# Multiclass Classification-based Side-channel Hybrid Attacks on Strong PUFs

Wei Liu, Ruiming Wang, Xuyan Qi, Liehui Jiang, and Jing Jing

***Abstract*—Physical unclonable functions (PUFs) are promising solutions for low-cost device authentication; hence, ignoring the security of PUFs is becoming increasingly difficult. Generally, strong PUFs are vulnerable to classical machine learning (ML) attacks; however, classical ML attacks do not perform well on strong PUFs with complex structures. Side-channel analysis (SCA) hybrid attacks provide efficient approaches to modeling XOR APUF. However, owing to the inadequate exploitation of all available data, recent SCA hybrid attacks may fail on novel PUF designs, such as MPUF and iPUF. Thus, herein, we introduce a method that combines challenge-response pairs with side-channel information to construct challenge-synthetic-feature pairs (CSPs) via feature cross, thereby making it possible to model strong PUFs through multiclass classification. We propose multiclass classification-based SCA hybrid attacks to model strong PUFs with complex structures. When provided with CSPs, the proposed hybrid attacks use a feed-forward neural network with a softmax activation function to build combined models of PUFs. The combined models predict class labels for given challenges and then reveal responses through simple mappings from these labels. Experimental results show that the proposed attacks could model 16-XOR APUF, (128,5)-MPUF, (8,8)-iPUF, and (2,16)-iPUF with accuracies exceeding 94%. Compared with state-of-the-art modeling techniques, the proposed attack has advantages in terms of modeling accuracy, time cost, and the size of required training data.**

***Index Terms*—Physical unclonable function, side-channel analysis, multiclass classification, synthetic feature, feed-forward neural network.**

## I. Introduction

PHYSICAL unclonable functions (PUFs) are circuits that exploit the inherent randomness introduced by the manufacturing process to give the physical entity a unique trust anchor [1]. Owing to their inherent properties, including being tamperproof, lightweight, and resistant to physical attacks [2], [3], PUFs have received significant attention in the security of internet of things (IoT) applications.

Manuscript received August 5, 2021; revised November 8, 2021 and January 15, 2022; accepted January 31, 2022. This work was supported in part by the National Natural Science Foundation under Grant 61871405 and Grant 61802431. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ulrich Rührmair. (*Corresponding author: Jing Jing.*)

The authors are with the State Key Laboratory of Mathematic Engineering and Advanced Computing, Zhengzhou 450002, China. (e-mail: weiliu_cs@hotmail.com; wangruimin2013nian@163.com; jsjwl_qxy@163.com; jianglie hui@163.com; jingjing_cs@hotmail.com).

Digital Object Identifier 10.1109/TIFS.2022.3152393

According to the number of challenge-response pairs (CRPs), PUFs can be classified as follows: weak PUFs and strong PUFs [4]. Among strong PUFs, the most widely studied structures are the arbiter PUF (APUF) and its variants. In addition to the APUF, well-known arbiter-based PUFs include the XOR APUF [5], lightweight secure PUF (LSPUF) [6], feed-forward PUF (FFPUF) [7], obfuscated PUF (OB PUF) [8], controlled PUF [9], multiplexer-based PUF (MPUF) [10], and the recent interpose PUF (iPUF) [11].

Although defined as unclonable, strong PUFs are vulnerable to machine learning (ML) modeling attacks. Classical ML attacks can deduce intricate mapping relationships between challenges and responses, and subsequently imitate the intrinsic physical behavior of PUFs. APUF, XOR APUF, LSPUF, and FFPUF can be modeled using various ML techniques, e.g., logistic regression (LR) [12], support vector machine (SVM), covariance matrix adaptation evolution strategy (CMA-ES), and deep learning [13], [14]. Even some novel PUF designs, e.g., (128,4)-MPUF and (4,4)-iPUF, are vulnerable to deep neural network (DNN) attacks [15], [16]. Currently, ML attacks are considered to be the most significant threats against PUF designs [17].

However, classical ML attacks cannot be performed on arbiter-based PUFs with complex structures. Here, complex structures mean those adopting nonlinear elements or containing a large number of APUFs as instances. The complexity of arbiter-based PUF is approximated according to the number of APUFs it contains. The complexity of classical ML attacks exponentially increases with the number of APUFs in a target. The time overhead and low accuracy make classical ML attacks impractical against arbiter-based PUFs with complex structures.

Side-channel analysis (SCA) provides an alternative approach to modeling strong PUFs [18]. Unlike classical ML attacks that build mathematical models of PUFs, SCA hybrid attacks employ ML algorithms to build side-channel models of PUFs. Typically, the complexities of side-channel models are polynomial with the number of APUFs in PUF designs. Thus, SCA hybrid attacks can crack PUFs that are difficult to break by ML techniques separately.

However, recent SCA hybrid attacks involve several limitations. 1) Most SCA hybrid attacks fail to demonstrate extensibility on some novel PUF designs, e.g., MPUF and iPUF. They are restricted to attacking a few types of strong PUFs, e.g., XOR APUF and LSPUF [19], [20]. For most

arbiter-based PUFs, even if the internal parameters are learned, the responses remain unknown; 2) SCA hybrid attacks face difficulties in building highly accurate side-channel models for PUFs with complex structures. When attacking XOR APUF and LSPUF, recent SCA hybrid attacks often require a two-step optimization process [20] or a restart [21] to achieve the optimal solution. In addition, even the recently introduced combined reliability attack cannot break iPUF with a large number of APUFs [22]; 3) The data available in SCA hybrid attacks are not effectively exploited. It is worth noting that side-channel models do not include any response information of targeted PUFs. Modeling without using responses of PUFs may be the main reason for the above limitations.

Thus, to address these limitations, we propose a feed-forward neural network (FNN) SCA hybrid attack, using the multiclass classification based on synthetic features to model arbiter-based PUFs accurately[1]. The main idea is to utilize more types of data in modeling PUFs. We introduce a method to construct synthetic features through feature crossing for using all available data together. Unlike previous studies, CRPs and side-channel information are combined into challenge-synthetic feature pairs (CSPs), which have a significantly better predictive ability than either CRPs or side-channel information. The proposed hybrid attack uses the FNN with softmax to train combined models that classify CSPs according to their synthetic features. The learned combined model outputs a class label for a given challenge, and then the response can be revealed through a simple mapping from the label. The experimental results demonstrate that the proposed hybrid attack can successfully model arbiter-based PUFs with complex structures (including XOR APUF, MPUF, and iPUF). In addition, the proposed attack does not require mathematical models of the targeted PUFs.

Our primary contributions are summarized as follows.

1) To the best of our knowledge, this paper is the first to introduce multiclass classification into PUF modeling. In previous studies, PUF modeling attacks were always performed using binary classification. We propose a method based on feature crossing to combine features extracted from different types of data, which open avenues for modeling strong PUFs via multiclass classification. The experimental results demonstrate that multiclass classification based on synthetic features provides more accurate predictions in PUF modeling than binary classification based on a single feature provides.
2) We propose an FNN SCA hybrid attack to build combined models of arbiter-based PUFs. The proposed hybrid attack outperforms both classical ML and SCA hybrid attacks in modeling MPUF (and variants) and iPUF.
3) We present the first successful attempt to model arbiter-based PUFs with complex structures, including (8,8)-iPUF and (2,16)-iPUF, which were previously considered difficult to break.

[1] The source code of the proposed work is available online at https://github.com/newwayclwy/MSA-on-PUFs.

The remainder of this paper is organized as follows. Related studies are discussed in Section II, and Section III introduces notations and provides preliminary information about arbiter-based PUFs. The power side-channel model of strong PUF is analyzed in Section IV. Section IV also presents the method for feature combination and the proposed FNN power analysis hybrid attack. Section V presents the experimental results and analysis, and Section VI discusses the proposed method and directions for future studies. Finally, the paper is concluded in Section VII.

## II. Related Works

Current ML attacks on strong PUFs can be categorized as classical ML attacks and SCA hybrid attacks. Classical ML attacks learn from CRPs to build mathematical models of the PUF, which directly reveals the response. In contrast, SCA hybrid attacks build side-channel models based on challenges and the corresponding side-channel information. Then, the responses are deduced from the learned internal parameters or predicted side-channel information.

### *A. Classical ML Attacks*

The classical ML attack was introduced in 2004, where the APUF was shown to be vulnerable to SVMs [23]. In 2010, Rührmair proposed several ML attacks [24], including LR, SVM, evolution strategy, and deep learning attacks, to model the XOR APUF, FFPUF, and LSPUF. Next, several ML attacks were introduced to realize higher accuracies or to model new variants of strong PUFs [25]. Subsequently, novel arbiter-based PUF architectures, such as MPUF and iPUF, were introduced for resistance to ML attacks.

LR is considered the most efficient attack against XOR APUF [11] and can also be used for modeling iPUF. Owing to the interposed bit on the lower layer of iPUF, LR cannot be performed directly on iPUF. However, a divide-and-conquer attack based on LR modeled (6,6)-iPUF with an accuracy of 83% and (1,7)-iPUF with an accuracy of 97% [26].

CMA-ES provides a heuristic updated direction to take the current model to the target model in each iteration. As a derivative-free optimization method, CMA-ES is less efficient than derivative-based methods like LR. However, as a heuristic algorithm, CMA-ES may perform well when attacking PUF without knowing its mathematical model [27]. In addition, OB PUF and random PUF, with challenges or subresponses hidden, cannot resist heuristic algorithms [28].

Recent studies have shown that DNN attacks are easy-to-use and powerful methods to model PUFs. Deep learning modeling attacks have broad applicability to different PUFs. Most strong PUFs designed for security purposes are insufficiently robust against deep learning-based attacks. The feed-forward back propagation neural network attack has been applied successfully to the XOR APUF, FFPUF, Multi-APUF, and XOR BRPUF [13], [14], [29], [30]. In addition, approximation

attacks based on ANNs had been proposed to model MPUF (and its variants) and modeled the (64,4)-MPUF with 95% accuracy [15]. An FNN-based attack against arbiter-based PUFs successfully modeled (4,4)-iPUF with 97.68% accuracy and (128,5)-MPUF with 96.4% accuracy [16].

It is difficult for almost all classical ML attacks to model strong PUFs with complex structures. For instance, LR and FNN threaten the security of 6-XOR APUF, (6,6)-iPUF, and (1,7)-iPUF; however, 16-XOR APUF, (8,8)-iPUF, and (2,16)-iPUF have demonstrated resistances to LR and FNN due to the subexponential relationship between complexity and amount of training data required for modeling [24], [26].

### *B. SCA Hybrid Attacks*

SCA attacks provide another means to model strong PUFs, and they have particular advantages when modeling arbiter-based PUFs with a large number of APUFs. Some SCA attacks, e.g., reliability-based analysis [31] and the photonic emission attack [32], can be conducted alone; however, they can only be performed on APUF. SCA hybrid attacks, which combine SCAs with ML, are more effective and time-efficient than ML attacks alone. Representative SCA hybrid attacks include reliability-based ML attacks and power analysis hybrid attacks.

Reliability-based ML attacks build PUF reliability models that demonstrate the relationship between the reliability of responses and internal parameters [21]. The measured reliability data and challenges are provided to CMA-ES as training data to learn the internal parameters of PUFs, and fault injection is frequently applied to accelerate the data collection process [33]. The reliability-based CMA-ES has successfully broken XOR APUF and LSPUF. Recently, a gradient-based reliability hybrid attack was proposed for iPUF modeling [22]. This attack combines reliability data, weight constraints, and LR into a single optimization objective and successfully modeled (7,7)-iPUF and (1,10)-iPUF.

ML attacks based on power analysis may be the most famous SCA hybrid attacks. The power consumption of CMOS circuits is an observable physical characteristic that can be exploited as a side-channel. Different methods have been designed to analyze power leakages, e.g., simple power analysis (SPA) and correlation power analysis (CPA). A CPA-based CMA-ES that uses power correlation coefficients as the fitness function was proposed to model controlled PUFs and LSPUFs [19]. In addition, an SPA-based hybrid attack [20] adopts a gradient-based algorithm similar to LR to learn the power side-channel model of XOR APUF. Here, the power side-channel information, which is the cumulative number of logical ones in the output of APUFs before the XOR gate, is used for training in the form of challenge-power side-channel information pairs (CPPs). Both CPA-based and SPA-based hybrid attacks can tackle XOR APUF and LSPUF with polynomial complexity in the number of APUFs as the instances; however, classical ML attacks have exponential complexity. The memristor crossbar PUF, whose mathematical model is similar to APUF, is also vulnerable to a combined optimization-theoretic and SCA hybrid attack [34].

The reported SCA hybrid attacks seldom attack arbiter-based PUFs besides XOR APUF and LSPUF. The reliability models and power side-channel models of PUFs only demonstrate the relationships between challenges and side-channel information. Thus, the original relations between the responses and challenges are not included in side-channel models.

As discussed previously, classical ML attacks and SCA hybrid attacks cannot model strong PUFs with complex structures. Thus, we propose a solution to use all available information effectively and a method to model arbiter-based PUFs with complex structures.

## III. PRELIMINARIES

This section introduces the notations used herein. In addition, an overview of arbiter-based PUFs as targets, including the XOR APUF, MPUF, and iPUF, is presented.

### *A. Notations*

The following notations are used throughout this paper. Vectors are in boldface italic, and scalars are denoted in regular font. Sets are denoted by uppercase letters, and variables are italicized. Superscript T denotes the transpose operation. Table I describes the notations of parameters.

TABLE I
Notations of Parameters

| Symbol | Description |
|---|---|
| $\boldsymbol{c}$ | Challenges of PUF |
| $\boldsymbol{w}$ | The parameter vector |
| $\boldsymbol{\Phi}$ | The feature vector, a transformation form of challenge $\boldsymbol{c}$ |
| $r$ | Responses of PUF, usually single bit. It represents the values of feature **R** |
| $n$ | Number of challenge bits |
| $l$ | Number of APUFs as the components in an arbiter-based PUF |
| $k$ | Number of APUFs used for selection in MPUF |
| $\Delta D_n$ | The delay difference between two paths in the $n_{\text{th}}$ stage in APUF |
| $\oplus$ | XOR operation or modulo-2 addition |
| $\otimes$ | The Kronecker product |
| $\times$ | The Cartesian product |
| $\cdot$ | The arithmetic product |
| **R** | Response as a feature |
| **P** | Side-channel information as a feature |
| $p$ | Values of feature **P**, the power analysis result, can be used as the class label directly |
| **S** | Synthetic feature combined by feature **R** and feature **P** |
| $s$ | Values of feature **S** |
| $label(\ )$ | The function that reveals the class label |
| $\varepsilon(x)$ | The Heaviside function |
| $d_{res}$ | The bias between the predicted response and the real one |
| $d_{sc}$ | The bias between the predicted side-channel information and the real one |

### *B. Arbiter PUF*

The APUF may be the most widely studied strong PUF among silicon PUF implementations. Although proven vulnerable to several modeling attacks, APUFs are used as elements for secure APUF designs owing to their simplicity and low overhead.

The APUF quantizes the difference between the propagation delays of two reconfigurable paths, which are determined by challenges. As shown in Fig. 1, each bit in challenge $\boldsymbol{c} =$

$(c_1, c_2, \ldots, c_n)$ controls a stage to determine whether a path segment is crossed or parallel. A signal feeding into the input side transmits along two paths and then arrives at the arbiter (typically a D flip-flop). The arrival sequence, i.e., the delay difference, determines the response $r$. The behavior of the APUF can be described via the linear additive delay model. The final delay difference between two paths $\Delta D_n$ can be expressed as follows.

$$\Delta D_n = \boldsymbol{w}^T \boldsymbol{\Phi} \tag{1}$$

The parameter vector w= $(w_1, w_2, \cdots w_{n+1})$ represents the propagation delay, which is determined in the manufacturing process. In addition, the feature vector $\boldsymbol{\Phi} = (\phi_1, \phi_2, \cdots \phi_{n+1})$ is a transformation form of challenge $\boldsymbol{c}$, and the transformation function is described as follows.

$$\phi_i = \begin{cases} \prod_{j=i}^{n}(1-2c_j), & 1 \le i \le n \\ 1, & i = n+1 \end{cases} \tag{2}$$

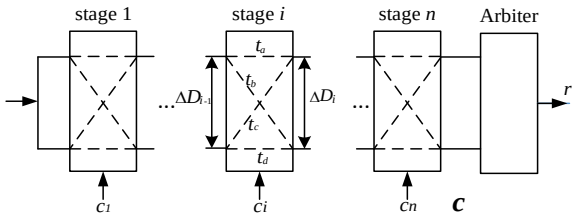



**Fig. 1.** Schematic of an APUF with *n*-bit challenge $\boldsymbol{c}$.

Note that response $r$ is determined by the sign of the final delay difference and can be expressed as follows.

$$r = \varepsilon(\Delta D_n) = \varepsilon(\boldsymbol{w}^T \boldsymbol{\Phi}) \tag{3}$$

Equation (3) shows that the APUF can be expressed as a hyperplane, and the determination of this hyperplane allows the prediction of PUF. When sufficient CRPs are collected, the APUF model can be accessed by either cryptanalysis or ML modeling.

### C. XOR APUF

XOR gates are added to PUF designs to overcome the vulnerabilities of plain APUF, either by increasing the nonlinearity or obfuscating the intermediate results. An $l$-XOR APUF comprises $l$ APUFs in parallel, and each APUF has a unique parameter vector $\boldsymbol{w}$. Challenges are sent to all APUFs in the XOR APUF directly, and the responses of APUFs are XORed to form the output. Fig. 2 shows the XOR APUF architecture.

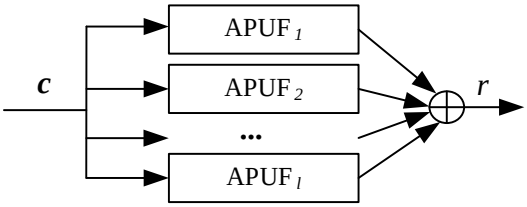



**Fig. 2.** Architecture of $l$-XOR APUF.

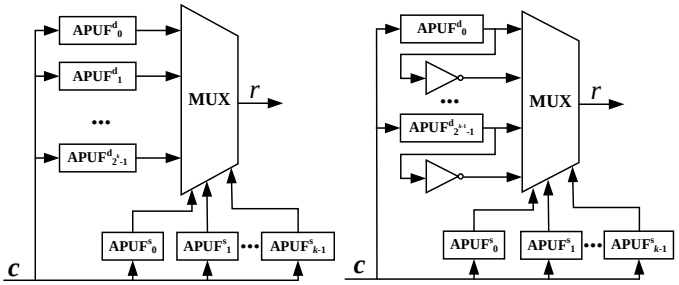
Taking a technical convention that the response of APUF is '−1' when the response is zero, the mathematical model of XOR APUF is built as follows.

$$r_{XOR} = \prod_{i=1}^{l} sign(\boldsymbol{w}^T \boldsymbol{\Phi}) = sign(\prod_{i=1}^{l} \boldsymbol{w}_i^T \boldsymbol{\Phi}_i)$$
$$= sign(\otimes_{i=1}^{l} \boldsymbol{w_i}^T \otimes_{i=1}^{l} \boldsymbol{\Phi_i}) = sign(\boldsymbol{w_{XOR}^T \Phi_{XOR}}) \tag{4}$$

The dimension of the vector $\boldsymbol{w}_{XOR}^T$ is $(n+1)^l$, which means the XOR APUF can be expressed as a hyperplane in an $(n+1)^l$-dimensional space. The modeling difficulty increases exponentially as the scale of the XOR APUF expands.

### D. MPUF

XOR APUF usually suffers from reliability defection. MPUF designs based on multiplexers and multiple APUFs are proposed to overcome the reliability problem while maintaining robustness against cryptanalysis and ML modeling. The family of MPUFs has a basic design and two variants (MPUF, cMPUF, and rMPUF). Fig. 3(a) shows the architecture of the $(n,k)$-MPUF, which employs a $2^k$:1 multiplexer with $k$ select lines connected to the output of $k$ APUFs. The $n$-bit challenge $\boldsymbol{c}$ is sent to all APUFs simultaneously, and the selected APUF outputs the response. The cMPUF variant, whose architecture is shown in Fig. 3(b), was proposed to improve robustness against cryptanalysis. In cMPUF, half of the data inputs of multiplexers are the complement of the other data inputs. The rMPUF variant was proposed to resist reliability-based ML attacks. As shown in Fig. 3(c), an $(n,k)$-rMPUF consists of k stages of 2:1 multiplexers, where selection inputs of all 2:1 multiplexers are generated by independent APUFs. Thus, $(n,k)$-rMPUF employs more APUF instances compared to basic $(n,k)$-MPUF.



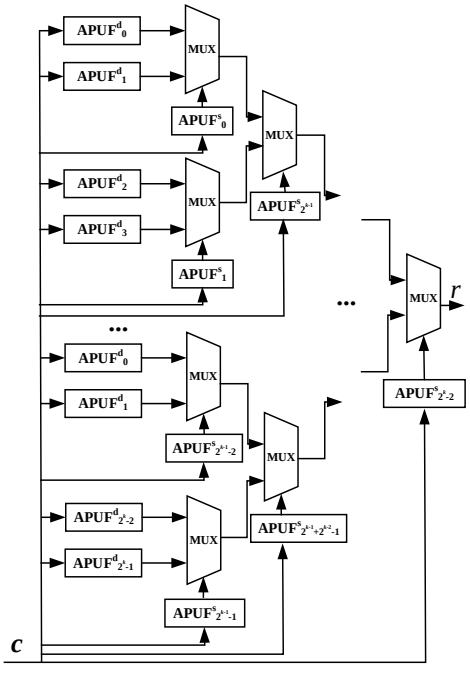




**Fig. 3.** Architectures of $(n,k)$-MPUF and its variants.

### *E. iPUF*

The iPUF comprises two layers of XOR APUFs. Fig. 4 shows the ($x$,$y$)-iPUF architecture. Here an $n$-bit challenge $\boldsymbol{c}$ is applied to both layers, and the 1-bit response $r_U$ of $x$-XOR APUF on the upper layer is interposed between two subchallenges of the $y$-XOR APUF on the lower layer to form an ($n$+1)-bit challenge. Then, the final response $r$ is determined by the lower layer. The iPUF is a promising replacement for XOR APUF and has been shown to be invulnerable to reliability-based CMA-ES and LR both theoretically and experimentally.

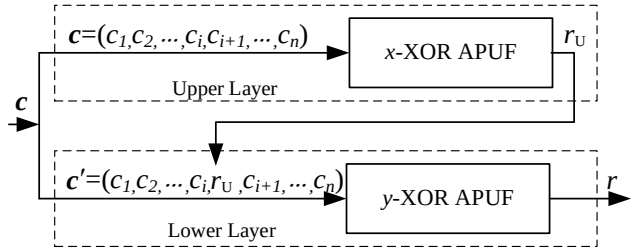



**Fig. 4.** Architecture of ($x$,$y$)-iPUF.

## IV. SCA Hybrid Attack Based on Multiclass Classification

Herein, we present a detailed analysis of the power side-channel model of arbiter-based PUFs and then introduce a method to combine CRPs with side-channel information via feature crossing. We then elaborate on the FNN SCA hybrid attack based on multiclass classification. Finally, the hyperparameters of the FNN are described.

### *A. Analysis of Power Side-Channel Model*

Dynamic power analysis of PUFs applies power tracing to determine the transition from zero to one of the latches, which are fundamental elements acting as arbiters in APUFs. When a latch transitions to one, there is a current peak. The amount of drawn charges, which is the integration of the current curve, is linearly proportional with the number of latches output one. For PUF designs that employ more than one APUF in parallel, by measuring the amount of current drawn from the supply voltage during any latch transition, the cumulative number of APUFs that respond with one can be determined [35].

For an arbiter-based PUF containing $l$ APUFs, the power side-channel information takes ($l$+1) values, representing the cumulative numbers of APUFs that respond with one. The power side-channel information here is not the original power consumption data collected by equipment or software, but the results of power analysis on collected power traces. The response of the $i^{\text{th}}$ APUF can be represented by $\varepsilon(\boldsymbol{w}_i^T\boldsymbol{\Phi}) \in \{0,1\}$. The power side-channel model of the arbiter-based PUF can be formulated by (5), which demonstrates the relationship among challenges, the internal parameters, and the power side-channel information.

$$p = \sum_0^l \varepsilon(\boldsymbol{w}_i^T\boldsymbol{\Phi}) \tag{5}$$

The power side-channel model reveals the statistical number of APUFs with the responses of logical ones. The model can be applied to all arbiter-based PUFs, even those complemented with other logic elements, such as XOR gates or multiplexers.

Without considering any other elements except APUFs, the power side-channel model is more straightforward than the mathematical model; thus, it is easier to learn by ML algorithms. ML computation times are improved from exponential to polynomial in the number of APUFs. For a few specific PUFs, e.g., XOR APUF and LSPUF, when the power side-channel model is learned, the responses can be predicted by the model directly. The responses of XOR APUFs can be calculated as follows.

$$r_{XOR} = p \bmod 2 \tag{6}$$

Unfortunately, the same methods do not work on other arbiter-based PUFs. In most cases, it is difficult to describe the relationships between the power side-channel information and the responses in mathematical formulas. Therefore, even if an optimal solution of the internal parameters is learned, we can predict how many APUF output logical ones for a given challenge; however, the response remains unknown. This may explain why only a few ML power analysis hybrid attacks on other PUFs have been reported.

Although the power side-channel model fails to reveal the response, the power side-channel information is helpful in terms of modeling the arbiter-based PUFs. According to the mathematical and power side-channel models, the response and power side-channel information are relevant to the internal parameters. Therefore, side-channel information as relative data fed into ML algorithms may improve modeling accuracy and reduce the scale of the training data. The main objectives of this study are to establish relationships between responses and side-channel information and use CRPs and side-channel information together in attacks.

### *B. Synthetic Feature Construction*

To exploit all available data, we propose a method to combine CRPs with side-channel information via feature crossing. As the combination of two or more features, the synthetic feature provides predictive abilities beyond what a single feature can provide individually.

In the SCA hybrid attack scenario, each challenge possesses two types of implicit features given by the targeted PUF, i.e., response **R** and side-channel information **P**. Challenges can be classified into two classes according to their responses. The classification model is equivalent to the mathematical model of the targeted PUF. Feature **P** can be related to any side-channel, e.g., power, electromagnet, and even reliability. In this study, we focus on the power side-channel. Supposing that the feature **P** takes values from a discrete set $[0, l]$, and each value represents an individual side-channel state. A challenge with an implicit feature **P** valued $\mathrm{p}_i$ means that $\mathrm{p}_i$ APUFs respond with one while the targeted PUF operates on the challenge. According to the values of **P**, challenges can be classified into ($l$+1) classes; therefore, multiclass classification ML algorithms may be the optimal choice to learn side-channel models.

| **Synthetic Feature Construction** |
|---|
| **Precondition:** $\mathbf{R} = \{\mathrm{r}_i \mid \mathrm{r}_0 = 0, \mathrm{r}_1 = 1, 0 \le i \le 1, i \in N\}$ |

$$\mathbf{P} = \{p_j | p_j \in Q, 0 \le j \le l, j \in N\}$$
$$\textbf{Feature cross: } \mathbf{S} = \mathbf{R} \times \mathbf{P}$$
$$= \{(r, p) | r \in \mathbf{R} \wedge p \in \mathbf{P}\}$$
$$= \{s_k | s_k = (r_i, p_j), k = i * (l+1) + j, 0 \le k \le 2l+1, k \in N\}$$

The synthetic feature **S** is a combination of feature **R** and feature **P**. As the Cartesian product of two individual features, synthetic feature **S** takes 2($l$+1) possible values. Here, a challenge with synthetic feature $s_0$ can be interpreted as follows. When the challenge is sent into the PUF, the observed response is zero, and the numerical value measured from a specific side channel is $p_0$.

| Response | Class label | One-hot code (2-element) |
|---|---|---|
| $r_0$ | 0 | 01 |
| $r_1$ | 1 | 10 |

| Side-channel information | Class label | One-hot code((l+1)-element) |
|---|---|---|
| $p_0$ | 0 | 0…001 |
| $p_1$ | 1 | 0…010 |
| … | … | … |
| $p_l$ | $l$ | 1…000 |

| Side-channel information \ Response | $r_0$ | $r_1$ |
|---|---|---|
| $p_0$ | $s_0$ | $s_{l+1}$ |
| $p_1$ | $s_1$ | $s_{l+2}$ |
| … | … | … |
| $p_l$ | $s_l$ | $s_{2l+1}$ |

| Synthetic feature | Class label | One-hot code (2(l+1)-element) |
|---|---|---|
| $s_0$ | 0 | 0…001 |
| $s_1$ | 1 | 0…010 |
| … | … | … |
| $s_{2l+1}$ | 2$l$+1 | 1…000 |

**Fig. 5.** Synthetic feature construction represented in one-hot coding.

In the feature combination process, class labels are used rather than values, and one-hot encoding is recommended to represent class labels. In practice, real-valued features seldom cross directly. Instead, one-hot feature vectors representing the class labels of features are crossed. Fig. 5 illustrates the combination of two features represented in one-hot coding. Here, response **R** produces separate two-element feature vectors due to the binary nature of PUFs, and the side-channel information **P** produces separate ($l$+1)-element feature vectors according to the values. The synthetic feature **S**, formed by taking a Cartesian product of the individual class labels, is also presented in the class label and represented by a 2($l$+1)-element one-hot vector.

Combining two features creates a synthetic feature and establishes particular logical conjunctions of responses and side-channel information. Thus, CRPs and side-channel information can be combined to construct CSPs. When synthetic feature **S** is adopted as the classification criterion, challenges are classified into 2($l$+1) independent classes. Multiclass classification provides a more precise classification than binary classification. The model built through multiclass classification based on synthetic features acts as a combination of mathematical and side-channel models. The learned model predicts a class label on a given challenge, and both the response and power side-channel information can be extracted from the predicted label. After developing the synthetic feature, we focus on predicting response via multiclass classification.

### *C. Multiclass Classification-based Hybrid Attack*

Based on the multiclass classification, we design ML SCA hybrid attacks to build combined models of arbiter-based PUFs. Here, we take the ML power analysis hybrid attack as an example. Challenge ***c*** and its corresponding synthetic feature $s$ are assembled to construct CSPs as training data, in which the synthetic feature $s$ exists in the form of class labels. The attack process involves the following three stages.

**Pretreatment stage**: Training data are collected as follows.

1) A randomly selected challenge ***c*** is sent to the PUF, and then the corresponding response $r$ is observed. Simultaneously, the amount of current drawn from the supply voltage is measured, and then the cumulative number $p$ of APUFs that respond with a logical one is determined via SPA.

2) According to the proposed feature combination method, the synthetic feature $s$ is calculated as the Cartesian product of the response $r$ and power side-channel information $p$. Here, all features are represented by class labels, and the values of features **R** and **P** can be used directly as class labels. The one-to-one mapping rule among labels follows the definition expressed in (7).

3) Repeat steps 1) and 2) until a sufficient number of CSPs are collected.

**Learning stage**: Multiclass classification algorithms are adopted in the proposed attack. When the classes are not mutually exclusive, some individual LR classifiers are required, and these use the one-against-one or one-against-all strategies. If the class is exclusive, multiclass classification is generally handled using a softmax regression classifier, which results in one out of the entire classes.

As training data, CSPs are input into ML algorithms. After training, a combined PUF model is built, and this model processes the function of both the side-channel and mathematical models.

**Prediction stage:** A class label $label(s')$ is predicted by the learned model on a given challenge. The final response $r'$ can be extracted from $label(s')$ according to the rules described in (8). In addition, the power side-channel information can also be deduced from the predicted label, although predicting the power side-channel information is not the focus of our study.

**Multiclass Classification-based Hybrid Attack**

**I. Pretreatment stage**

**Input:** CRPs, P= $\{0,1,\dots,l\}$

**Process:** **S**= **R** × **P**

$$label(s) = label(r) \cdot (l+1) + label(p) \quad (7)$$

**Output:** CSPs

**II. Learning stage**

**Input:** CSPs

**Process:**

Training by multiclass classification ML algorithms

**Output:** A combined model

**III. Prediction stage**

**Input:** Challenge ***c***

**Process:** Send challenge into the combined model

**Intermediate result:**

Predicted class label $label(s')$

**Process:**

$$r' = 0 \;\; label(p') = label(s') \;\; if \; label(s') < l + 1$$
$$r' = 1 \;\; label(p') = lable(s') - l - 1 \;\; if \; label(s') \geq l + 1 \quad (8)$$

**Output:** Predicted response $r'$

The proposed hybrid attack is applicable to all arbiter-based PUFs without requiring knowledge of the mathematical models. According to the proposed hybrid attack, ML algorithms for multiclass classification should be appropriately selected.

### *D. FNN for Multiclass Classification*

Feature crossing on massive datasets is an efficient strategy to learn highly complex models, and neural networks represent another strategy. The FNN with softmax as the activation function is a recommended method for multiclass classification tasks. An FNN comprises an input layer, several hidden layers, and an output layer. Here, information only travels forward in the FNN, through the input layer, then through the hidden layers (if present), and finally arrives at the output layer.

According to the complexity of targets, the FNN in hybrid attacks adopts one or two hidden layers. Here, the complexity of an arbiter-based PUF is evaluated according to how many APUFs it contains. When the number of APUFs is greater than 10, two hidden layers are suggested. Otherwise, a single hidden layer may be sufficient. ReLU is employed as the activation function of the input layer and hidden layers for free from the gradient-vanishing effect. The softmax function is adopted as the activation function of the output layer for multiclass classification. An FNN architecture implemented is shown in Fig. 6.

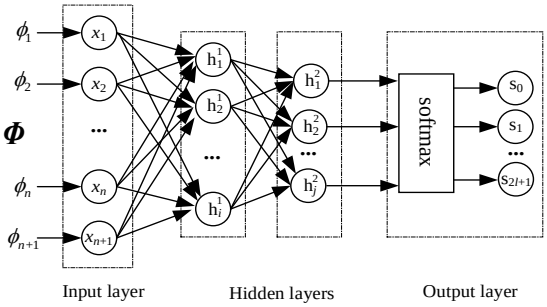



**Fig. 6.** An FNN architecture for multiclass classification.

In this study, binary_crossentropy (rather than categorical _crossentropy) is employed as the cross-entropy loss function to optimize the model. Adam and RMSprop optimizers both work well in the FNN. As a summary, the hyperparameters adopted in the hybrid attack are presented in Table II.

TABLE II
Hyperparameter Values

| Hyperparameter | Values |
|---|---|
| Activation function of the output layer | Softmax |
| Activation functions of the input layer and hidden layers | ReLU |
| Number of hidden layers | 1 or 2 |
| Loss function | Binary_crossentropy |
| Optimizer | Adam or RMSprop |

## V. Experimental Results and Analysis

In this section, we describe the experimental setup and present the experimental results to demonstrate the effectiveness and superiority of the proposed method compared to state-of-the-art methods. We also present an analysis of the partially correct prediction in multiclass classification and discuss the major factors in the proposed methods that contribute to the good performances.

### *A. Experimental Setup*

The uniqueness of FPGA-based APUF implementations is relatively poor [36]. In these experiments, we collected CRPs and power traces from simulation models built with PSPICE. The component named line ideal is adopted to implement the propagation delay. A single stage of APUF is constructed by combining two ideal lines with a 2:1 multiplexer. A D flip-flop is adopted as the latch. As a device, the APUF model uses the selection pins of multiplexers for inputting challenges and output response from the D flip-flop. All delay parameters of APUFs are random numbers generated by MATLAB, and so do the challenges.

We collected the power trace by measuring the amount of current drawn from the supply voltage during the simulated PUF operating. By putting a current marker to the GND pin or the VCC pin of the supply voltage, the current trace can be measured. During the power consumption measurements, a voltage marker on the output of the D flip-flop presents the response simultaneously. We calculated the integration of the collected current trace as the amount of drawn charges. After subtracting the amount of charges normally drawn in case of a floating load, the amount of drawn charges was linearly proportional with the number of latches output one. The power side-channel information, which was the cumulative number of APUFs that respond with one, could be revealed by a division. In our study, we followed the power analysis method proposed by Rührmair *et al*. [20]. A more detailed process can be found in the reference.

The proposed hybrid attack was implemented using Python 3.7.3, Keras 2.4.3, and TensorFlow 2.3, and all experiments were conducted on a laptop with a 2.2 GHz, six-core Intel I7-8750 processor, and 32 GB RAM.

### *B. Modeling Attacks and Results Analysis*

We selected the XOR APUF, MPUF (and its variants), and iPUF to verify the effectiveness of the proposed method. Here, all modeled PUFs were with 128-bit challenges. For each PUF, the generated dataset was divided into training data (80%) and test data (20%).

We compared the proposed method to five corresponding studies: the traditional LR attack [12], the traditional FNN attack [16], the SPA-based LR attack [20], the LR-based divide-and-conquer attack (LDA) [26], and the most recent gradient-based reliability attack (GRA) [22]. An overview of the comparisons was provided in Table III, and more detailed information was presented according to PUF types. As shown

in Table III, the proposed attack is the only one that can model all listed complex PUFs.

Unlike other attacks, each case of the proposed attack has two accuracies in Tables IV, V, and VI. The former accuracy is for response prediction, and the latter one in the parentheses is the accuracy of multiclass classification by FNN. Note that a correct class label prediction means that both response and power side-channel information are predicted precisely. However, a wrong prediction that only mistakes the power side-channel information may indicate the response correctly. The influence of these partially correct predictions is illustrated explicitly in Section V-C.

TABLE III
Comparisons of Attacks on Complex PUFs

| Target / Method | XOR APUF | | | MPUF | | iPUF | | |
|---|---|---|---|---|---|---|---|---|
| | 6 | 12 | 16 | (128,4) | (128,5) | (6,6) | (8,8) | (2,16) |
| Proposed | √ | √ | √ | √ | √ | √ | √ | √ |
| LR [12][37] | √ | × | × | × | × | × | × | × |
| FNN [11] [16] | √ | × | × | √ | √ | × | × | × |
| SPA-based LR [20] | √ | √ | √ | × | × | × | × | × |
| GRA [22] | √ | × | × | × | × | √ | × | × |
| LDA [26] | √ | × | × | × | × | √ | × | × |

Diverse attacks require various training data. The classical ML attacks, e.g., LR and FNN, use CRPs for training, and the SPA-based LR attack requires CPPs. The proposed method requires CSPs. CRP, CPP, and CSP are different combinations of challenge, response, and power side-channel information. Here, CRP represents the challenge and response, CPP means the challenge and power side-channel information, and a CSP comprises all three elements. Note that the GRA requires both challenges and the corresponding reliability information. For the convenience of comparing the training data scales, we use CRP to represent all data types uniformly in all tables.

*1) XOR APUF*

Fig. 7 presents the prediction accuracies of $l$-XOR APUFs ($l$ = 4, 6, and 8) achieved by the proposed hybrid attack. All tested XOR APUFs are successfully modeled with accuracies exceeding 95%, and the modeling difficulty increases with the parameter $l$. In addition, more CRPs are required to train models of higher accuracies. When modeling 8-XOR APUF, the proposed method requires $2 \times 10^5$ CRPs to achieve 96.43% accuracy. However, three times the number of CRPs are required to improve the performance to 97.96%.

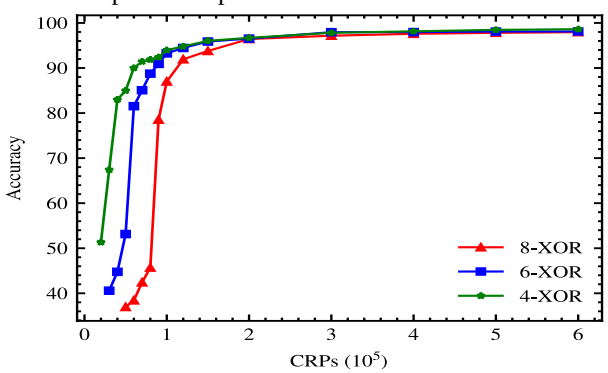



**Fig. 7.** Prediction accuracies of $l$-XOR APUFs (challenge length $n$=128 bits, the FNN used single hidden layer, iterations = 200).

Table IV lists the modeling accuracies of XOR APUFs. The proposed method successfully modeled 16-XOR APUF, which only was broken by the SPA-based LR attack before. Compared to the SPA-based LR attack, the proposed method shows an advantage in time costs but requires more CRPs to achieve the same accuracy.

The traditional LR attack is considered one of the most efficient attacks against XOR APUF. The FNN adopted in the proposed attack uses a black box learning approach; on the contrary, the traditional LR uses a precise mathematical model of the XOR APUF. That is to say, compared to the proposed attack, the traditional LR needs less training data when modeling the same XOR APUF. However, the traditional LR has difficulties in modeling 8-XOR APUF. Furthermore, the modeling complexity of the proposed attack linearly increases with the number of APUFs in the target; in contrast, the complexity of classical ML attacks increases exponentially.

TABLE IV
Modeling Accuracy for XOR APUFs

| No. of XOR | Method | No. of CRPs ($10^5$) | Prediction accuracy | Training time |
|---|---|---|---|---|
| 4 | Proposed | 2.55 | 98.1% (98.1%) | 2 min 13 s |
| | LR [12] | 0.24 | 99% | 2 h 52 min |
| | FNN [16] | 2.55 | 97.8% | 8 min 30 s |
| 5 | Proposed | 6.55 | 98.4% (98.4%) | 6 min 49 s |
| | LR [12] | 5 | 99% | 16 h 36 min |
| | FNN [16] | 6.55 | 97.87% | 29 min 21 s |
| 6 | Proposed | 3 | 97.6% (97.6%) | 2 min 34 s |
| | LR [12] | —— | —— | —— |
| | FNN [16] | 12 | —— | —— |
| 8 | Proposed | 3 | 97.2% (97.2%) | 2 min 30 s |
| | SPA-based LR [20] | 0.516 | 98.0% | 13 min |
| 12 | Proposed | 3 | 96.9% (96.9%) | 4 min 51 s |
| | SPA-based LR [20] | 0.774 | 97.3% | 47 min |
| 16 | Proposed | 6 | 97.8% (97.8%) | 9 min 52 s |
| | SPA-based LR [20] | 1.032 | 97.5% | 2 h 28 min |

The accuracies of modeling 6-XOR APUF by LR and FNN [12], [16] were not reported here because either the training time exceeds the expected boundary or the amount of required training data is greater than the limit for practical applications.

*2) MPUF*

The results of modeling MPUF and its variants are presented in Table V. The accuracies in the parentheses, which are for predicting class labels, are lower than the accuracies of the traditional FNN. However, by considering the partially correct predictions, we found that the proposed method ultimately outperformed the traditional FNN in predicting response.

TABLE V
Modeling Accuracy for MPUFs

| Target | Method | No. of CRPs ($10^5$) | Prediction accuracy | Training time |
|---|---|---|---|---|
| (128,3)-MPUF | Proposed | 1.12 | 98.57% (93.17%) | 2 min 19 s |
| | FNN [16] | 1.12 | 97.50% | 3 min 23 s |
| (128,4)-MPUF | Proposed | 1.84 | 98.30% (90.83%) | 3 min 35 s |
| | FNN [16] | 1.84 | 96.49% | 16 min 10 s |
| (128,5)-MPUF | Proposed | 3.12 | 97.56% (81.08%) | 9 min 52 s |
| | FNN [16] | 3.12 | 96.40% | 22 min 43 s |
| (128,5)-rMPUF | Proposed | 6 | 93.66% (44.91%) | 33 min 37 s |
| | | 8 | 96.18% (49.35%) | 33 min 43 s |
| | FNN [16] | 4 | 95.45% | 32 min 27 s |
| (128,5)-cMPUF | Proposed | 2.15 | 98.69% (90.36%) | 5 min 10 s |
| | FNN [16] | 2.15 | 96.36% | 10 min 13 s |

An exception occurred in modeling (128,5)-rMPUF. The proposed method need more data than traditional FNN to achieve high accuracies in the response prediction. The results also revealed that most of the responses remained correct even though over half of the class labels were wrongly predicted by the proposed method. Too many APUFs as components maybe accounts for the poor performance in multiclass classification. A ($n$,$k$)-rMPUF contains $(2^{k+1}-1)$ APUFs. However, only $(k+1)$ APUFs in the ($n$,$k$)-rMPUF contribute to the response once a time. According to the power side-channel model, the modeling complexity of a ($n$,$k$)-rMPUF approximately equals that of a $(2^{k+1}-1)$-XOR APUF. Contrarily, according to the mathematical model, the modeling complexity of a ($n$,$k$)-rMPUF equals that of a $(k+1)$-XOR APUF. Thus, it is more suitable to attack rMPUF by adopting traditional FNN to build the mathematical model when the number of APUFs exceeds 40.

*3) iPUF*

Table VI lists iPUF modeling accuracies obtained by four methods. The proposed method modeled iPUFs with better performances than state-of-the-art attacks, which meant higher accuracies, fewer CRPs, and smaller time costs.

The designers of iPUF suggested adopting practical design parameters as (1,10)-iPUF with the interposed bit in the middle. To test the effectiveness of the proposed method, we selected a more complex target (2,16)-iPUF, and successfully modeled it.

TABLE VI
Modeling Accuracy for iPUFs

| Parameter ($x$,$y$) | Method | No. of CRPs ($10^5$) | Prediction accuracy | Training time |
|---|---|---|---|---|
| (4,4) | Proposed | 6.47 | 97.89% (96.37%) | 7 min 21 s |
| | FNN [16] | 6.47 | 97.68% | 32 min 17 s |
| (5,5) | Proposed | 10 | 98.23% (97.22%) | 11min 58 s |
| | FNN [16] | 12 | —— | —— |
| | LDA [26] | 10 | 98.00% | 14 min 36 s |
| (7,7) | Proposed | 6 | 96.03% (94.76%) | 6 min 23 s |
| | LDA [26] | 40 | 74% | 17 h 12 min |
| | GRA [22] | 6 | 81% | 6 h 42 min |
| (1,10) | Proposed | 6 | 96.27% (96.26%) | 6 min 17 s |
| | GRA [22] | 8 | 89% | 2 h 5 min |
| (8,8) | Proposed | 6 | 95.38% (94.00%) | 5 min 45 s |
| (2,16) | Proposed | 6 | 94.52% (93.01%) | 10 min 50s |

## C. Analysis of Partially Correct Prediction

The prediction by multiclass classification reveals both response and power side-channel information. Considering the partially correct predictions, prediction accuracies of response should be higher than that of the class label. To analysis actual accuracies of response prediction, we verified 12000 samples of prediction errors when attacking 16-XOR APUF, (128,5)-MPUF, (8,8)-iPUF, and (2,16)-iPUF separately. As shown in Table VII, the partially correct rates of different PUFs vary from 0.2% to 90.02%. The MPUF modeling has exceptionally high rates. The response prediction accuracy of (128,5)-MPUF finally reached 98.64% because over 90% of incorrect labels map to the correct responses.

TABLE VII
Calibrated Modeling Accuracy of Proposed Attack

| Target | Prediction accuracy of label | Partially correct rate | Prediction accuracy of response |
|---|---|---|---|
| 16-XOR APUF | 95.03%[a] | 0.20% | 95.03% |
| (128,5)-MPUF | 86.27%[b] | 90.02% | 98.64% |
| (8,8)-iPUF | 94.00% | 23.06% | 95.38% |
| (2,16)-iPUF | 93.91% | 10.00% | 94.52% |

95.03%[a]: achieved using $3\times10^5$ CRPs
86.27%[a]: achieved using $16\times10^5$ CRPs

For further analysis, we introduce two parameters $(d_{res}, d_{sc})$ to quantify prediction bias in response and side-channel information individually. These two parameters are defined as follows:

$$d_{res} = \begin{cases} true, & if\ r' = r \\ false, & if\ r' \neq r \end{cases} \tag{9}$$

$$d_{sc} = label(p') - label(p) \tag{10}$$

The $d_{res}$ reveals whether the response is correctly predicted. The $d_{sc}$ represents the bias between the predicted power side-channel information and the real one, concretely, the difference in the number of APUFs respond with one. The relationships among $label(p')$, $r'$, and $label(s')$ are illustrated in (8). Based on these two parameters, we analyze the prediction errors that occurred in modeling four PUFs. Statistics of prediction errors are exhibited by 3-dimensional histograms in Fig. 8.

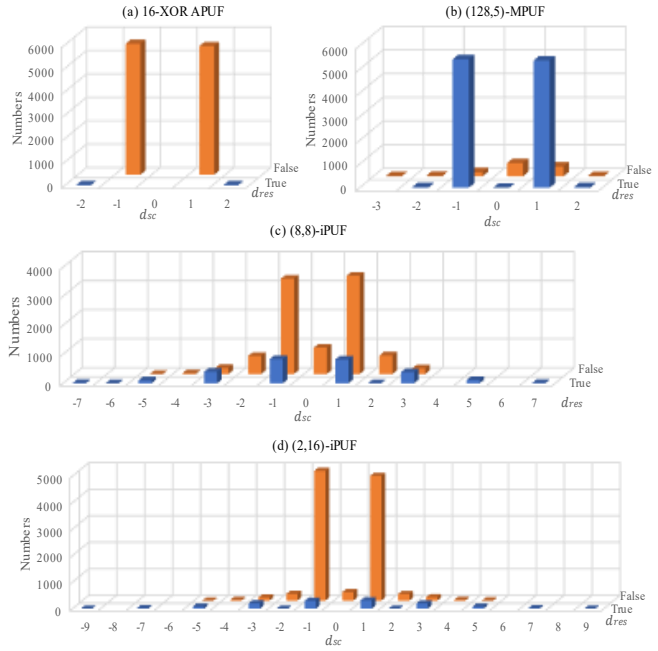



**Fig. 8.** Statistics of prediction errors in modeling four arbiter-based PUFs.

According to the probability theory, the most common error in prediction is mistaking a single APUF response, corresponding to the value '1' and '-1' of $d_{sc}$. This single mistake may occur at any APUF with the same probability owing to the randomness of PUF internal parameters. In addition, the probability of mistaking two APUF responses is much smaller than that of mistaking only one.

For XOR APUF, if the combined model mistakes a single APUF response, this error will lead to a prediction error of XOR APUF response. Nevertheless, if two responses of APUFs are wrongly predicted, the response of XOR APUF will remain correct due to the XOR operation.

In Fig. 8(b), a large number of prediction errors fall into the category of the wrong side-channel information prediction with the right response prediction. The value '1' and '-1' of $d_{sc}$ mean that an APUF response is incorrectly predicted; however, if this APUF is not selected by multiplexers or not used for selection, this wrong prediction will not influence the response prediction of MPUF. During an operation of (128,5)-MPUF, only six APUFs out of the total 37 decide the final response. That is why over 90% of wrong predictions still map to the correct response.

According to the two-layer architecture of iPUF, when the response of APUF in the upper layer is wrongly predicted, the iPUF response prediction will be affected with a probability of 50%. However, if the same misprediction occurs in the lower layer, the iPUF response will be wrongly predicted. As shown in Fig. 8(c), when $d_{sc}$ takes value '1' or '-1', the partial correct predictions make up about a quarter of the total perdition errors in modeling (8,8)-iPUF.

### *D. Comparisons of Classifications Based on Different Features*

To exploit the major factor in the proposed method that contributes to its good performance, we designed experiments to model the 16-XOR APUF, (128,5)-MPUF, (8,8)-iPUF, and (2,16)-iPUF in different ways. For each PUF type, we conducted three FNN-based modeling attacks for classification based on different features.

Attack A: The training data were classified according to the synthetic feature. Here, CSPs were used for training, and the proposed hybrid attack was adopted to build the combined PUF model.

Attack B: The training data were classified according to the side-channel information. Here, CPPs were used for training. The FNN power analysis attack, as a variant of the proposed hybrid attack, was adopted to build the power side-channel model of PUF via multiclass classification.

Attack C: The training data were classified according to the response. The FNN modeling attack, which is a form of the traditional ML attacks, was adopted to build the mathematical model of the PUF.

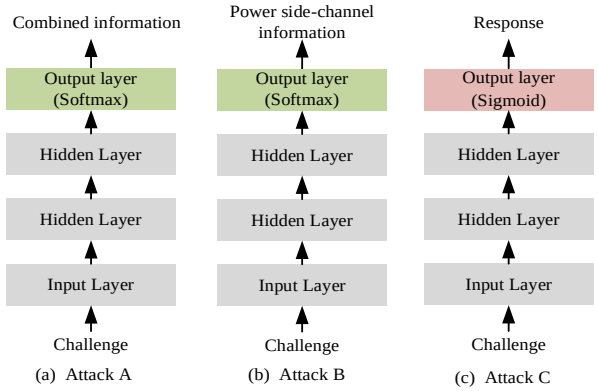



**Fig. 9.** Three FNN-based modeling attacks on PUFs.

The amount of training data was kept the same for all FNN-based modeling attacks. All FNNs used in these attacks has the same structure, which means the number of hidden layers and the FNN hyperparameters were kept constant. The only difference is in the activation function of the output layer. Sigmoid was employed as the output layer activation function to predict response directly, and Softmax was selected for multiclass classification.

Table VIII lists the modeling accuracies of classifications based on different features. The predictions extracted from the predicted labels achieve higher accuracies than the direct predictions of response or power side-channel information. Responses and power side-channel information are relevant to the internal parameters of arbiter-based PUFs. Providing all available data to FNN improves the modeling accuracy of the internal parameters, thus improving the prediction accuracy of the response and the side-channel information. In addition, the experimental results are consistent with previous studies [16], which demonstrate that the traditional FNN cannot model 16-XOR APUF, (8,8)-iPUF, and (2,16)-iPUF.

TABLE VIII
Modeling Accuracy of Classification Based on Different Features

| Target | Model | Attack | Output | Accuracy |
|---|---|---|---|---|
| 16-XOR APUF | Combined model | A | Combined | 95.03% |
| | | | Side-channel[c] | 95.03% |
| | | | Response[c] | 95.03% |
| | Side-channel model | B | Side-channel | 94.95% |
| | Mathematical model | C | Response | 49.98% |
| (128,5)-MPUF | Combined model | A | Combined | 86.27% |
| | | | Side-channel[c] | 86.87% |
| | | | Response[c] | 98.63% |
| | Side-channel model | B | Side-channel | 82.73% |
| | Mathematical model | C | Response | 96.79% |
| (8,8)-iPUF | Combined model | A | Combined | 94.00% |
| | | | Side-channel[c] | 94.04% |
| | | | Response[c] | 95.38% |
| | Side-channel model | B | Side-channel | 87.37% |
| | Mathematical model | C | Response | 49.97% |
| (2,16)-iPUF | Combined model | A | Combined | 93.91% |
| | | | Side-channel[c] | 93.91% |
| | | | Response[c] | 94.52% |
| | Side-channel model | B | Side-channel | 86.35% |
| | Mathematical model | C | Response | 49.90% |

Superscript c indicates that these values was extracted from the labels predicted by the combined model.

The experimental results demonstrate that multiclass classification based on synthetic features mainly contributes to the superior performance of the proposed method. It also implies that the constructed CSPs provide more predictive information than CRPs and CPPs individually.

## VI. DISCUSSION

Here, we analyze the extensibility of the proposed method and discuss feature selection principles based on information theory. In addition, we compare the proposed hybrid attack to recent power side-channel attacks to evaluate its effectiveness. Finally, we discuss potential future work.

### *A. Analysis of Feature Combination*

The proposed method for feature combination can be extended to other side channels (besides the power side-channel) through a few adjustments. For example, a reliability-based hybrid attack through multiclass classification may outperform recent reliability-based attacks. However, the reliability data

usually take a wide range of values. The combination of CRPs and reliability data may suffer from the curse of dimensionality (CoD).

Generally, it is preferable to have more data when exploring scientific questions; however, the proliferation of data introduced by feature combinations may result in CoD [38], which hinders ML algorithms from detecting genuine relationships. In practice, as a direct embodiment of CoD, an increase in sparsity makes it much more challenging to collect the required data. In addition, too many features may cause overfitting because the flexibility of prediction equations is determined in part by the number of involved variables. Because feature combination suffers from the threatening of CoD, the selection of features for combinations may influence the performance of the proposed hybrid attack.

According to the information theory, whether features are recommended to cross is primarily determined by mutual information. Mutual information is a quantity that measures the relationship between two features. Notably, the prerequisite of correlation between two features is essential for subsequent modeling. There are several recommendations based on mutual information for feature combination in PUF modeling.

1) The mutual information between the response and the optional feature should be greater than zero. The mutual information between two features is zero if and only if the two features are statistically independent. A feature independent of the response cannot provide any information for the response.

2) Even if two features are independent but are both relevant to response, their combination can be beneficial to modeling. A typical example is the XOR function with two inputs. However, it may be challenging to find two independent features in PUF modeling scenarios.

3) When the mutual information between two features equals the entropy of a single feature, i.e., one feature can be deducted directly from the other, their combination cannot eliminate the uncertainty of the single feature. For XOR APUF, the mutual information between the response and power side-channel information is equal to the entropy of the response, and a combination of features does not improve modeling performance compared to the side-channel feature used individually.

### *B. Comparison of Power Side-channel Attacks*

There are three approaches to make use of power side-channel information in PUF modeling.

The first approach, i.e., particular CPPs, can be directly used for modeling by applying cryptanalysis [35]. The term "particular" means that $p$ takes the value of zero or $l$, implying that all APUFs in the attack target output logical zero or logical one. From a particular CPP, one equation for each APUF can be derived. When a sufficient number of particular CPPs are collected, models for each APUF can be built individually by solving these equations. Here, assume that the parameters of APUFs are independent identically distributed, and the probability of a particular CPP appearing in all CPPs is $1/2^l$. As $n$ individual equations are required for simultaneous equations, $n \times 2^l$ CPPs must be collected to search for a particular CPP on average.

Second, the power side-channel model was proposed as an alternative to the mathematical model. ML algorithms, e.g., LR, SVM, and ES, use CPPs for training rather than CRPs. A power side-channel model, which also contains the predicted internal parameters for each APUF, can be obtained when a sufficient number of CPPs is collected. The number of CPPs required for training is much less than that for cryptanalysis.

Both cryptanalysis and ML learning can learn the internal parameters of all APUFs; however, neither can predict the positions of these APUFs. Here, there are $l!$ possibilities of position permutations, and all cases should be tested and compared to determine the optimal one. However, the time and computing resources required are unaffordable for these two methods to attack arbiter-based PUFs with complex structures. Therefore, they can only be applied to PUFs whose responses are not influenced by the positions of APUFs. Representative examples include the XOR APUF and LSPUF. In addition, reliability-based hybrid attacks also cannot predict the positions of the APUFs as instances.

The last approach is the proposed FNN power analysis attack. With the utilization of all accessible data, the proposed method can be applied to all arbiter-based PUFs, even without knowing the mathematical models. Responses of the arbiter-based PUFs are relevant to both the internal parameters and the positions of APUFs. Power analysis provides extra information about the internal parameters of APUFs. Provided with CSPs, the combined model built by FNN learn the internal parameters as well as the positions of APUFs; thus, it predicts response and power side-channel information.

The proposed attack possesses the merits of availability and extensibility. Several anti-SCA techniques are proposed to avoid the power leakage of APUFs [39]. Adding an extra arbiter to produce a complementary response can mitigate power leakage. Due to the difference between the capacitive load, the dual-rail logic only increases the required side-channel measurements but cannot entirely avoid the SCA attack. Although the power consumption of arbiter-based PUFs implemented on an FPGA is too low to make power analysis possible, measuring the current repeatedly on the same challenge is an effective way to overcome this issue. However, dual-rail solutions remain vulnerable to electromagnetic analysis [40]. The electromagnetic leakage model of PUFs is similar to the power leakage model; therefore, the proposed method can be applied to hybrid attacks based on electromagnetic analysis through a few adjustments. The proposed method maintains effectiveness by changing some measurement instruments when considering strong PUFs with dual-rail logic protection.

The drawbacks of SCA on PUFs may be the extra effort required to collect the side-channel information. Some PUF-based protocols, such as the lockdown protocol [41] and the deception protocol [42], inherently prevent attackers from collecting sufficient data for modeling. A restriction on the number of authentication events limits the number of collected CSPs, and the proposed modeling attacks cannot be applied directly to these protocols. However, combined with attacks

reported in the literature [43], the proposed method may provide a potential route to attack protocols based on arbiter-based PUFs with complex structures.

### C. Future Work

We found that the evaluation of the security of strong PUFs is more complex than expected, e.g., by applying the FNN SCA hybrid attack, we require more CRPs and time to model (128,5)-MPUF than to model (4,4)-iPUF. Considering the traditional FNN or LR, (4,4)-iPUF is more challenging to crack than (128,5)-MPUF. The difference in modeling difficulty is derived from which model the attackers choose to build. There are three standard models: the mathematical model, which explains the relationship between the challenge and response; the power side-channel model, which reveals the variation of power leakage during the PUF operation; the reliability model, which focuses on unstable CRPs. The latter two models belong to side-channel models, which play an essential part in modeling PUFs with hidden intermediate responses. For ($n$,$k$)-MPUF, when parameter $k$ increases, the complexity of the power side-channel model increases exponentially, whereas the mathematical model continues to grow linearly with $k$. It is difficult to say which type of PUF is more secure. All kinds of attacks are recommended to be taken into account when evaluating the security of PUFs.

In our future studies, we shall focus on adopting the proposed technique to construct multi-side-channel hybrid attacks. The conceived attacks would use training data containing more than two features, e.g., response, power side-channel information, and reliability. This work is expected to enhance the understanding that providing more information to ML algorithms typically results in better prediction accuracies. Thus, multi-side-channel hybrid attacks based on multiclass classification deserve further study.

## VII. Conclusion

In this paper, we have proposed a method for feature combination and FNN SCA hybrid attacks to model strong PUFs with complex structures. Multiclass classification is effective in terms of modeling strong PUFs by utilizing side-channel information, which, to the best of our knowledge, has not been discussed previously in the literature. Combining CRPs with power side-channel information makes it possible to model PUFs via multiclass classification, and this can be easily extended to other side-channel attacks. The hybrid attack described in this paper is computationally feasible and reasonably time-efficient for most arbiter-based PUF designs, and it is particularly suitable for strong PUFs with complex structures. Therefore, new mechanisms to resist the FNN SCA hybrid attack examined in this work should be implemented in future strong PUF designs.

## Acknowledgment

The authors would like to thank Prof. Wenlin Zhang, Prof. Rui Chang, Dr. Yang Wang, Dr. Lin Li, Dr. Xiaolin Zhang, and Dr. Pei Cao for their constructive and helpful comments.

## References

[1] Y. Gao, S. F. Al-Sarawi, and D. Abbott, "Physical unclonable functions," *Nature Electronics*, vol. 3, pp. 81–91, 2020.

[2] C. Herder, M. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, pp. 1126–1141, 2014.

[3] C. 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, pp. 32–62, 2017.

[4] T. McGrath, I. E. Bagci, Z. Wang, U. Roedig, and R. Young, "A PUF taxonomy," *Applied physics reviews*, vol. 6, p. 011303, 2019.

[5] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 2007 44th ACM/IEEE Design Automation Conference*, pp. 9–14, 2007.

[6] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," in *ICCAD 2008*, pp. 670-673, 2008.

[7] J. Lee, D. Lim, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," *2004 Symposium on VLSI Circuits. Digest of Technical Papers*, pp. 176–179, 2004.

[8] Y. Gao, G. Li, H. Ma, S. Al-Sarawi, O. Kavehei, D. Abbott, and D. Ranasinghe, "Obfuscated challenge-response: A secure lightweight authentication mechanism for PUF-based pervasive devices," *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 1–6, 2016.

[9] Y. Gao, H. Ma, S. Al-Sarawi, D. Abbott, and D. Ranasinghe, "PUF-FSM: A controlled strong PUF," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, pp. 1104–1108, 2018.

[10] D. P. Sahoo, D. Mukhopadhyay, R. Chakraborty, and P. H. Nguyen, "A multiplexer-based arbiter PUF composition with enhanced reliability and security," *IEEE Transactions on Computers*, vol. 67, pp. 403–417, 2018.

[11] 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 Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 4, pp. 243–290, 2019.

[12] U. Rührmair, J. Sölter, 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, pp. 1876–1891, 2013.

[13] J. Huang, M. Zhu, B. Liu, and W. Ge, "Deep learning modeling attack analysis for multiple FPGA-based APUF protection structures," in *Proc. 2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pp. 1–3, 2018.

[14] 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. 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 204–209, 2019.

[15] J. Shi, Y. Lu, and J. Zhang, "Approximation attacks on strong PUFs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, pp. 2138–2151, 2020.

[16] P. Santikellur, A. Bhattacharyay, and R. Chakraborty, "Deep learning based model building attacks on arbiter PUF compositions," *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 566, 2019.

[17] R. Elnaggar and K. Chakrabarty, "Machine learning for hardware security: Opportunities and risks," *Journal of Electronic Testing*, vol. 34, pp. 183–201, 2018.

[18] Y. Li, J. Shen, W. Liu, and W. Zou, "A survey on side-channel attacks of strong PUF," in *ICAIS2020*, pp. 74-85, 2020.

[19] G. Becker and R. Kumar, "Active and passive side-channel attacks on delay based PUF designs," *IACR Cryptol. ePrint Arch.*, vol. 2014, p. 287, 2014.

[20] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, and W. Burleson, "Efficient power and timing side channels for physical unclonable functions," in *CHES2014*, pp. 476–492, 2014.

[21] G. Becker, "The gap between promise and reality: On the insecurity of XOR arbiter PUFs," in *CHES2015*, pp. 535–555, 2015.

[22] J. Tobisch, A. Aghaie, and G. Becker, "Combining optimization objectives: New modeling attacks on strong PUFs," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 357–389, 02 2021.

[23] D. Lim, J. Lee, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, pp. 1200–1205, 2005.

[24] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," *in Proc. the 17th ACM Conference on Computer and Communications Security*, pp. 237–249, 2010.
[25] D. P. Sahoo, P. H. Nguyen, D. Mukhopadhyay, and R. Chakraborty, "A case of lightweight PUF constructions: Cryptanalysis and machine learning attacks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, pp. 1334–1343, 2015.
[26] 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 Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, pp. 97–120, 2019.
[27] J. Ye, Q. Guo, Y. Hu, H. Li, and X. Li, "Modeling attacks on strong physical unclonable functions strengthened by random number and weak PUF," in *Proc. 2018 IEEE 36th VLSI Test Symposium (VTS)*, pp. 1–6, 2018.
[28] G. Becker, "On the pitfalls of using arbiter-PUFs as building blocks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, pp. 1295–1307, 2015.
[29] W. Ge, J. Huang, B. Liu, M. Zhu, and Y. Cao, "A deep learning modeling attack method for MISR-APUF protection structures," in *Proc. 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 398–402, 2018.
[30] M. Khalafalla, M. Elmohr, and C. Gebotys, "Going deep: Using deep learning techniques with simplified mathematical models against XOR BR and TBR PUFs (attacks and countermeasures)," in *Proc. 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 80–90, 2020.
[31] J. Delvaux and I. Verbauwhede, "Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise," in *Proc. 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 137– 142, 2013.
[32] S. Tajik, E. Dietz, S. Frohmann, H. Dittrich, D. Nedospasov, C. Helfmeier, J. Seifert, C. Boit, and H. Hübers, "Photonic side-channel analysis of arbiter PUFs," *Journal of Cryptology*, vol. 30, pp. 550–571, 2016.
[33] J. Delvaux and I. Verbauwhede, "Fault injection modeling attacks on 65 nm arbiter and RO sum PUFs via environmental changes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, pp. 1701–1713, 2014.
[34] Y. Liu, Y. Xie, C. Bao, and A. Srivastava, "A combined optimization-theoretic and side-channel approach for attacking strong physical unclonable functions," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, pp. 73–81, 2018.
[35] A. Mahmoud, U. Rührmair, M. Majzoobi, and F. Koushanfar, "Combined modeling and side channel attacks on strong PUFs," *IACR Cryptol. ePrint Arch.*, vol. 2013, p. 632, 2013.
[36] Y. Hori, H. Kang, T. Katashita, A. Satoh, S. Kawamura, and K. Kobara, "Evaluation of physical unclonable functions for 28-nm process field programmable gate arrays," J. Inf. Process., vol. 22, pp. 344–356, 2014.
[37] J. Tobisch and G. Becker, "On the scaling of machine learning attacks on PUFs with application to noise bifurcation," in *RFIDsec2015*, pp. 17–31, 2015.
[38] N. S. Altman and M. Krzywinski, "The curse(s) of dimensionality," *Nature Methods*, vol. 15, pp. 399–400, 2018.
[39] Y. Nozaki and M. Yoshikawa, "Countermeasure of lightweight physical unclonable function against side-channel attack," in *Proc. 2019 Cybersecurity and Cyberforensics Conference (CCC)*, pp. 30–34, 2019.
[40] A. Aghaie and A. Moradi, "TI-PUF: Toward side-channel resistant physical unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3470–3481, 2020.
[41] M. Yu, M. Hiller, J. Delvaux, R. Sowell, S. Devadas, and I. Verbauwhede, "A lockdown technique to prevent machine learning on PUFs for lightweight authentication," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, pp. 146–159, 2016.
[42] C. Gu, C.-H. Chang, W. Liu, S. Yu, Y. Wang, and M. OrNeill, "A modeling attack resistant deception technique for securing lightweight-PUF based authentication," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, pp. 1183–1196, 2021.
[43] J. Delvaux, "Machine-learning attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUF-FSMs," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 2043–2058, 2019.

**Wei Liu** received B.S. and M.S. degrees from Zhengzhou Institute of Science and Technology, Zhengzhou, China, in 2004 and 2007, respectively. He is currently pursuing a Ph.D. degree at the State Key Laboratory of Mathematic Engineering and Advanced Computing, Zhengzhou, China.

Since 2014, he has been a Lecturer at the State Key Laboratory of Mathematic Engineering and Advanced Computing. His research interests include physical unclonable functions, hardware security, and artificial intelligence security.

**Ruimin Wang** received a B.S. degree in Computer Science from Henan Normal University, Zhengzhou, China, in 2005 and an M.S. degree in Computer Science from Northwestern Polytechnical University, Xi'an, China, in 2008.

She is currently an Associate Professor at the State Key Laboratory of Mathematical Engineering and Advanced Computing. Her research interests include embedded system security, IoT security, and device identification.

**Xuyan Qi** received a B.S. degree in communication engineering from Lanzhou Jiaotong University, Lanzhou, China, in 2007 and an M.S. degree in integrated circuit engineering from Wuhan University, Wuhan, China, in 2012. She is currently pursuing a Ph.D. degree in computer science and technology at the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China.

She is currently an Associate Professor at the State Key Laboratory of Mathematical Engineering and Advanced Computing. Her research interests include malware detection and cybersecurity.

**Liehui Jiang** received an M.S. degree in computer science and technology from Zhengzhou Institute of Science and Technology in 1994 and a Ph.D. degree from Zhengzhou Institute of Science and Technology, Zhengzhou, China, in 2007.

He is currently a Professor and Ph.D. supervisor at the State Key Laboratory of Mathematic Engineering and Advanced Computing, Zhengzhou, China. His research interests include computer architecture, reverse engineering, and cybersecurity.

**Jing Jing** received a B.S. degree in computer science and technology from the Zhengzhou Institute of Science and Technology, Zhengzhou, China, in 2002, received an M.S. degree in computer applications and a Ph.D. degree in computer software and theory from the Zhengzhou Institute of Science and Technology, Zhengzhou, China in 2007 and 2014, respectively.

From 2002 to 2004, she was a Lecturer at the Institute of Foreign Languages, Luoyang, China. She is currently an Associate Professor at the State Key Laboratory of Mathematical Engineering and Advanced Computing. Her research interests include embedded systems, IoT security, and cyber-physical systems.