Breaking XOR Arbiter PUFs with Chosen Challenge Attack

Niloufar Sayadi*†, Phuong Ha Nguyen‡, Marten van Dijk*†§, Chenglu Jin*
*CWI Amsterdam, The Netherlands
†Vrije Universiteit Amsterdam, The Netherlands
‡eBay, CA, United States
§University of Connecticut, CT, United States

***Abstract*—The XOR Arbiter PUF was introduced as a strong PUF in 2007 and was broken in 2015 by a Machine Learning (ML) attack, which allows the underlying Arbiter PUFs to be modeled individually by exploiting reliability information of the measured responses. To mitigate the reliability-based attacks, state-of-the-art understanding shows that the reliability of individual Arbiter PUFs and the overall XOR Arbiter PUF can be boosted to an arbitrarily high level, thus rendering all known reliability-based ML attacks infeasible; alternatively, an access control interface around the XOR Arbiter PUF can prevent the same challenge-response pairs from being accessed repeatedly, thus eliminating the leakage of reliability information.**

We show that, for the first time, a perfectly reliable XOR Arbiter PUF can be successfully attacked in a divide-and-conquer manner, meaning each underlying Arbiter PUF in an XOR Arbiter PUF can be attacked individually. This allows us to attack large XOR Arbiter PUFs efficiently, even without reliability information or any side-channel information. Our key insight is that, instead of reliability information, the responses of highly correlated challenges also reveal how close the responses are to the response decision boundary. This leads to a *chosen challenge attack* on XOR Arbiter PUFs by carefully choosing correlated challenges to measure and aggregate the collected information. We validate our attack by using PUF simulation, as well as an XOR Arbiter PUF implemented on FPGA. We also demonstrate that our chosen challenge methodology is compatible with the state-of-the-art combined gradient-based multi-objective optimization attack. Finally, we discuss an effective countermeasure that can prevent our attack but with a relatively large area overhead compared to the PUF itself.

***Index Terms*—XOR Arbiter PUFs, PUF Modeling Attacks, Chosen Challenge Attacks, Reliability-based Attacks**

I. Introduction

The unique process manufacturing variation on every chip can be leveraged to create a Physical Unclonable Function (PUF) as a fingerprint for the device [1]. A vector of bits is applied to the PUF as the PUF challenge, and a unique response is generated by the PUF as output. PUFs are useful for device authentication and secret key management [2]–[4].

One classical design of silicon PUFs is Arbiter PUF (APUF), which measures the delay differences of two competing paths determined by a challenge and produces a one-bit response depending on which path is faster [1]. Ideally, the challenge-response pairs (CRPs) of an APUF are only determined by process variation and thus unpredictable to an attacker. However, the behavior of any APUF can be easily modeled by machine learning (ML) attacks after collecting many CRPs of the APUF [5]. A precise mathematical model of a PUF allows the attacker to violate the security properties of the PUF and thus successfully impersonate a legitimate user in authentication or retrieve secret keys managed by the PUF.

XOR Arbiter PUF (XOR APUF) was introduced as a security enhancement of the APUF design. It XORs multiple parallel APUF response bits together to form a response bit of the XOR APUF. In addition to being a strong component of many state-of-the-art strong PUFs, e.g. iPUF [6], XOR APUF is one of the few PUFs with practical applications such as PUF-based RFID tags [7]. XOR APUF has been considered a promising candidate for secure strong PUF design since their introduction in 2007 [3] until Becker presented a reliability-based ML modeling attack that can model the individual underlying APUFs in 2015 [7]. The reliability-based attack relies on the fact that the reliability of a CRP under the measurement noise reveals whether the response is close to the response decision boundary where the value of the response bit will be flipped. This extra information allows attackers to implement a divide-and-conquer strategy to attack individual APUFs in an XOR APUF rather than the whole XOR APUF [7].

One limitation of the reliability-based attack is that it has to access the reliability information of individual CRPs by repeated measurement. Following this line of thinking, countermeasures have been proposed to thwart or prevent the leakage of reliability information by designing super reliable XOR APUFs [8] or blocking repeated measurements using an access control interface [9].

In this paper, we introduce a novel *divide-and-conquer* strategy to model the XOR APUF without the reliability information of CRPs or any side-channel information of the PUF. Therefore, the XOR APUF design is completely broken in both reliable and unreliable cases. Note that the proposed attack requires the attackers to choose the challenges applied to the PUF; however, it does not mean that we assume a stronger adversarial model than the reliability-based attack because the reliability-based attack is also one kind of chosen challenge attack, as it chooses to measure the same CRPs repeatedly.

In our attack, we choose a few challenges around a randomly selected challenge with only a one-bit difference in their Ψ vectors (a Ψ vector is a transformed challenge based on Eq. 2), and then we check how many surrounding challenges

will lead to a flipped response bit from the response of the centroid challenge. The flipping probability reveals how close the centroid challenge is to the response decision boundary, where the response bit flips. Relying on this fact, we propose a chosen challenge attack that can model individual APUFs in an XOR APUF and thus break the security of XOR APUF completely, even in the case when state-of-the-art reliability-based attacks fail.

Our experimental results in Sec. IV-B show that the proposed attack can model individual APUFs in a perfectly reliable XOR APUF with a high prediction accuracy, while the reliability-based attacks fail to work [7]. As shown in our experiments in Sec. IV-C, our attack can also model the XOR APUF when the PUF is realistically unreliable. Finally, we also validate our attack by attacking XOR APUFs implemented on a real FPGA.

A. Contributions

We make the following contributions in this paper:

- We introduce a novel chosen challenge attack on unreliable and, more importantly, perfectly reliable XOR-Arbiter PUFs.
- We evaluate the effectiveness of the proposed attack using a PUF simulation under various conditions, including various noise levels, sampling rates, and the number of flips.
- We evaluate the compatibility of the proposed chosen-challenge methodology with the latest combined Logistic Regression (LR) and reliability-based attack proposed in [10] and provide a combined LR non-flipping-probability-based attack, which is applicable for both reliable and unreliable PUF.
- We demonstrate the practicality of our attack by attacking a real XOR APUF implemented on an FPGA.
- We discuss an effective countermeasure against our attack.
- As a proof of concept, the source code of our Chosen Challenge attack is publicly available on GitHub.[1]

B. Organization

We present the necessary background of APUF, XOR APUF, and the reliability-based attacks in Sec. II. The proposed chosen challenge attack is discussed in Sec. III. Sec. IV presents the experimental results and a fair comparison with the state-of-the-art reliability-based attacks. An effective countermeasure is discussed in Sec. V, followed by some other related work in Sec. VI. The paper concludes in Sec. VII.

II. Background

A. Arbiter PUFs and XOR Arbiter PUFs

An Arbiter PUF (APUF) is a strong PUF that consists of n consecutive delay stages that lead to an arbiter. The stages are identical 2-to-1 multiplexers that lead the top and bottom signals based on the challenge bits that are applied to their $select$ inputs. In the last stage, the top and bottom signals have an accumulated delay difference due to the differences in the delay of every stage introduced by process variations. Finally, the arbiter outputs a '1' or '0' bit depending on which signal arrives at the arbiter earlier.

The behavior of an APUF can be captured by a *Linear Additive Delay Model* [2]

$$\Delta = w[0]\Psi[0] + ... + w[n]\Psi[n] = \langle W,\ \Psi \rangle, \tag{1}$$

where W is the weight vector, Ψ is the parity (or feature) vector and n is the number of challenge bits or the number of delay stages. The weight vector W defines the character of the APUF, and it is determined only by process variations. For $\Delta \geq 0$, the response $r = 1$, otherwise $r = 0$. The parity vector Ψ is solely based on the challenge vector c in the following way:

$$\Psi[n] = 1 \ and \ \Psi[i] = \prod_{j=i}^{n-1}(1 - 2c[j]), i = 0, ..., n-1. \tag{2}$$

A k-XOR APUF is constructed by k APUFs that are fed by the same challenge, and their response bits are XORed together. Unlike APUFs, XOR APUFs have a non-linear model due to the added XOR. Thus, it was generally believed that XOR APUFs are much harder to model than APUFs, and the difficulty grew exponentially in k before the reliability-based attack was known [5].

B. Reliability-Based Attacks on XOR APUFs

Unlike the classical machine learning modeling attacks on PUFs that use response bits directly for training, in a reliability-based modeling attack, the reliability information derived from repeated measurement of CRPs is used for modeling. The response to each challenge is measured multiple times to compute the reliability of the CRP. This reliability gives information about the delay difference Δ of an APUF component in an XOR APUF under the evaluated challenge. This is because when $|\Delta|$ is smaller than a threshold value ϵ, the CRP is unreliable, and the response can easily be flipped between 0 and 1; otherwise, the CRP is reliable, and the response is consistent. Also, in an XOR APUF, if one of the underlying APUFs is unreliable, then the overall XOR APUF is also unreliable. This is why the reliability information reveals the delay information of the individual APUF and allows a divide-and-conquer strategy to model individual APUFs in an XOR APUF. In [7], Becker used a popular evolution strategy-based optimization algorithm, Covariance Matrix Adaptation Evolution Strategy (CMA-ES), to learn the weight vector of the APUF along with the threshold value ϵ. For this purpose, a set of challenges C_i is randomly generated, and the reliability R_i is measured for every challenge using Eq. 3, where M is the number of measurements per CRP.

$$R_i = |\frac{M}{2} - \sum_{i=1}^{M} r_i| \tag{3}$$

Then, Becker used CMA-ES to optimize the PUF model by maximizing the Pearson correlation coefficient of the measured reliability and the predicted reliability based on the PUF

[1] https://github.com/niloufarsyd/Chosen_Challenge_Attack

models in the current iteration of the CMA-ES algorithm. The algorithm will converge to the model of an underlying APUF in the attacked XOR APUF. By executing the CMA-ES algorithm multiple times, all APUF models in the XOR APUF will be modeled eventually.

An enhanced reliability-based attack on XOR APUF was presented by Nguyen *et al.*, which utilizes a more accurate reliability model to improve the prediction accuracy or attack efficiency [6]. Tobisch *et al.* demonstrated that it is also possible to use a gradient-based optimization to launch reliability-based attacks and to combine the objective of the reliability-based attack and that of the classical CRP-based attack [10].

III. Proposed Attack on XOR APUFs

A. Motivation

To the best of our knowledge, the reliability-based attack is still the most efficient attack on XOR APUFs because the attackers can exploit more fine-grained information than just the CRPs used in classical ML attacks. However, if no reliability information is available to the attacker, then none of the reliability-based attacks [6], [7], [10] would work, and the attackers will have to use the best-known classical attacks [5], [11]–[15].

As shown in [8] and [16], by implementing a majority voting at the end of every APUF in an XOR APUF, one can boost the reliability of the APUFs and the overall XOR APUF arbitrarily. Although this defense may incur a performance overhead, it is very effective in mitigating the reliability-based attack because the attacker cannot find a noisy CRP to derive its reliability. Furthermore, the challenge obfuscation proposed in [16] is based on DES permutation, which is reversible and known to adversaries. Another potential countermeasure that could paralyze the reliability-based attacks is the (logically) erasable PUF interface [9], [17]. Using such an interface, the access to each individual CRP can be controlled, and then, it is impossible to remeasure the response of an erased challenge anymore, e.g., the interface can automatically erase the measured challenge after one or a certain number of measurements. Thus, it prevents the attacker from deriving the reliability information even if the underlying PUF is still noisy. Our proposed attack can defeat both of the countermeasures since we do not rely on the reliability information anymore.

B. Adversarial Model

We assume the same adversarial model as the reliability-based attack [7]. The attackers can apply any challenges they want to the PUF and only get the responses of the queried challenges back, i.e., no other side channel information, like power or timing information [18]. The only difference is that we *choose* to measure correlated CRPs, but the reliability-based attacks *choose* to measure the same CRPs repeatedly.

C. Chosen Challenge Attack

Key Idea. In this strategy, we focus on manipulating Ψ and investigate how it affects the delay model (Eq. 1). Suppose that there is a flip in the Ψ vector of the delay model; then, this flip will lead to a different coefficient for one of the weight vector elements. Note that a challenge c consists of only 0 or 1, but a Ψ vector only contains 1 or -1, according to Eq. 2. Thus, after one flip in Ψ (from 1 to -1 or from -1 to 1), the delay difference Δ would be changed from Δ_1 to Δ_2, as shown in Eq. 4. Whether this change leads to a flipped response bit reveals whether Δ flips its sign, i.e., whether $|\Delta| < 2|w[1]|$, in the case of Eq. 4. This is the key enabler of our attack on APUFs.

$$\begin{cases} \Delta_1 = w[0] - w[1] - w[2] + ... + w[n] \\ \Delta_2 = w[0] + w[1] - w[2] + ... + w[n] \end{cases} \tag{4}$$

This relation also allows us to attack individual APUFs in an XOR APUF. According to the analysis provided in Appendix A, if there is a flip in Ψ, it is unlikely that the response of an APUF will be flipped. But if the response of one APUF is flipped, this flip will be propagated to the output of the XOR and be observable by the attacker, assuming this flip in Ψ does not flip the responses of the other APUFs. Of course, multiple APUF responses can be flipped occasionally, but we will use the Pearson correlation coefficient as a robust indicator to deal with the "noise" introduced by multiple APUF response flips.

Detailed Steps. Algorithm 1 shows the pseudo-code of the proposed chosen challenge attack, and it models the PUF using the flipping probability information. To compute the flipping probability, we first choose a challenge c randomly in the challenge space and then convert it to the corresponding Ψ

Algorithm 1 Chosen Challenge Attack

1: **procedure** Chosen Challenge Attack
2: **for** $t \leftarrow 1$ to N **do** ▷ Training Data Collection
3: randomly select a challenge c
4: transform c into Ψ based on Eq. 2
5: $r \leftarrow PUF(c)$
6: **for** $i \leftarrow 1$ to m **do**
7: randomly generate Ψ_i s.t. $d_H(\Psi, \Psi_i) = 1$
8: transform Ψ_i into c_i by inverse Eq. 2
9: $r_i \leftarrow PUF(c_i)$
10: **end for**
11: $F_t = 1 - \frac{\sum_{i=o}^{m-1} |r_i - r|}{m}$
12: $\mathcal{Z} = \{(\Psi_i, F_i)\}_{i=1}^{N}$
13: **end for**
14: generate K random models: $\{W_1,...,W_j,...,W_K\}$
15: **for** $j \leftarrow 1$ to K **do** ▷ Optimization
16: **for** $h \leftarrow 1$ to N **do**
17: $F'_{j,h} = |\Delta| = |\langle W_j,\ \Psi_h \rangle|$
18: **end for**
19: $F'_j = \{F'_{j,h}\}_{h=1}^{N}$
20: $\rho_j = Correlation(F, F'_j)$
21: **end for**
22: CMA-ES uses L out of the K models of W with the highest ρ to generates K new models.
23: repeat lines (15-22) for T iteration to maximize ρ and output the optimized model W.
24: **end procedure**

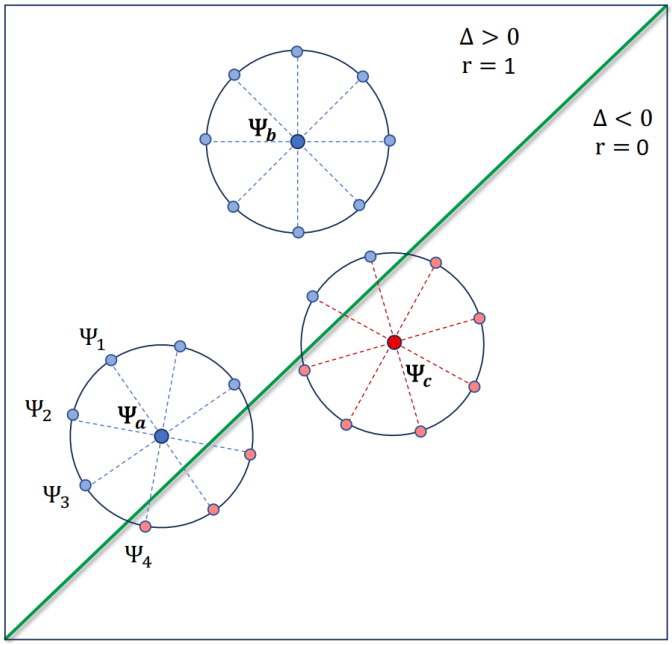

Fig. 1. Geometric illustration of the chosen Ψ vectors and their relation with the linear response decision boundary in the Ψ space. Note that this diagram is for illustration purposes only, as the Ψ space is never a 2-D space. The decision boundary is always linear due to the linear delay model in Eq. 1.

according to Eq. 2. Then we randomly sample m Ψ_i vectors that have a Hamming distance of 1 from Ψ:

$$d_H(\Psi, \Psi_i) = 1. \tag{5}$$

The Hamming distance of 1 means only one bit is different between Ψ and Ψ_i. Practically speaking, if one bit $\Psi[i]$ is flipped, then two consecutive bits ($c[i-1]$ and $c[i]$) should be flipped in the corresponding challenge c, except for $i = 0$, then only $c[0]$ needs to be flipped.

Fig. 1 illustrates the main idea of our attack. For any randomly chosen Ψ (e.g., Ψ_a, Ψ_b, or Ψ_c), the m sampled points with $d_H = 1$ are on the circular radius to the centroid of Ψ. If the Ψ is close to the response decision boundary, it is more likely that the responses of the sampled neighboring Ψ_i vectors are different from the response of Ψ.

Note that in Algorithm 1 and in our source code, the non-flipping probability value is used. The non-flipping probability is calculated as in Eq. 6.

$$F = 1 - \frac{\sum_{i=1}^{m} |r_i - r|}{m} \tag{6}$$

The responses r_i and r in the Eq. 6 are the PUF responses of the sampled neighboring Ψ_i and the centroid Ψ, respectively, and m is the number of sampled ψ_i around a centroid Ψ.

Thus, after sampling many random challenges and their surrounding challenges, we get a dataset of (Ψ, F) pairs as Eq. 7.

$$\mathcal{Z} = \{(\Psi_1, F_1), ...(\Psi_i, F_i), ..., (\Psi_N, F_N)\}, \tag{7}$$

where N is the size of the training dataset in our attack.

Then, the CMA-ES algorithm is used to find the optimal W model for individual APUF in an XOR APUF. To start the procedure, K random models are generated. For each model W_j, the Δ is calculated for each Ψ vector included in $\mathcal{Z}$, and the predicted non-flipping probability $F^{'}$ is computed according to Eq. 8. The expected linear relationship of F and F' will be discussed in Sec. III-D.

$$F^{'} = |\Delta| = |\langle W, \Psi \rangle| \tag{8}$$

The objective function of CMA-ES is the Pearson correlation coefficient ρ_j between the measured non-flipping probability $(F_1,...,F_N)$ and the predicted non-flipping probability $(F_1^{'},...,F_N^{'})$. Inside the CMA-ES algorithm, an L number of w APUF models with the highest ρ_j value are kept in each iteration to generate a new generation of K models. After T iterations, the CMA-ES outputs the model with the highest Pearson correlation coefficient ρ as the best model. When attacking an XOR APUF, the CMA-ES algorithm would converge to one of the individual APUFs in the XOR APUF. We can repeat the whole process, and eventually, all the APUFs of an XOR APUF will be modeled after a sufficient number of CMA-ES runs; therefore, the whole model of XOR APUF is revealed.

Time Complexity. The modeling time complexity of the proposed Chosen Challenge Attack is shown in Eq. 9. The timing complexity is determined by the number of centroid challenges or training challenge sets (N), the number of challenges sampled around each centroid challenge (m), the number of CMA-ES iterations (T), the number of models populated in each CMA-ES generation (K), and the size of Ψ (n). The term Nm in the time complexity is due to the training data collection phase of the attack, and the term $TKNn$ is due to the modeling optimization phase, as we can see in Algorithm 1.

$$\mathcal{O}(Nm + TKNn) \tag{9}$$

D. A Unified Theoretical Foundation of Our Attack and the Reliability-based Attacks

As shown in [6], [19], the non-flipping probability F (or reliability) of a CRP in an APUF, under some perturbations, has a relationship with the delay difference Δ of the CRP and the standard deviation σ of the perturbations as follows.

$$\Delta/\sigma = \sum_{i=0}^{n} (w[i]/\sigma)\Psi[i] = -\Phi^{-1}(F) \tag{10}$$

In a reliability-based attack, the perturbations come from environmental noise, and thus σ is the standard deviation of the environmental noise with a normal distribution $\mathcal{N}(0, \sigma^2)$. Similarly, in our proposed chosen challenge attack, the perturbations come from the flipped Ψ bit, so σ is the standard deviation of the perturbation ($2W[i]$) caused by one flip in Ψ. Note that each weight vector component $W[i]$ in an APUF is believed to follow a normal distribution $\mathcal{N}(0, \sigma^2/4)$ if we define σ to be the standard deviation of the random perturbation in this attack. This normal distribution assumption of the PUF weights $W[i]$ is widely used in APUF simulations and is validated in real APUF implementations in the past research [5]–[7], [11], [12].

Eq. 10 can be further approximated as Eq. 11 if $F \in [0.1, 0.9]$.

$$|\Delta/\sigma| \approx F \tag{11}$$

Due to the linear relationship between the measured non-flipping probability F and the delay model Δ, which is also the predicted non-flipping probability F', we can select the best-fitting models among random models and then optimize them in every iteration of the CMA-ES algorithm to generate an accurate model of an underlying APUF.

E. Comparison with Reliability-based Attacks

The proposed attack shares many similarities with the reliability-based attack in [6], which is an enhanced version of the attack in [7]. Both our attack and the reliability-based attacks need to choose the CRPs to be measured, and they all exploit the fact that the non-flipping probability (or the reliability) has a linear relationship with the delay difference Δ. Most importantly, both our attack and the reliability-based attack can learn individual APUFs in an XOR APUF in a divide-and-conquer manner.

Actually, the proposed chosen challenge attack can be viewed as a generalization of the reliability-based attack. As we will show in Sec. IV-D, our chosen attack does not have to restrict the hamming distance to be 1; the known reliability-based attacks are a special case of our chosen challenge attack when the hamming distance is 0, and the perturbation is caused by small environmental noise instead of flips in Ψ. This is why our attack methodology is more widely applicable, including attacking perfectly reliable XOR APUFs.

IV. Experimental Evaluation

In our experiments, we use a commercial laptop with Intel(R) Core(TM) i9-10885H CPU and 16.0 GB RAM for the simulations, and we use a Nexys 4 DDR DIGILENT board for the FPGA XOR APUF implementations. All the APUFs or XOR APUFs being attacked are 64-bit long. Some of our source codes are modified from the GitHub repository[2] of the enhanced reliability-based attack [6]. In all of our simulations, the weights of all APUFs are constructed based on the normal distribution $\mathcal{N}(0, 1)$.

A. Validating the Linear Relation between Non-Flipping Probability F and the Delay Difference Δ

The main reason behind our proposed attack is the linear relation between the delay differences of an APUF and its output reliability/non-flipping probability, as we analyze in Sec. III-D. To validate our theory (Eq. 11) empirically, we plot the absolute value of the delay difference $|\Delta|$ and the non-flipping probability of the responses for 300,000 different centroid challenges of a random APUF. In Fig. 2, We can see the linear growth of non-flipping probability with respect to the increasing $|\Delta|$. Fig. 2 (b) presents a subset of the data with $F < 1$ to see the linear relationship between $|\Delta|$ and non-flipping probability more clearly. Fig. 2 (a) includes all the 300,000 CRPs experimental results, and because this figure includes those data with the delay difference value exceeding a certain threshold, the corresponding non-flipping probability F is one; it is expected because with high $|\Delta|$ the response of an APUF becomes very reliable and not vulnerable to perturbations.

[2]https://github.com/scluconn/DA_PUF_Library

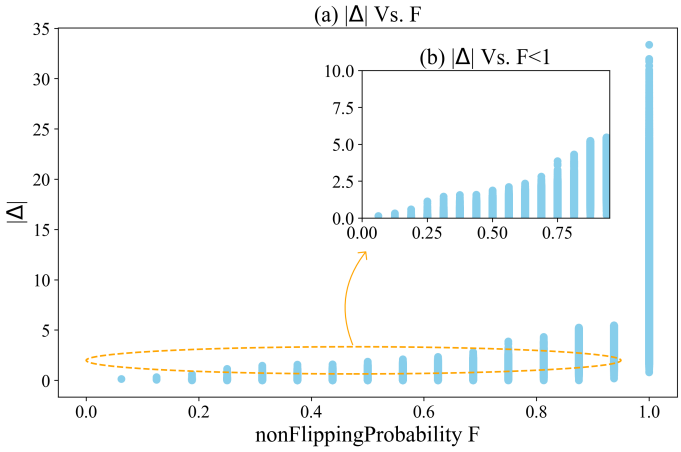

Fig. 2. (a): The scatter plot of the absolute delay difference $|\Delta|$ of a random arbiter PUF versus its non-flipping probability F for 300,000 CRPs. (b): Focusing on the data when the non-flipping probability F is less than one.

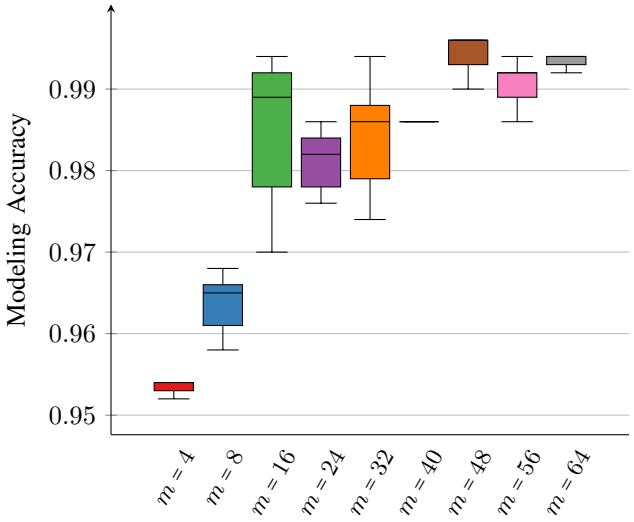

Fig. 3. Prediction Accuracy of the Chosen Challenge Attack on 64-bit 2-XOR APUF with different number of samples m.

B. Attacking Perfectly Reliable PUFs in Simulation

We want to study the sensitivity of the modeling accuracy w.r.t. the number of samples (m) around every centroid challenge. We test various numbers of samples on a perfectly reliable 2-XOR APUF using the proposed attack, each with 150×10^3 (Ψ, F) pairs. As shown in Fig. 3, we test the numbers of samples from 4 to 64, and we see the modeling accuracy increasing with the increasing number of samples. As the green box illustrates in Fig. 3, using 16 samples or more, the prediction accuracy of individual APUFs would be considerably high, and by increasing the number of samples

TABLE I
THE MODELING ACCURACY OF THE INDIVIDUAL APUFS IN A SIMULATED RELIABLE XOR APUF.

#XORs	Training Dataset size	Modeling Accuracy of our Attack	Modeling Accuracy of Becker Attack [7] *	Duration† of our Attack	Duration‡ of Becker Attack
1	20×10^3	98.6% - 99.0%	98.3% - 99.3%	0.5 min	6 min
4	150×10^3	97.6% - 98.6%	99.1% - 99.7%	4.5 min	27 min
8	300×10^3	95.8% - 98.0%	99.1% - 99.7%	20 min	58 min
16	500×10^3	93.6% - 94.8%	98.7% - 99.6%	21 min	94 min

* Since Becker's attack does not work on a reliable PUF, we cannot make a fair comparison. The column fourth presents the results of Becker's attack on an *unreliable* 128-bit PUF reported in [7].
† Duration times correspond to the average duration time of each CMA-ES run.
‡ Duration times correspond to the average time of Becker attack per run taken from [7]. This column is not comparable with our attack duration because they were measured under different devices and different XOR APUF stages.

TABLE II
THE MODELING ACCURACY OF OUR PROPOSED ATTACK FOR THE INDIVIDUAL APUFS IN A SIMULATED RELIABLE XOR APUF.

#XORs	Training Dataset size	Modeling Accuracy of our Attack	Duration†
1	20×10^3	99.2% - 99.6%	0.5 min
4	80×10^3	96% - 96.5%	3 min
8	160×10^3	97% - 98.5%	6 min
16	320×10^3	94% - 94.3%	13 min

† Duration times correspond to the average duration time of each CMA-ES run.

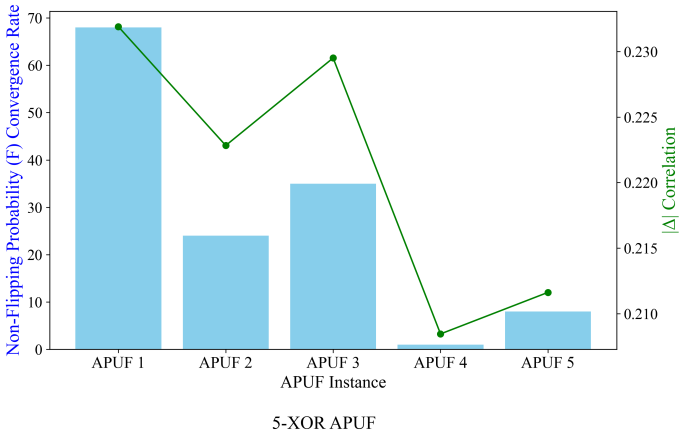

Fig. 4. The right Y-axis: The correlation between the delay difference $|\Delta|$ value of constituent APUFs and the overall non-flipping probability of the 5-XOR APUF. The left Y-axis: The convergence rate of each constituent APUF. The chosen challenge attack is performed on a 5-XOR APUF with 150 CMA-ES runs using the same dataset.

from 16 to 64, there is not much remarkable improvement in accuracy. Therefore, we use 16 samples for the rest of our experiments. It seems to be a good trade-off between the modeling accuracy and the sampling/computational cost.

Comparison with the reliability-based attacks. The results in Table I show that the proposed chosen challenge attack can attack individual APUFs in a perfectly reliable XOR APUF with high accuracy. In contrast, the reliability-based attacks do not work in the same condition. The prediction accuracy of the original Becker's attack (directly copied from [7]) on noisy XOR APUFs with the same number of XORs and the same number of challenge-reliability pairs is also shown in Table I.

Scalability. We show the scalability of our proposed attack in Table. II. When the size of the training dataset is linearly scaled with respect to the number of XORs, we see a slight performance degradation in the modeling accuracy, likely due to the more complex structure in larger XOR APUFs. However, the attacks on the 16-XOR APUF can still be considered successful in a practical setting.

Convergence Rate. Next, we study the convergence rate of each APUF in the random process of CMA-ES in our attack. The convergence rate represents the chance of an individual APUF being successfully modeled in an XOR APUF in our attack. It indicates the susceptibility of individual APUFs to the modeling attack. We attack a reliable 64-bit 5-XOR APUF by our attack with 150 independent CMA-ES runs using the same dataset. We can clearly see that the correlation between $|\Delta|$ of each APUF and the overall output non-flipping probability, or in short $|\Delta|$ correlation, is strongly related to the convergence rate of each APUF in Fig. 4. The convergence rate for 150 independent CMA-ES runs is defined as the fraction of runs that successfully identify an accurate APUF model. Success is determined by achieving a Pearson correlation coefficient (ρ) greater than 0.95 between the predicted and observed non-flipping probabilities of the individual APUF. When an APUF has a strong $|\Delta|$ correlation, the CMA-ES optimization process has a higher chance of convergence to that APUF model. Also, we notice that even if some of the APUFs have a relatively low $|\Delta|$ correlation in a given dataset, it is still possible for the attack to converge to them.

C. Attacking Realistically Unreliable PUFs in Simulation

The proposed attack works not only with reliable PUFs but also on realistically unreliable PUFs. In every noisy APUF simulation, noisy APUF weights are independently drawn from a normal distribution with $\sigma_{Noise} * \sigma$ as the standard deviation, where $\sigma = 1$. In this way, the proposed physical

TABLE III
THE MODELING ACCURACY OF THE INDIVIDUAL APUFS IN A SIMULATED NOISY XOR APUF.

#XORs	Training Dataset Size	σ_{Noise}	Modeling Accuracy of our Attack	Modeling Accuracy of Attack in [6]	PUF Reliability	Duration† of our Attack	Duration† of Attack in [6]
1	20×10^3	0.0055	98.6% - 99.6%	99.2% - 99.6%	99%	0.6 min	0.5 min
		0.01	99.0% - 99.4%	100% - 100%	98%	0.6 min	0.5 min
		0.025	98.8% - 99.5%	99.3% -99.7%	97%	1.2 min	0.5 min
3	150×10^3	0.0055	97.0% - 98.2%	99.2% - 99.6%	98%	4.6 min	4 min
		0.01	98.0% - 99.0%	99.6% - 100%	96%	4.8 min	4 min
		0.025	98.9% - 99.4%	98.8% - 99.5%	92%	4.8 min	4.5 min
5	300×10^3	0.0055	98.8% - 99.4%	98.2% - 98.6%	96%	9.2 min	8.7 min
		0.01	97.8% - 98.2%	99.4% - 99.6%	94%	9.4 min	9 min
		0.025	96.5% - 98.8%	99.1% - 99.6%	86%	17.6 min	7.5 min
7	300×10^3	0.0055	96.0% - 98.4%	99.4% - 99.8%	95%	10 min	9.3 min
		0.01	98.0% - 98.4%	99.2% - 99.4%	92%	11 min	9.5 min
		0.025	96.5% - 97.1%	99.6% - 99.3%	82%	21.8 min	9.5 min
15	500×10^3	0.0055	93.0% - 95.2%	99.4% - 99.8%	91%	22 min	20 min
		0.01	94.2% - 97.4%	99.8% - 99.8%	85%	23 min	18 min
		0.025	93.0% - 93.8%	99.1% - 99.7%	66%	21 min	18.5 min

* σ_{noise} values are taken from the noise levels used in Footnote 2
† Duration times correspond to the average duration time of each CMA-ES run.

error is converted to the mathematical model of APUFs in influencing the delay difference Δ of APUF in the simulation. As shown in Table III, when σ_{Noise} = 0.0055 and even with more noise as σ_{Noise} = 0.01 and σ_{Noise} = 0.025, the accuracy of the chosen challenge attack is considerably high, and the XOR APUF could be modeled. Note that the noise-free weight vectors of APUFs are drawn from $\mathcal{N}(0,1)$ in the simulation. The three noise level parameters σ_{Noise} are inherited from the noisy PUF simulation[2], and they correspond to three reliability levels measured by [6] on their APUF FPGA implementations. We also reproduced the results of the enhanced reliability-based attack on the same targets using the authors' open-source codes[2] so that the two attacks can be properly compared in Table III. In the reliability-based attack, each challenge-reliability pair is derived from 17 measurements, just like each (Ψ, F) pair is derived from 17 CRPs in our chosen challenge attack. The measured reliability of each PUF is also reported in Table III. It is measured as the percentage of CRPs that do not flip within 10 measurements. Note that we only evaluate our attacks on odd number XOR APUFs in all the following experiments to avoid the influence of the systemic bias in the responses of even number XOR APUFs [20].

From Table III, we notice that, in general, the proposed attack has slightly lower accuracy than the enhanced reliability-based attack. This is because the perturbation in Δ in the reliability-based attack is smaller than the perturbation introduced by one flip in Ψ in our attack. Smaller perturbation means a lower chance of flipping the final response under the perturbation. This means that in the dataset collected for the chosen challenge attack, it is more likely that more than one APUF response gets flipped, and this is considered noise in our attack and negatively affects the training.

Upon detailed examination of Table III, it is observed that for both attack methodologies, an increase in the standard deviation of noise from 0.0055 to 0.01 slightly enhances the modeling accuracy in general. This improvement can be potentially attributed to the expansion of the σ_{Noise} parameter within APUF weights, which broadens the range of Δ's variations. Consequently, a greater number of challenges are likely to result in flipped responses. As demonstrated in Fig. 2, the increased variability in $|\Delta|$ leads to more challenges contributing to the linear relation analysis, thereby enriching the dataset with more informative data points for more effective modeling.

TABLE IV
THE MODELING ACCURACY OF THE INDIVIDUAL APUFS IN A SIMULATED XOR APUF WITH 4 DIFFERENT LEVELS OF NOISE UNDER THE 2-BIT FLIP ATTACK.

#XORs	Training Dataset Size	σ_{Noise}	Modeling Accuracy	Duration†
1	20×10^3	0	98.4% - 99.0%	1 min
		0.0055	99.4% - 99.8%	1 min
		0.01	99.2% - 99.8%	1 min
		0.025	98.4% - 99.2%	1 min
3	150×10^3	0	94.4% - 96.0%	7.5 min
		0.0055	98.2% - 98.8%	10 min
		0.01	97.0% - 99.2%	7 min
		0.025	97.6% - 99.0%	7 min
5	300×10^3	0	91.2% - 93.0%	16 min
		0.0055	97.4% - 97.6%	23 min
		0.01	97.0% - 98.0%	15 min
		0.025	94.6% - 98.2%	15 min
7	300×10^3	0	92.8% - 95.2%	14 min
		0.0055	93.8% - 94.6%	15.5 min
		0.01	92.4% - 92.8%	20 min
		0.025	91.0% - 96.0%	14 min

† Duration times correspond to the average duration time of each CMA-ES run.

D. More Bit Flips Attacks in Simulation

One may propose to use a programmable access control PUF interface [17] to implement a more complex access control policy, which keeps a list of erased CRPs and blocks the measurement of all CRPs of d_H = 1 with an erased CRP. Hence, In the next step, we aim to extend the methodology of our proposed attack and investigate the resulting impact of flipping more bits in Ψ on the modeling accuracy.

We choose the challenges around the proposed challenge that have two bits flip in their corresponding Ψ vector. Hence, by extending to two-bit flips, there would be $\frac{64(64-1)}{2}$ = 2016 available Ψ_i vectors with d_H=2 to be selected from, instead

TABLE V
THE MODELING ACCURACY OF COMBINED LR-RELIABILITY ATTACK AND COMBINED LR-NONFLIPPING PROBABILITY ATTACK ON XOR APUFS.

#XORs	Training Dataset Size	σ^2_{Noise} †	Modeling Accuracy R [10]	Modeling Accuracy F	PUF Reliability	#Trials	#Epochs
4	20×10^3	0.5	87.0%	89.3%	90%	6	25
		0.25	91.8%	90.3%	92%		
		0.1	93.1%	94.4%	95%		
		0	98.7%*	97.9%	100%		
6	40×10^3	0.5	84.0%	84.4%	85%	6	25
		0.25	84.6%	87.5%	89%		
		0.1	90.9%	90.7%	93%		
		0	50.7%*	96.0%	100%		
8	60×10^3	0.5	79.0%	77.1%	82%	6	25
		0.25	83.8%	83.8%	87%		
		0.1	87.1%	87.9%	92%		
		0	52.7%*	94.4%	100%		
10	100×10^3	0.5	74%	75.0%	78%	12	25
10	200×10^3	0.25	79%	78.5%	84%	12	40
		0.1	84%	83.8%	89%		
		0.025	89%	88.5%	94%		
		0	50.3%*	93.0%	100%		

† The PUF noise defined in [10] and subsequently in section IV-E is added to Δ; Hence, the interpretation of the value σ_{Noise} is slightly different from our other experiments where the noise influences every weight vector of APUFs as a structural noise. However, the reliability metric is consistent among the experimental results and offers a reference for understanding the effect of changing σ_{Noise}.

* For simulating the combined LR-Reliability attack when σ_{Noise} is zero since computing the reliability is not possible without noise, we disable the reliability term in the loss function. The training dataset sizes (numbers of CRPs) are multiplied by 17.

of just 64 vectors, if we still attack 64-bit APUFs. This extension strongly reduces the chance of using an access control interface to successfully defend against our attack because it would be too costly for an interface to block accessing all 2016 available Ψ_i around an erased challenge. In other words, erasing one challenge will effectively result in erasing 2016 challenges from the PUF, and thus, the challenge space will be depleted much faster.

Table IV presents the modeling accuracy of individual APUFs in an XOR APUF with four different noise levels σ_{Noise} when 16 challenges with 2-bit flipped (d_H = 2) in Ψ are sampled around each centroid challenge. The results show that the XOR APUFs still could be modeled with high accuracy values. However, as expected, the accuracy is lower than that of the one-flip attacks in Tables I and III because likely more multiple APUF response flips occur in the training dataset, which cancels out some of the individual response flips and adds more noise into the training dataset. According to Table IV, with the growing size of XORs, the modeling accuracy decreases, and this decrease is steeper than what occurs in Tables I and III. The reason is that as the size of the perturbation becomes larger through higher hamming distances, the higher the number of XORs, the higher the probability of crossing the decision boundaries of multiple APUFs.

E. Compatibility of non-flipping probability-based Attack with Combined gradient-based Attack

The combined multi-objective attack framework presented in [10] integrates the strengths of the direct modeling by CRPs and the reliability-based approach through a gradient-based optimization framework with multiple objectives. The authors demonstrated that combining the reliability attacks, weight constraints, and Logistic Regression (LR) into a single optimization framework enhances the efficiency of modeling attacks against strong PUFs, including XOR APUF and interpose PUF (iPUF). This combined method allows the simultaneous exploitation of both responses and reliability information, leading to a more effective attack on iPUFs compared to traditional approaches that just rely on CMA-ES or gradient-based methods without this integration.

This combined methodology inspired us to introduce non-flipping probability into the combined attack on XOR APUF. Moreover, we want to make sure that our chosen challenge methodology to exploit non-flipping probability information can integrate with other objectives effectively in the combined attack framework. Furthermore, tailoring the combined attack to employ F information specifically for chosen challenge attacks enables a more efficient attack when the reliability information is less accessible for the attackers. Our experimental investigations reveal that incorporating non-flipping probability F within the combined attack methodology yields a similar level of efficiency with [10] in predicting XOR APUFs behaviors across different configurations and sizes. To be precise, we use the objective of optimizing the correlation between $|\Delta|$ of individual APUFs and F to replace the objective of optimizing the correlation between the measured reliability and the predicted reliability in [10].

Table V shows the simulation results of combined LR reliability-based attack [10] on XOR APUF and our proposed combined LR non-flipping probability-based attack on XOR APUFs in the same configurations and with multiple noise levels.[3] This result shows the inherent stability and wide applicability of the non-flipping probability-based approach. For computing non-flipping probability, 16 samples around a centroid challenge are collected. According to the same configuration, in reliability-based attacks, each challenge is measured 16+1=17 times to assess the response's reliability. The same efficiency of the combined LR non-flipping probability-based

[3]Our implementation of the combined attack is modified from the source codes provided by the authors of [10] at https://github.com/jtobi/puf-simulation/.

TABLE VI
THE MODELING ACCURACY OF COMBINED LR-RELIABILITY ATTACK AND COMBINED LR-NONFLIPPING PROBABILITY ATTACK ON IPUFS.

(x,y)	Training Dataset Size	σ^2_{Noise}	Modeling Accuracy R [10]	Modeling Accuracy F	PUF Reliability	#Epochs	Batch Size
(1,4)	40×10^3	0.5	87.0%	86.5%	90%	15	256
		0.25	90.7%	89.6%	92%		
		0.1	92.8%	93.4%	95%		
		0	98.7%*	96.2%	100%		
(1,6)	150×10^3	0.5	82.0%	83.3%	85%	15	256
		0.25	87.8%	86.8%	88%		
		0.1	90.6%	90.1%	93%		
		0	50.1%*	96.6%	100%		
(1,8)	200×10^3	0.5	77.0%	78.8%	81%	25	256
		0.25	82.2%	81.9%	86%		
		0.1	87.5%	85.9%	91%		
		0	50.1%*	93.3%	100%		
(1,10)	500×10^3	0.5	73.0%	73.2%	77%	25	256
		0.25	77.5%	78%	82%		
		0.1	83.0%	82.8%	88%		
(1,10)	800×10^3	0.025	89.5%	87.1%	94%	25	256
		0	50.5%*	91.3%	100%		

* For simulating the combined LR-Reliability attack when σ_{Noise} is zero since computing the reliability is not possible without noise, we disable the reliability term in the loss function. The training dataset sizes (numbers of CRPs) are multiplied by 17.

attack with the combined LR reliability-based attack highlights the potential of the proposed method as a new standard for assessing APUF vulnerabilities.

Similar to our previous experiment, we demonstrate the feasibility of a successful attack on a reliable PUF without noise using our combined attack. As we can see in Table V, an accuracy of 93% is produced by a combined LR non-flipping probability-based attack on 10-XOR APUF with σ_{Noise} = 0. To make a fair comparison with the modeling accuracy of the combined LR-Reliability attack, we should also use the combined LR-Reliability attack to attack perfectly reliable XOR APUFs. Since it is impossible to measure reliability when σ_{Noise}=0, we disable the reliability term in the loss function of the attack, and only the response prediction accuracy is considered for this part of the experiments. Moreover, because we do not need to repeatedly measure the same CRPs on a reliable PUF, we multiply the corresponding training dataset size by the number of samples m+1 = 17 for a fair comparison. We see that when the number of XOR is 6 or larger, our attack achieves a significant improvement in the modeling accuracy compared with the combined LR-Reliability attack.

Furthermore, the combined LR-non-flipping probability attack targeting iPUF [6] provides the same successful prediction accuracy as the combined LR-Reliability attack when the iPUF is not reliable. Table VI includes the simulation results of the combined LR-non-flipping probability attack on iPUF along with the combined LR-Reliability attack on iPUF for comparison. To run the LR-Reliability attack on reliable iPUF, we also disable the reliability term in the loss function, and the training dataset size for these experiments is 17 times that of the corresponding LR-non-flipping probability attack. The combined LR-Reliability attack is not successful on (1,6)-iPUF and larger iPUFs when there is no noise.

In the experimental results in Tables V and VI, a different trend from the experimental results of previous subsections emerges concerning the relationship between the increasing noise level and the modeling accuracy. Previously, in Tables III and IV, the decrease in the noise level results in a reduction of modeling accuracy for both non-flipping probability-based and reliability-based CMA-ES ML attacks. This can be attributed to the finer sensitivity of these attacks to perturbations, which are less prevalent in a lower-noise PUF, thereby reducing the effectiveness of the modeling. Conversely, Tables V and VI show that under LR-combined attack strategies, the modeling accuracy becomes higher when there is a reduction in the noise standard deviation. The reason is the direct presence of CRPs in the loss function of the combined attack. Thus, the type of attack and its underlying mechanics significantly influence how noise impacts modeling accuracy. The standalone non-flipping probability-based or reliability-based ML attacks suffer from reduced noise, while combined approaches benefit from it.

F. Attacking FPGA Implemented PUFs

Last but not least, we evaluate the proposed attack on XOR APUFs implemented on an FPGA. Table VII shows the modeling accuracy of the individual APUFs under the proposed attack, and its high accuracy validates the effectiveness of our attack in practice. We target different sizes of XOR APUF up to 9 XORs. High prediction accuracy values can be achieved for constituent APUFs of XOR APUF with the chosen challenge attack. Even though the modeling accuracy falls slightly when increasing the size, it stands above 96% for the 9-XOR APUFs.

V. COUNTERMEASURES

One straightforward but effective countermeasure against our attack is to add a one-way function (OWF) before an XOR APUF and enforce that the one-way function and the PUF must be evaluated as a whole: $R = PUF(OWF(C))$ as suggested in [4], [21], [22]. Then, an attacker cannot choose any specific PUF challenges to be evaluated anymore, due to the one-wayness of the added one-way function [23]. Hence, with a specific Hamming distance value, it is impossible to measure the non-flipping probability for the chosen challenge

TABLE VII
THE MODELING ACCURACY OF THE INDIVIDUAL APUFS IN A REAL XOR APUF ON XILINX FPGA.

#XORs	Training Dataset Size	Modeling Accuracy	CRP Collection Time	Simulation Time†
1	20×10^3	99.0%-99.3%	$\simeq 15h$	$\simeq 0.5 - 1$ min
3	75×10^3	98.0%-98.4%	$\simeq 57h$	$\simeq 2 - 3$ min
5	75×10^3	97.3%-98.0%	$\simeq 57h$	$\simeq 2 - 3$ min
7	75×10^3	96.6%-97.6%	$\simeq 57h$	$\simeq 2 - 3$ min
9	75×10^3	96.2%-96.3%	$\simeq 57h$	$\simeq 2 - 3$ min

† Simulation time is the average duration time of each CMA-ES run.

attack. A similar idea has been proposed in [21], which uses a hash function to control the input of the PUF and prevent selecting the challenge required for exploiting informative parameters.

However, this countermeasure may not be ideal to implement in a lightweight application scenario, e.g., RFID tags, due to the area overhead incurred by a one-way function. According to [24], there may be 1000-10000 gates on a basic RFID tag, but only 200-2000 are allocated specifically for security. The lightweight PHOTON hash function requires 865 gate equivalence, according to [25], which is still a relatively large portion of the whole security overhead. The occupied area for the hash function varies in different cases, but it is still considerably large compared with the small area footprint of a PUF.

Furthermore, this countermeasure does not prevent reliability-based attacks since reliability information relies only on the repetition of the measurement for the same challenge and does not depend on any specifically chosen challenges. Therefore, the OWF alone cannot resist the reliability-based attack on XOR APUF, and more countermeasures should be combined with it. As a strong countermeasure against the chosen challenge attack, reliability-based attacks, and combined attacks, a novel, secure, strong PUF design is needed.

Another potential countermeasure for countering ML attacks is response obfuscation, where the PUF's output is intentionally poisoned under hidden conditions to mislead modeling attacks. For example, Wang et al. [26] proposed an adversarial PUF that uses a secret trigger signal, derived from the challenge and a secret control word, to inject misleading responses. However, keeping this control word secret is not trivial and introduces overhead. Also, it should not contradict the fundamental goal of using PUF to avoid on-chip digital secrets.

As a strong countermeasure against the chosen challenge attack, reliability-based attacks, and combined attacks, a novel, secure, strong PUF design is needed.

VI. OTHER RELATED WORK

Existing Chosen Challenge Attacks. Prior to this work, some other studies on (adaptively) chosen challenge attacks on APUF have been proposed. For example, active learning methods can improve the efficiency of modeling by collecting informative CRPs, i.e., the CRPs with the highest uncertainty [27]. Adaptively choosing CRPs to measure based on optimization theory for reducing the needed number of CRPs is used in [28]. However, both the chosen challenge attacks above can only attack single APUFs, not XOR APUFs.

The most recent and relevant study is a chosen challenge attack exploiting the output transitions of an XOR APUF [29]. The attack intentionally collects pairs of CRPs, which flip the response bit of an XOR APUF with limited input flips in raw challenges (instead of the Ψ vectors in our attack). However, the authors directly used the collected CRPs for machine learning training to attack an XOR APUF as a whole. Thus, they only achieved up to 50% reduction in the required number of CRPs in attacking up to 5-XOR APUFs, while our attack can attack individual APUFs instead of the XOR APUF as a whole and demonstrate successful attacks on larger XOR APUFs.

The Effect of Single-bit Flips in Challenges. The relationship of input bit flips and output bit flips of an XOR APUF has been well studied in the form of strict avalanche criteria, predictability test, and input sensitivity [29]–[31]. However, the understanding has been mainly used for assessing the machine learning resilience of a PUF [32], [33] and for inspiring new PUF design ideas [34]. A relevant recent work analyzed the sensitivity of strong PUF responses (including XOR Arbiter PUFs, XOR Bistable Ring PUFs, and FeedForward Arbiter PUFs) to a single-bit flip in challenges as security metrics to enhance the design assessments [33]. In our attack, we introduce a single-bit flip in Ψ instead of in the challenge and study the effect of bit flips in Ψ.

Classical ML Attacks on XOR APUFs. Several recent attempts have been made to attack XOR APUFs using classical machine learning attacks, meaning the CRPs are used directly in training [5], [12]–[15], [35]. A deep feedforward neural network-based modeling attack is proposed in [13] targeting Arbiter PUF and multiple compositions of APUF, including XOR-APUF up to 6 APUFs. Wisiol et al. presented an efficient neural-network-based modeling attack on XOR APUF using challenge-response pairs [15]. However, the attack does not scale well with the number of XORs in an XOR APUF. Another artificial neural network-based attack proposed by [35] characterizes the structure of XOR APUF up to 5 APUFs. Neural network-based tools could further reduce the computational complexity of ML attack on XOR APUF using CP-decomposition-based tensor regression network [14]. Hongming *et al.* [36] proposed a generic attack framework for modeling multiple XOR APUFs and other delay-based PUFs using a mixture-of-PUF-experts neural architecture, without structural knowledge. However, this attack requires a large number of CRPs and scales only up to 7-XOR in their experiments. The mentioned modeling attacks utilize passive learning methods that analyze the PUF behavior based on a set of CRPs that are collected randomly without a specific plan. In general, Neural networks-based attacks on XOR APUF, as detailed in [14], [15], present a more general framework compared to other methods like reliability-based attacks and non-flipping probability-based attacks. However, despite their broad applicability, the complexity of these neural network methods cannot scale linearly with respect to the number of XORs and struggles with large XOR APUFs beyond a certain

point.

Recently, CalyPSO [11], an enhanced search optimization framework inspired by Particle Swarm Optimization (PSO), has been demonstrated to model delay-based PUFs effectively, including XOR APUF variants without relying on reliability or side-channel information. The CalyPSO method provides a successful attack on the largest XOR APUFs modeled to date, up to 20-XOR APUFs, but the attack still takes the XOR APUF as a whole and optimizes all the APUF models together. Moreover, their experiments were conducted on a supercomputer as Intel(R) Xeon(R) Gold 6226 CPU @ 2.70 GHz with 96 cores, 2 threads per core, 12 cores per socket, 256GB DRAM, and each experiment was spread across 4 physical cores through Python's *multiprocessing.Pool* while our research has been done on a commercial laptop, as mentioned earlier in Section IV.

A recent study [37] demonstrated that challenge-response obfuscation (CRO) can be bypassed using deep learning techniques. This method specifically targets PUFs that utilize CRO and uncovers both the original challenges and the behavior of the PUF by passively collecting CRPs. While their research emphasizes obfuscated versions rather than standard XOR APUFs, it exposes the weaknesses of CRO-based defenses when confronted with deep learning models. In contrast, our attack targets XOR APUFs directly using chosen-challenge queries, representing a complementary attack strategy on standard PUFs.

Generally speaking, the existing classical PUF ML attacks that only use CRPs directly for training do not scale well with a large number of XORs. To the best of our knowledge, only the reliability-based attack and our attack can use a divide-and-conquer approach to simplify the optimization problem regardless of the size of XOR APUFs.

VII. Conclusion

The XOR APUF is a component of many state-of-the-art strong PUFs (e.g., iPUF [6]) and one of a few PUF designs deployed in practice (e.g., in RFID tags [7]); Therefore, it is very important to analyze its security in different dimensions and conditions. In this work, we use the non-flipping probability of chosen challenges instead of the reliability for attacking individual APUFs in XOR APUFs. Our method successfully attacks different sizes of XOR APUF in both reliable and unreliable conditions with high prediction accuracy. The unified theoretical foundation of our attack and the reliability-based attack also shows that our attack is a generalization of the reliability-based attacks, and thus, our attack method is applicable in more scenarios, including perfectly reliable PUFs. The compatibility of the chosen challenge method with the combined multi-objective attack is another proof of the generality of non-flipping probability. We validate our attack experimentally using simulation and FPGA implementations.

Acknowledgments

We would like to thank the valuable comments from the anonymous reviewers. Marten van Dijk and Chenglu Jin are (partially) supported by project CiCS of the research programme Gravitation which is (partly) financed by the Dutch Research Council (NWO) under the grant 024.006.037.

References

[1] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, V. Atluri, Ed. ACM, 2002, pp. 148–160. [Online]. Available: https://doi.org/10.1145/586110.586132

[2] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 13, no. 10, pp. 1200–1205, 2005. [Online]. Available: https://doi.org/10.1109/TVLSI.2005.859470

[3] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th Design Automation Conference, DAC 2007, San Diego, CA, USA, June 4-8, 2007*. IEEE, 2007, pp. 9–14. [Online]. Available: https://doi.org/10.1145/1278480.1278484

[4] D. Gurevin, C. Jin, P. H. Nguyen, O. Khan, and M. van Dijk, "Secure remote attestation with strong key insulation guarantees," *IEEE Transactions on Computers*, 2023.

[5] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, Eds. ACM, 2010, pp. 237–249. [Online]. Available: https://doi.org/10.1145/1866307.1866335

[6] 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. [Online]. Available: https://doi.org/10.13154/tches.v2019.i4.243-290

[7] G. T. Becker, "The gap between promise and reality: On the insecurity of XOR arbiter pufs," in *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, ser. Lecture Notes in Computer Science, T. Güneysu and H. Handschuh, Eds., vol. 9293. Springer, 2015, pp. 535–555. [Online]. Available: https://doi.org/10.1007/978-3-662-48324-4_27

[8] N. Wisiol and M. Margraf, "Why attackers lose: design and security analysis of arbitrarily large XOR arbiter pufs," *J. Cryptogr. Eng.*, vol. 9, no. 3, pp. 221–230, 2019. [Online]. Available: https://doi.org/10.1007/s13389-019-00204-8

[9] C. Jin, W. P. Burleson, M. van Dijk, and U. Rührmair, "Erasable pufs: Formal treatment and generic design," in *Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security Workshop, ASHES@CCS 2020, Virtual Event, USA, November 13, 2020*, C. Chang, U. Rührmair, S. Katzenbeisser, and P. Schaumont, Eds. ACM, 2020, pp. 21–33. [Online]. Available: https://doi.org/10.1145/3411504.3421215

[10] J. Tobisch, A. Aghaie, and G. T. Becker, "Combining optimization objectives: New modeling attacks on strong pufs," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 2, pp. 357–389, 2021. [Online]. Available: https://doi.org/10.46586/tches.v2021.i2.357-389

[11] N. Mishra, K. Pratihar, S. Mandal, A. Chakraborty, U. Rührmair, and D. Mukhopadhyay, "Calypso: An enhanced search optimization based framework to model delay-based pufs," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2024, no. 1, pp. 501–526, 2024.

[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, no. 11, pp. 1876–1891, 2013.

[13] P. Santikellur, A. Bhattacharyay, and R. S. Chakraborty, "Deep learning based model building attacks on arbiter puf compositions," *Cryptology ePrint Archive*, 2019.

[14] P. Santikellur and R. S. Chakraborty, "A computationally efficient tensor regression network-based modeling attack on xor arbiter puf and its variants," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 6, pp. 1197–1206, 2020.

[15] N. Wisiol, B. Thapaliya, K. T. Mursi, J. Seifert, and Y. Zhuang, "Neural network modeling attacks on arbiter-puf-based designs," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2719–2731, 2022. [Online]. Available: https://doi.org/10.1109/TIFS.2022.3189533

[16] N. N. Anandakumar, M. S. Hashmi, and M. A. Chaudhary, "Implementation of efficient xor arbiter puf on fpga with enhanced uniqueness and security," *IEEE Access*, vol. 10, pp. 129 832–129 842, 2022.

[17] C. Jin, W. P. Burleson, M. van Dijk, and U. Rührmair, "Programmable access-controlled and generic erasable PUF design and its applications," *J. Cryptogr. Eng.*, vol. 12, no. 4, pp. 413–432, 2022. [Online]. Available: https://doi.org/10.1007/s13389-022-00284-z

[18] 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 *Cryptographic Hardware and Embedded Systems–CHES 2014: 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings 16*. Springer, 2014, pp. 476–492.

[19] J. Delvaux and I. Verbauwhede, "Side channel modeling attacks on 65nm arbiter pufs exploiting CMOS device noise," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2013, Austin, TX, USA, June 2-3, 2013*. IEEE Computer Society, 2013, pp. 137–142. [Online]. Available: https://doi.org/10.1109/HST.2013.6581579

[20] N. Wisiol and N. Pirnay, "Short paper: XOR arbiter pufs have systematic response bias," in *Financial Cryptography and Data Security - 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10-14, 2020 Revised Selected Papers*, ser. Lecture Notes in Computer Science, J. Bonneau and N. Heninger, Eds., vol. 12059. Springer, 2020, pp. 50–57. [Online]. Available: https://doi.org/10.1007/978-3-030-51280-4_4

[21] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas, "Controlled physical random functions," in *18th Annual Computer Security Applications Conference (ACSAC 2002), 9-13 December 2002, Las Vegas, NV, USA*. IEEE Computer Society, 2002, pp. 149–160. [Online]. Available: https://doi.org/10.1109/CSAC.2002.1176287

[22] M. van Dijk and C. Jin, "A theoretical framework for the analysis of physical unclonable function interfaces and its relation to the random oracle model," *Journal of Cryptology*, vol. 36, no. 4, p. 35, 2023.

[23] J. Katz and Y. Lindell, *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.

[24] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Advances in Cryptology–CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005. Proceedings 25*. Springer, 2005, pp. 293–308.

[25] H. Maleki, R. Rahaeimehr, C. Jin, and M. Van Dijk, "New clone-detection approach for rfid-based supply chains," in *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2017, pp. 122–127.

[26] S.-J. Wang, Y.-S. Chen, and K. S.-M. Li, "Modeling attack resistant pufs based on adversarial attack against machine learning," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 2, pp. 306–318, 2021.

[27] Y. Wen and Y. Lao, "PUF modeling attack using active learning," in *IEEE International Symposium on Circuits and Systems, ISCAS 2018, 27-30 May 2018, Florence, Italy*. IEEE, 2018, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ISCAS.2018.8351302

[28] Y. Liu, Y. Xie, C. Bao, and A. Srivastava, "An optimization-theoretic approach for attacking physical unclonable functions," in *Proceedings of the 35th International Conference on Computer-Aided Design, ICCAD 2016, Austin, TX, USA, November 7-10, 2016*, F. Liu, Ed. ACM, 2016, p. 45. [Online]. Available: https://doi.org/10.1145/2966986.2967000

[29] C. Lin and M. Chen, "Learning from output transitions: A chosen challenge strategy for ML attacks on pufs," in *IEEE/ACM International Symposium on Low Power Electronics and Design, ISLPED 2023, Vienna, Austria, August 7-8, 2023*. IEEE, 2023, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ISLPED58423.2023.10244524

[30] F. Ganji, S. Amir, S. Tajik, D. Forte, and J. Seifert, "Pitfalls in machine learning-based adversary modeling for hardware systems," in *2020 Design, Automation & Test in Europe Conference & Exhibition, DATE 2020, Grenoble, France, March 9-13, 2020*. IEEE, 2020, pp. 514–519. [Online]. Available: https://doi.org/10.23919/DATE48585.2020.9116316

[31] P. H. Nguyen, D. P. Sahoo, R. S. Chakraborty, and D. Mukhopadhyay, "Security analysis of arbiter PUF and its lightweight compositions under predictability test," *ACM Trans. Design Autom. Electr. Syst.*, vol. 22, no. 2, pp. 20:1–20:28, 2017. [Online]. Available: https://doi.org/10.1145/2940326

[32] F. Ganji, D. Forte, and J. Seifert, "Pufmeter a property testing tool for assessing the robustness of physically unclonable functions to machine learning attacks," *IEEE Access*, vol. 7, pp. 122 513–122 521, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2938408

[33] W. Stefani, F. Kappelhoff, M. Gruber, Y.-N. Wang, S. Achour, D. Mukhopadhyay, and U. Rührmair, "Strong puf security metrics: Sensitivity of responses to single challenge bit flips," *Cryptology ePrint Archive*, 2024.

[34] F. Ganji, S. Tajik, P. Stauss, J. Seifert, M. M. Tehranipoor, and D. Forte, "Rock'n'roll pufs: crafting provably secure pufs from less secure ones (extended version)," *J. Cryptogr. Eng.*, vol. 11, no. 2, pp. 105–118, 2021. [Online]. Available: https://doi.org/10.1007/s13389-020-00226-7

[35] 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, no. 10, pp. 2138–2151, 2019.

[36] F. Hongming, O. Millwood, P. Gope, J. Miskelly, and B. Sikdar, "Attacking delay-based pufs with minimal adversarial knowledge," *IEEE Transactions on Information Forensics and Security*, 2024.

[37] N. Hassan and U. Chatterjee, "Machine learning attacks on challenge-response obfuscations in strong pufs," in *2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2024, pp. 361–372.

[38] J. Sölter, "Cryptanalysis of electrical pufs via machine learning algorithms," Ph.D. dissertation, MSc thesis, Technische Universität München, 2009.

APPENDIX A
THEORETICAL ANALYSIS OF THE FLIPPING PROBABILITY GIVEN A FLIPPED $\Psi[i]$

In this part, we investigate the probability of an APUF response flip when there is a sign flip in the feature vector Ψ.

Analysis of a Concrete Manufactured APUF: Let us assume a fixed weight vector $w[\cdot]$ corresponding to a manufactured APUF. According to Eq. 1, the delay difference of an APUF is expressed as:

$$\begin{aligned}\Delta &= D + \Psi[j]\cdot w[j], \text{ where}\\ D &= \sum_{i=0,\neq j}^{n} \Psi[i]\cdot w[i].\end{aligned}$$

When flipping $\Psi[j]$ from -1 to $+1$ or vice versa, the sign of Δ flips if and only if, for $w = w[j]$, $D + w$ and $D - w$ have a different sign. Recall that we assume a perfectly reliable PUF, i.e., without measurement noise[4] (otherwise, even if $D + w$ and $D - w$ have the same sign but with one of them close to zero, measurement noise may induce a response bit flip with significant probability). Since the sign of Δ directly corresponds to the response bit, we have a response bit flip with probability

$$P_{\texttt{flip}} = \Pr\big[(D+w>0 \wedge D-w<0) \vee (D+w<0 \wedge D-w>0)\big],$$

or equivalently $\Pr[|w| > |D|]$. Here, the probability is over the random selection of the challenges in the attack, and this corresponds to a uniform selection of each $\Psi[i]$ from $\{-1,+1\}$ with the exception of $\Psi[n] = 1$.

According to the law of large numbers we have that D is normal distributed with mean $w[n]$ and standard deviation $\hat{\sigma}$ with

$$\hat{\sigma}^2 = \sum_{i=0,\neq j}^{n-1} w[i]^2.$$

[4]If we do assume some measurement noise, for example, in the case when we restrict access to the same challenge-response pair repeatedly, then we can merge its distribution into the distributions of D and w which are analyzed in the remainder of this appendix.

Each of the delay stages $w[i]$ results from a manufacturing process that is assumed to be independent and follow the same normal distribution $\mathcal{N}(0,\sigma^2)$ [5], [38]. Therefore, again by the law of large numbers, $\hat{\sigma}^2$ is a good estimate of $(n-1)\sigma^2$. We have $\hat{\sigma}^2 \approx (n-1)\sigma^2$ with estimation noise having a mean of 0 and standard deviation $O(\sqrt{n})$. We conclude that approximately

$$D - w[n] \sim \mathcal{N}(0,(n-1)\cdot\sigma^2).$$

Expectation of the Flipping Probability: We can now compute the expectation of $P_{\texttt{flip}}$ over all possible realizations/manufacturing of $w = w[j] \sim \mathcal{N}(0,\sigma^2)$ and $v = w[n] \sim \mathcal{N}(0,\sigma^2)$:

$$\begin{aligned}
\mathbb{E}[P_{\texttt{flip}}] &= \mathbb{E}[\Pr[|w| > |D|]] \\
&= 2\cdot\int_{v=-\infty}^{\infty}\frac{e^{-\frac{v^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\cdot\int_{D=0}^{\infty}\frac{e^{-\frac{(D-v)^2}{2(n-1)\sigma^2}}}{\sqrt{2\pi(n-1)\sigma^2}} \\
&\qquad\cdot\int_{w=D}^{\infty}\frac{e^{-\frac{w^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\,\mathrm{d}w\,\mathrm{d}D\,\mathrm{d}v \\
&= 2\cdot\int_{D=0}^{\infty}\frac{e^{-\frac{D^2}{2n\sigma^2}}}{\sqrt{2\pi n\sigma^2}}\cdot\int_{w=D}^{\infty}\frac{e^{-\frac{w^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\,\mathrm{d}w\,\mathrm{d}D
\end{aligned}$$

because adding two normal distributed random variables $v \sim \mathcal{N}(0,\sigma^2)$ and $D - v \sim \mathcal{N}(0,(n-1)\sigma^2)$ yields the random variable $D \sim \mathcal{N}(0,n\sigma^2)$ in the derivation. Substitution of w for $w = D\cdot x$ simplifies integration by scaling w relative to D:

$$\begin{aligned}
\mathbb{E}[P_{\texttt{flip}}] &= 2\cdot\int_{D=0}^{\infty}\frac{e^{-\frac{D^2}{2n\sigma^2}}}{\sqrt{2\pi n\sigma^2}}\cdot\int_{x=1}^{\infty}\frac{e^{-\frac{D^2x^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\,D\,\mathrm{d}x\,\mathrm{d}D \\
&= 2\cdot\int_{x=1}^{\infty}\int_{D=0}^{\infty}\frac{e^{-\frac{D^2}{2\sigma^2}(x^2+\frac{1}{n})}}{2\pi\sqrt{n}\cdot\sigma^2}D\,\mathrm{d}D\,dx \\
&= \int_{x=1}^{\infty}\frac{1}{\pi\sqrt{n}(x^2+\frac{1}{n})}dx \\
&= \int_{x=1}^{\infty}\frac{\sqrt{n}}{\pi(nx^2+1)}dx.
\end{aligned}$$

Now, let $y = \sqrt{n}x$, giving:

$$\mathbb{E}[P_{\texttt{flip}}] = \int_{y=\sqrt{n}}^{\infty}\frac{1}{\pi(y^2+1)}dy = \frac{1}{2} - \frac{1}{\pi}\cdot\arctan(\sqrt{n}).$$

The expected value $\mathbb{E}[P_{\texttt{flip}}] = \frac{1}{2} - \frac{1}{\pi}\cdot\arctan(\sqrt{n})$ relates inversely proportional to the expected number of challenges required for observing at least one response bit flip when flipping one $\Psi[j]$ in a Ψ vector. For n equal to 64, 128, and 256, $\mathbb{E}[P_{\texttt{flip}}]$ is approximately 0.0395, 0.0280 and 0.0198. The expected flipping probability decreases as we increase n, and for larger n our attack becomes less effective (unless we add some optimizations, e.g., flipping multiple Ψ elements).

Interpretation with respect to Attack Efficiency: In our attack, we select random challenges that lead to normally distributed D, as guaranteed by the law of large numbers. For a given randomly chosen challenge, when looking at the joint probability indicating which neighboring Ψ among all neighboring Ψ around a centroid Ψ will lead to a response bit flip, each of the bit flips is positively correlated. This is because all the neighboring Ψ have closely correlated D, which is likely to be small if any response bit flip occurs. Therefore, for smaller D, we expect to see more response bit flips. This is exploited by our implemented CMA-ES attack.

Appendix B
Flipping Position Sensitivity Study

In a further experiment, we evaluated the response sensitivity to Ψ-bit flip across different positions. We conducted an experiment to study the response flipping probabilities based on single-bit flips in the feature vector Ψ. Fig. 5 presents the response flipping probability as a function of the bit-flip positions. For each XOR configuration ($k = 1, 4, 7, 15$), the response flipping probability is measured for all 64 feature vector elements ($\Psi[i]$) under the single-bit flip condition. As demonstrated, the response flipping probability remains roughly on the same level regardless of the position i of the flipped $\Psi[i]$ bit. The variation of the probabilities on the different positions is due to the randomly sampled W vector used in this run of the experiment. Note that this probability is different from the response flipping probability caused by single-bit flips in challenges, and they have different characteristics. For example, for a single APUF, the response flipping probability increases monotonically when the flipped challenge bit moves from the first bit to the last bit (the closest to the arbiter) [33]. Furthermore, as k increases, the overall response flipping probability also increases, reflecting the combined effects of multiple APUFs in the XOR structure. With an increasing k, the variation of the overall sensitivity gets lower and its value gets closer to 50%. This observation further demonstrates the necessity of using Pearson correlation as a robust metric against the "noisy" flips introduced by the other APUFs when attacking one APUF. Theoretically, the probability of a flip in the XOR-APUF response, considering k-XOR APUFs and assuming each APUF response flip is **independent**, is shown in equation 12:

$$\begin{aligned}
P_{\text{XOR_flip}} &= \sum_{i=0,\ \text{odd}}^{k}\binom{k}{i}\cdot P_{\text{flip}}^{i}\cdot(1-P_{\text{flip}})^{k-i} \\
&= \frac{1}{2}\cdot\left(1-(1-2P_{\text{flip}})^{k}\right) \qquad (12)
\end{aligned}$$

For k from 1 to 15 and $P_{\text{flip}} = 0.04$, the $P_{\text{XOR\_flip}}$ grows toward 0.5 which is validated in Fig. 5.

As a further conclusion, if we consider the case where access to the same CRP is restricted so that the classical reliability-based attack using CMA-ES does not work, we still have the measurement noise. Notably, larger n leads to increased measurement noise amplified by the number of APUFs in the XOR construction. Hence, practical XOR APUF designs are constrained to use relatively small values of n (number of APUF delay stages) and k (#XORs) to ensure stability. Accordingly, Appendix A together with Appendix B demonstrate that our proposed attack is successful for practical implementations in access-restricted XOR APUF design.

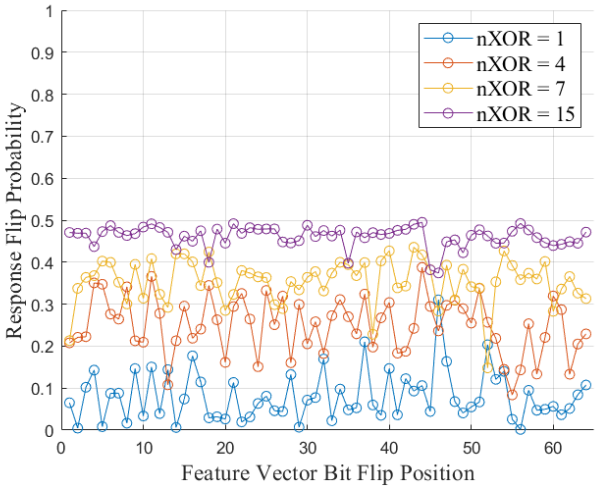

Fig. 5. k-XOR APUF response flipping probability w.r.t. a single feature vector bit flip across different positions for different k-XOR configurations.

APPENDIX C
MORE COMPARISON

Table VIII provides a comprehensive overview of the most relevant modeling attacks on XOR APUFs and related delay-based PUFs.

TABLE VIII
COMPARISON OF MODELING ATTACKS ON XOR APUF

Reference	**Training Data**	**Challenge Collection**	**Max #XOR**	**Targeted PUFs**	**Divide and Conquer Strategy**	**Remarks**
Ruhrmair et al. [5]	CRPs	Random	5	APUF, XOR APUF, FF-APUF, RO PUF	No	Evolution strategies and logistic regression on simulated CRPs
Ruhrmair et al. [12]	CRPs	Random	5	APUF, XOR APUF, FF-APUF, RO PUF	No	Extension of earlier work [5]; applied to real chips
Becker [7]	Challenge-Reliability Pairs	Random	32	XOR APUF	Yes	Models individual APUFs in XOR APUF using Reliability-based CMA-ES
Liu et al. [28]	CRPs	Chosen	1	APUF, MXbarPUF	No	Adaptive challenge selection via Chebyshev center; effective on linearly modeled PUFs
Wen et al. [27]	CRPs	Chosen	1	APUF	No	Active learning on APUFs; uncertainty sampling and query-by-bagging strategies
Santikellur et al. [13]	CRPs	Random	6	APUF, XOR APUF, MPUFs, iPUF	No	Deep Neural Networks (NNs) on CRPs
Shi et al. [35]	CRPs	Random	5	XOR APUF, MPUFs	No	Logical and global approximation attack using ANNs
Santikellur et al. [14]	CRPs	Random	8	XOR APUF	No	A CP-decomposition based tensor regression NN
Wisiol et al. [15]	CRPs	Random	11	XOR APUF, FF PUF, XOR FF-APUF	No	Deep Multilayer Perceptron Attack; surpasses LR attack
Lin et al. [29]	CRPs	Chosen	5	XOR APUF, iPUF, XOR AAPUF†	No	Differential Chosen Challenge ML Attack for training full XOR APUF model based on output transitions.
Hongming et al. [36]	CRPs	Random	7	XOR APUF, XOR FF-APUF, iPUF	No	Mixture-of-PUF-experts NN; minimal assumptions
Mishra et al. [11]	CRPs	Random	20	XOR APUF, iPUF, LP-PUF, FF-APUF, BR-PUF	No	CalyPSO: An Evolution Strategy algorithm inspired by Particle Swarm Optimization (PSO); evaluated on a supercomputing setup
Hassan et al. [37]	CRPs	Random	4	$Mn_{s1,s2,s3}$ (functionally equivalent to XOR APUF), RSO PUF ‡	No	Key extraction and impersonation attack on obfuscation-based PUFs: RSO PUF and $Mn_{s1,s2,s3}$ APUF using CMA-Es and ML
Our Work	Challenge-Non-Flipping Probability Pairs	Chosen	15	XOR APUF	Yes	Models individual APUFs in XOR APUF using CMA-ES; No reliability or side-channel data needed; works on reliable/unreliable PUFs

‡ RSO PUF: Random Set-based Obfuscation PUF, † AAPUF: Adversarial APUF