Towards Attack Resilient Arbiter PUF-Based Strong PUFs

Nils Wisiol
nils.wisiol@tu-berlin.de
Technische Universität Berlin

ABSTRACT

We present the LP-PUF, a novel, Arbiter PUF-based, CMOS-compatible strong PUF design. We explain the motivation behind the design choices for LP-PUF and show evaluation results to demonstrate that LP-PUF has good uniqueness, low bias, and fair bit sensitivity and reliability values. Furthermore, based on analyses and discussion of the LR and splitting attacks, the reliability attacks, and MLP attack, we argue that the LP-PUF has potential to be secure against known PUF modeling attacks, which motivates a discussion of limitations of our study and future work with respect to the LP-PUF.

1 INTRODUCTION

Strong Physical Unclonable Functions (PUFs) have the potential to remove the need for secure non-volatile memory in hardware security tokens like ID documents or credit cards. Unfortunately, strong PUF proposals based on optical tokens require large and extremely sensitive measurement setups Pappu et al. [13]. Alternative proposals using integrated circuits are easy to manufacture when using CMOS technology Gassend et al. [8], but strong PUF designs implemented in CMOS circuits could not yet reliably withstand modeling attacks. As successful modeling attacks enable an attacker to impersonate the PUF token much like is the case in a successful key retrieval attack in the classical setting, modeling attack resilient designs are prerequisite to the deployment of PUF-based security tokens.

In this paper, we propose a novel PUF circuit based on the CMOS-compatible and well-studied Arbiter PUF Gassend et al. [8]. We argue why the design has the potential to withstand known modeling attacks and present simulation results that indicate that a reliable implementation of the design is possible. In more detail, our contributions are:

- The presentation of the novel, CMOS-compatible *LP-PUF* circuit, which is motivated by recent attacks on Arbiter PUF-based PUFs, and a discussion of the motivation to the design (Sec. 3).
- A simulation-based assessment of the fundamental PUF metrics of the LP-PUF (Sec. 3.2).
- A first broad, but promising security analysis of the LP-PUF, with respect to
  - the LR attack [14, 18, 23] and it's variant, the splitting attack [22], both in an analytical and empirical fashion (Sec. 4.1);
  - the reliability attack [2, 17], analytically, but based on empirical evidence on the behavior of simulated LP-PUF instances (Sec. 4.2);
  - the MLP attack [23], in an empirical fashion (Sec. 4.3).
- A discussion of limitations of our preliminary analyses and suggestions for future work (Sec. 5).

2 BACKGROUND

The Arbiter PUF [8] inspired the design of the XOR Arbiter PUF [16], which in turn inspired the design of the Interpose PUF. Each step in this evolution was motivated by by an successful attack on the predecessor design. For all designs, mathematical models based on the *additive delay model* are available that are able to model the behavior of the involved circuit closely. The n-bit k-XOR Arbiter PUF can be modeled as a Boolean function[1] $f : \{-1, 1\}^n \to \{-1, 1\}$ parameterized by $k \cdot n$ real values $W \in \mathbb{R}^{k \times n}$ with represent the intrinsic physical properties of a PUF instance by

$$c \mapsto f(c) = \prod_{l=1}^{k} \left(\operatorname{sgn} \sum_{i=1}^{n} W_{l,i} \cdot x_i \right),$$

where the *feature vector* $x \in \{-1, 1\}^n$ is a function of the challenge defined as $x = \left(\prod_{j=1}^{i} c_j \right)_i$, i.e. $x_1 = c_1 c_2 \cdots c_n$ and $x_2 = c_2 \ldots c_n$, ... and $x_n = c_n$. Writing the model function using the feature vector x rather than the given challenge c is a crucial insight for running modeling attacks; we will refer back to it below. The additive delay model model is motivated by the physics of the Arbiter PUF; for a detailed motivation of the model we refer the reader to Wisiol et al. [23, Appendix A].

For the purpose of simulation of noisy responses of Arbiter PUFs, i.e. $k = 1$, a Gaussian Δ_N with zero mean and prescribed variance is used,

$$c \mapsto f(c) = \operatorname{sgn} \left(\Delta_N + \sum_{i=1}^{n} W_i \cdot x_i \right).$$

This extends to XOR Arbiter PUFs by adding independently chosen noise for each involved Arbiter PUF. We refer to this as the the Arbiter PUF noise model [6].

To study the security of PUFs, we use an attacker model where the attacker gets physical access to the PUF for a limited amount of time after it was manufactured. Afterwards, the PUF is then passed on to the legitimate user.

3 PROPOSED DESIGN

The strong PUF circuit proposed in this paper is an advancement of the Interpose PUF design [12]. The Interpose PUF followed a design strategy similar to the Feed-Forward Arbiter PUF [8] by including challenge bits that have been generated internally, i.e. not been given as part of the challenge. While this does not change the fact that the Arbiter PUF response can be effectively modeled by a linear threshold function, it is supposed to mitigate modeling attacks by depriving the attacker of the knowledge of all input bits.

[1] For the sake of convenience, we use $\{-1, 1\}$ to model Boolean values both for challenges and responses. This enables us to write the the Arbiter PUF function as the sign of a scalar product and the XOR of Boolean values as the (real) product, which in turn generalizes to a differentiable function. However, it is just a question of convenient presentation, all arguments in this work also apply when using $\{0, 1\}$ to model Boolean values. For conversion, use the group homomorphism $\varphi : \{-1, 1\} \to \{0, 1\}, c \mapsto 1/2 - 1/2c$.

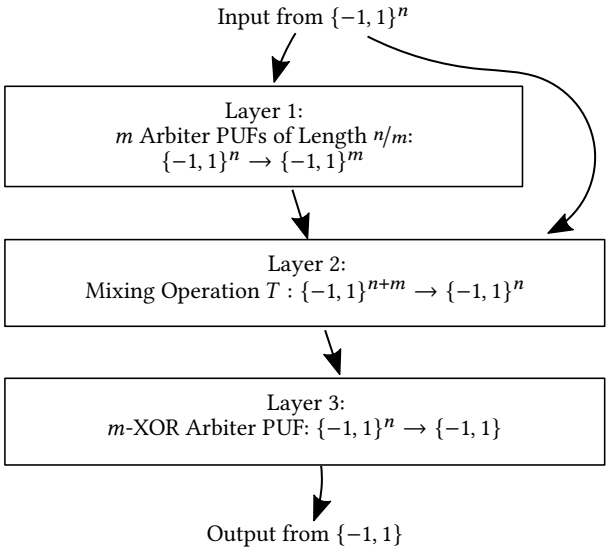

Figure 1: The LP-PUF design, parameterized by the challenge length n and the additional security parameter m.

We present the LP-PUF in the form of three layers, parameterized by a challenge length $n \in \mathbb{N}$ and an additional security parameter $m \in \mathbb{N}$ which must be a divisor of n.

(1) In Layer 1, the LP-PUF generates m *private challenge* bits. To that end, the (*public*) challenge $c = (c_1, \ldots, c_n)$ to the PUF is split into m partitions of equal length by cutting it into m blocks $(c_1, \ldots, c_m), (c_{m+1}, \ldots, c_{2m}), \ldots, (c_{n-m}, \ldots, c_n)$. Each block is fed into an individual Arbiter PUF of challenge length n/m, generating m response bits which are not part of the input, but an instance-specific function of the challenge. Note that the Arbiter PUFs in this layer are chosen deliberately short.
(2) In Layer 2, the the LP-PUF mixes the *private challenge* with the *public challenge* by computing a function $T : \{-1,1\}^{n+m} \to \{-1,1\}^n$, where each output bit of T is the parity (XOR) of exactly one of the public inputs and an individual subset of size $m/2$ of the m private inputs. We chose the involved subsets uniformly at random at design-time. This operation thus does not depend on the given PUF instance, but is a design-constant.
(3) In Layer 3, the n-bit challenge computed in Layer 2 is fed into an ordinary n-bit m-XOR Arbiter PUF, which produces the final output bit of the LP-PUF.

3.1 Motivation

Inspired by the Feed-Forward Arbiter PUF and Interpose PUF, we carefully designed the LP-PUF to use easy-to-model building blocks combined with attacker-unknown outputs (in Layer 1) and attacker-unknown inputs (in Layer 3) to build a composite PUF which is resistant to known modeling attacks.

There are various motivations for the different aspects of our design. To mitigate a splitting attack (originally on the Interpose PUF [22]), we introduced the use of more than one "interpose bit" as well as the mixing operation in Layer 2. This drastically reduces the chance of the attacker to guess the feature vector required to learn the XOR Arbiter PUF of Layer 3. We detail on the mitigation of the splitting attack in Sec. 4.1.

By reducing the attacker-knowledge about the input to Layer 3, the mixing operation of Layer 2 also mitigates the reliability-based attacks [2, 17] on Layer 3. This is detailed in Sec. 4.2.

The use of Arbiter PUFs in our design is to facilitate a CMOS-compatible design, which allows for fabrication of the LP-PUF using standard design processes. It also benefits from literature available on implementation [such as 6] and a well-studied model of its behavior (see 2).

The use of short Arbiter PUFs in Layer 1 is motivated by the hope that short Arbiter PUFs can be implemented such that it generates very reliable responses. In Sec. 4.2, we detail on this. In Sec. 5, we discuss potential problems with this choice with respect to chosen-challenge attacks.

This yields an overall structure that vaguely resembles a substitution-permutation-network, which are used in block cipher design. Specifically, and in contrast to proposals such as the Lightweight Secure PUF and Permutation PUF [10, 20], the LP-PUF employs a scheme where the attacker cannot compute the first or last operation in the network. Furthermore, in an advancement of the Interpose PUF design, by introducing the mixing operation in Layer 2, the LP-PUF combines operations of each low complexity, but from different "realms", albeit limited to only one and a half "rounds".

Alternatives and extensions of these design choices include to use several rounds, i.e. to introduce a second mixing layer, and/or to not use the original challenge input in deeper layers. We did not study these variations in great detail due to concerns with respect to the reliability of implementations. However we believe that the security of the construction would greatly benefit from such modifications.

3.2 Metrics

The metrics in this chapter are fundamental requirements to every strong PUF design. A high *uniqueness* of a PUF design shows that two randomly chosen instances of this design indeed behave differently. If they show correlation or similarity in their behavior, an attacker can use this fact to guess PUF responses of one instance with the assistance of another, unrelated PUF instance. A low *bias* is required to reduce the probability that the attacker guesses correctly by choosing the most likely response to the minimum possible. (For PUFs with more than one response bit, which are not studied in this work, this generalizes to the notion of *min-entropy*.) A proper *bit sensitivity* is needed to prevent attackers from predicting responses when the response to a related challenge is known. Finally, a high *reliability* is of the PUF implementation is needed for the design to be usable, as for low reliability, PUF responses are no longer reproducible, depriving the PUF of its key feature. For formal definitions of the measured quantities, we refer the reader to the definitions given in pypuf [21].

The results shown in this section justify the hope that the LP-PUF can fulfill these requirements. The LP-PUF shows high uniqueness, low bias, bit sensitivity similar to that of an XOR Arbiter PUF, and fair reliability in our simulations.

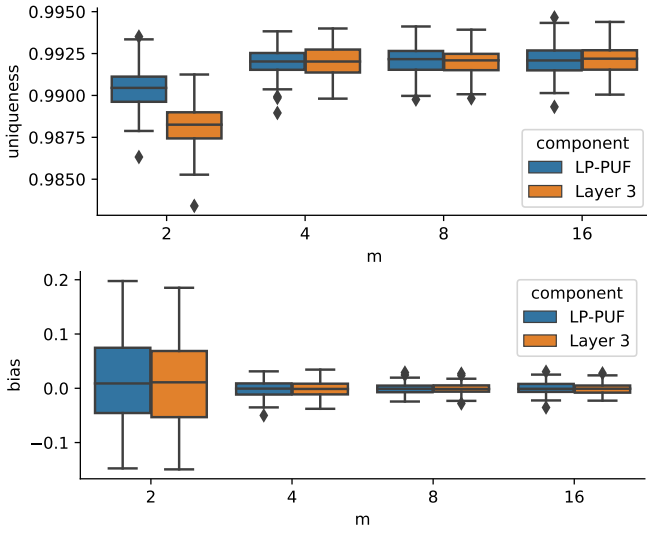

Figure 2: Uniqueness and bias of the proposed LP-PUF, measured in noise-free simulations and for $n = 64$ challenge bits across different values of the m parameter. For comparison, the corresponding metrics are also shown for Layer 3 of the LP-PUF, which consists of a traditional n-bit m-XOR Arbiter PUF.

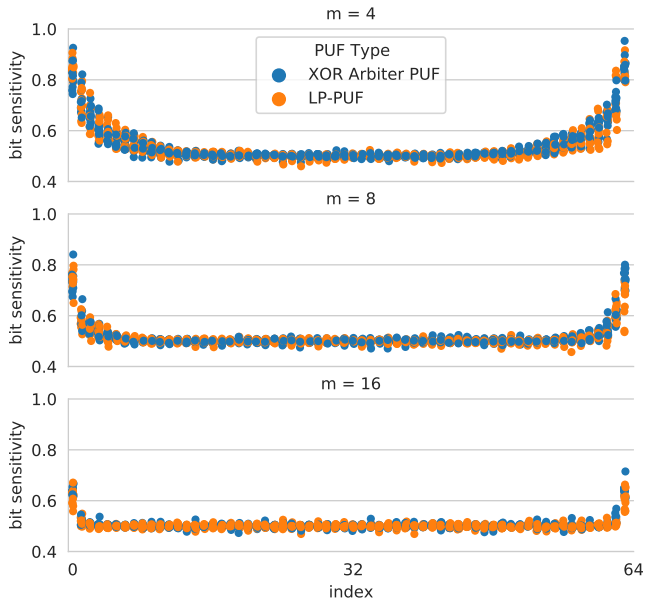

Figure 3: Bit sensitivity values for the LP-PUF and XOR Arbiter PUF for both 64-bit challenges. Values of $1/2$ are ideal.

In Fig. 2, we show the uniqueness and bias values, compared to a baseline given by the XOR Arbiter PUF in Layer 3 of the LP PUF. In all studied cases, the LP-PUF shows the same or better uniqueness and bias distribution as the XOR Arbiter PUF.

In Fig. 3, we show the bit sensitivity for LP-PUF and XOR Arbiter PUFs, which are very similar, and could be improved by adjusting the mixing operation in Layer 2.

The reliability of the LP-PUF must be studied in more detail, as it is crucial for the feasibility of LP-PUF implementations. (It is easy to come up with a PUF design that is resilient to modeling attacks when reliability is not an issue.)

As a design composed of several building blocks, the reliability of the LP-PUF is a function of the reliability of the involved building blocks. Solely composed of Boolean logic, Layer 2 is assumed to be fully reliable. For the XOR Arbiter PUF in Layer 3, commonly believed reliability values can be found in the literature [2, 6, 22] if 64-bit challenges are employed. For the (short) Arbiter PUFs in Layer 1, however, to the extend of our knowledge, no reliability estimate is available in the literature. (There are some arguments to justify an increase in stability for very long Arbiter PUFs [22].) The established noise model used for Arbiter PUFs [6], unfortunately, does not allow reliability predictions for longer or shorter challenges, as it remains unclear how much noise is introduced by the n stages used in the Arbiter PUF and how much noise is due to the one arbiter element. Other factors engineering factors which might change with increasing challenge length are also not taken into account by the commonly used Arbiter PUF noise model. We conclude that the reliability of Arbiter PUFs with challenge lengths other than the usual 64 and 128 bit remains an open research question. In lack of better options, we assume that the reliability of the LP-PUF Layer 1 will be in between 99.8% and 87.7%.

In Fig. 4, we study the reliability of the LP-PUF based on simulations and as a function of the reliability of Layers 1 and 3. For Layer 1, we give the average reliability of the m Arbiter PUFs in use, but remark that there is little variance. For Layer 3, we give the reliability of the single output bit of Layer 3, as measured individually, i.e., with challenges directly applied to Layer 3, disregarding layers 1 and 2.

We conclude that assuming a 96.3% reliability for Layer 1 and a 79% reliability of Layer 3, the LP-PUF is conceivable for at least $m = 8$, as the total reliability in this case is estimated at 73%. While this reliability is within the acceptable range for a basic authentication protocol based on pre-recorded challenges, we remark that an even lower reliability could make the protocol inefficient or shrink the security margin against attackers using models with weak prediction accuracy. Hence, to obtain a definite answer on the feasibility of the LP-PUF design, a study of the reliability of real-world data will be necessary.

4 SECURITY ANALYSIS

4.1 Logistic Regression / Splitting Attack

As the LP-PUF is an extension of the Interpose PUF [12], we first consider an extension of the splitting attack [22]. The splitting attack reduced the security level of an $(1, k)$-Interpose PUF to that of a k-XOR Arbiter PUF and thereby demonstrated that the security advantage of the Interpose PUF is less than what had been previously claimed.

The original splitting attack employs the LR attack [14], which is a well-established and well-studied analysis tool in the field of Arbiter-based PUFs [18, 20, 23]. However, the employed modeling

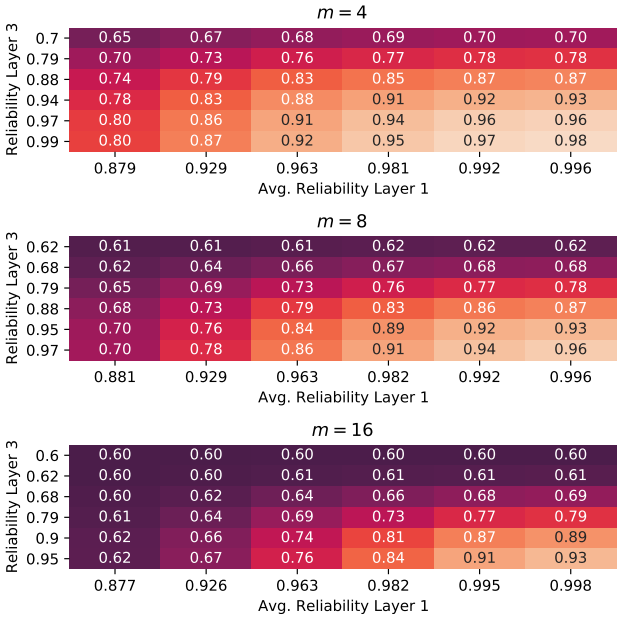

Figure 4: Simulated reliability of the 64-bit LP-PUFs depending on reliabilities of the building blocks.

algorithm of the splitting attack can be swapped out for alternatives, e.g. for a Multilayer Perceptron (MLP) attack [23]. In any case, the splitting attack is based on CRPs collected from the composite PUF under attack, but attacks the building blocks of the PUF separately.

We review the procedure of the splitting attack on the Interpose PUF. It is conducted in two steps which may be repeated to increase the resulting accuracy. The attack is prepared by collecting CRP data of the Interpose PUF, i.e. challenges to the upper layer and responses of the lower layer. The attacker has no knowledge of the *intermediate* challenges, i.e. the responses of the upper layer and challenges applied to the lower layer. Furthermore, an XOR Arbiter PUF model of appropriate size of each the upper and the lower layer of the Interpose PUF are initialized but not yet trained.

(1) **Lower Layer Training.** Using the model of the upper layer, for each challenge of the CRP data, a guess for the output of the upper layer, and thus for the intermediate challenge is generated. With the resulting set of guessed intermediate challenges and recorded responses from the CRP data, the model for the layer lower is trained.
(2) **Upper Layer Training.** Based on the model of the lower layer, the responses of the upper layer are estimated: For each challenge from the available CRP data, for all possible responses of the upper layer, the corresponding intermediate challenges are computed and evaluated on the model of the lower layer; the responses are compared to the recorded responses of the CRP data. If for any given challenge, only one variant matches the recorded CRP data, then it is assumed that the corresponding response of the upper layer is the correct one. With the resulting set of guessed responses of the upper layer and recorded challenges from the CRP data, the upper layer is trained.

Pseudocode for the attack is given by Wisiol et al. [22].

For the analysis of the splitting attack, we analyze the probability that the attacker guesses the *feature vectors* x required for training correctly. In case of the Interpose PUF, the attacker guesses the single response bit of the upper layer, which is directly fed into the middle challenge bit of the lower layer. Due to the nature of the feature vector required for training models of XOR Arbiter PUFs (see Sec. 2), this challenge bit appears in the first $n/2 + 1$ features x_i for training. So, while the attacker has to guess many features, they are highly correlated. The probability to guess an entire n-feature vector correctly is 50%. The probability to guess individual feature bits correctly is approx. 75% on average (50% for the feature bits including the interpose bit, 100% for the feature bits not including it).

We note that it is not sufficient to extend the Interpose PUF with a number of l interpose bits to mitigate this attack. One could think that the guessing probability of the attacker is degraded to 2^{-l}. We show that this is not the case. If there were two interpose bits c_i and c'_i in the middle of the lower layer, then the first $n/2$ features of the lower layer all include the XOR of c_i and c'_i — a value that the attacker can still guess with probability 50%; so not much is gained in this setting. Similar arguments apply for any number of interpose bits. Distributing these interpose bits across the challenge of the lower layer, i.e. not only interposing in the middle, opens up other attack surfaces, as outlined by the original authors [12].

In the LP-PUF design, the mixing operation in Layer 2 is aimed at removing correlations from the feature bits to minimize the guessing advantage. The goal is that the attacker can guess feature bits correctly only with probability 50%, and feature vectors only with probability approximately 2^{-m}. We confirmed this in our simulations. The measured guessing probabilities for feature bits were at 53% and 50% for $m = 4$ and $m \geq 8$, respectively. The measured probabilities for guessing feature vectors correctly were 13% for $m = 4$ and 0.7% for $m = 8$ and $< 1/10{,}000$ for $m = 16$. In this way, the LP-PUF provides a way to introduce almost m bit of entropy in the challenges to Layer 3.

We did not study the guessing probability for each feature bit separately, but remark that if the attacker is able to guess single feature bits with higher probability, or finds correlations between the feature bits, then using this knowledge may enable the attacker to increase their guessing probability.

The reduced guessing probability for the features to the model of Layer 3 constitutes itself in a significant increase of required CRPs for successfully training a model. In the case of $m = 4$, the (adapted) splitting attack requires approx. 500,000 CPRs to train a high-accuracy model, compared to 60,000 CRPs for the (m, m)-Interpose PUF and 30,000 CRPs for the m-XOR Arbiter PUF. We believe that the reason that the training succeeds at all is that the probability to guess feature vectors correctly is still at 13%, which means that guessing errors can be averaged out over large sets of CRPs. However, as the guessing advantage of the attacker declines exponentially with m, we also expect an exponential increase of the required CRPs in m. Unfortunately, we also have seen in Sec. 4.2

that the reliability of the LP-PUF suffers greatly from an increase of m.

Nevertheless, there is hope that the LP-PUF could find a sweet spot that mitigates the splitting attack, while at the same time provides sufficiently reliable responses, e.g. for $m = 8$ or $m = 16$. Note that while the XOR Arbiter PUF also suffers from decreasing reliability in it's security parameter, the known reliability-attacks [2, 17] cannot be mitigated by increasing the XOR Arbiter PUF size.

4.2 Reliability Attack

In the past, reliability attacks have targeted Arbiter PUFs [6], XOR Arbiter PUFs [2, 17], and the Interpose PUFs [17]. The observation fundamental to all of these attacks is that the reliability of an Arbiter PUFs response to a given challenge is a function of the delay difference corresponding to this challenge. The smaller the absolute value of the delay difference, the higher the unreliability. This means that Arbiter PUFs can be identified not only by their response behavior, but also by their reliability. In the case of single Arbiter PUFs, it is sufficient so obtain an approximate solution to a system of linear equations. In the case of XOR Arbiter PUFs, evolution strategies or gradient-decent machine learning algorithms can be employed to find high accuracy models. These approaches are based on the Pearson correlation of the measured reliability of target PUF and model; the higher the correlation, the more accurate the model will be.

In principle, all Arbiter PUFs in a composite design can be target of a reliability-based attack if the attacker can correlate any measurable reliability to the reliability of the target Arbiter PUF. In case of the XOR Arbiter PUF, it was found that the XOR Arbiter PUF's output reliability is correlated with the reliability of the individual Arbiter PUFs [2]. Similarly, the output of the Interpose PUF has reliability correlation with the lower layer [17].

To analyze the vulnerability of the LP-PUF towards reliability-based attacks, we thus study the reliability correlation of Layer 1 and Layer 3 with the attacker-measurable reliability of responses at the LP-PUF output. Based on our simulations, we could not find significant correlations of output reliability and Layer 1, as shown in Fig. 5. Instead, the correlation shows values that are also measured when compared to an entirely unrelated PUF. (The increase with m can be explained as we show the *maximum* correlation to any of the m individual reliability vectors of Layer 1.) This result is expected and applies similarly to the Interpose PUF.

The reliability correlation of Layer 3 with the output of the LP-PUF is high for small values of m and indicates that an attack for these values of m will be possible. However, as we increase m, the correlation vanishes, with $m = 8$ and $m = 16$ hardly showing any difference when compared to the correlation with an unrelated PUF instance. We conclude that increasing m will mitigate current versions of the reliability attack.

We note that the attack is *not* mitigated by removing unreliable challenges from Layer 3, or by improving the reliability of the implementation. (Due to the nature of the Arbiter PUF, we believe such an approach to be not promising.) Instead, by decreasing attacker knowledge of the challenge applied to Layer 3, we remove the attackers ability to meaningfully correlate the measured reliability, which prevents an application of evolution-strategies or gradient descent machine learning algorithms. Still, the theoretical analysis of the reliability-based attacks is quite thin, and we are afraid that there could be a way to adapt the attack to account for the mixing operation in Layer 2, especially since the attacker can choose the individual inputs to the PUFs in Layer 1.

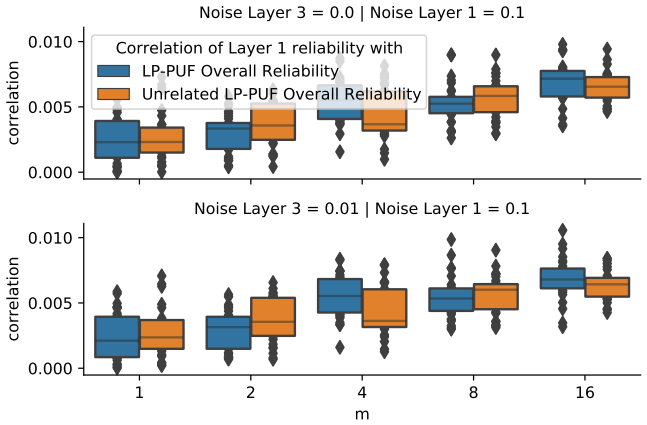

Figure 5: The correlation of the attacker-observable overall reliability of a given $n = 64$ challenge bit LP-PUF with the reliability of Layer 1. To account for all response bits of Layer 1, the maximum correlation in each instance is taken. If a large correlation of Layer 1 reliability and LP-PUF reliability can be established, an attacker could attempt a reliability-based modeling attack on Layer 1.

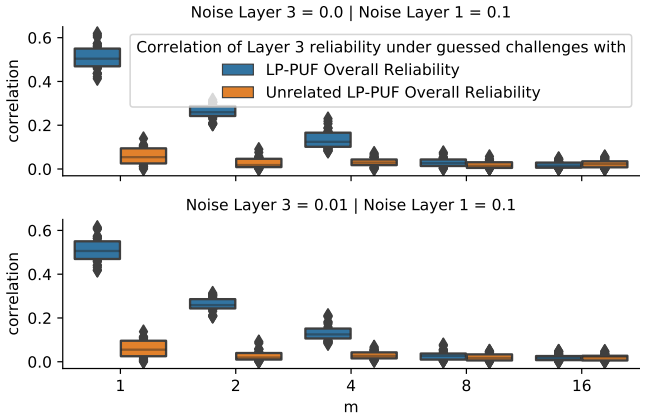

Figure 6: The correlation of the attacker-observable overall reliability of a given $n = 64$ challenge bit m-LP-PUF with the reliability of it's Layer 3 under attacker-guessed challenges. High correlations pave the way to conduct a reliability-based attack on Layer 3 (as done on the Interpose PUF Tobisch et al. [17]). The absence of high correlation for the LP-PUF is *not* caused by increasing the reliability of Layer 3, but by reducing the ability of the attacker to guess Layer 3 input bits.

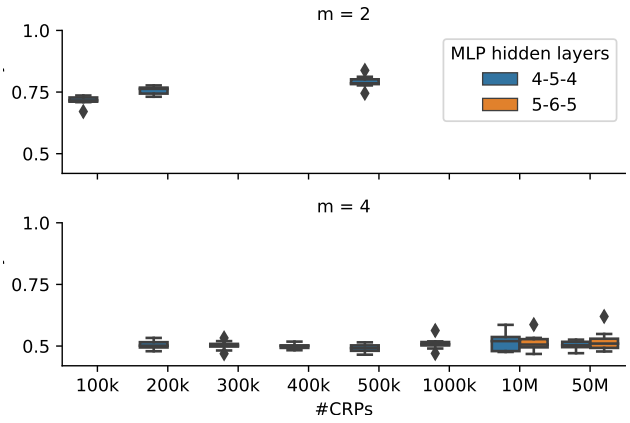

Figure 7: Prediction accuracy of the MLP attack by Wisiol et al. [23] run on 64-bit m-LP-PUF. The neural network, training parameters, and – perhaps most importantly – the features were not modified from the original version. Due to the attacker-unknown challenge to Layer 3 of the LP-PUF, it is unclear how the attacker should adjust the features.

4.3 MLP Attack

The latest addition to the toolbox in strong PUF security analysis are attacks based on multilayer perceptrons (MLP) [1, 11, 15, 22, 23]. These attacks have the advantage over the LR attack that no exact model is required, while at the same time, the required CRPs for modeling XOR Arbiter PUF and Interpose PUF is decreased. On the downside, even a successful modeling attack will not allow much insight in the inner workings of the attack, as it cannot be expected that the trained weights of the model can be interpreted. As such, it is well-suited for quick and preliminary analysis of novel PUF designs such as the LP-PUF.

We first consider the MLP attack on the full 64-bit LP-PUF, i.e. without applying any technique similar to the splitting attack discussed in Sec. 4.1. Similar experiments have shown that straightforward extensions of the Interpose PUF can be attacked [22].

While for $m = 2$, we were able to obtain models with an accuracy around 80% (reminiscent of the first step of the splitting attack), already for $m = 4$ we did not achieve any significant success, even when using 50 million CPRs. (The 64-bit 4-XOR Arbiter PUF requires merely 150,000 CRPs [23].) We can conclude that either we chose inappropriate network parameters (we tried networks which have been shown to be able to attack 4-XOR Arbiter PUFs and 5-XOR Arbiter PUFs), or that the MLP attack might not be able to infer the features required to model Layer 3. As evidence for the latter case, it was reported that MLP is also unable to train a model given the *challenges*, instead of the Arbiter PUF *features* [15].

An MLP-based splitting attack on the LP-PUF is also conceivable, as it has been demonstrated against the Interpose PUF [23]. However, it faces the same difficulties in guessing the feature bits for the model of Layer 3, and is hence largely covered by our arguments in Sec. 4.1. Given that the MLP attack has been shown to reduce the number of required CRPs [23], this may also apply to the splitting attack discussed in this paper.

5 LIMITATIONS AND FUTURE WORK

The design of the LP-PUF and the results presented on its metrics and security properties aim at making the case that the LP-PUF and related designs are worth to be studied in more detail; the analysis presented here is by no means exhaustive.

To conduct a more rigorous security analysis, a formal model of the LP-PUF, based on the additive delay model, should be derived. Due to the verbose definition of the mixing operation in Layer 2, the use of a computer algebra system such as sage is necessary. To the extend of our knowledge, no such analysis has ever been done on a PUF. Such a formal model will serve as a basis for a formalization of some of the arguments made above, e.g. for the decreasing chances of the attacker to guess the feature bits when m is increased. It could also allow for a more rigorous choice of the mixing operation, rather than just using randomly chosen subsets and allow for an improvement in the bit sensitivity of the LP-PUF.

Likewise, a model for the reliability of the LP-PUF needs to be developed, to make sure that attacks based on the correlation of reliability behavior cannot be adjusted to somehow work around the mixing operation in Layer 2 (see Sec. 4.2).

To increase the trustworthiness of our failed modeling attempts using machine learning algorithms, it will also be necessary to revisit the chosen hyperparameters and argue in more detail that also hyperparameter optimization will not enable the attacker to obtain a model of the LP-PUF or parts thereof.

As mentioned above, it is uncertain if Arbiter PUFs of short length can be reliably build; the commonly used noise model of the Arbiter PUF is ill-suited to make a prediction. This can be clarified by studying the behavior of short Arbiter PUFs in real hardware or by replacing them with an alternative solution.

Adjusting the design of Layer 1 may also be indicated to defend against attackers that choose challenges, instead of using challenges that are chosen uniform at random. Given the standard attacker model for PUFs, where the attacker gets physical access to the PUF for a limited amount of time, this is certainly a concern for the LP-PUF. Alternatives to Layer 1 could try to limit the freedom of the attacker in choosing which challenges are applied to which Arbiter PUF; at the very least, they should remove known weaknesses in the bit sensitivity of the Arbiter PUFs in Layer 1.

Not included in this work is an analysis of the LP-PUF with respect to its PAC learnability. While here, we cannot expect to obtain a negative result, the known proofs of learnability should be applied to the new setting to verify that no known attack applies. As a first step, the PUF-G framework [4] and the PUFMeter [7] should be applied to the LP-PUF.

Finally, even though we argue that a PUF design needs to withstand all scrutiny in an idealized form, i.e. in simulation, eventually also an implementation needs to be analyzed with the same precision. To that end, FPGA or ASIC data has to be collected. Due to the highly specific nature of the mixing operation in Layer 2 which generates the challenge to Layer 3, none of the publicly available Arbiter PUF measurement data is suited for this task.

To facilitate future work, we publish the simulation of the LP-PUF as well as all analyses of this work at https://github.com/nils-wisiol/LP-PUF/ under a free license.

6 RELATED WORK

To alleviate the machine learning attacks on (XOR) Arbiter PUFs, several works suggest to preprocess the challenge before it is applied to the hardware. Wisiol et al. [20] report that, with respect to the LR attack, the number of required CRPs may increase when such preprocessing is applied and propose the Permutation PUF, which employs an easy-to-implement preprocessing. While the complexity is increased, their results also show that the attack remains possible even if a pseudorandom number generator is employed as preprocessing method. Zhuang et al. [26] report that the MLP attack can be mitigated by applying challenge preprocessing, a finding that may relate to above-mentioned hypothesis that MLP is unable to compute the features required for modeling [15].

In a similar branch of work, Delvaux demonstrated that ad-hoc solutions to challenge preprocessing or combination of different PUF types into one can easily fall victim to specialized attacks [5].

Some works suggest to select only a subset of all challenges of a given PUF [19, 24], but such selections may bias the PUF responses or expose information that can help to the attacker.

Other composite designs of Arbiter PUFs include the IPN [9], where the composition of PUFs is changed "from time to time" before an attacker can collect enough data to train a model.

Some works moving to new implementations and avoid the Arbiter PUF entirely. An alternative CMOS-compatible PUF build from Strong Subthreshold Current Arrays shows good metrics [25], but the security analysis is based on too few CRPs to allow final conclusions. Charlot et al. [3] use Hybrid Boolean Networks as strong PUF and demonstrate promising security properties by showing that a modeling of their design by the use of PUFmeter [7] failed. However, a more detailed analysis using MLP or physically inspired modeled was not done.

7 ACKNOWLEDGEMENTS

We would like to thank Stefan Katzenbeisser and Phuong Ha Nguyen for insightful comments on the weak PUF in Layer 1 used in an earlier version of LP-PUF, as well as Roel Maes for inspiring further experiments on the noise resilience of the LR attack. We also thank Johannes Tobisch for pointing out weaknesses in an earlier version of the security analysis with regard to reliability-based attacks.

REFERENCES

[1] A. O. Aseeri, Y. Zhuang, and M. S. Alkatheiri. 2018. A Machine Learning-Based Security Vulnerability Study on XOR PUFs for Resource-Constraint Internet of Things. In *2018 IEEE International Congress on Internet of Things (ICIOT)*. 49–56.

[2] Georg T. Becker. 2015. The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs. In *Cryptographic Hardware and Embedded Systems – CHES 2015 (Lecture Notes in Computer Science)*, Tim Güneysu and Helena Handschuh (Eds.). Springer Berlin Heidelberg, 535–555.

[3] N. Charlot, D. Canaday, A. Pomerance, and D. J. Gauthier. 2021. Hybrid Boolean Networks as Physically Unclonable Functions. *IEEE Access* 9 (2021), 44855–44867.

[4] D. Chatterjee, D. Mukhopadhyay, and A. Hazra. 2020. PUF-G: A CAD Framework for Automated Assessment of Provable Learnability from Formal PUF Representations. In *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. 1–9.

[5] J. Delvaux. 2019. Machine-Learning Attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUF–FSMs. *IEEE Transactions on Information Forensics and Security* 14, 8 (Aug. 2019), 2043–2058.

[6] Jeroen Delvaux and Ingrid Verbauwhede. 2013. Side Channel Modeling Attacks on 65nm Arbiter PUFs Exploiting CMOS Device Noise. In *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium On*. IEEE, 137–142.

[7] Fatemeh Ganji, Domenic Forte, and Jean-Pierre Seifert. 2019. PUFmeter a Property Testing Tool for Assessing the Robustness of Physically Unclonable Functions to Machine Learning Attacks. *IEEE Access* 7 (2019), 122513–122521. https://ieeexplore.ieee.org/document/8819883/

[8] Blaise Gassend, Daihyun Lim, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. 2004. Identification and Authentication of Integrated Circuits. *Concurrency and Computation: Practice and Experience* 16, 11 (Sept. 2004), 1077–1098. https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.805

[9] Hongxiang Gu and Miodrag Potkonjak. 2021. Evolution-Strategies-Driven Optimization on Secure and Reconfigurable Interconnection PUF Networks. *Electronics* 10, 5 (Jan. 2021), 537. https://www.mdpi.com/2079-9292/10/5/537

[10] Mehrdad Majzoobi, Farinaz Koushanfar, and Miodrag Potkonjak. 2008. Lightweight Secure PUFs. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design (ICCAD '08)*. IEEE Press, Piscataway, NJ, USA, 670–673. http://dl.acm.org/citation.cfm?id=1509456.1509603

[11] Khalid T. Mursi, Bipana Thapaliya, Yu Zhuang, Ahmad O. Aseeri, and Mohammed Saeed Alkatheiri. 2020. A Fast Deep Learning Method for Security Vulnerability Study of XOR PUFs. *Electronics* 9, 10 (Oct. 2020), 1715. https://www.mdpi.com/2079-9292/9/10/1715

[12] Phuong Ha Nguyen, Durga Prasad Sahoo, Chenglu Jin, Kaleel Mahmood, Ulrich Rührmair, and Marten van Dijk. 2019. The Interpose PUF: Secure PUF Design against State-of-the-Art Machine Learning Attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (Aug. 2019), 243–290. https://tches.iacr.org/index.php/TCHES/article/view/8351

[13] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. 2002. Physical One-Way Functions. *Science* 297, 5589 (Sept. 2002), 2026–2030. http://science.sciencemag.org/content/297/5589/2026

[14] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. 2010. Modeling Attacks on Physical Unclonable Functions. In *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS '10)*. ACM, New York, NY, USA, 237–249. http://doi.acm.org/10.1145/1866307.1866335

[15] Pranesh Santikellur, Aritra Bhattacharyay, and Rajat Subhra Chakraborty. 2019. Deep Learning Based Model Building Attacks on Arbiter PUF Compositions. (2019), 10.

[16] G. Edward Suh and Srinivas Devadas. 2007. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *Proceedings of the 44th Annual Design Automation Conference (DAC '07)*. ACM, New York, NY, USA, 9–14. http://doi.acm.org/10.1145/1278480.1278484

[17] Johannes Tobisch, Anita Aghaie, and Georg T. Becker. 2021. Combining Optimization Objectives: New Modeling Attacks on Strong PUFs. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (Feb. 2021), 357–389. https://tches.iacr.org/index.php/TCHES/article/view/8798

[18] Johannes Tobisch and Georg T. Becker. 2015. On the Scaling of Machine Learning Attacks on PUFs with Application to Noise Bifurcation. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer, 17–31.

[19] S.-J. Wang, Y.-S. Chen, and K. S.-M. Li. 2021. Modeling Attack Resistant PUFs Based on Adversarial Attack against Machine Learning. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* (2021), 1–1.

[20] Nils Wisiol, Georg T. Becker, Marian Margraf, Tudor A. A. Soroceanu, Johannes Tobisch, and Benjamin Zengin. 2020. Breaking the Lightweight Secure PUF: Understanding the Relation of Input Transformations and Machine Learning Resistance. In *Smart Card Research and Advanced Applications (Lecture Notes in Computer Science)*, Sonia Belaïd and Tim Güneysu (Eds.). Springer International Publishing, Cham, 40–54.

[21] Nils Wisiol, Gräbnitz, Christoph, Mühl, Christopher, Zengin, Benjamin, Soroceanu, Tudor, Pirnay, Niklas, and Mursi, Khalid T. 2021. Pypuf. Zenodo. https://zenodo.org/record/3901410

[22] Nils Wisiol, Christopher Mühl, Niklas Pirnay, Phuong Ha Nguyen, Marian Margraf, Jean-Pierre Seifert, Marten van Dijk, and Ulrich Rührmair. 2020. Splitting the Interpose PUF: A Novel Modeling Attack Strategy. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (June 2020), 97–120. https://tches.iacr.org/index.php/TCHES/article/view/8584

[23] Nils Wisiol, Khalid T. Mursi, Jean-Pierre Seifert, and Yu Zhuang. 2021. *Neural-Network-Based Modeling Attacks on XOR Arbiter PUFs Revisited*. Technical Report 555. https://eprint.iacr.org/2021/555

[24] Chen Zhou, Keshab K. Parhi, and Chris H. Kim. 2017. Secure and Reliable XOR Arbiter PUF Design: An Experimental Study Based on 1 Trillion Challenge Response Pair Measurements. In *Proceedings of the 54th Annual Design Automation Conference 2017 on - DAC '17*. ACM Press, Austin, TX, USA, 1–6. http://dl.acm.org/citation.cfm?doid=3061639.3062315

[25] Haoyu Zhuang, Xiaodan Xi, Nan Sun, and Michael Orshansky. 2020. A Strong Subthreshold Current Array PUF Resilient to Machine Learning Attacks. *IEEE Transactions on Circuits and Systems I: Regular Papers* 67, 1 (Jan. 2020), 135–144.

[26] Yu Zhuang, Khalid T. Mursi, and Li Gaoxiang. 2021. A Challenge Obfuscating Interface for Arbiter PUF Variants against Machine Learning Attacks. *arXiv:2103.12935 [cs]* (March 2021). arXiv:cs/2103.12935 http://arxiv.org/abs/2103.12935