Taking the 9-bit DCH structure as an example, when the underlying structure is an original APUF, it only uses 13 more slices (3 for the 9-bit LFSR, 7 for the 7 branch arbiters, and 3 for the three 8:1 MUXs). Similarly, when the underlying structures are 2-XOR APUF and 3-XOR APUF, an additional 14 and 13 slices are needed, respectively. Since the proposed design is based on APUF or XOR APUF, and additional hardware consumption is needed for configuring the branches. Therefore, it will inevitably need more hardware resources than the underlying PUF structure. However, when increasing the number of branches, the number of hardware units required will not be multiplied as XOR APUF does.

4.2 Security Analysis

In this subsection, we will show the resistance of the DCH PUF structure to modeling attacks. All the simulations in this work are performed on an Intel Xeon E5-2695 processor, 16 cores with 64GB are were used for each run.

In the existing configurable PUF designs, the configuration signals are input from the outside of the device. This mechanism enables the device side to change the configuration on demand, and the verifier (server) does not need to use a dedicated authentication method. However, the configuration signals from external input will leak information to attackers, and also have the risk of being tampered with. In contrast, in the proposed design, configuration signals which are internally controlled cannot be obtained and manipulated from outside. From the mathematical model discussed in the previous section, when calculating the DCH PUF response, the delay vector and the real-time configurations (*i.e.* the real-time state of the LFSR) are required. According to general assumptions, an attacker knows the mathematical model of the PUF and the structure of the LFSR, and can collect CRPs without limitation. However,

the attacker still does not know the real-time state of the LFSR, as the LFSR cannot be measured directly. In the first modeling experiment, we assume that the attacker already knows the real-time configurations. The modeling attack results using CMA-ES are shown in Fig. 8. Later, we will show the cost of obtaining the real-time configurations.

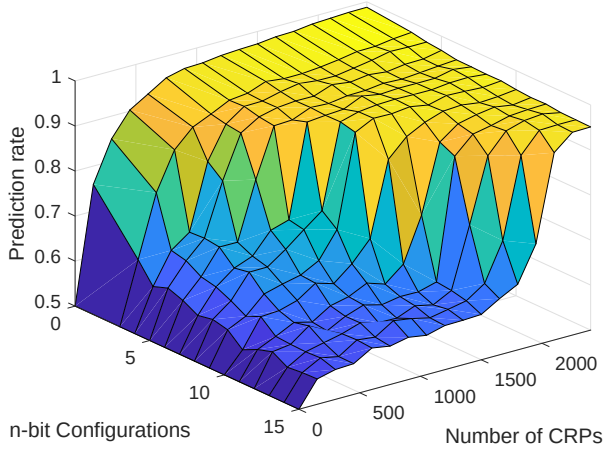

Fig. 8: CMA-ES modeling results of DCH PUFs with real-time configurations.

In Fig. 8, each point represents the average prediction rate over 20 runs, where the 0-bit configuration refers to the conventional APUF design. For this design, it can be seen that when the real-time configurations are known, only a small amount of CRPs are needed to model the DCH structure efficiently. Even for the 15-bit DCH PUF structure, only 2,400 CRPs are needed to achieve a modeling accuracy of about 95%. In addition, in this experiment, the modeling time of each DCH PUF structure (based on APUF) is less than one minute.

In an on-board implementation, the LFSR may be in any state. This means that the real-time configurations of the DCH PUF are unknown. Hence, the attacker needs to pre-process the collected CRPs to obtain the real-time configurations. Here we provide two pre-processing methods, namely circular matching and CRP regrouping. For the DCH PUF structure with n-bit LFSR, assume that the LFSR can provide a maximum length sequence.

1) Circular matching. The collected CRPs are placed in order. The 2^n-1 sequential states (configuration signals) of the LFSR are calculated according to the feedback function. Next, circular matching is performed on the CRP set to obtain 2^n-1 sets of CRPs and configurations. These sets are used as the training data and input into the ML algorithm in turn, until the prediction rate is higher than 90%. The modeling results are shown in Table 3.

The time cost T_{match} of modeling the dynamically configurable structure consists of two parts:

$$T_{match} = T_{DCH} + \frac{2^n-1}{2} T'_{DCH}, \tag{18}$$

where T_{DCH} is the time cost of training with correct configurations, and T'_{DCH} is the time cost of training with incorrect configurations, which requires about 2,000 iterations. Since the required matching times are uniformly distributed from $1 \sim 2^n-1$, the matching times are directly taken as half of the number of configurations 2^n-1, and then multiplied it by T'_{DCH}. It can be seen that the number of CRPs needed to model the DCH PUF is only slightly more than that needed to model an XOR APUF of the same scale. But the time cost (calculation cost) is much higher than for the XOR APUF.

TABLE 3: CMA-ES Modeling Results of XOR APUFs and DCH PUFs with Pre-processing of the CRPs

PUF structure	Pre-process	Number of CRPs	Prediction rate	Time cost
APUF	NA	800	94.35%	8 sec
2-XOR APUF	NA	3×10^3	94.73%	30 sec
2-XOR lightweight secure	NA	5×10^3	95.27%	2.6 min
3-XOR APUF	NA	8×10^4	94.30%	9.9 min
3-XOR lightweight secure	NA	8×10^4	98.45%*	2.3 hrs
4-XOR APUF	NA	6×10^5	96.43%*	1.9 hrs
9-bit DCH APUF	Circular matching	1.6×10^3	96.16%	43 sec + 1.2 hrs
10-bit DCH APUF	Circular matching	2×10^3	95.64%	51 sec + 2.3 hrs
11-bit DCH APUF	Circular matching	2.2×10^3	96.36%	1 min + 6.8 hrs
9-bit DCH 2-XOR	Circular matching	8×10^3	96.06%	7.2 min + 12.3 hrs
10-bit DCH 2-XOR	Circular matching	1×10^4	95.70%	8.7 min + 29.6 hrs
11-bit DCH 2-XOR	Circular matching	1×10^4	95.11%	9.5 min + 64.4 hrs
9-bit DCH 3-XOR	Circular matching	2.4×10^5	98.12%*	7.5 hrs + 32 days
10-bit DCH 3-XOR	Circular matching	2.4×10^5	97.87%*	7.9 hrs + 68 days
11-bit DCH 3-XOR	Circular matching	2.4×10^5	98.06%*	8.7 hrs + 149 days
9-bit DCH APUF	Regroup	2×10^5	94.55%	8 sec + 13.7 min
10-bit DCH APUF	Regroup	4×10^5	94.32%	8 sec + 27.3 min
11-bit DCH APUF	Regroup	8×10^5	93.78%	8 sec + 54.6 min
9-bit DCH 2-XOR	Regroup	1.3×10^6	94.36%	2.6 min + 4.5 hrs
10-bit DCH 2-XOR	Regroup	2.6×10^6	95.06%	2.6 min + 9 hrs
11-bit DCH 2-XOR	Regroup	5.1×10^6	94.84%	2.6 min + 18 hrs
9-bit DCH 3-XOR	Regroup	2×10^7	97.35%	2.3 hrs + 10 days
10-bit DCH 3-XOR	Regroup	4×10^7	97.67%	2.3 hrs + 20 days
11-bit DCH 3-XOR	Regroup	8×10^7	96.89%	2.3 hrs + 40 days

* CMA-ES running with 9,000 iterations. (Default with 5,000 iterations.)

2) CRP Regroup. The CRPs collected are numbered from 1 to $2^n - 1$ in sequence. And put the CRPs with same sequence number together to form $2^n - 1$ new CRP sets, which are used for training respectively. For the simplest configuration (*i.e.* the traditional XOR APUF structure), there must be one set corresponding to it. CMA-ES is used for training, and the results are also shown in Table 3.

The time cost $T_{regroup}$ is also obtained in two parts:

$$T_{regroup} = T_{XOR} + \frac{2^n - 1}{2} T'_{XOR}, \tag{19}$$

where T_{XOR} is the time cost of modeling the traditional XOR APUF with a correct CRP set, and T'_{XOR} is the time cost of modeling with an incorrect CRP set. Since the time needed to select the correct set is uniformly distributed in the range of $1 \sim 2^n - 1$, we directly take half of the number of sets $2^n - 1$, and then multiply by T'_{XOR}. It can be seen that, compared with the circular matching method, the regrouping method saves a certain amount of time, but the number of CRPs needed is very large. With an increase of *n* bits, the number of CRPs needed and the training time double.

From these two experiments, it can be seen that the designed DCH PUF structures have a good resistance to the CMA-ES ML modeling attack. Also, due to the particularity of the proposed design of the DCH structure, it cannot be attacked by other techniques typically applied against conventional PUF designs, as described next.

Resistance to DNN Attack. As a black box algorithm, DNNs do not need to know the mathematical model of a PUF. Simply by inputting a certain number of CRPs, a DNN attack can effectively simulate the relationship between the challenges and responses of a PUF, and predict its response accurately [20]. This is based on the premise that the relationship between the challenges and responses of the PUF structure is fixed (static). In our proposed dynamic design, after generating a one bit response, a new configuration will be generated to form a new PUF structure. In other words, all adjacent responses are generated by different PUF structures. Even if one challenge is input multiple times, the same DCH structure will generate different responses. Therefore, DNNs cannot model the DCH structure.

Resistance to LR Attack. Rührmair *et al.* [21] implemented an effective modeling attack against APUFs and XOR APUFs using LR, which is based on a special form of the APUF and XOR APUF's output delay calculation. The DCH structure cannot be modeled using this approach, as discussed next. We will first analyze how LR works when attacking APUF and XOR APUF.

First of all, in APUF, the total difference in the delays (referred to as the decision delay) determined by the arbiter is shown in Eq. (2). When LR is used for training, the attacker uses the predicted APUF delay vector to calculate the predicted decision delay, and then inputs a sigmoid function to get a delay difference with the target response, which can be used to modify the predicted delay vector. The search space in this method is much smaller than that for CMA-ES, which is also the reason why LR can attack APUF and XOR APUF very efficiently.

In the *m*-stage *n*-XOR APUF, the responses of individual APUFs need to be determined first, and then XORed to obtain the final response. The LR algorithm cannot be used to model this APUF variant. However, the model of XOR APUF can be transformed, as follows:

$$R_{xor} = \begin{cases} 0, \Delta_{xor}(m) > 0 \\ 1, \Delta_{xor}(m) \leq 0 \end{cases}, \tag{20}$$

$$\Delta_{xor}(m) = \prod_{i=1}^{n} \Delta_{apuf_i}(m) \ . \tag{21}$$

Instead of XORing the multiple responses after the arbiters make their decisions, the delay differences of the individual APUFs can be multiplied, using the product as the decision delay of the XOR APUF. The final output can be determined by this decision delay (in the previous section, the self-XOR mathematical model we provided also uses this calculation method). Similar as the approach taken for APUF, during the training phase, the total decision delay is input into a sigmoid function to get the deviation between the current prediction delay vector and the target delay vector, and using this deviation the prediction model can be modified effectively.

This LR-based algorithm is efficient, but the format of its fitness function limits its application. In the proposed DCH structure, many configurations contain two overlapping FF loops. For such a structure, the calculation of final response cannot be transformed in a similar way as that detailed above, and further in [8] and [31], it is also reported that LR can only model non-overlapping FF APUF structures with fewer than 2 FF loops. Therefore, the LR-based ML algorithm in [21] cannot be used to model the DCH structure.

Resistance to Reliability-based CMA-ES Attack. The reliability-based CMA-ES is an attack method proposed by Becker [24], which uses reliability information from the PUF structure instead of CRPs. In multi-XOR APUF structures, if a classical ML algorithm based on challenge response pairs is used for modeling, the number of CRPs needed will increase exponentially with an increase in the number of XORs. Therefore, when n is large, such as n>10, the XOR APUF is considered to be sufficiently resistant [18]. However, when using reliability-based CMA-ES to model the XOR APUF, the number of challenge-reliability pairs needed only increases linearly.

In this attack, each challenge is used repeatedly *m* times, and the proportion of different responds in the statistical results is obtained. The challenge-reliability pairs reflect the switching possibility of the PUF response when different challenges are applied.

1) In APUF, it can be further inferred that the larger the switching probability, the closer the final decision delay is to 0. Therefore, when unstable challenges are used, the final decision delay can be constrained to different ranges around 0 according to the different switching probabilities.

2) In *n*-XOR APUFs, if the final response has flipped, it implies that at least one individual APUF response has flipped. Therefore, in the unstable challenge set collected, each challenge is responsible for at least one individual APUF's flip. In other words, some challenges in this set will cause the same individual APUF to flip. In the training process, only one APUF delay vector is generated randomly, and its decision delay is limited to 0. After enough iterations,

the vector may converge to one of the individual APUFs. By using the same challenge-reliability pairs set and running this algorithm multiple times, the delay vectors of multiple individual PUFs of the XOR APUF can be obtained.

Next, following the same approach, we will analyze if the reliability-based CMA-ES can be applied to the DCH structure. First, if a response bit is flipped in the APUF-based DCH structure, there are two reasons that can cause the switching: 1) the response of the main path including the FF loops is flipped; 2) the response of the SX branch is flipped. This shows that there are two kinds of unstable challenges in the set of challenge-reliability pairs, one is to make the decision delay of the main path approach 0, and the other is to make the decision delay of the SX branch approach 0. According to the steps, one of the two delay vectors is needed for training. However, unlike the decision delay calculation of APUF and XOR APUF, in the DCH structure, the decision delay calculation of the main path and SX branches must know the exact configuration information. This means that if an attacker tries to model the DCH PUF structure by using the reliability-based CMA-ES algorithm, they must first know the configuration of each challenge-reliability pair. However, the configuration information of the DCH structure cannot be obtained directly by an attacker. Therefore, we claim that the proposed DCH PUF structure is resistant to reliability-based CMA-ES modeling attacks.

5 Dynamic Matching Authentication Protocol

The various configurations in the proposed DCH PUF structure also bring challenges to its application in authentication protocols. In this section, we design an authentication protocol based on the dynamic DCH structure.

5.1 Enrollment Phase

In enrollment phase, the server first sends a sufficient number of challenges to the device side. The device sends the response and the corresponding configuration signal provided by the LFSR to the server. The server then trains the PUF model according to the response signal and the corresponding configuration, and stores the LFSR's complete sequence. After the modeling is complete, the server/device communications link is removed. This enrollment process is shown in Fig. 9. It should be noted that the signals from LFSR are not necessary for sever to model the PUF, but with them, the modeling complexity is greatly reduced (refer to Section 4.2).

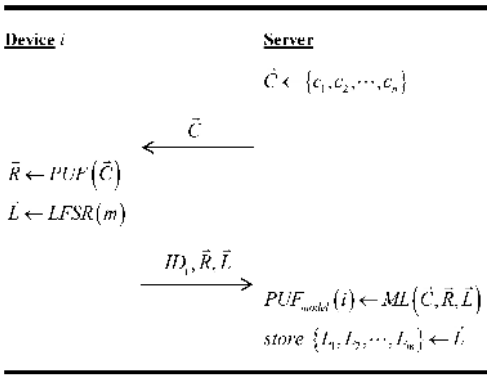

Fig. 9: Enrollment phase.

5.2 Authentication Phase

During the authentication phase, as outlined in Fig. 10, the server first sends a random nonce, $Nonce_b$ to the device. The device sends its ID_i and another random nonce, $Nonce_a$ to the server. The device side concatenates the two nonces and uses $\{Nonce_a \parallel Nonce_b\}$ as the challenge. According to the agreed response bit-length N_{R0}, the device side generates consecutive N_{R0} bit responses R_0 from the PUF. Next, R_0 is placed at a random location in R_a, determined by a random index ind_2, with the remainder of R_a padded with random numbers. R_a is then sent to the server.

The server first verifies whether the ID_i of the device is in its stored list. Then using $\{Nonce_a \parallel Nonce_b\}$ as the challenge, it calculates the response under all LFSR states according to the PUF model, and generates consecutive R_b values. After receiving R_a, N_{R0}-bit substrings are matched with R_b to get the maximum matching degree T. If T is greater than the matching threshold T_{th} set by the system, the authentication of the server to the device passes; otherwise, it fails. Next, the location index $ind'_2 = \max(match_{N_{R0}}(R_a, R_b))$ of the real response in R_a is calculated and sent to the device, here $match_{N_{R0}}(R_a, R_b)$ means calculating the matching degrees between all N_{R0} substrings in R_a and R_b.

After receiving ind'_2, if $|ind_2 - ind'_2| > N_{th}$, the authentication of the device to the server passes, otherwise it fails.

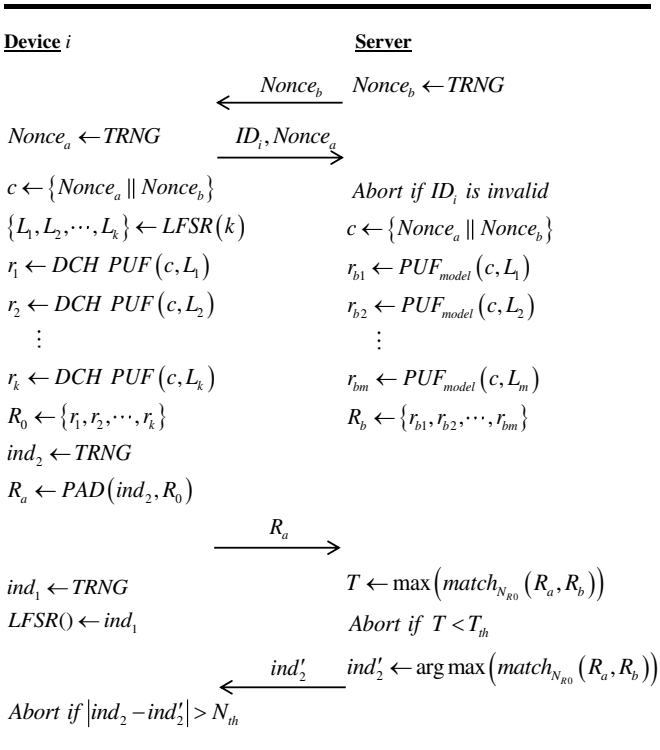

Fig. 10: Authentication phase.

TABLE 4: Recommended Protocol Parameters

Parameters	Symbol	Value
Bit-length of dynamic configurations	N_{LFSR}	9
Bit-length of substring	N_{R0}	64 ∼ 512
Bit-length of authentication bits	N_{Ra}	512
Matching threshold	T_{th}	0.95 ∼ 0.8
Position error threshold	N_{th}	5

Note that the authentication string, R_a, provided by the device includes the substring, R_0 from the real response and padding with random bits generated by a TRNG, as shown in Fig. 11. The starting position of the substring R_0 is determined by the number of clock cycles ind_1 and the previous status S_{last} of the LFSR. After each generation of R_a, the device inputs a random number of ind_1 clock cycles to the LFSR to ensure that the next generated substring is random in the full response string R_{full}.

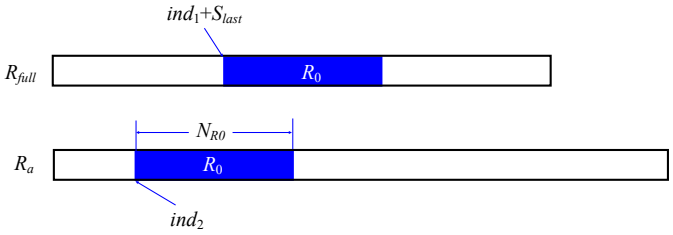

Fig. 11: Authentication bits string.

The proposed protocol differs from the Slender protocol [28], which also is a pattern matching based protocol:

- The proposed protocol uses the DCH PUF structure as the underlying PUF, and the random substrings of the responses are therefore generated from random configurations of the PUF. However, the Slender protocol uses a static PUF structure, and hence, its random substrings are generated from only one PUF structure.
- The proposed protocol generates ind_1 by inputting a random number of clock signals to its LFSR. The Slender protocol uses a TRNG to generate the random index ind_1.

6 Security Analysis of the Protocol

Recommended parameters for the proposed protocol are shown in Table 4. Based on these parameters, we will analyze the security of the protocol in this section. It is worth noting that since the underlying PUF structure we used is different from Slender protocol [28], the BERs of the implemented structures are not the same level. This means that the error tolerance required by the agreement is different. Therefore the parameters proposed in this paper are quite different from Slender protocol.

6.1 ML Attack Resistance

Classial ML attack. To perform ML-based modeling of the proposed DCH PUF, multiple groups of delay vectors are randomly generated for one iteration of the protocol and the group closest to the target delay vector is retained. To calculate which is closest to the target, challenges from the collected CRP set are substituted into the mathematical model of the PUF, and the most consistent response set is calculated. The fitness function, f is:

$$f = \sum_{i=1}^{n} \min\left(HD_{N_{R0}}\left(R'_b, R_0\right)\right), \tag{22}$$

where R'_b is the complete response calculated according to the training model. The fitness function calculates the Hamming distance between all N_{R0}-bit substrings in R'_b and R_0. This calculation needs CRPs from the PUF structure, that is, the real response substring, R_0, from the authentication output in the protocol. However, as the proposed protocol uses random substrings, random positions and string padding, an attacker cannot know which part of the authentication bits string is the real response. If a sufficient number of correct responses are obtained by guessing, the number of possible combinations is:

$$N_{comb} = \left(N_{Ra}\left(N_{Ra} - N_{R0}\right)\right)^{\frac{N_{\min}}{N_{R0}}}. \tag{23}$$

If the underlying structure is a 2-XOR APUF, and $N_{R0} = 64$ is selected, according to Table 3, the number of CRPs needed for modeling is 9,000, and $N_{comb} = (512 \times (512 - 64))^{\frac{9000}{64}} = 229,376^{125}$. Such a large number makes it practically impossible to guess the correct configuration using exhaustive searching.

New ML attack against pattern matching. In [29], Becker proposed a new fitness function of the CMA-ES algorithm, where the Hamming weight of the authentication bits string is used instead of the direct response to reflect the accuracy of the prediction model. The new fitness function is:

$$f' = corr\left(HW\left(\vec{R}'_b\right), HW\left(\vec{R}_a\right)\right), \tag{24}$$

where $HW\left(\vec{R}_a\right)$ represents a vector composed of Hamming weights of all elements of the vector $\vec{R}_a$, and

$$HW\left(\vec{R}_a\right) = \left\{HW\left(R_{a1}\right), HW\left(R_{a2}\right), \cdots\right\}, \tag{25}$$

and equivalently for $HW\left(\vec{R}'_b\right)$. Since the Hamming weight calculation is independent of the random position of the set, the introduction of ind_1 and ind_2 cannot resist this modeling attack. With this method, the Slender protocol can be broken at a relatively lower cost [29].

We use the same fitness calculation from [29] to attack the protocol proposed in this paper. Table 5 shows the simulation results using MATLAB 2017b to perform the CMA-ES modeling. The inputs are blocks of authentication bits strings, where 1 block = 512 bits. The first four runs use the recommended parameters, and the last two runs use the parameters recommended in the Slender protocol. The experimental results show that Becker's method [29]

TABLE 5: CMA-ES Modeling Results of Proposed Protocol When HW is Used as Fitness Function

N_{Ra}	N_{R0}	Inputs	Iterations	Prediction Rate
512	64	10^4	5000	52.38%
512	64	2×10^4	9000	51.06%
512	256	10^4	5000	51.55%
512	256	4×10^4	9000	50.65%
1762	1250	10^4	5000	52.34%
1762	1250	4×10^4	9000	51.73%

cannot break the proposed protocol. In the Slender protocol, different random substrings of the responses are generated by its only underlying PUF. Therefore, the HW vectors of the substrings and full responses have a certain correlation. However, in the proposed protocol, the different random substrings of the responses come from different PUF structures (that is the different configurations of the DCH PUF), and the HW vectors of the substrings are no longer related to each other, which makes this method valid.

6.2 Random Guessing

The probability of successfully guessing the PUF response is related to the bit-length of the authentication bits string, the bit-length of the dynamic configuration values, the bit-length of the substrings, and the matching threshold. The probability that an attacker can determine the correct authentication response with random guessing is:

$$P_{guess} = N_{Ra} \cdot (N_{Ra} - N_{R0}) \sum_{i=T_{th} \cdot N_{R0}}^{N_{R0}} \binom{N_{R0}}{i} (\frac{1}{2})^i (\frac{1}{2})^{N_{R0}-i}. \quad (26)$$

It will take an attacker $N_{Ra} \cdot (N_{Ra} - N_{R0})$ attempts to correctly guess the valid bit-string. Therefore, if the bit-length of the substring N_{R0} is too small, the probability of a correct guess is increased. We have completed experiments for different N_{R0} values when $N_{Ra} = 512$, and each authentication output is randomly guessed 10,000 times. The results are shown in Table 6. When N_{R0} is less than or equal to 32, there will be a certain probability that the attacker will correctly guess all the valid bits, which will lead to a false acceptance. When N_{R0} increases, the proportion of the number of bits guessed correctly by attackers decreases. When $N_{R0} = N_{Ra} = 512$, at most 317 bits are guessed correctly, accounting for only 61.91%.

6.3 Error Tolerance

Another advantage of the proposed pattern matching based protocol is its high error tolerance. In other words, the accuracy of its authentication does not depend on error correction. By adjusting the threshold value for the degree of matching, an honest device can be identified at a certain bit error rate. The error tolerance of the proposed protocol is also shown in Table 6. For authentication bits strings of $N_{R0} = 64$, the maximum matching over 10,000 random guesses is 54 bits, which means that when there are 10 unstable bits in the real response substring, R_0 of an honest device, by setting the matching threshold to $T_{th} = \frac{54}{64} = 84.38\%$, the honest device can still be identified while an incorrect acceptance of a dishonest device can be avoided. Also, as the bit-length R_0 increases, the error tolerance of the protocol improves.

TABLE 6: Random Guess Authentication Bits String Matching Results

N_{R0}	Maximum Matching Bit Number	Error Tolerance	False Acceptance
16	16	-	99.99%
32	32	-	0.02%
48	43	10.42%	0
64	54	15.62%	0
128	96	25.00%	0
192	135	29.69%	0
256	174	32.03%	0
320	205	35.00%	0
384	243	36.72%	0
448	281	37.28%	0
512	317	38.09%	0

7 Discussion and Conclusion

Discussion. The dynamically configurable PUF design presented in this paper can be utilised with existing structures, such as the XOR APUF (shown in Fig. 5) and the recently proposed interpose PUF [18] to improve security. It can also be used with the CRO PUF [32] and RRO PUF [11]. In a DCH PUF, the dynamic configuration signal is obtained by a LFSR and, although an attacker can derive the feedback function by analyzing the structure of the LFSR, the real-time configuration for each CRP remains unknown. This new dynamic authentication mechanism provides strong resistance to modeling attacks.

The proposed pattern matching based protocol has the following advantages: good error tolerance, no disclosure of the complete responses, and further increased resistance to ML attacks by the random selection of substrings and random padding. However, there are also potential disadvantages: for the same challenge, there are many combinations (any sufficient substring length of the complete response) that can be authenticated. This reduces the accuracy requirements of cloned PUF devices to a certain extent, and should be handled appropriately when designing protocols.

In the ML-based attacks mentioned above, CMA-ES is the most flexible one. It can be said that any strong PUF structure with a mathematical model will be threatened by CMA-ES. Its traditional approach is based on the information provided by CRPs, and that the fitness function of the CMA-ES algorithm is very flexible, which enables the derivation of more effective versions. For example, [24] proposed the use of a reliability-based CMA-ES algorithm, which is much more efficient than the traditional ML algorithms when attacking *n*-XOR APUFs with a large *n*. Moreover in [29], the new variant of CMA-ES with HW-based fitness function was capable of breaking the Slender protocol.

Conclusion. Almost all existing strong PUF structures have a static response behaviour. The proposed DCH structure achieves a dynamic PUF using a LFSR. The dynamic design does not conflict with existing structures, rather it

can further improve their security. We also design a corresponding authentication protocol for the proposed dynamic PUF structure. The protocol based on matching pattern not only has strong resistance to traditional ML algorithms, but also resist the new variant of the CMA-ES algorithm. Overall, we demonstrate that the dynamic response mechanism and pattern matching can greatly increase the security of strong PUFs in authentication applications.

Reconfiguration is an effective mechanism to improve security for strong PUF structures, and can be achieved in various methods. However, it is challenging to realize the device identification and authentication without leaving enough information for attackers. At present, PUF design is facing high reliability requirements, and error correction mechanisms often consume a lot of hardware resources. With this concern, it is preferable to use an error-tolerant authentication protocol than to design highly reliable structure with a high cost. Moreover, by setting appropriate thresholds, another concerned aging issue can also be addressed by applying error-tolerant authentication protocol. Considering the incompatibility between different PUF structures and different authentication protocols, the collaborative design between PUF structure and authentication protocol will be a meaningful research direction, and which is also our future research direction.

ACKNOWLEDGMENT

This work has been supported by the National Natural Science Foundation China (62022041 and 61771239).

REFERENCES

[1] C. H. Chang, Y. Zheng, and L. Zhang, "A Retrospective and a Look Forward: Fifteen Years of Physical Unclonable Function Advancement," *IEEE Circuits and Systems Magazine*, vol. 17, no. 3, pp. 32-62, Aug. 2017.

[2] C. Herder, M. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," in *Proc. of the IEEE*, vol. 102, no. 8, pp. 1126-1141.

[3] U. Chatterjee, V. Govindan, R. Sadhukhan, D. Mukhopadhyay, R. S. Chakraborty, D. Mahata, and M. M. Prabhu, "Building PUF Based Authentication and Key Exchange Protocol for IoT Without Explicit CRPs in Verifier Database," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 3, pp. 424-437, May. 2019.

[4] U. Rührmair and D. E. Holcomb, "PUFs at a glance," in *Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE)*, Dresden, 2014, pp. 1-6.

[5] C. Gu, N. Hanley, and M. O'Neill. 2017, "Improved Reliability of FPGA-Based PUF Identification Generator Design," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 10, no. 3, pp. 1-23, Jul. 2017.

[6] M. Rostami, M. Majzoobi, F. Koushanfar, D. S. Wallach, and S. Devadas, "Robust and Reverse-Engineering Resilient PUF Authentication and Key-Exchange by Substring Matching," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 1, pp. 37-49, Mar. 2014.

[7] P. Gope, J. Lee, and T. Q. S. Quek, "Lightweight and Practical Anonymous Authentication Protocol for RFID Systems Using Physically Unclonable Functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831-2843, Nov. 2018.

[8] U. Rührmair, F. Sehnke, J. Solter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proc. of the 17th ACM conference on Computer and communications security*, Oct. 2010, pp. 237-249.

[9] Y. Lao and K. K. Parhi, "Statistical Analysis of MUX-Based Physical Unclonable Functions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 5, pp. 649-662, May. 2014.

[10] Y. Lao and K. K. Parhi, "Reconfigurable architectures for silicon Physical Unclonable Functions," in *Proc. of 2011 IEEE International Conference on Electro/Information Technology*, Mankato, MN, USA, 2011, pp. 1-7.

[11] W. Liu, L. Zhang, Z. Zhang, C. Gu, C. Wang, M. O'Neill, and F. Lombardi, "XOR-based Low-cost Reconfigurable PUFs for IoT Security," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 3, pp. 1-21, Apr. 2019.

[12] A. Spenke, R. Breithaupt, and R. Plaga, "An arbiter PUF secured by remote random reconfigurations of an FPGA," in *Proc. of International Conference on Trust and Trustworthy Computing*, Aug. 2016, pp. 140-158.

[13] M. Ender, A. Moradi, and C. Paar, "The Unpatchable Silicon: A Full Break of the Bitstream Encryption of Xilinx 7-Series FPGAs," in *Proc. of 29th USENIX Security Symposium*, Boston, MA, USA, Aug. 2020.

[14] J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *Proc. of 2004 Symposium on VLSI Circuits. Digest of Technical Papers*, Honolulu, HI, USA, 2004, pp. 176-179.

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

[16] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," in *Proc. of IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, USA, Nov. 2008, pp. 670-673.

[17] W. Liang, S. Xie, J. Long, K. C. Li, and K. Li, "A Double PUF-based RFID Identity Authentication Protocol in Service-centric Internet of Things Environments," *Information Sciences*, vol. 503, pp. 129-147, Nov. 2019.

[18] P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair, and M. van Dijk, "The Interpose PUF: Secure PUF Design Against State-of-the-art Machine Learning Attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 4, pp. 243-290, Aug. 2019.

[19] I. Goodfellow, Y. Bengio, and A. Courville. "Deep Learning," *The MIT Press*, 2016.

[20] M. Khalafalla and C. Gebotys, "PUFs Deep Attacks: Enhanced modeling attacks using deep learning techniques to break the security of double arbiter PUFs," in *Proc. of 2019 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, Florence, Italy, 2019, pp. 204-209.

[21] U. Rührmair, J. Solter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF modeling attacks on simulated and silicon data," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1876-1891, 2013.

[22] N. Hansen, "The CMA Evolution Strategy: A Tutorial," CoRR, abs/1604.00772, 2016.

[23] J. Tobisch and G. T. Becker, "On the scaling of machine learning attacks on PUFs with application to noise bifurcation," in *Proc. of International Workshop on Radio Frequency Identification: Security and Privacy Issues*, Springer, 2015, pp. 17-31.

[24] G. T. Becker, "The gap between promise and reality: On the insecurity of xor arbiter PUFs," in *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems*, Berlin, Heidelberg, Sep. 2015, pp. 535-555.

[25] N. Wisiol, C. Mühl, N. Pirnay, P. H. Nguyen, M. Margraf, J. Seifert, M. van Dijk, and U. Rührmair, "Splitting the interpose PUF: A novel modeling attack strategy," *IACR Cryptology ePrint Archive*, vol. 2019, pp. 1473.

[26] U. Rührmair and M. van Dijk, "PUFs in Security Protocols: Attack Models and Security Evaluations," in *Proc. of 2013 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 2013, pp. 286-300.

[27] C. Gu, C. H. Chang, W. Liu, S. Yu, Q. Ma, and M. O'neill, "A Modeling Attack Resistant Deception Technique for Securing PUF based Authentication," in *Proc. of 2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, Xi'an, China, 2019, pp. 1-6.

[28] M. Rostami, M. Majzoobi, F. Koushanfar, D. S. Wallach, and S. Devadas, "Robust and Reverse-Engineering Resilient PUF Authentication and Key-Exchange by Substring Matching," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 1, pp. 37-49, March 2014.

[29] G. T. Becker, "On the Pitfalls of Using Arbiter-PUFs as Building

Blocks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1295-1307, Aug. 2015.
[30] Y. Wang, C. Wang, C. Gu, Y. Cui, M. O'Neill, and W. Liu, "Theoretical Analysis of Delay-Based PUFs and Design Strategies for Improvement," in *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, Sapporo, Japan, 2019, pp. 1-5.
[31] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Testing techniques for hardware security," in *Proc. of 2008 IEEE International Test Conference*, Oct. 2008, pp. 1-10.
[32] W. Liu, Y. Yu, C. Wang, Y. Cui, and M. O'Neill, "RO PUF Design in FPGAs with New Comparison Strategies," in *Proc. of 2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, Lisbon, Portugal, 2015, pp. 77-80.

Yale Wang received the B.S degree in Automation from Luoyang Institute of Science and Technology, Luoyang, China, in 2013 and his M.S degree in Information Engineering from Henan University of Science and Technology, Luoyang, China, in 2016. He is currently pursuing the Ph.D. degree in Electrical and Information Engineering at Nanjing University of Aeronautics and Astronautics. His research interests mainly include physical unclonable functions (PUFs), machine learning based modeling attacks on PUFs and countermeasures.

Chenghua Wang received the B.S and M.S degrees from Southeast University, Nanjing, China, in 1984 and 1987, respectively. In 1987, he joined the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China, where he became a Full Professor in 2001. He has published six books and over 100 technical papers in journals and conference proceedings. His current research interests include design and test of integrated circuits, circuits and systems for communications, and signal processing. He was a recipient of over 10 teaching and research awards at the national and provincial level.

Chongyan Gu (S'14–M'16) received the Ph.D. degree from Queen's University Belfast, Belfast, U.K., in 2016. She is currently a Lecturer (Assistant Professor) in the School of EEECS at Queen's University Belfast, and a member of Center for Secure Information Technologies (CSIT) with in the Institute of Electronics Communications and Information Technologies (ECIT). Dr. Gu is an expert in hardware security. Her research into physical unclonable function (PUF) has been utilised as part of a security architecture for electronic vehicle (EV) charging systems, licensed by LG-CNS, South Korea, and was also licensed for evaluation by Thales, U.K.. Her team was the overall winner of INVENT 2015, a competition to accelerate the commercialisation of innovative ideas. She has co-authored two research book chapters on the topics of "Lightweight Cryptographic Identity Solutions for the Internet of Things" and "Approximate Computing and Its Application to Hardware Security" published by IET in 2016 and Springer in 2018, respectively. She has successfully organised two conference special sessions (IEEE APCCAS in 2018 and ACM GLSVLSI in 2020). She was invited to give tutorial/talks to international conferences, such as, IEEE ASP-DAC 2020 on the topic of practical PUF design on FPGA. Her current research interests include physical unclonable functions (PUFs), security in/for approximate computing, true random number generator (TRNGs), hardware Trojan detection and machine learning attacks.

Yijun Cui received the B.S and Ph.D. degree in information engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2010 and 2019, respectively. He was a Visiting Ph.D. Student with the Data Security System Group, Centre of Secure Information Technologies, Queen's University Belfast, U.K., from 2014 to 2015. His current research interests are mainly in the hardware security.

Máire O'Neill (M'03-SM'11) is currently Director of the UK Research Institute in Secure Hardware and Embedded Systems (RISE). She is Chair of Information Security and is Research Director of Data Security Systems at the Centre for Secure Information Technologies (CSIT), Queen's University Belfast. She also leads the EU H2020 SAFEcrypto (Secure architectures for Future Emerging Cryptography) project (www.safecrypto.eu). She previously held an EPSRC Leadership Fellowship (2008-2014) and was a former holder of a Royal Academy of Engineering research fellowship (2003-2008). She has received numerous awards for her research work which include a 2014 Royal Academy of Engineering Silver Medal and British Female Inventor of the Year 2007. She has authored two research books and has over 130 peer-reviewed conference and journal publications. She is an Associate Editor for IEEE TC and IEEE TETC and is an IEEE Circuits and Systems for Communications Technical committee member. She is a member of the Royal Irish Academy and a Fellow of the Irish Academy of Engineering. She is also a senior member of the IEEE and a member of the IET and IACR. Her research interests include hardware cryptographic architectures, lightweight cryptography, side channel analysis, physical unclonable functions, post-quantum cryptography and quantum-dot cellular automata circuit design.

Weiqiang Liu (M'12-SM'15) received the B.Sc. degree in Information Engineering from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China and the Ph.D. degree in Electronic Engineering from the Queen's University Belfast (QUB), Belfast, UK, in 2006 and 2012, respectively. In Dec. 2013, he joined the College of Electronic and Information Engineering, NUAA, where he is currently a Professor and the Vice Dean of the college. He has published one research book by Artech House and over 100 leading journal and conference papers. His paper was selected as the Highlight Paper of IEEE TCAS-I in the 2021 January Issue and the Feature Paper of IEEE TC in the 2017 December issue. He has been awarded the prestigious Excellent Young Scholar Award by National Natural Science Foundation of China in 2020. He serves as the Associate Editors for IEEE Transactions on Circuits and System I: Regular Papers (2020.1-2021.12), IEEE Transactions on Emerging Topics in Computing (2019.5-2021.4) and IEEE Transactions on Computers (2015.5-2019.4), an Steering Committee Member of IEEE Transactions on VLSI Systems (2021.1-2022.12). He is the program co-chair of IEEE ARITH 2020, and also technical program committee members for ARITH, DATE, ASAP, ISCAS, ASP-DAC, ISVLSI,GLSVLSI, SiPS, NANOARCH, AICAS and ICONIP. He is a member of CASCOM and VSA Technical Committee of IEEE Circuits and Systems Society. His research interests include approximate computing, hardware security and VLSI design for digital signal processing and cryptography.