Received: 25 June 2020 | Revised: 18 December 2020 | Accepted: 21 December 2020

DOI: 10.1049/cds2.12042

IET Circuits, Devices & Systems

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

MARPUF: physical unclonable function with improved machine learning attack resistance

Somanath Tripathy | **Vikash Kumar Rai** | **Jimson Mathew**

Indian Institute of Technology Patna, Patna, Bihar, India

Correspondence

Vikash Kumar Rai, Indian Institute of Technology Patna, Patna, Bihar, India.
Email: vikash.pcs15@iitp.ac.in

Abstract

Nowadays, physical unclonable functions (PUFs) are emerging as one of the key building blocks for device authentication and key generation. Although PUF is very useful in the area of hardware security, it is vulnerable to machine learning modelling attacks (ML-MA) by modelling the challenge-response pairs (CRPs) behaviour. To this end, this study proposes a novel PUF named MARPUF, which gives good resistance to machine learning (ML) attacks. The study proposed a MARPUF design, where the mapping of CRPs is randomized by implementing two-round challenges to meet the randomness requirements for ML resistance. Ssome of the popular ML techniques are used to test the ML attack resistance and compare the results with some existing PUFs. We evaluate the performance of the PUF against various parameters like reliability, uniformity, uniqueness, *etc*. The hardware cost analysis shows that MARPUF requires lesser hardware than the existing ML-MA resistant PUFs.

1 | INTRODUCTION

Emerging computing and connected systems face significant challenges due to the rapid evolution of cyber-physical systems (CPS) and various security threats. One of the major thrusts towards CPS design is the evolution of the internet of things (IoT) and its security layer design aspects. First of all, the system design has to be capable of dealing with the heterogeneity of interactions and resistant against different major attack scenarios. While the IoT paradigm gives a number of opportunities to designers and consumers, it creates new challenges in terms of security, trust, and privacy in the computing and communication methodologies used in these devices. When it comes to IoT security, physical unclonable function (PUF) is emerging as one of the key building blocks for device authentication and other supporting functions such as key generation. A PUF is a hardware-specific unique identity or a digital fingerprint of an integrated circuit (IC) or a device under consideration. It is a challenge-response mechanism that gives a unique response for each challenge [1-3] applied to the device under the authentication process. PUF exploits manufacturing process variation inside ICs or the device to generate unique responses. The unique response of an IC can be used in a variety of applications in the area of hardware security like secret key generation, intellectual property protection, device authentication, and radio-frequency identification tag to detect counterfeit ICs, etc.

In general, PUFs are categorized into two types. Delay-based PUFs like ring oscillator PUF (RO PUF) [4], arbiter PUF [5], and glitch PUF (Anderson PUF) [6], etc. are referred as strong PUF, while the memory-based PUF like SRAM PUF [7-10] is generally considered as weak PUF. The strong PUF is preferentially used in direct authentication schemes due to its large challenge-response pairs (CRPs), which make it highly unpredictable. Cryptographic key generation schemes may use weak PUFs. Among various kinds of PUFs, RO-PUF and arbiter PUF are the two most popular PUF architectures. Arbiter PUF is based on the delay-time difference of the two signals. It has a serial connection of multiple stages. Each stage consists of two multiplexers. Each signal may propagate along two paths through every stage based on the selection bit. Here, the selection bit is called a challenge. The output of the last stage determines the response bit on the basis of the faster signal among the two signals. An RO-PUF produces CRP based on the frequency of the two ring oscillators. A challenge is given to multiplexers and based on that, the ring oscillators

are selected. Then, these selected ring oscillators are connected to different counters. After a certain time interval, the counter values for both the counters are captured. The PUF responses are calculated based on the faster ring oscillator among the selected ring oscillators.

The unpredictability of the CRPs behaviour is a crucial feature for the security of a PUF, and the protocols build on it. Meanwhile, machine learning (ML) techniques become a handy tool to model the PUF CRPs behaviour [11]. They become a vital threat to strong PUFs. In machine learning modelling attacks (ML-MA), a set of CRPs are used to build a strong PUF module, which is later used to predict the response to a new challenge. Dubrova et al. in [12] proposed a lightweight ML-MA resistant PUF that consists of a linear feedback shift register (LFSR) and arbiter PUF. Another modelling attack-resistant PUF is presented in [13] by Nguyen et al., which is a two-round PUF. It uses n-bit arbiter PUF and the response of the PUF is added to the initial challenge to make the new challenge of size $n + 1$-bit. This new challenge is given to the $n + 1$ arbiter PUF again to generate the final response. Though they analysed their schemes in terms of hardware cost and showed that their schemes required less hardware, but Dubrova et al. in [12] tested their scheme only against LR and Nguyen used some reliability-based techniques. Komurcu et al. in [14] presented two methods in RO PUF with enhanced CRP sets and analysed the area efficiency and uniqueness of the design. This study also proposed three different uses scenario for the PUF that generates enhanced CRPs sets.

We propose a lightweight PUF named MARPUF which is also resistant to various ML attacks. The major contributions of this work are as follows.

- We design and simulate a lightweight ML resistant PUF architecture named MARPUF. A small size PUF is used to generate intermediate challenge, which acts as an input of the PUF in the next round. Thus the direct relationship between CRPs is concealed. Hence it becomes difficult for the attacker to develop an attack model of the PUF design because the attacker cannot know the intermediate challenge.
- We evaluate the efficiency of the proposed PUF based on widely accepted parameters including uniqueness, uniformity, bit-aliasing, reliability, steadiness, probability of misidentification (PMSID), and entropy.
- We perform the modelling attack on MARPUF using widely used ML techniques like LR, support vector machine (SVM), and Naive Bayes and compare modelling attack results with some of the existing PUFs. MARPUF is found to be better resistant to such modelling attacks.
- We conduct the hardware cost analysis for MARPUF and compare it with the existing modelling attack resistant PUFs and found that MARPUF consumes relatively lesser hardware than that of the existing schemes.

The organization of this study is as follows. In Section 2, the background and related literature regarding PUFs are provided. The architecture for ML resistant PUF is presented in Section 3. Section 4 provides the simulation and results analysis of the proposed design. We conclude this study in Section 5.

2 | BACKGROUND AND RELATED WORK

2.1 | Ring oscillator PUF

RO PUF [4] consists of ring oscillators as its backbone component. The ring oscillators frequencies are different due to the manufacturing and material variations. Two ring oscillators RO_1 and RO_2, are selected from a set of ring oscillators using a challenge and connected to counters. After a certain time interval, the counter values are compared to collect the output using a simple rule. If the counter value (RO_1) >counter value (RO_2) the output is 0 and 1 otherwise.

2.2 | Arbiter PUF

The arbiter PUF [5, 15] consists of an array of stages. Each stage of the array is comprised of two MUXes and connected to its next stage. Two signals start from the first stage at the same time, and the signal paths are decided by the MUXes selection bits, which also called the challenge to the PUF. Since there are process variations during the manufacturing of the chip, two signals would not take the same amount of time to go through the last stage. At last, an arbiter is connected, which takes one signal as data input and other as clock input, and based on the temporal relation between clock and data signal, it produces the output bit, which is called the PUF response.

2.3 | Modelling attack on PUF

Modelling attack on PUF has an assumption that the attacker somehow collects a subset of all CRPs and attempts to build a model using these CRPs. In modelling attacks, computer algorithms try to predict the PUF responses with high modelling attack accuracy. Some ML techniques are used to learn the parameters of a PUF circuit. As an example, an N-stage arbiter PUF can be treated as a linear additive delay model. The time delay between two stages can be modelled using some efficient ML techniques using some CRPs. The time delay difference Δ can be evaluated as Equation (1).

$$\Delta = w^T Q \tag{1}$$

where delay vector is denoted by w for every segment of the arbiter PUF and Q is a function of k-bit challenge C.

$$Q(C) = (Q^1(C), \ldots, Q^k(C), 1)^T \tag{2}$$

where $Q^l(C) = \prod_{i=l}^{m}(1 - 2c_i)$ for $l = 1, \ldots, k$. It is to be noted that the function Q to map the challenges to the real values

would be different for different types of PUF. The primary focus of an attacker is to approximate the value of w, which could be a real delay vector. In ML attack, the delay information w for each stage can be learned with good modelling attack accuracy if an attacker possesses a sufficient number of CRPs.

Recently, many researchers have widely studied the modelling attack over different types of PUF, including RO PUF. Wang et al. in [16] studied various kinds of modelling attacks in detail on RO PUF and arbiter PUF. They used logistic regression (LR) and neural network ML algorithms to conduct modelling attacks. The experimental result clearly shows that they are successfully able to model these PUFs with a modelling attack accuracy of around 90%. However, the authors have reported a dual-mode feedback PUF on the reconfigurable platform, which can behave as either an RO PUF or a bistable ring (BR) PUF to mystify the attacker. This PUF has a unique feature that it works in both cases. If it has an even number of inverters, it works as a BR PUF. If it consists of an odd number of inverters, it works as a reconfigurable RO PUF. As long as the attacker would be unaware of the working mode of the PUF, it would remain the modelling resistant PUF. This PUF showed the resistance against modelling attacks, with the modelling attack accuracy reduced to around 70%. Saha et al. [17] presented another modelling attack on RO PUF using genetic programming. They have implemented evolutionary computation to build an accurate model for the field programmable gate arrays (FPGA)-based RO PUF. The authors have used LR and neural network ML algorithms to conduct this modelling attack. The experimental result clearly shows that they can successfully model these PUFs with a modelling attack accuracy of around 90%.

Gabriel et al. in [18] used some widely accepted ML algorithms like artificial neural network (ANN), SVM to study the effectiveness of these algorithms on a 64-stage arbiter PUFs realized in 65nm complementary metal oxide semiconductor (CMOS). This work has shown that from a training set of only 500 CRPs, a 90% accurate model can be built, and only 5000 CRPs are required to perfectly model a PUF design. Gabriel et al. in [18] proposed a new methodology to study the implications of these attacks and conclude that a simple 64-stage PUF is not secure for challenge-response authentication. In [19], Ulrich et al. presented modelling attacks on different PUFs. They have used LR ML algorithms to model arbiter PUFs on a given set of CRPs. In this proposed scheme, for 64-bit PUF, 18000 CRPs are required to achieve the modelling attack accuracy of more than 90%. Similarly, for 128-bit PUF, 32000 CRPs are required to achieve the modelling attack accuracy of more than 90%. Some other modelling attacks are discussed in [20,22]. Tanaka et al. in [20] developed a novel PUF architecture that is resilient against ML attacks. This PUF is based on a BR. They analyzed the convergence through analytical formulations. The vital feature of this PUF is the convergence time of the BR is nonlinearly dependent on the variations in the threshold voltage of the transistors. A coin-flipping PUF architecture is proposed using this nonlinearity, which consists of an RO and a BR. The instantaneous value of the RO is captured as and when the BR paired to it gets converge. This captured value served as PUF response. Ma et al. in [23] presented an ML resistant PUF named multi-PUF, which is based on the challenge obfuscation. In this design, any n weak PUFs and a strong PUF can be used to generate one-bit response. In particular, n- picoPUFs are used, each of which produced one-bit response. An intermediate n-bit binary string $C_0C_1\ldots C_n$ is used to XOR with each response of the picoPUF, and as a result, an n-bit string is produced. This n-bit string is given to the challenge to strong PUF, and one-bit output is collected as a final response. The performance of the design is evaluated, and uniqueness and uniformity are calculated as 40.60% and 37.03%, respectively. The ML attack resistance is evaluated against LR and covariance matrix adaptation and evolution strategy (CMA-ES) modelling techniques. The modelling attack accuracy of the LR is found to be 50%, and for the CMA-ES the modelling attack accuracy is calculated as 80%. Cui et al. proposed multiplexer-based multi-PUF in [24] which is resistant to LR and SVM ML techniques.

Khalafalla et al. in [25] pushed the boundaries of the ML attack by introducing deep learning techniques against double arbiter PUFs. In this, the attacker came up with the attack with high modelling attack accuracy.

2.4 | Design requirements

PUF is a challenge–response system which produces a distinct set of outputs (*responses*) corresponding to a set of inputs (*challenges*). A better PUF design should have the following design requirements:

- **Uniqueness:** Uniqueness is an essential feature of a PUF which is used to uniquely identify a particular chip among a group of identical chips. To evaluate the uniqueness, Hamming distance (HD) between a pair of PUF identifier is used. If R_p and R_q are n-bit responses of two chips, p and q ($p \neq q$), respectively, for same challenge C, the average interchip HD among r chips is defined as:

$$Uniqueness = \frac{2}{r(r-1)} \sum_{p=1}^{r-1} \sum_{q=p+1}^{r} \frac{HD(R_p, R_q)}{n} \times 100\% \quad (3)$$

The ideal value for uniqueness is 50% for a unique PUF.

- **Uniformity:** PUF response bits consist of 0s and 1s. Evaluation of the proportion of 0s and 1s in the response bit sequence is called the uniformity. The ideal value for uniformity is 50% for a truly random PUF response. The percentage Hamming weight (HW) is used to calculate the uniformity of an n-bit PUF identifier using the following formula:

$$Uniformity = \frac{1}{n} \sum_{l=1}^{n} R_{p,l} \times 100\% \quad (4)$$

where $R_{p,l}$ is the l^{th} binary bit of an n-bit response from a chip p.

- **Reliability:** Reliability is defined as the efficiency of a PUF to reproduce its response bits while applying the same challenge. Intrachip HD of different PUF instances is used to measure the reliability. An n-bit response R_p is taken as reference response at normal operating parameters. Then, n-bit responses R'_p are collected at various operating conditions. m number of such n-bit samples are collected, and the average of the intrachip HD is calculated as follows:

$$HD_{INTRA} = \frac{1}{m}\sum_{t=1}^{m}\frac{HD(R_p, R'_{p,t})}{n} \times 100\% \quad (5)$$

where $R'_{p,t}$ is the t^{th} sample of R'_p. HD_{INTRA} shows the average number of unreliable PUF response bits. So, we can calculate the reliability of a PUF as follows:

$$Reliability = 100\% - HD_{INTRA} \quad (6)$$

The ideal value of reliability should be 100%.

- **Bit-aliasing:** Different chips may produce identical responses for same challenge due to bit-aliasing. This is not a desirable characteristic for an ideal PUF. The percentage HW of l^{th} bit of the PUF identifier across r devices is defined as the bit-aliasing which can be calculated as:

$$Bit - Aliasing = \frac{1}{r}\sum_{p=1}^{r} R_{p,l} \times 100\% \quad (7)$$

where $R_{p,l}$ is the l^{th} binary bit of an n-bit response from a chip p. Ideally, bit-aliasing should be 50%.

- **Steadiness:** According to Hori et al. [26] the steadiness is defined as the degree of bias of a response bit towards 0 or 1 over S sample. The ideal value for the steadiness is 100%. Lesser value of steadiness would produce lesser correctness. The desired value for steadiness is 100%.

$$Steadiness(S_n) = 1 + \frac{1}{K \cdot L}\sum_{k=1}^{K}\sum_{l=1}^{L} log_2 max(P_{n,k,l}, 1 - P_{n,k,l}) \quad (8)$$

where K is total number of identifier per chip, L is total number of response bit, and $P_{n,k,l} = \frac{1}{S}\sum_{s=1}^{S} r_{n,k,s,l}$. where r is the response bits, n is ithe ndex of a chip, k is the index of an identifier, s is the index of sample, and l is the index of response bits.

- **PMSID:** PMSID is introduced by Maiti et al. [27], which measures the likelihood of PUF being falsely identified as another PUF due to some noise in the PUF response bits. PMSID is defined as:

$$\sum_{h=0}^{L}\left[\binom{L}{h}0.5^h(1-0.5)^{L-h} \cdot \sum_{h/2}^{h} p^{h/2}(1-p)^{h-h/2}\right] \quad (9)$$

where L is the length of the response bits, p is the fraction of the unreliable bits, and h is the HD between the two PUF responses ($h \leq L$). PMSID value should be 0% for an ideal PUF.

3 | MARPUF: THE PROPOSED PUF

In this section, we present MARPUF, our proposed PUF with improved ML resistance. MARPUF operates in two rounds (for obtaining response corresponding to the challenge), to reduce linear dependency and so the modelling attack accuracy of modelling attacks get reduced. The architecture of MARPUF is as shown in Figure 1 and its working principle is discussed below.

- The n-bit challenge (C) is divided into k subsets (C_1, C_2, C_3,…,C_k) of m−bit each, such that $n = m*k$.
- The parity bit for each bit position from every subset is calculated that formed a bit sequence as shown in Figure 2, which is calculated as follows:

$$Parity(P^i) = (C_1^i \oplus C_2^i \oplus C_3^i \oplus ... \oplus C_k^i) \quad (10)$$

where i = 1,2,3,…m. The corresponding bits from each position from every subset are Ex-ORed to calculate parity bit and finally generate an m-bit binary sequence. The objective of this parity bit calculation is to break the challenge-response relationship. Thus it becomes hard for the attacker to realize a relation due to the two-round iteration.

- This resultant m-bit sequence C' is served as challenge to the m-bit PUF, and the response R′ is collected as the output.
- The PUF response R′is Ex-ORed with C_1, C_2, C_3, …,C_k as $C_1' = C_1 \oplus R'$, $C_2' = C_2 \oplus R'$, $C_3' = C_3 \oplus R'$,…, $C_k' = C_k \oplus R'$.
- Now, C_1', C_2', C_3', …, C_k' are combined and given as challenge input to the n-bit PUF in the second round.
- The final result is collected and combined as n-bit response R.

The use of parity bits to derive a new challenge makes the relation between challenge and response nonlinear, which makes it difficult for learning techniques to model the PUF design. The responses generated in-between are essentially needed to generate the final response, but the responses produced in between are not known to the attacker. Hence, the attacker would not be able to build the model of this PUF with high modelling attack accuracy. Moreover, the first round challenge is an m-bit challenge, so it requires a lesser amount of resource compared to the case when the challenge is $m*k$ bits. So even if the PUF is two rounds, it would consume lesser resources.

4 | SIMULATION AND RESULT ANALYSIS

The simulation of the circuit is performed using synopsys HSPICE simulator, using a 65-nm predictive technology

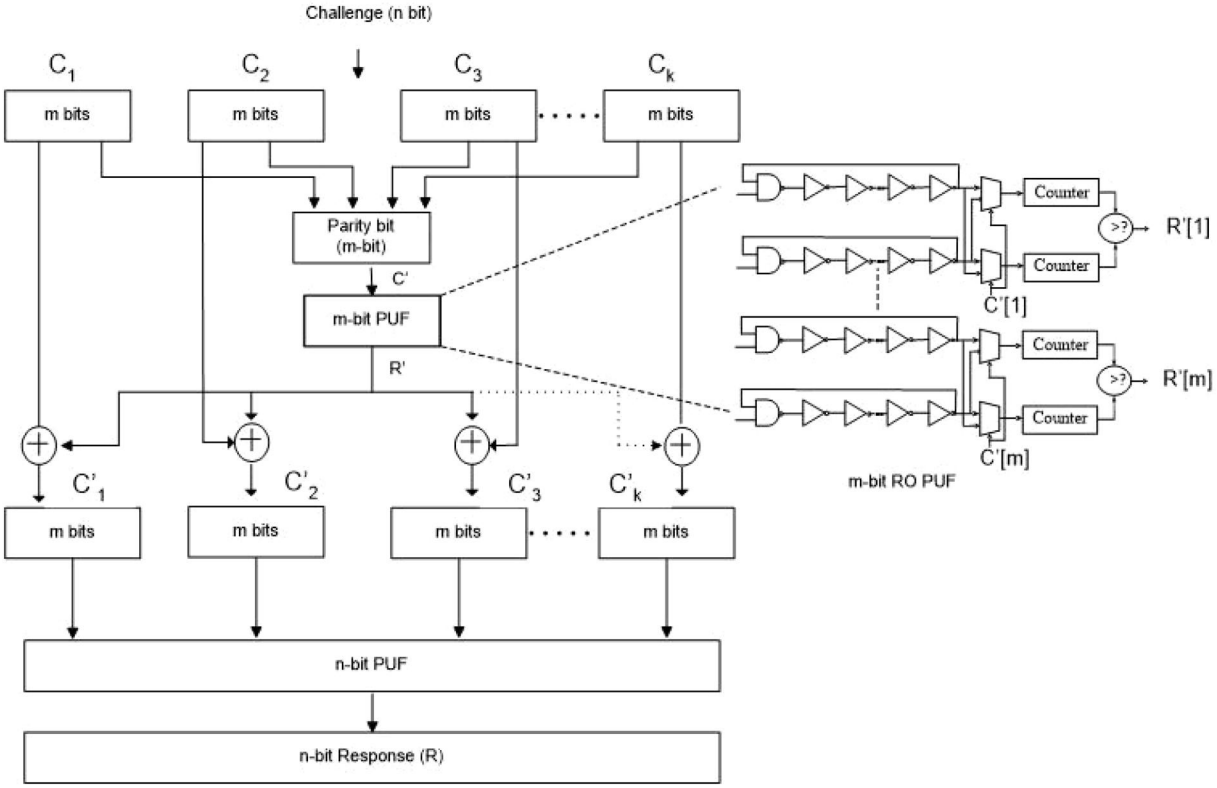
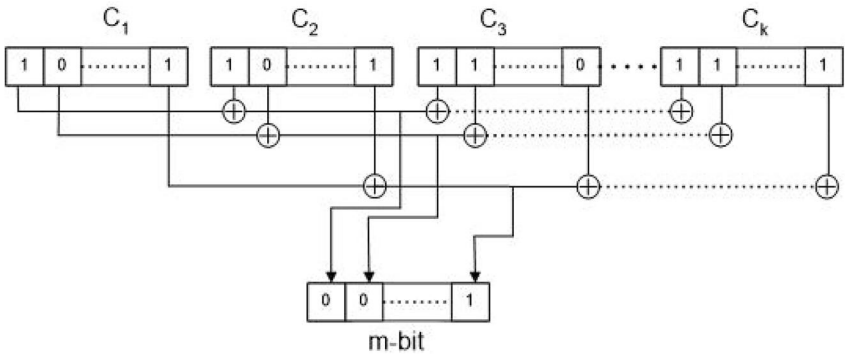


FIGURE 1 MARPUF architecture

FIGURE 2 Intermediate challenge generation



model for the CMOS devices. In our simulation, we assumed approximately 10%–20% variations in the basic parameters like V_{th}, supply voltage, etc. The PUF circuit used here in simulation is an RO PUF. Though we have used here RO PUF as a building block, our scheme can adopt to work with other PUFs like arbiter PUF as well.

4.1 | Efficiency evaluation

The efficiency of MARPUF is evaluated on the ground of widely used parameters like entropy, uniqueness, uniformity, reliability, bit-aliasing, steadiness, and PMSID. The results of PUF performance evaluations are compiled in Table 1.

- **Entropy:** The entropy analysis is carried out using the bit strings generated from the 2000 instances of the PUF, as discussed in [28]. Entropy is defined by Equation (11). The probability of 0s and 1s is computed for each binary string of 16 bits, then the entropy is calculated for each binary string using Equation (11). The entropy result for MARPUF and some other PUFs is shown in Figure 3.

$$H_n(X) = \left(- \sum_{x \epsilon (0,1)^n} p(X) log_2 p(X) \right) \tag{11}$$

$$H_{min}(X) = -log_2 \left(max \left(p_X \right) \right) \tag{12}$$

TABLE 1 Different PUFs efficiency evaluation

PUF schemes parameters	Uniformity	Uniqueness	Reliability	Bit-aliasing	Steadiness	PMSID	Entropy
Ideal value	50%	50%	100%	50%	100%	0%	1
MARPUF	48.23%	47.12%	95.16%	46.23%	94.32%	0.03%	0.928
RO PUF [4]	50.56%	40.12%	86.34%	50.56%	84.3%	0.86%	0.656
Wang et al. [16]	44.65 %	45.37%	88.5 %	44.61%	89.2%	1.12%	0.879
Arbiter PUF [5]	55.69%	42.12%	88.64%	43.23%	86.3%	0.47%	0.625

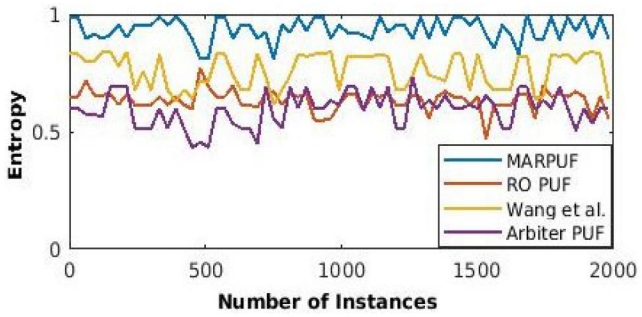


FIGURE 3 Entropy calculation for different PUFs

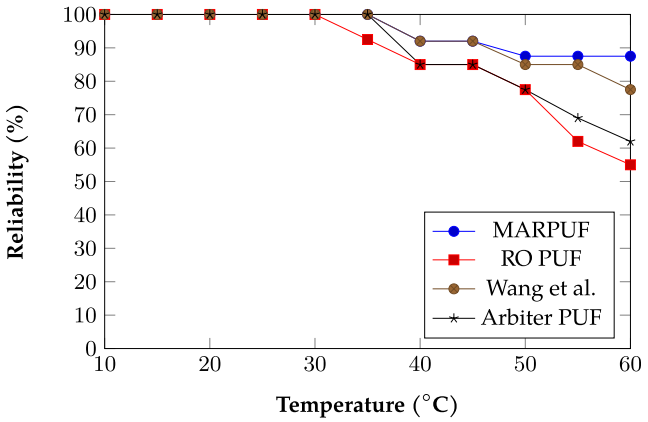


FIGURE 4 Reliability of different PUFs at different temperatures

The entropy for two round MARPUF is calculated as 0.928, whereas the entropy for the usual RO PUF is calculated as 0.656. Furthermore, we have also calculated MinEntropy, which is defined in Equation (12). MinEntropy for the proposed MARPUF is found to be 0.692, while for RO PUF, it is 0.613. Thus, MARPUF has a high value of entropy.

- **Uniformity:** We assessed the uniformity of the PUF using 20000 different response bits obtained by simulation of the MARPUF circuit. We calculated the percentage HW of the response sequence to obtain the uniformity. It has been calculated as 48.23%, while the ideal value is 50%.
- **Uniqueness:** The uniqueness of the PUF has been evaluated using different instances of the same PUF. We took the two different instances of the same PUF and obtained two different response sets from both PUF instances. Then, we calculated the average HD between the two response sets to obtain the uniqueness of the PUF. The ideal value for the uniqueness of a PUF must be 50%, whereas the calculated value for MARPUF is 47.12%.
- **Bit-aliasing:** Bit-aliasing is another metric for PUF efficiency measurement, which shows identical bits from different chips. We collected the response sets from different PUFs and calculated the HW of each bit of the response. It is found that bit-aliasing is 46.3%, while the ideal value for bit-aliasing is 50%.
- **Steadiness:** We have also calculated the steadiness, which shows the degree of the bias of response bits towards 0 and 1. The value calculated for steadiness is 94.6% for MARPUF. The value of steadiness must be 100% for an ideal PUF.
- **PMSID:** We have also evaluated our PUF against another parameter called the PMSID, which indicates the probability of a PUF identifying as another PUF. The PMSID value is calculated to be 0.03%. The ideal value for PMSID is 0.
- **Reliability:** Reliability is one of the most significant features of a PUF. The temperature and voltage values are varied, and responses of different PUF instances are captured to test the behaviour of the PUF in different operating conditions. We varied the temperature ranges from 10°C to 60° C and tested the reliability of the PUF as discussed in [29]. We considered 25°C as the reference value. The reliability of various PUF designs against different temperature values is depicted in Figure 4. It can be observed that, for MARPUF, the reliability is 100% from 10°C to 35°C and after 35°C, it has been decreased to 87.5%. For RO PUF, the reliability decreased after 30°C and went down to 55% at 60°C. The reliability for the arbiter PUF is 100% till 35°C, and then it decreased to 62% at 60°C. Similarly, for Wang et al. [16] the reliability remains 100% till 35°C and reached up to 78% at 60°C. Furthermore, we varied the supply voltage from 1.1 volts to 1.4 volts and captured the responses of the different PUF designs over these different voltage values. 1.25 volt is taken as a reference voltage value. The reliability of various PUF designs against different voltage values is depicted in

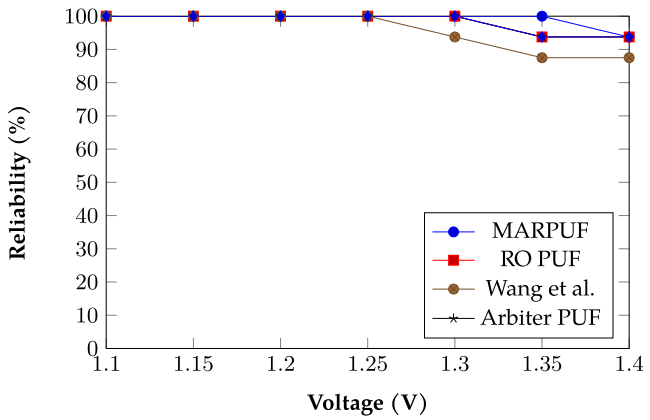


FIGURE 5 Reliability of different PUFs at different voltages

Figure 5. It has been seen that after 1.35 volts, the reliability for MARPUF has been decreased to 93.75%. The reliability for RO PUF is 100% till 1.25 volts, and after that, it has been reduced to 87.5% at 1.4 volts. For arbiter PUF the reliability remains 100% up to 1.3 volts, and then it decreased to 93.75%. Similarly, the reliability for Wang et al. [16] stayed at 100% till 1.3 volts and then reduced to 93.75%.

4.2 | Modelling attack analysis

Here, we describe the modelling attack results on various PUFs, including the proposed MARPUF. We have implemented the most popular ML techniques like LR, Naive Bayes, SVM, random forest, and ANN to test our PUF design. These techniques are well established for the ML attacks and widely used in [16], [12], and [30]. We first collected the CRPs from the HSPICE simulation of the circuit and used these CRPs to test the modelling attack accuracy against different modelling techniques. We have also simulated arbiter PUF, RO PUF, and Wang et al. PUF structure [16] and performed the modelling attack using the same ML techniques to conduct a comparative study.

The LR ML modelling attack results for the arbiter PUF, RO PUF, Wang et al. PUF [16], and also for MARPUF are depicted in Figure 6. We varied the training size from 100 to 20000 to study the attack results. The y-axis represents the modelling attack accuracy of the attack, where the maximum value that could be reached is one, which indicates that the PUF response bits can be perfectly predicted. The modelling attack accuracy of an ideal ML resistant PUF is 0.5, which is equivalent to a random guess. It can be observed in Figure 6 that modelling attack accuracy is nearly 0.9 for arbiter and RO PUF, which indicates that LR can successfully predict the responses for RO PUF and arbiter PUF with a modelling attack accuracy of nearly 90%. The modelling attack accuracy for PUF described in [16] is 66.9%. Meanwhile, it is observed that the modelling attack accuracy reduced to 0.535 in the case

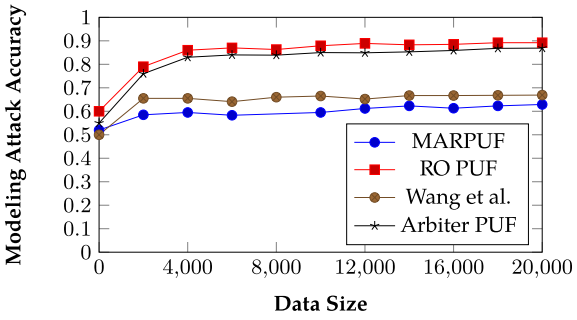


FIGURE 6 LR modelling attack on various 128-bit PUF

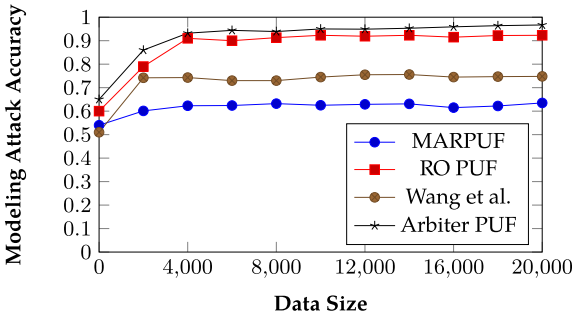


FIGURE 7 SVM modelling attack on various 128-bit PUF

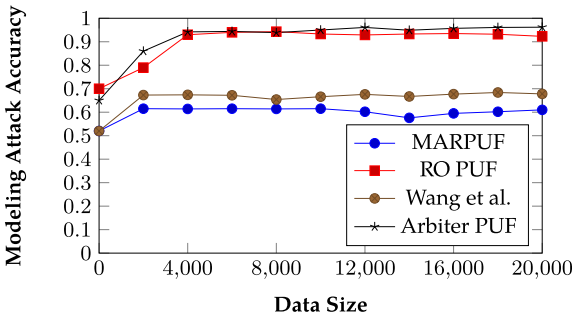


FIGURE 8 Naive Bayes modelling attack on various 128-bit PUF

of MARPUF. LR can predict the responses with a modelling attack accuracy of 53.5%.

We also used SVM, Naive Bayes, random forest, and ANN ML algorithms to perform the modelling attack. The results for SVM are depicted in Figure 7. Here, it can be observed that SVM modelling attack accuracy reaches up to 96.7% for the arbiter PUF and 92.2% for RO PUF. However, SVM could model Wang et al. PUF [16] with a modelling attack accuracy of 74.8%. MARPUF reduced the modelling attack accuracy up to 57.1%. Similar results are found while using the Naive Bayes algorithm, as shown in Figure 8. It can clearly be observed that Naive Bayes modelling attack accuracy reaches up to 96.2% for the arbiter PUF and 94.3% for RO PUF. However, Naive Bayes could model Wang et al. PUF [16] with a modelling attack accuracy

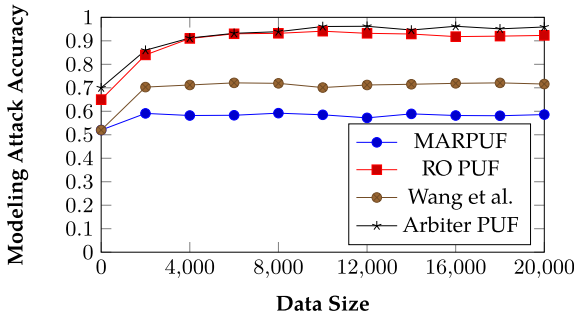


FIGURE 9 ANN modelling attack on various 128-bit PUF

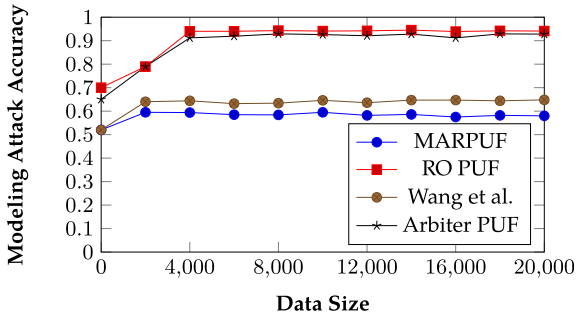


FIGURE 10 Random forest modelling attack on various 128-bit PUF

of 67.8%. MARPUF reduced the modelling attack accuracy up to 56.3%. The ANN modelling attack results are shown in Figure 9. Here, it can be observed that ANN modelling attack accuracy reaches up to 92.1% for the arbiter PUF and 89.48% for RO PUF. However, ANN could model Wang et al's PUF [16] with a modelling attack accuracy of 71.8%. On the other hand, MARPUF reduced the modelling attack accuracy up to 58.3%. The results of the random forest modelling attack are depicted in Figure 10. It can successfully predict the responses for RO PUF and arbiter PUF with a modelling attack accuracy of 90.57% and 88.58%, respectively. The modelling attack accuracy for PUF described in [16] is 63.7%. Meanwhile, it is observed that in the case of MARPUF it can predict the responses with a modelling attack accuracy of 57.98% only.

The reason for the lesser modelling attack accuracy on MAPRPUF is due to its internal architecture, which has two rounds of operation. The first round generates a response using the original challenge set, which serves as a new challenge set in the second round. This new challenge set is applied to the second round of the PUF; thus the challenge is obfuscated. So, the linear dependency between the original challenge and the PUF response is reduced. Therefore the accuracy of the modelling attack is reduced. The modelling attack accuracy for 16-bit, 32-bit, and 64-bit MARPUF is also studied along with 128-bit PUF size and compared with some of the existing PUFs. The comparison result is shown in Table 2. It can be clearly observed from the table that MARPUF shows better ML attack resistance.

TABLE 2 ML-MA accuracy on various PUFs

PUF schemes	LR attack				SVM attack				Naive Bayes attack				ANN attack				Random forest attack			
ML techniques	**16-bit**	**32-bit**	**64-bit**	**128-bit**	**16-bit**	**32-bit**	**64-bit**	**128-bit**	**16-bit**	**32-bit**	**64-bit**	**128-bit**	**16-bit**	**32-bit**	**64-bit**	**128-bit**	**16-bit**	**32-bit**	**64-bit**	**128-bit**
MARPUF (proposed)	54.4%	54.9%	55.2%	53.5%	55.2%	54.5%	57.1 %	57.3%	56.2%	56.9%	56.2%	56.3%	58.4%	58.2%	58.9%	58.3%	58.2%	58.4%	57.3 %	58.98%
RO PUF [4]	88.7%	87.4%	89.2%	89.2%	91.7%	92.2%	91.2%	92.2%	93.6%	92.3%	93.8%	94.3%	89.4%	89.9%	88.8%	89.48%	89.2%	90.5%	90.1 %	90.57%
Wang et al. [16]	65.7%	66.2%	66.7%	66.9%	75.1%	74.7%	73.6%	74.8%	69.2%	70.1%	68.7%	67.8%	70.4%	70.9%	71.2%	71.8%	63.2%	63.5%	62.1 %	63.7%
Arbiter PUF [5]	87.1%	88.4%	87.2%	86.9%	95.1%	96.1%	95.3%	96.7%	95.7%	96.7%	95.6%	96.2%	92.4%	93.9%	93.2%	92.1%	90.2%	89.5%	90.1 %	88.58%

TABLE 3 Hardware cost comparison of MARPUF with other PUFs

PUF schemes parameters	PUF size	Common hardware (GE)	Additional hardware (GE)	Total GE
MARPUF	128	128-bit arbiter PUF (646)	136 XOR gates, 8-bit arbiter PUF (386)	1032
Wang et al. [16]	128	-	256 MUXes, 16 NAND gates, 16 AND gates, and 256 inverter	1064
CRC-PUF [12]	128	128-bit arbiter PUF (646)	128-bit LFSR (510)	1156
Interpose PUF [13]	128	128-bit arbiter PUF (646)	129-bit arbiter PUF (651)	1297

4.3 | Hardware cost analysis

In this subsection, we discuss the hardware cost of the proposed MARPUF, and also we compare it with some state-of-the-art ML resistant PUFs. We use gate equivalent (GE) as a measurement unit. GE is a technology-independent unit to measure the circuit area. The area of the smallest 2-input NAND gate is normally considered as GE for CMOS technology. The gate parameters to calculate the GE are given in [12]. The GE for a 2-input NAND gate is 1, for 2-input AND gate is 1.5, for 2-input XOR 2.5, for 2-to-1 MUX 2.5, and for flip-flop 6.25.

In MARPUF design, there are one m-bit PUF, one n-bit PUF, and $m \cdot (k-1) + k$ XOR gates (where $n = m * k$) for intermediate challenge generation. So, for 128-bit MARPUF, we have m=8 and k=16, and if we take arbiter PUF as a basic PUF, we have one 8-bit arbiter PUF and one 128-bit arbiter PUF along with 136 XOR gates. Hence, the hardware cost would be 340 GE for 136 XOR gates, 46 GE for 8-bit arbiter PUF, and 646 GE for 128-bit arbiter PUF. So, the total cost would be 1032 GE.

The cyclic redundancy check PUF (CRC PUF) design discussed in [12] consists of one n-bit LFSR and one n-bit arbiter PUF. The total GE for 128-bit CRC PUF is calculated by considering one 128-bit LFSR apart from basic 128-bit arbiter PUF. The 128-bit LFSR cost would be 510 GE and 646 GE for arbiter PUF. Hence, total GE for 128-bit CRC PUF is 1156. Interpose PUF proposed in [13] consists of one n-bit arbiter PUF and one $(n + 1)$ bit arbiter PUF.

For 128-bit interposing PUF there is 129-bit arbiter PUF, which costs 651 GE along with 646 GE for basic 128-bit arbiter PUF. The total GE becomes 1297 for this PUF. Wang et al. PUF scheme [16] consists of 256 MUXes, 16 NAND gates, 16 AND gates, and 256 inverters. So, total GE for this PUF scheme is calculated as 1064. The hardware cost (GE) comparison is presented in Table 3, which clearly shows that MARPUF requires lesser hardware cost in terms of circuit area in comparison to the other three PUF designs.

4.4 | Discussion

We proposed a PUF design with improved ML attack resistance. Reliability of MARPUF is evaluated against various temperature as shown in Figure 4 and voltage values as shown in Figure 5, and concluded that it is better than the existing PUF designs [4, 5, 16]. The modelling attack accuracy is carried out using some popular techniques, and the simulation results showed that the MARPUF has lesser modelling attack accuracy than the existing schemes. In other words, the MARPUF is more resistant to ML attacks. We also compared the proposed MARPUF with other PUF schemes like CRC PUF [12], interpose PUF [13], and Wang et al. PUF [16] to evaluate the hardware cost and it is observed from Table 3 that the proposed MARPUF has the edge over the said PUF designs. Furthermore, MARPUF requires two clock cycles to generate the output as PUF response, which is higher than the conventional PUF like RO PUF and arbiter PUF designs, as MARPUF involves two rounds of operation. But considering the benefits of MARPUF, which is its ML attack resistance, this limitation may be considered negligible.

5 | CONCLUSION

PUF is one of the critical hardware security primitives, and an ML-MA on a PUF is a potential threat to the security protocols and applications. We proposed a novel PUF architecture called MARPUF, which would efficaciously prevent the modelling attacks. We tested the MARPUF against multiple ML approaches and found that the proposed MARPUF reduced the modelling attack accuracy up to 53%. We also performed the hardware cost analysis of MARPUF and observed that it requires lesser additional hardware. The FPGA implementation of MARPUF is in progress.

ORCID

Somanath Tripathy https://orcid.org/0000-0002-6964-2648
Vikash Kumar Rai https://orcid.org/0000-0003-0284-0823

REFERENCES

1. Gassend, B., et al.: Silicon physical random functions. In: Proceedings of the 9th ACM conference on computer and communications security, pp. 148–160. ACM, Washington, DC USA (2002)
2. Gassend, B., et al.: Controlled physical random functions. In: 18th Annual computer security applications conference, 2002 Proceedings, pp. 149–160. IEEE, Las Vegas, NV, USA (2002)
3. Lee, J.W., et al.: A technique to build a secret key in integrated circuits for identification and authentication applications. In: 2004 Symposium on VLSI circuits. Digest of technical papers (IEEE Cat. No. 04CH37525), pp. 176–179. IEEE, Honolulu, HI, USA (2004)
4. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: 2007 44th ACM/IEEE design automation conference, pp. 9–14. IEEE, San Diego, California, USA (2007)

5. Lim, D., Lee, J.W., Gassend, B., Suh, G.E., Van.Dijk, M., Devadas, S.: Extracting secret keys from integrated circuits. IEEE Trans Very Large Scale Integr VLSI Syst. 13(10), 1200–1205 (2005)
6. Anderson, J.H.: A PUF design for secure FPGA-based embedded systems. In: Proceedings of the 2010 Asia and South Pacific design automation conference, pp. 1–6. IEEE Press, Taipei Taiwan (2010)
7. Holcomb, D.E., et al.: Initial SRAM state as a fingerprint and source of true random numbers for RFID tags., Proc Conf. RFID Security 7, 01 (2007)
8. Holcomb, D.E., Burleson, W.P., Fu, K.: Power-up SRAM state as an identifying fingerprint and source of true random numbers. IEEE T Comput. 58(9), 1198–1210 (2008)
9. Maes, R., Tuyls, P., Verbauwhede, I.: Low-overhead implementation of a soft decision helper data algorithm for SRAMPUFS. In: International workshop on cryptographic hardware and embedded systems, pp. 332–347. Springer, Lausanne, Switzerland (2009)
10. Guajardo, J., et al.: FPGA intrinsic PUFs and their use for IP protection. In: International workshop on cryptographic hardware and embedded systems, pp. 63–80. Springer, Vienna, Austria (2007)
11. Rührmair, U., et al.: Modeling attacks on physical unclonable functions. In: Proceedings of the 17th ACM conference on computer and communications security, pp. 237–249. ACM, Chicago, Illinois, USA (2010)
12. Dubrova, E., et al.: CRC-PUF: a machine learning attack resistant lightweight PUF construction. In: 2019 IEEE European symposium on security and privacy workshops (EuroS&PW), pp. 264–271. IEEE, Stockholm, Sweden (2019)
13. Nguyen, P.H., et al.: The interpose PUF: Secure PUF design against state-of-the-art machine learning attacks. IACR Transactions on Cryptographic Hardware and Embedded Systems, 243–290 (2019)
14. Kömürcü, G., Pusane, A.E., Dündar, G.: Enhanced challenge-response set and secure usage scenarios for ordering-based ring oscillator-physical unclonable functions. IET Circ Dev Syst. 9(2), 87–95 (2015)
15. Devadas, S., et al.: Design and implementation of PUF-based unclonable RFIDICS for anti-counterfeiting and security applications. In: 2008 IEEE international conference on RFID, pp. 58–64. IEEE, Las Vegas, Nevada, USA (2008)
16. Wang, Q., Gao, M., Qu, G.: A machine learning attack resistant dual-mode PUF. In: Proceedings of the 2018 on Great Lakes Symposium on VLSI, pp. 177–182. ACM, Chicago, IL, USA (2018)
17. Saha, I., Jeldi, R.R., Chakraborty, R.S.: Model building attacks on physically unclonable functions using genetic programming. In: Hardware-Oriented Security and Trust (HOST), 2013 IEEE international symposium on, pp. 41–44. IEEE, Austin, TX, USA (2013)
18. Hospodar, G., Maes, R., Verbauwhede, I.: Machine learning attacks on 65nm arbiter PUFs: Accurate modeling poses strict bounds on usability. In: WIFS, pp. 37–42 Costa Adeje, Spain (2012)
19. Rührmair, et al.: PUF modeling attacks on simulated and silicon data. IEEE Trans. Inf. Forensics Secur. 8(11)
20. Tanaka, Y., et al.: Coin flipping PUF: A novel PUF with improved resistance against machine learning attacks. IEEE Trans. Circuits Syst. II Express Briefs. 65(5), 602–606 (2018)
21. Liu, Y., et al.: A combined optimization-theoretic and side-channel approach for attacking strong physical unclonable functions. IEEE Trans. Very Large Scale Integr. VLSI Syst. 26(1), 73–81 (2018)
22. Amsaad, F., et al.: Reliable delay based algorithm to boost PUF security against modeling attacks. Information. 9(9), 224 (2018)
23. Ma, Q., et al.: A machine learning attack resistant multi-PUF design on FPGA. In: 2018 23rd Asia and South Pacific design automation conference (ASP-DAC), pp. 97–104. IEEE, San Francisco, CA, USA (2018)
24. Cui, Y., et al.: Lightweight modeling attack-resistant multiplexer-based multi-PUF (MMPUF) design on FPGA. Electronics. 9(5), 815 (2020)
25. Khalafalla, M., Gebotys, C.: PUFs deep attacks: enhanced modeling attacks using deep learning techniques to break the security of double arbiter PUFs. In: 2019 Design, automation & test in Europe conference & exhibition (DATE), pp. 204–209. IEEE, Florence, Italy (2019)
26. Hori, Y., et al.: Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs. In: 2010 International conference on reconfigurable computing and FPGAs, pp. 298–303. IEEE, Washington, DC, USA (2010)
27. Su, Y., Holleman, J., Otis, B.P.: A digital 1.6 pj/bit chip identification circuit using process variations. IEEE J. of Solid-State Circuit. 43(1), 69–77 (2008)
28. Che, W., et al.: Analysis of entropy in a hardware-embedded delay puf. Cryptography. 1(1), 8 (2017)
29. Maiti, A., Kim, I., Schaumont, P.: A robust physical unclonable function with enhanced challenge-response set. IEEE Trans. Inf. Forensics Secur. 7(1), 333–345 (2011)
30. Alamro, M.A., et al.: Examination of double arbiter PUFs on security against machine learning attacks. In: 2019 IEEE international conference on Big Data (Big Data), pp. 3165–3171. IEEE, Los Angeles, CA, USA (2019)

How to cite this article: Tripathy, S., Rai, V.K., Mathew, J.: MARPUF: physical unclonable function with improved machine learning attack resistance. IET Circuits Devices Syst. 15(5), 465–474 (2021). https://doi.org/10.1049/cds2.12042