Feed-Forward XOR PUFs: Reliability and Attack-Resistance Analysis

S. V. Sandeep Avvaru and Keshab K. Parhi

Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA
avvar002@umn.edu, parhi@umn.edu

ABSTRACT

Physical unclonable functions (PUFs) can be used to generate unique signatures of integrated circuit (IC) chips. XOR arbiter PUFs (XOR PUFs), that typically contain multiple standard arbiter PUFs as their components, are more secure than standard arbiter PUFs. This paper proposes design of feed-forward XOR PUFs (FFXOR PUFs) where each component PUF is a feed-forward arbiter PUF (FF PUF). Arbiter PUFs suffer from two main drawbacks: vulnerability to modeling attacks and degraded reliability. It is shown that FFXOR PUFs cannot be accurately modeled if the number of component PUFs is more than 5 or 6. We also state that the number of machine learning runs required to learn a model using evolutionary strategies increases by a factor of $N^2/2$ if N -stage FF PUFs with one loop are used as components instead of standard arbiter PUFs. In general, for FF PUFs with k loops (one intermediate arbiter), it would increase by a factor of $\binom{N}{k+1}$. We also show that the use of a *thresholding strategy* can increase the reliability of FFXOR PUFs by about 30% for a 15% noise level.

KEYWORDS

Hardware Security; XOR PUF; Feed-Forward PUF; Feed-Forward XOR PUF; Artificial Neural Network; Reliability; Security

ACM Reference Format:

S. V. Sandeep Avvaru and Keshab K. Parhi. 2019. Feed-Forward XOR PUFs: Reliability and Attack-Resistance Analysis. In *Proceedings of Great Lakes Symposium on VLSI 2019 (GLSVLSI '19)*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3299874.3318019>

1 INTRODUCTION

Physical unclonable functions (PUFs) are integrated circuits used to securely authenticate devices or generate cryptographic keys without the involvement of non-volatile memories. PUFs harvest natural randomness in manufacturing process variations to create unique signatures for every integrated circuit (IC) chip. Ever since they were first introduced in the early 2000s, numerous PUF structures have been proposed [5]. Arbiter PUFs have been shown to be vulnerable to modeling attacks [2, 10]. Despite this limitation, they are used as building blocks in variants like XOR arbiter PUFs, lightweight secure PUFs, and composite PUFs because of their low area

overhead and availability of a large set of *challenge-response pairs* (CRPs). Two of the biggest issues in this context are low reliability [7] and vulnerability to attacks [10]. Reliability is a measure of the variation of the response signature, for a set of challenges, as the noise is varied. FF PUFs and XOR PUFs [11] have been suggested as more secure alternatives to arbiter PUFs. But the added security comes at the cost of reduced reliability. One of the goals of this paper is to evaluate XOR PUFs in terms of their resilience against attacks (security) and against noise (reliability).

Authors in [3, 6, 10, 12] have presented methods to attack XOR PUFs using machine learning. One major issue with XOR PUFs is that their models can be expressed as linear decision boundaries [10] which makes it easier to learn an accurate model. In this paper, we assume a black-box approach to modeling XOR PUFs. Additionally, the existing studies [3, 4, 6, 10, 12] are only limited to XOR PUFs with standard arbiter PUFs as elements, called *standard XOR PUFs*. We simulate and analyze XOR PUFs with FF PUFs as elements, called *feed-forward XOR PUFs* (FFXOR PUFs), and assess their security and reliability. Since FF PUFs are inherently nonlinear, a linear approximation similar to the cases of standard XOR PUFs or Interpose PUFs is not applicable. Artificial neural networks (ANNs) are used to learn the models and the amount of data an attacker requires to learn an accurate predictive model is estimated. It is important to note that our work is limited to modeling attacks and does not focus on side-channel attacks. To the best of the authors' knowledge, no prior paper has either proposed or analyzed the security and reliability aspects of FFXOR PUFs.

The rest of the paper is organized as follows. The results of security analysis are presented and discussed in Section II. Reliability analysis is presented in Section III. Section IV concludes the paper.

2 SECURITY ANALYSIS

In this work, Standard arbiter PUFs and FF PUFs, used as components for XOR PUFs, are simulated using additive linear delay model. For each stage, the delay difference is sampled from an independent identical Gaussian distribution with zero mean and unit variance ($\mathcal{N}(0, 1)$) [3, 9, 10]. The intermediate and final bias values are sampled from $\mathcal{N}(0.1, 1)$. The non-zero mean of the bias accounts for the skew in total path delay due to the arbiter and other factors like the cumulative effect of noise and bias at each stage [2].

For security analysis, XOR PUFs and FFXOR PUFs with number of levels $l = 2$ to $l = 8$ are simulated and their neural network models are trained. 32-stage MUX PUFs were used as the components for each XOR PUF. FF PUFs containing a feed-forward loop whose intermediate response is computed at the 15th stage (N_1) and is fed into the 25th stage (N_2) were used as components of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GLSVLSI '19, May 9–11, 2019, Tysons Corner, VA, USA

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6252-8/19/05...\$15.00

<https://doi.org/10.1145/3299874.3318019>

Figure 1: Prediction accuracy as function of training data size and number of XOR levels. The number levels is varied from $l = 2$ to $l = 8$. The FF PUFs contain one loop from $N_1 = 15$ to $N_2 = 25$. The number of stages (N) is 32.

FFXOR PUFs. XOR PUFs with $l \geq 8$ were found to be attack resistant. For each of these configurations, data for 10 PUF instances were generated. For each instance, 1.2 million challenge-response pairs (CRPs) are extracted. 1 million CRPs are used to train the model and 100,000 CRPs are used for testing and validation each. For every PUF instance, several models were trained by increasing the size of the training set from 100k to 1M and the prediction accuracy on an independent test-set is reported. The randomly chosen test challenges are kept consistent across all the models to have an unbiased comparison. For configurations with standard arbiter PUFs as components, multilayer perceptrons with 2 hidden layers were implemented with 60 and 30 neurons in each hidden layer, respectively. Since ANNs with 2 hidden layers were unable to accurately model FFXOR PUFs, 3 hidden layer structures with 120, 30 and 15 neurons, respectively, were trained. The training and testing errors are compared for each model to make sure that there is no over-training.

2.1 Results and Discussion

Prediction accuracies of standard arbiter XOR PUFs were evaluated by increasing the size of the training set from 100,000 CRPs to 1 million CRPs. Figure 1(a) shows how prediction accuracies of attack models vary with respect to training size for FFXOR PUFs. The number of levels is varied from $l = 2$ to $l = 8$. As expected, the prediction accuracy increases with increase in the training size and smaller XOR PUFs are easier to model as compared to higher number of XOR inputs (say $l = 7, 8$). Each accuracy value is the median of 10 PUF instances. PUFs with $l = 2, 3, 4$ achieve more than 90% accuracy while the models for PUFs with $l \geq 5$ are not as accurate. The maximum accuracy values attained using 1 million CRPs as a function of number of component PUFs in the circuit are presented in Figure 1(b). The model accuracies of standard and feed-forward XOR PUFs can be observed. Each value is presented in the form of a box-plot of 10 instances. It can be observed that FFXOR PUFs require significantly more resources to attack as compared to standard XOR PUFs. This follows from the fact that the number of ANN parameters required to train standard arbiter PUFs is $33 \times 60 \times 30 \times 1 = 59,400$ and for FF PUFs, it is $33 \times 120 \times 30 \times 15 \times 1 = 1,782,000$. Note that the input layer has 33

parameters, including the bias, since the input size is 32 bits. This costs more computational resources and processing time for the attacker. This difference is expected to be more significant when dealing with XOR PUFs with more than 128 or 256 bit-challenges. The results in Figure 1(b) also show that the standard XOR PUFs studied in this paper can be trained with more than 95% prediction accuracy up to 7 levels while FFXOR PUFs with 5 levels only attain a maximum prediction accuracy of 80% and the models for more than 5 levels are just as good as random guessing (50% accuracy). These observations offer an important insight that more secure XOR PUFs can be designed with much less area overhead by replacing standard arbiter PUFs with FF PUFs as elements.

Logistic regression based attack strategies as shown in [10] are not valid for FFXOR PUFs since the component PUFs are not linear. Even though we know that alternatives such as evolutionary strategies or multi-layer perceptrons can be used to train FF PUFs, it has been shown that training FF PUFs is much harder compared to training standard arbiter PUFs [1, 10]. For example, a 128-bit standard arbiter PUF requires 2.10 seconds while a FF PUF requires 3:15 hours using evolutionary strategy [10]. In recent work by Becker, it has been shown that XOR PUFs can be attacked by using evolutionary strategies [3]. However, this method may not be applicable to attack FFXOR PUFs unless the internal arbiter output locations are known to the attacker. The attacker could potentially examine all possible FF loop positions and identify the design via trial and error. But this requires a significant increase in the attack time. For FFXOR PUFs composed of N -stage FF PUFs as components, there are $\frac{N(N-1)}{2}$ possible FF loop placements for *one* loop. Therefore, the number of machine learning runs increases by 496 times for 32 bits and 2016 times for 64 bits. In general, for FF PUFs with one intermediate arbiter and K loops, it would increase by a factor of $\binom{N}{K+1}$. For a 6-loop FF PUF, this number increases to 621 million.

3 RELIABILITY ANALYSIS

3.1 Definition of Reliability

Depending on the challenge bits, the path delays of the two paths of an arbiter PUF can be comparable thereby causing the arbiter to enter a meta-stable state. Under these circumstances, small variations in source voltage, ambient temperature or other environmental and measurement noise can affect the response. The *reliability* of a PUF under noise is mathematically defined in terms of its intra-chip variation (or intra-chip Hamming distance) [9]. Assume that R is an n -bit response vector used as a reference and the noisy response-vectors are denoted by R' . An empirical estimate of intra-chip variation P_{intra} is given by,

$$P_{intra} = \frac{1}{m} \sum_{i=1}^m \frac{HD(R, R'_i)}{n} \quad (1)$$

where m is the number of noisy responses. Reliability can be computed as $(1 - P_{intra}) \times 100\%$.

For reliability analysis, noisy responses were generated by adding Gaussian noise with varying standard deviation to the delay difference at each stage. As a result, the distribution is modified to $\Delta_i \sim \mathcal{N}(0, 1) + \mathcal{N}(0, (\sigma_n)^2)$. Here, σ_n denotes the standard deviation of the noise. Once again, XOR PUFs with 2 levels to 8 levels were evaluated for the reliability analysis. To emulate the effect of

Figure 2: Reliability vs. number of levels for 32-stage standard and FFXOR PUFs.

environmental and measurement noise, Gaussian noise with zero mean and a standard deviation corresponding to 5%, 10% and 15% noise levels is added to the delay difference at each stage and to the arbiter delay [9]. *Noise level* is defined as the ratio of the standard deviation of the noise to the standard deviation of delay differences at each stage. For a given design configuration, 100 PUF instances were generated. 1000 randomly chosen unique challenges are used to extract a 1000-bit response-vector for each instance. 100 noisy response vectors are generated for each of the 100 PUF instances for the purpose of computing reliability.

3.2 FFXOR PUFs with Single-Loop FF PUFs

FF PUFs that were used as the components for the XOR PUFs consist of a FF loop placed between stages $N_1 = 10$ and $N_2 = 20$. XOR PUFs with number of XORs (l) varying from 2 to 8 are studied. The change in reliability with increase in the number of XOR levels is shown in Figure 2. It can be observed that the reliability decreases as the number of levels increases and that FFXOR PUFs have less reliability compared to standard XOR PUFs of the same size. Moreover, the reliability values of FFXOR PUFs drop at a slightly higher rate with respect to the value of l . These values can be used to extrapolate the trend to estimate how reliability scales with the increase in the number of arbiter PUFs used. This demonstrates that reliability of XOR PUFs can be as low as 60% (for $l = 8$). We considered 64-bit feed-forward PUFs ($N = 64$) with the intermediate output measured at the 10th stage ($N_1 = 10$) and feeding into the 40th stage ($N_2 = 40$) and verified that they have similar reliability to 32-stage PUFs shown in Figure 2. It has been shown in [9] that reliability of an N -stage FF PUF depends on $\arctan(\sqrt{\frac{N_2-1}{N-N_2+1}})$ which is equal to 0.88 and 0.895 for the 32-bit and the 64-bit FF PUFs, respectively.

3.3 Soft-Response Thresholding

For a given challenge, *soft-response* is defined as the probability that the response is 1 ($Pr(R = 1)$) under environmental variations [13]. The soft-responses were computed by applying the same challenge 100 times under the presence of noise. As a convention, we use a 90% threshold to measure the consistency of a response. That is, if the response is 0 or 1 in 90% of the cases, it is considered a *stable response*. Otherwise, it is referred to as an *unstable response*. A fundamental limitation of XOR PUFs is that a large portion of the responses could be unstable, resulting in low reliability as illustrated

in Figure 2. For a FFXOR PUF with 8 levels, it was observed that 25.9% of the responses are stable.

We know that soft-responses can be computed by repetitive measurement of responses. Soft-responses of the 8-level FFXOR PUF were computed by assigning each challenge 100 times. A subset of challenges were then chosen by thresholding the soft-responses using 90-10 thresholding. The challenges that generate stable responses are referred to as *stable challenges*. Instead of using a set of 1000 randomly chosen challenges, 1000 stable challenges were used to compute the reliability. Table 1 shows that reliability of XOR PUFs during authentication can be significantly increased by thresholding the soft-responses and identifying stable challenges. For example, the reliability of an 8-level FFXOR PUF (with one loop) can be increased from 58.9% to 91.3% (assuming 15% noise).

3.4 FFXOR PUFs with Multi-Loop FF PUFs

As a counter-measure to the lower reliability of FF PUFs compared to standard arbiter PUFs, a *modified FF PUF* (MFF PUF) structure was proposed in [9]. The idea is to have the intermediate arbiter output feeding into two consecutive stages instead of one. FFXOR PUFs with MFF PUFs as the components were simulated and their reliability was computed. Furthermore, we extend the analysis to FFXOR PUFs consisting of FF PUFs with multiple loops. Results of the reliability analysis of 8-level FFXOR PUFs up to 8 loops is presented in Table 1. The proportion of stable challenges and the reliability values computed before and after thresholding are shown for different noise levels. These values were computed using 90% threshold and each value is the median of 100 PUF instances. The results show that MFF PUFs can be used to attain reliability close to standard XOR PUFs. In case of PUFs with one intermediate arbiter, having 1, 3 or 5 loops results in less reliability compared to having even number of loops. The reliability and the proportion of stable challenges can be substantially low, 53% (approx.) and 1-3%, respectively, when multiple intermediate arbiters are used. Two such examples can be noticed in the Table. However, having MFF loops leads to better reliability. The results confirm that the soft-response thresholding strategy can be employed to significantly increase the reliability of response signatures to at least 89%. It is also important to note that XOR PUFs have a huge set of challenges to choose from, i.e., 2^N challenges for N -bit PUFs. For 64-bit XOR PUFs, if we assume that a mere 1% of the challenges are stable, the server still has 1.8×10^5 trillion (1% of 2^{64}) stable challenges.

4 CONCLUSION

This paper shows that FFXOR PUFs are better alternatives to standard XOR PUFs, in terms of their attack-resistance. Standard XOR PUF and FF XOR PUFs with 32-bit challenges, up to 8 XOR levels are evaluated using multi-layer perceptrons and the results are compared. It is worth noting that reliability based machine learning attacks have been shown to outperform other machine learning attacks on XOR PUFs. This is because they utilize a *divide-and-conquer approach* which reduces the number of model parameters from $l(N+1)$ to $N+1$ for XOR PUFs with l components of N -stages each. But, the divide-and-conquer approach used in reliability based attacks is dependent on the fact that reliability of a response bit depends equally on each of the component PUFs [3]. However, this

Table 1: Percentage of Stable Challenges and Reliability Before and After Thresholding. Threshold = 90%.

No. of Arbs/ Loops	FF Loops Locations	Noise Level = 5%			Noise Level = 10%			Noise Level = 15%		
		% Stable	Rel.(%) Before	Rel.(%) After	% Stable	Rel.(%) Before	Rel.(%) After	% Stable	Rel.(%) Before	Rel.(%) After
0/0	-	65.5	84.63	97.36	40.69	73.48	95.26	23.88	65.67	93.45
1/1	10 → 40	52.1	78.65	96.2	24.73	66.09	93.45	10.63	58.93	91.26
1/2	10 → 40, 41 (MFF)	63.78	83.98	97.21	38.94	72.63	94.95	22.62	65.18	93.09
1/2	10 → 30, 40	58.45	81.35	96.75	32.07	69.67	94.25	16.54	62.4	92.41
1/3	10 → 30, 40, 50	52.14	78.66	96.26	24.65	65.99	93.41	10.5	59.07	91.34
1/4	10 → 30, 31, 40, 41	62.74	83.27	97.18	37.70	72.21	94.82	21.45	64.58	92.86
1/4	10 → 20, 30, 40, 50	54.81	79.92	96.43	27.73	67.61	93.76	12.85	60.16	91.75
1/5	10 → 20, 30, 40, 50, 60	52.34	78.74	96.15	24.61	66.37	93.37	10.6	59.03	91.28
1/6	10 → 30, 31, 40, 41, 50, 51	61.81	83.12	97.1	36.64	71.73	94.75	20.61	64.26	92.7
1/6	10 → 30, 35, 40, 45, 50, 55	56.52	80.49	96.52	29.44	68.44	94	14.35	61.11	91.96
1/8	10 → 20, 21, 30, 31, 40, 41, 50, 51	61.43	82.72	96.96	35.76	71.52	94.59	19.74	63.93	92.72
1/8	10 → 20, 25, 30, 35, 40, 45, 50, 55	54.82	79.67	96.51	27.55	67.48	93.79	20.73	60.39	91.68
4/4	10 → 20; 15 → 30; 25 → 40 35 → 50	33.47	70.40	94.66	9.98	58.74	91.44	2.72	53.42	89.5
4*/8	10 → 20, 21; 15 → 30, 31; 25 → 40, 41; 35 → 50, 51;	57.73	81.14	96.75	33.09	70.04	94.30	18.18	63.02	92.56
4/4	15 → 55; 20 → 50; 25 → 45; 30 → 40	26.17	66.92	94.01	6.16	56.21	90.89	1.45	52.48	88.80
4*/8	15 → 55, 56; 20 → 50, 51; 25 → 45, 46; 30 → 40, 41;	58.01	81.26	96.85	33.36	70.18	94.52	18.36	63.32	92.58

* The intermediate arbiters are connected to modified FF loops.

would not be the case if *heterogeneous* FFXOR PUFs are used, making them more attack resistant. In a heterogeneous FFXOR PUF, the component PUFs are non-identical, i.e., these components could be different FF PUF structures with different number of arbiters and loops. Therefore, future work should investigate heterogeneous FFXOR PUFs. The effect of number of XOR levels and the number of FF loops on the reliability of FFXOR PUFs under different noise levels is presented and a soft-response thresholding strategy is demonstrated as an effective countermeasure to degraded reliability. Reliability values as low as 52% were increased to 89%. Future efforts will be directed towards studying the aging effects of FFXOR PUFs [8].

ACKNOWLEDGMENTS

This research has been supported by the National Science Foundation under grant number CNS-1441639 and the Semiconductor Research Corporation under contract number 2014-TS-2560.

REFERENCES

- [1] S. V. S. Avvaru, C. Zhou, C. H. Kim, and K. K. Parhi. 2017. Predicting hard and soft-responses and identifying stable challenges of MUX PUFs using ANNs. In *Midwest Symposium on Circuits and Systems (MWSCAS)*. 934–937.
- [2] S. V. S. Avvaru, C. Zhou, S. Satapathy, Y. Lao, C. H. Kim, and K. K. Parhi. 2016. Estimating Delay Differences of Arbiter PUFs Using Silicon Data. In *Proceedings of Design, Automation and Test in Europe (DATE)*. 543–546.
- [3] G. T. Becker. 2015. The gap between promise and reality: On the insecurity of XOR arbiter PUFs. In *Int. Workshop on Cryptographic Hardware and Embedded Systems*. 235–255.
- [4] F. Ganji, S. Tajik, and J. Seifert. 2015. Why Attackers Win: On the Learnability of XOR Arbiter PUFs. In *Trust and Trustworthy Computing*, Mauro Conti, Matthias Schunter, and Ioannis Askoxylakis (Eds.). Springer Int. Publishing, Cham, 22–39.
- [5] C. Herder, M. Yu, F. Koushanfar, and S. Devadas. 2014. Physical Unclonable Functions and Applications: A Tutorial. In *Proc. of the IEEE*, Vol. 102. 1126–1141.
- [6] G. Hospodar, R. Maes, and I. Verbauwhede. 2012. Machine learning attacks on 65nm Arbiter PUFs: Accurate modeling poses strict bounds on usability. In *IEEE International Workshop on Information Forensics and Security (WIFS)*. 37–42.
- [7] S. Katzenbeisser, Ü. I. Kocabaş, V. Rožić, A. R. Sadeghi, I. Verbauwhede, and C. Wachsmann. 2012. PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon. In *Proceedings of 14th Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. 283–301.
- [8] A. Koyli, S. V. S. Avvaru, C. Zhou, C. H. Kim, and K. K. Parhi. 2018. Effect of aging on linear and nonlinear MUX PUFs by statistical modeling. In *Asia and South Pacific Design Automation Conference (ASP-DAC)*. 76–83.
- [9] Y. Lao and K. K. Parhi. 2014. Statistical Analysis of MUX-Based Physical Unclonable Functions. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 33, 5 (May 2014), 649–662.
- [10] U. Rührmair, J. Sölter, F. Sehnke, Xiaolin Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas. 2013. PUF Modeling Attacks on Simulated and Silicon Data. *IEEE Trans. on Information Forensics and Security* 8, 11 (Nov 2013), 1876–1891.
- [11] G. E. Suh and S. Devadas. 2007. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *44th ACM/IEEE DAC*.
- [12] C. Zhou, K. K. Parhi, and C. H. Kim. 2017. Secure and reliable XOR arbiter PUF design: An experimental study based on 1 trillion challenge response pair measurements. In *54th ACM/IEEE DAC*. 1–6.
- [13] C. Zhou, S. Satapathy, Y. Lao, K. K. Parhi, and C. H. Kim. 2016. Soft Response Generation and Thresholding Strategies for Linear and Feed-Forward MUX PUFs. In *Proc. of Int. Symposium on Low Power Electronics and Design (ISLPED)*.