Computers & Security 168 (2026) 104958

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Full length article

PUF optimization for IoT device authentication

Raúl Aparicio-Téllez [a,*], Miguel Garcia-Bosque [a], Guillermo Díez-Señorans [a,b], Concepcion Aldea [a], Santiago Celma [a]

[a] Group of Electronic Design (GDE), I3A, University of Zaragoza, Spain
[b] Centro Universitario de la Defensa (CUD), Zaragoza, Spain

ARTICLE INFO

Keywords:
Authentication
Optimization
Physical unclonable function (PUF)
Reliability
Ring oscillator (RO)

ABSTRACT

Two critical aspects in IoT are security and resource consumption. Physically Unclonable Functions (PUFs) are hardware security primitives that exploit the inherent manufacturing variations of integrated circuits to generate unique identifiers. Due to their uniqueness and resistance to cloning, they are widely used for IoT device authentication. In this work, a novel approach to enhance the identifiability of compensated measurement PUFs is introduced. This method involves applying specific weight masks to the parameters extracted from the PUF entropy source before conducting comparisons to determine the output bit. This technique has been tested on several types of PUFs constructed using public datasets. As a result, it has been observed that the Equal Error Rate (*EER*) can be greatly improved, up to two orders of magnitude. The main advantage of this technique is that it does not modify the architecture of the compensated measurement PUF (which is interesting for IoT devices, as it does not require extra resources), while it also has proven to be generalizable, i.e., the optimal mask parameters can be found using a small set of devices and, then, generalized to a different bigger set of devices. This way, this proposal contributes to the development of novel authentication schemes for IoT devices, addressing both security and resource efficiency, critical issues in IoT environments.

1. Introduction

With the increasing growth of the Internet of Things (IoT), IoT networks must face a series of challenges including those resource limitations and a vast number of network nodes, making these systems vulnerable to security risks. Furthermore, cryptographic methods based on sharing secret keys have been proven to be vulnerable particularly in terms of protection against physical attacks and secure key generation and storage (Shamsoshoara et al., 2020; McGrath et al., 2019; Ning et al., 2020; Miah and Hossain, 2025; Kalam and Keshri, 2025). Traditionally, conventional cryptographic methods rely on storing a secret key in the memory of the IoT device, making it possible for an attacker who has access to the device to uncover the secret key.

To address this issue, Gassend et al. (2002) introduced for the first time the concept of "Physically Unclonable Function" (PUF). A PUF is a hardware security primitive that leverages the intrinsic variations occurring during the manufacturing process of devices to generate a kind of "digital fingerprint" that uniquely authenticates the IoT device (Maes, 2013; Lin and Liang, 2021). This key is only generated when needed for the authentication process, thus avoiding storing of secret information in non-volatile memories, as it is done in conventional cryptographic modules. This fact prevents the key from being exposed to several types of attacks, including invasive, semi-invasive, and side-channel attacks (Zerrouki et al., 2022).

A large number of PUFs use the compensated measurement technique, which consists of comparing parameters extracted from the PUF entropy source in pairs (Marchand et al., 2018; Li et al., 2020; Yao et al., 2021). This way, a much more stable response is obtained, as this technique contributes to reduce effects that similarly impact all entropy sources of the PUF, mainly variations of temperature and supply voltage changes of the devices (Maiti and Schaumont, 2011).

Among the diverse types of PUFs employed in IoT devices, delay-based PUFs and memory-based PUFs are the most widely implemented. Specifically, these include Arbiter PUF (Hemavathy and Bhaaskaran, 2023; Anandakumar et al., 2022); Ring Oscillator (RO) PUF (Baturone et al., 2023; Rahman et al., 2016); Latched Ring Oscillator (LRO) PUF (Della Sala et al., 2022); SRAM PUF (Ni et al., 2024; Shifman et al., 2019); and Configurable Tristate (CT) PUF (Zhang et al., 2022). Since these primitives are commonly used in authentication schemes for IoT devices (Mall et al., 2022; Farha et al., 2021; Modarres and Sarbishaei, 2023), the identifiability, which reflects how effectively a PUF can uniquely identify a device, is often a key metric to evaluate

* Corresponding author.
E-mail address: r.aparicio@unizar.es (R. Aparicio-Téllez).

https://doi.org/10.1016/j.cose.2026.104958
Received 30 July 2025; Received in revised form 11 March 2026; Accepted 7 May 2026
Available online 12 May 2026

PUF quality. To achieve a strong identifiability, a PUF must exhibit high reliability (the ability to consistently generate the same response to a specific challenge despite environmental or operational variations) and high uniqueness (the ability to produce distinct and unique responses to a given challenge across different devices).

One major research focus in PUF development has been the design and implementation of techniques to improve PUF reliability (Cook et al., 2023; Su et al., 2022; Santana-Andreo et al., 2024), especially in the face of environmental changes such as temperature or supply voltage fluctuations. Recent works have introduced strategies to enhance the reliability of specific PUF types, such as the Switched-Capacitor (SC) PUF (Wang et al., 2024) and PUFs based on Random Telegraph Noise (RTN) in transistors (Santos-Prieto et al., 2024). Importantly, improvements in reliability must be achieved without compromising the uniqueness of PUF, as both properties are essential for IoT device authentication (Idriss et al., 2021; Qureshi and Munir, 2022). In this regard, in Aparicio-Téllez et al. (2024), a novel approach to improve identifiability of the PUF was proposed. However, this approach required a significant amount of resources, a critical aspect for IoT devices, and its generalizability was not demonstrated.

In this work, we present a novel optimization technique to improve the identifiability of compensated measurement PUFs in IoT devices. Existing methods to enhance PUF identifiability often require significant hardware resources or per-device parameter tuning, which is impractical for resource-constrained IoT devices. Our approach addresses this limitation by optimizing the PUF parameters on a small subset of devices and generalizing the results to a larger population, eliminating the need for individual device tuning. This strategy not only improves identifiability — reducing the Equal Error Rate (EER) by up to two orders of magnitude in some cases — but also requires no additional hardware resources, making it highly suitable for practical IoT authentication schemes.

The main contributions of this work are:

- The development of a generalizable optimization technique for compensated measurement PUFs, which allows for improved identifiability using only a small training set of devices.
- The implementation and evaluation of this technique across multiple PUF types using public datasets, demonstrating significant identifiability improvements without additional resource overhead.

This paper is organized as follows: in Section 2, the compensated measurement technique and some important PUF metrics are introduced; in Section 3, the optimization technique is explained as well as its theoretical justification; Section 4, describes the optimization process; Section 5, applies the proposed technique to different types of PUFs and analyzes the identifiability improvement; finally, conclusions are drawn in Section 6.

2. Background

2.1. Related work

Regarding hardware architecture modifications, some works propose physical cell changes to inherently boost reliability. On the one hand, Su et al. (2022) designs a custom 8T SRAM cell to improve radiation tolerance and lower the Bit Error Rate (BER) by half compared to standard 6T cells. On the other hand, Wang et al. (2024) exploits the configurability of capacitors in an SC-PUF to suppress the offset of the sense amplifier, which is a major source of instability in advanced FinFET processes.

Regarding bit selection, some PUF optimization studies focus on identifying the most robust entropy units before PUF deployment. In particular, Santana-Andreo et al. (2024) uses Data Retention Voltage (DRV) as a predictive tool to select SRAM cells that will remain stable even after significant aging (considering BTI and the often-overlooked NC-HCI effects). Other works such as Aparicio-Téllez et al. (2023b), significantly increase the number of possible configurations in a CRO-PUF, allowing them to select pairs with such high frequency differences that the PUF achieves 100% reliability, potentially eliminating the need for complex Error Correcting Codes (ECC).

Other works like Santos-Prieto et al. (2024) and Wang et al. (2024) also address environmental stability: mainly temperature and voltage fluctuations. The former uses a temperature-aware tuning of the harvesting function based on the Arrhenius activation energy of defects, while the latter ensures that their SC-PUF units remain valid as anti-invasive attack shields across a wide temperature range. In Table 1, the main strategies discussed in all these works are compared, including hardware architecture modifications, bit selection, and stability techniques against environmental variations. Furthermore, together with Aparicio-Téllez et al. (2024), this is the only work that focuses on identifiability while simultaneously improving uniqueness and reliability.

Finally, a comparison of this work with Aparicio-Téllez et al. (2024) is shown in Table 2. As can be seen, the proposed method improves PUF identifiability more effectively, reducing the *EER* by up to two orders of magnitude, whereas the approach in Aparicio-Téllez et al. (2024) requires a very specific nonlinear transformation (NLT). Additionally, the proposed method does not require changes to the PUF architecture or extra hardware, unlike the NLTs. The optimization of mask values and block structure allows for more direct and efficient tuning compared to function coefficient optimization.

2.2. Compensated measurement technique

In PUFs that contain multiple identical entropy sources (such as an array of identical ring oscillators), a compensated measurement technique is often used to ensure reliable and stable PUF responses, despite potential environmental variations (e.g. temperature, voltage, and aging effects). With this technique, which was first introduced by Gassend et al. (2002), instead of directly using the raw measured values (such as the oscillation frequencies of each ring oscillator) to generate the PUF response, a pairwise comparison approach is employed.

With this technique, the entropy sources are grouped into pairs. For each pair, a binary response is derived by comparing the values of the parameters within that pair. Specifically, if the parameter (e.g., frequency) of the first source in a pair is higher than the parameter of the second source, the output bit b_{ab} is set to 1; otherwise, it is set to 0:

$$b_{ab} = \begin{cases} 1 & f_a > f_b \\ 0 & f_a \leq f_b \end{cases} \quad (1)$$

This relative comparison between pairs helps to minimize the impact of environmental fluctuations on the response, a critical requirement for ensuring reliable performance in IoT devices, where operating conditions can vary significantly. This makes the output more stable and reliable over time and across varying environments. In Fig. 1, the architecture of a PUF using the compensated measurement technique is shown.

2.3. Main PUF properties

To be suitable for IoT device identification and authentication, a PUF must satisfy two essential properties: reliability and uniqueness.

Reliability measures the ability of a PUF to consistently produce the same response to a given challenge under varying environmental conditions, such as temperature fluctuations or supply voltage variations in IoT devices. A commonly used metric to quantify PUF reliability is the intra-chip Hamming Distance (intra-*HD*). Given two responses, $Y_j(x)$ and $Y'_j(x)$, generated by the same PUF device j under the same

Table 1
Comparison of state-of-the-art PUF designs, their strategies, techniques, target metrics, and implementation platforms.

Work	PUF type	Primary strategy	Technique	Target metric	Platform/ Process
Su et al. (2022)	SRAM	Custom 8T Cell	Adding two cascode pMOS transistors to standard 6T cells	Reliability	28 nm FDSOI
Cook et al. (2023)	CRO	Configuration Search	Maximizing frequency difference through a massive search space	Reliability	FPGA (Artix-7)
Santana-Andreo et al. (2024)	SRAM	DRV-based Bit Selection	Identifying stable cells using Data Retention Voltage (DRV) metrics before deployment	Aging Resilience	65 nm CMOS
Santos-Prieto et al. (2024)	RTN-based	Evolutionary Selection and Tuning	Genetic algorithms for pairing and temperature-aware tuning	Reliability	65 nm CMOS
Wang et al. (2024)	Switched-Capacitor (SC)	Capacitor Reconfiguration	Optimal configuration of four capacitors + LSSA offset suppression	Reliability	12 nm FinFET
Aparicio-Téllez et al. (2024)	Different PUFs	Nonlinear Transformation	Applying polynomial transformations to parameters extracted from entropy source	Identifiability	Public databases and FPGA
This work	Different PUFs (RO, TERO, etc.)	Weight Masks	Applying specific additive values to parameters before bit comparison	Identifiability (Uniqueness + Reliability)	Public databases and FPGA

Table 2
Comparison between the approach in Aparicio-Téllez et al. (2024) and the proposed weight mask technique for improving PUF identifiability.

Aspect	Aparicio-Téllez et al. (2024)	This work
Basic idea and operation	Applies a NLT to each extracted parameter before comparison.	Applies a weight mask that modifies the parameters before performing the comparisons.
Transformation approach	Mathematical transformation applied individually to each parameter.	Structured modification of parameter blocks using a sliding mask.
Implementation	Requires hardware logic to compute the nonlinear function.	Does not modify the PUF architecture and some architectures require no additional hardware resources.
Optimization	Optimization of function coefficients through training.	Optimization of mask values and dimensions.
Identifiability improvement	Improvement depends on the selected transformation.	Can reduce the EER by up to two orders of magnitude.

challenge x, the intra-*HD* is defined as: $HD_j^{\text{intra}}(x) = HD(Y_j(x), Y_j'(x))$, where *HD* is the "Hamming Distance", which quantifies the number of differing bits between two binary sequences Y, Y' of the same length N. Ideally, this value should be 0%.

Uniqueness measures how distinct the responses of the same type of PUF are across different devices. The most commonly used metric to quantify uniqueness is the inter-chip Hamming Distance (inter-*HD*). Given two responses $Y_i(x)$ and $Y_j(x)$ generated by the same type of PUF in two different devices, i and j ($i \neq j$), under the same challenge x, the inter-*HD* is defined as: $HD_{i,j}^{\text{inter}}(x) = HD(Y_i(x), Y_j(x)) \quad i \neq j$, Ideally, its average value should be 50%.

In a PUF-based authentication scheme, the response is compared with a previously saved response from the IoT device. If the distance between them is less than or equal to a previously established identification threshold, t_{id}, the device is positively identified. Otherwise, the response is considered to be from a different device. Typically, the effectiveness of any authentication system is measured with False Rejection Rate (*FRR*) and False Acceptance Rate (*FAR*). While *FRR* gives the probability of an authentication attempt to result in a false rejection, *FAR* gives the probability of an authentication attempt to result

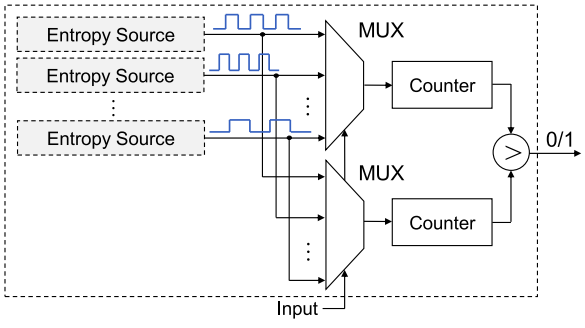

Fig. 1. Conventional compensated measurement PUF architecture.

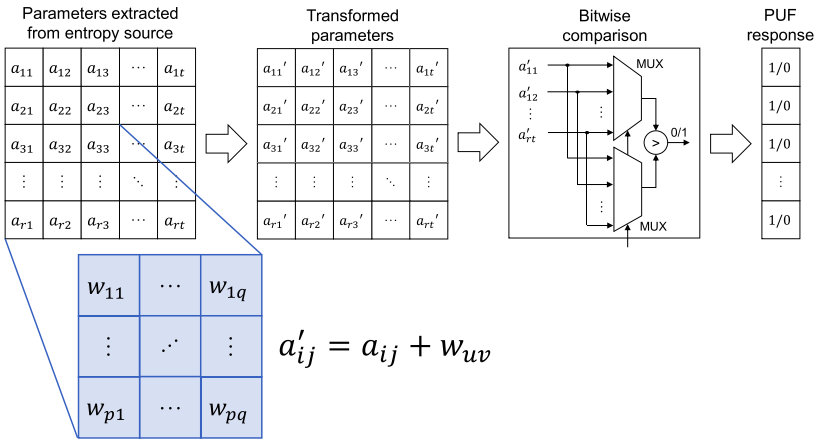

Fig. 2. Proposed PUF identifiability optimization technique.

in a false acceptance. Both parameters are mathematically defined:

$$FRR(t_{\text{id}}) = 1 - F_{\text{bino}}(\hat{p}_P^{\text{intra}}), \quad FAR(t_{\text{id}}) = F_{\text{bino}}(\hat{p}_P^{\text{inter}}), \tag{2}$$

where $F_{\text{bino}}(\hat{p}_P^{\text{intra}})$ and $F_{\text{bino}}(\hat{p}_P^{\text{inter}})$ are the Cumulative Distribution Function (CDF) of intra-*HD* and inter-*HD* respectively.

Both rates should be as low as possible, but they cannot be minimized at the same time as when one rate is improved, the other is worsened and vice-versa. Typically, an identification threshold value for which both *FAR* and *FRR* are approximately equal is selected. This threshold is the "equal error threshold" (t_{EER}). For discrete distributions, the t_{EER} is defined: $t_{EER} = \text{argmin}_t\{\max\{FAR(t_{\text{id}}), FRR(t_{\text{id}})\}\}$, and the corresponding probability is the "equal error rate" (*EER*):

$$EER = \max\{FAR(t_{EER}), FRR(t_{EER})\}, \tag{3}$$

From a practical point of view, identifiability, measured with the *EER*, is the most important property of a PUF as it provides the probability of an authentication attempt to result in a false rejection and false acceptance. Furthermore, it simultaneously combines the properties of uniqueness and reliability. This way, this work focuses on proposing a novel strategy specifically designed for IoT devices to improve the *EER* without requiring significant additional resources.

3. Proposal

3.1. Description

In this work, the main idea is to add a certain quantity to the parameters extracted from the PUF entropy source before performing the comparison to obtain the output bit. This way, given two parameters f_a, f_b, the novel output bit b'_{ab} result of the comparison is given by:

$$b'_{ab} = \begin{cases} 1 & f_a > f_b + \alpha \\ 0 & f_a \leq f_b + \alpha \end{cases} \tag{4}$$

α does not have to be the same for each parameter. Therefore, the value α added to each parameter is determined by the so-called created "weight masks".

Given a PUF with N_p parameters a_{ij}, it is possible to arrange them into a $r \times t$ matrix. As it can be seen in Fig. 2, the proposal consists of adding a certain value w_{uv} to the parameters a_{ij} obtained from the PUF entropy source. This way, the transformed parameters a'_{ij} are given by the expression:

$$a'_{ij} = a_{ij} + w_{uv} \quad \forall w_{uv} \in \mathbb{R}. \tag{5}$$

The added values w_{uv} are determined from a weight mask of size $p \times q$ with $p \leq r$, $q \leq t$ which is displaced around the original parameters matrix. As it can be seen in the example of Fig. 3(a), the weight mask shifts in blocks over the larger matrix, applying a single transformation to each parameter at each block position. Furthermore, as it is shown in Fig. 3(b), depending on the dimensions of the weight mask, when the mask is shifted it may extend beyond the bounds of the larger matrix. In those cases, the transformation is applied only to the parameters that remain within the weight mask. Once performed the transformation, all a'_{ij}'s are compared in pairs to obtain the PUF response. For simplicity, in this work, weight masks in the form of a $1 \times q$ array have been used.

This transformation is equivalent to initializing some of the memories that store the measured property of the PUF primitive to a certain value (e.g. frequency counters in case of the RO-PUF), as it can be seen in Fig. 4(a). In particular, when 1×2 or 2×1 masks are used, the compensated measurement PUF architecture remains unaltered (Fig. 4(b)) as only one of the counters is being initialized.

3.2. Theoretical justification

There are different ways to compare the parameters extracted from the PUF entropy source to obtain the binary response. One of the most widely used topologies involves comparing all parameters with each other, allowing a response of up to $N_p(N_p - 1)/2$ bits from N_p parameters. Another topology involves repeating one parameter in each comparison, thus obtaining responses of up to $N_p - 1$ bits. However, in both topologies, the resulting bits are not completely independent (Diez-Senorans et al., 2021). To achieve independent bits, a simple strategy is to compare parameters without repeating any in each comparison, thus obtaining responses of up to $N_p/2$ bits. This is the most common strategy and the one that will be used in this work. Furthermore, it is worth noting that the properties of inter-*HD* and intra-*HD* are dependent on the comparison topology used to construct the PUF.

In Aparicio-Téllez et al. (2023b), two general parameters to measure the uniqueness and reliability of PUF based on the compensated measurement technique were proposed. Unlike intra-*HD* and inter-*HD*, these parameters were independent on the comparison topology used. Let f_{ijk} be the parameter obtained from the entropy source of the PUF at repetition i, location j and device k. Furthermore, it is supposed that N_i repetitions are performed in N_j locations of N_k different devices.

This way, the reliability quality indicator is defined as the percentage of bits which meet the condition:

$$\left| \frac{\sigma_{\text{dif}}^{jj'k}}{\Delta F^{jj'k}} \right| < t_{\text{repr}} \tag{6}$$

where $F^{jk} = \frac{1}{N_i}\sum_{i=1}^{N_i} f_{ijk}$, σ_{rep}^{jk} is the standard deviation of F^{jk}, $\Delta F^{jj'k} = F^{jk} - F^{j'k}$, $\sigma_{\text{dif}}^{jj'k} = \sqrt{(\sigma_{\text{rep}}^{jk})^2 + (\sigma_{\text{rep}}^{j'k})^2}$, and t_{repr} is the threshold used to determine if a bit is reliable.

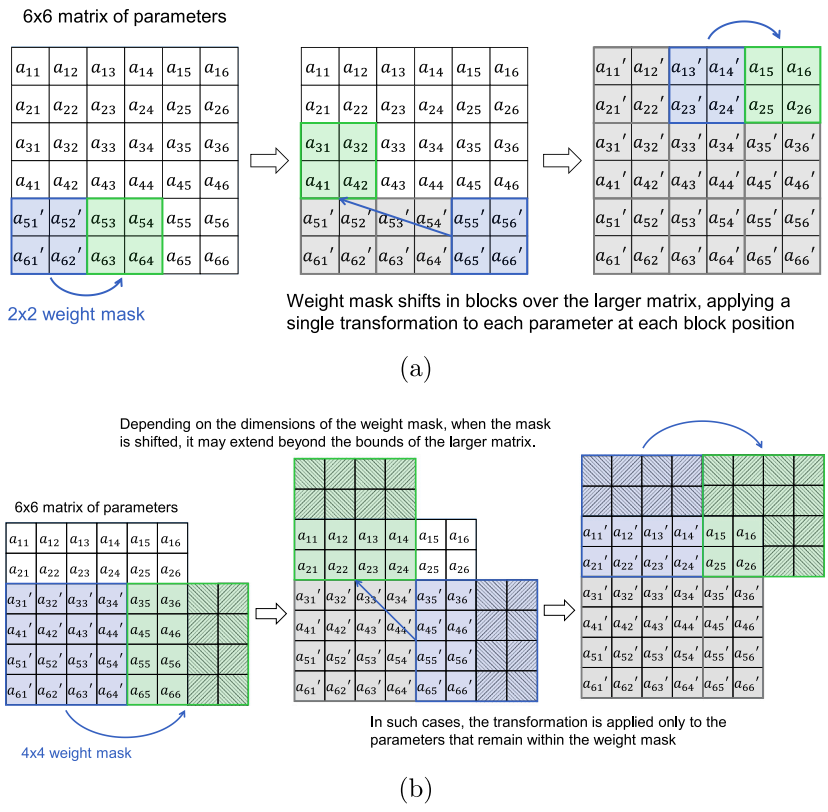

Fig. 3. Examples of matrix of 6 × 6 parameters using (a) 2 × 2 weight mask and (b) 4 × 4 weight mask.

Similarly, the uniqueness quality indicator was defined as the percentage of bits which meet the condition:

$$\left| \frac{\sigma_{\text{inter}}^{jj'}}{\overline{\Delta F^{jj'}}} \right| > t_{\text{uniq}} \tag{7}$$

where $\overline{\Delta F^{jj'}} = \frac{1}{N_k} \sum_{k=1}^{N_k} \Delta F^{jj'k}$, $\sigma_{\text{inter}}^{jj'}$ is the standard deviation of $\overline{\Delta F^{jj'}}$, and t_{uniq} is the threshold used to determine if a bit is unique or not. It is evident that a PUF must meet both conditions (6) and (7) simultaneously.

In this work, it is proposed to add certain values to the parameters obtained from the entropy source before performing the comparison to obtain the PUF response. To determine the effect on uniqueness and reliability, parameters in locations j remain unchanged $f_{ijk} \rightarrow f_{ijk}$ while a certain value α is added to parameters in locations j' $f_{ij'k} \rightarrow f_{ij'k}^{\alpha} = f_{ij'k} + \alpha$. This is equivalent to the architecture previously shown in Fig. 4(b). This way, $F_{\alpha}^{j'k} = F^{j'k} + \alpha$, $\sigma_{\text{rep}\ \alpha}^{j'k} = \sigma_{\text{rep}}^{j'k}$, $\Delta F_{\alpha}^{jj'k} = \Delta F^{jj'k} - \alpha$, $\sigma_{\text{dif}\ \alpha}^{jj'k} = \sigma_{\text{dif}}^{jj'k}$, and the novel condition of reliability is:

$$\left| \frac{\sigma_{\text{dif}}^{jj'k}}{\Delta F^{jj'k} - \alpha} \right| < t_{\text{repr}}^{\alpha}. \tag{8}$$

Similarly, $\overline{\Delta F_{\alpha}^{jj'}} = \overline{\Delta F^{jj'}} - \alpha$, $\sigma_{\text{inter}\ \alpha}^{jj'} = \sigma_{\text{inter}}^{jj'}$, and the novel condition of uniqueness is:

$$\left| \frac{\sigma_{\text{inter}}^{jj'}}{\overline{\Delta F^{jj'}} - \alpha} \right| > t_{\text{uniq}}^{\alpha}. \tag{9}$$

As it can be seen in (8) and (9), depending on the value of α, uniqueness or reliability can be enhanced although it is not possible to enhance both properties simultaneously. However, one property can be improved (e.g. reliability) and the other can be worsened (e.g. uniqueness) so that the overall result is an improvement of the PUF identifiability.

In Fig. 5, the *EER* of different pairs of μ^{intra} and μ^{inter} has been represented using 128-bit responses and assuming ideal binomial distributions. As it can be seen, it can be found a pair of points (A_1, B_1) where an improvement of the uniqueness and a worsening of the reliability results in an improvement of the PUF identifiability, for example:

$$A_1 = (1.3,\ 34.0,\ 1.67 \cdot 10^{-9}), \tag{10}$$

$$B_1 = (1.8,\ 48.5,\ 2.22 \cdot 10^{-14}). \tag{11}$$

Similarly, it can also be found a pair of points (A_2, B_2) where an improvement of the reliability and a worsening of the uniqueness results in an improvement of the identifiability, for example:

$$A_2 = (1.6,\ 49.5,\ 2.33 \cdot 10^{-15}), \tag{12}$$

$$B_2 = (0.5,\ 47.0,\ 1.30 \cdot 10^{-17}). \tag{13}$$

Based on this justification, identifiability improvement of different PUFs is measured using the so-called "Enhancement Factor" (EF), defined as the division between the *EER* without applying the transformation (EER_{before}) and after applying the transformation (EER_{after}):

$$EF = \frac{EER_{\text{before}}}{EER_{\text{after}}} \tag{14}$$

A value of $EF > 1$ will mean that the identifiability of the PUF has been improved while $EF \leq 1$ will mean that it has remained equal or worsened.

4. Methodology

4.1. Datasets

To test this proposal, four databases which provide the raw frequencies of different types of oscillators have been used to construct different types of PUFs:

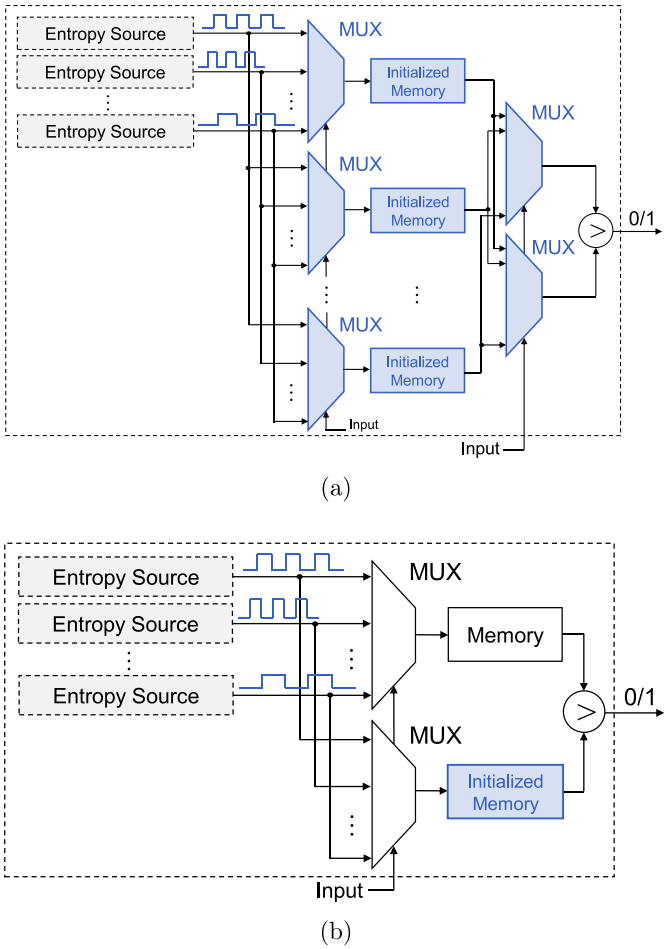

Fig. 4. Proposed PUF architectures based on the Compensated Measurement Technique. (a) Proposed optimization using constant masks. (b) Proposed optimization using constant masks with one parameter is equivalent to initialize odd (or even) counters to a certain pre-fixed value.

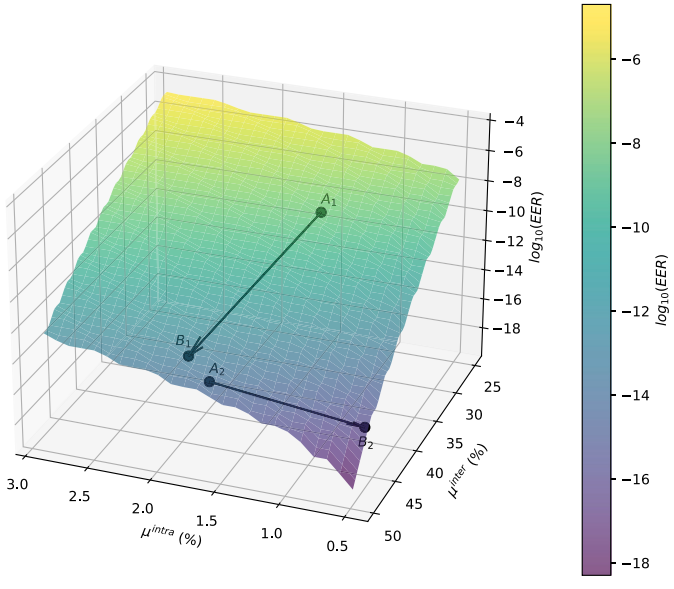

Fig. 5. *EER* depending on μ^{inter} and μ^{intra} using 128-bit PUF responses.

1. Maiti et al. database: This database has been obtained from Maiti et al. (2010). It contains the frequencies of 5-LUT ring oscillators in 125 different Spartan3E S500 FPGAs. Oscillators are distributed in a 16 × 32 matrix and each frequency has been measured 100 times. Moreover, each ring oscillator is implemented in a single Configurable Logic Block (CLB). This database will be referred as HOST10 database.
2. Wild et al. database: This database has been obtained from Wild et al. (2017). It contains the frequencies of three oscillation-based architectures used to construct different types of PUFs: Ring Oscillator PUF (RO-PUF), Transient Effect Ring Oscillator PUF (TERO-PUF), and three Loop-PUFs (Loop-Latch, Loop-Wire, and Loop-PDL). Each architecture has been implemented in a 16 × 80 matrix of 100 Artix-7 FPGAs. To analyze random temporal noise, the authors also measured these frequencies twice at standard nominal temperature. Furthermore, two more extra measurements can be extracted from this database for RO-PUF, TERO-PUF and Loop-Latch PUF frequencies. This database will be referred as FPL17 database.
3. Hesselbarth et al. database: This database has been obtained from Hesselbarth et al. (2018). It contains the frequencies of 6592 3-LUT ring oscillators implemented in 217 Artix-7 FPGAs. Each frequency has been measured 100 times in four different parts of the FPGA: Left-Lower (LL), Left-Upper (LU), Right-Lower (RL) and Right-Upper (RU). In this works, authors mention that the frequency of oscillators are strongly dependent on their location on the FPGA (LL, LU, RL or RU). Therefore, four different weak PUFs have been constructed in each part. This database will be referred as HOST18 database.
4. Aparicio et al. database: This data have been obtained from Aparicio-Téllez et al. (2023a). In this work, among other experiments, 200 11-LUT and 3-LUT ring oscillators were implemented in 40 different Artix-7 FPGAs. The frequency of each oscillator was measured 100 times. Furthermore, the location and routing of oscillators were fixed so that all of them were as identical as possible. This database will be referred as SNS23 database.

4.2. PUF response generation

Given the raw frequencies of the oscillators, the PUF response has been obtained for each case using software tools. Oscillators have been compared using the non-overlapping comparison topology thus ensuring that no correlation exists between consecutive bits of the PUF response and obtaining $N_p/2$ bit-length responses.

4.3. Optimization

They key idea is to use approximately 20% of FPGAs during the training process to determine the optimal values of the weight mask and then use the other 80% of data to validate the results using the weight mask obtained during the training process. The first dataset will be referred as "training dataset" while the second one will be the "validation dataset". This way, it can be ensured that the obtained optimization parameters using a small set of devices can be generalized to other bigger set of devices.

It is worth noting that the optimization stage is performed using only a subset of devices because the objective is to determine transformation parameters associated with the PUF architecture rather than with individual devices. Since all devices share the same hardware design and differ only due to manufacturing variability, a representative subset is sufficient to estimate parameters that can later be applied to the rest of the population. The proposed transformation does not alter the underlying entropy sources but only applies a deterministic mask before generating the response bit, thus preserving the intrinsic physical behavior of the PUF. This strategy is particularly suitable for IoT–cloud scenarios, where a large number of devices manufactured

Table 3
Number of FPGAs (N_{FPGA}), repetitions (N_{Meas}) and oscillators (N_{Osc}) of the complete, training and validation datasets.

Dataset	Data	HOST10 (Maiti et al., 2010)	FPL17 (Wild et al., 2017)	HOST18 (Hesselbarth et al., 2018)	SNS23 (Aparicio-Téllez et al., 2023a)
Full dataset	N_{FPGA}	125	100	217	40
	N_{meas}	100	4/2[1]	100	100
	N_{osc}	512	1280	6592	200
Training	N_{FPGA}	20	20	20	8
	N_{meas}	100	4/2[1]	100	100
	N_{osc}	256	256	256	200
Validation	N_{FPGA}	100	80	80	32
	N_{meas}	100	4/2[1]	100	100
	N_{osc}	256	256	256	200

[1]Using available public dataset, four repetitions can be extracted for RO, TERO and Loop-Latch PUFs at 22°C while only two for Loop-Wire and Loop-PDL PUFs.

with the same design must be authenticated by a centralized infrastructure, allowing the optimization to be performed once during enrollment and then generalized across the entire device population.

In Table 3, the number of data used for training and validation is shown as well as the total number of data of the dataset. As observed, for each database and type of PUF, a weak PUF with 256 oscillators has been constructed (resulting in 128-bit responses), except for the SNS23 database, where 200 oscillators were used, generating 100-bit responses. Although with the HOST10, FPL17, and HOST18 databases, longer responses could be generated, the choice to use 128-bit responses aims to strike a balance: ensuring the response is long enough to make the PUF secure, while not so long as to significantly increase the authentication time. This response length is also a standard practice in PUFs.

The optimization target will be the *EER*. The objective function to be optimized takes as input an array of length n: $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ and calculates the *EER* based on the transformed values obtained from the entropy source of the PUF. The parameter n corresponds to the number of parameters of the weight mask used in the optimization process. As parameters are being compared using the non-overlapping comparison topology, parameters with odd index remain unalterable. Therefore, a $1 \times q$ weight mask will have $n = q/2$ different parameters. In this case, optimizations have been carried out with up to $n = 4$ parameters. This is because it has been observed that using only $n = 1$ is sufficient to improve the identifiability of the PUF. Moreover, using a larger number of parameters complicates the optimization process and significantly increases the time required to complete the optimization. In this way, for each vector $\vec{\alpha}$, an *EER* value is obtained.

The optimization process is carried out using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (Fletcher, 2000), implemented in Python's "scipy.optimize" library. This method uses the gradient of the objective function, in this case $EER(\vec{\alpha})$, to converge more quickly to the solution. However, if the gradient is not provided, as in this case, the algorithm estimates it using finite differences. Compared to other algorithms such as the Nelder–Mead Simplex algorithm, the BFGS algorithm requires fewer function calls, even when estimating the gradient (SciPy, 2024). Additionally, to obtain a better solution, the optimization system with multiple starts has been implemented, generating a random array $\vec{\alpha}$ of size n in each start.

One of the aspects that requires special attention in this algorithm lies in its sensitivity to initial conditions. For this reason, the values of the databases have been scaled down by a certain reduction factor, which varies for each case, to ensure that the values are not too high. Additionally, the initial prediction from which the algorithm starts has been carefully selected, taking into account the order of magnitude of the differences in oscillator frequencies between FPGAs after applying the corresponding reduction factor. Furthermore, the local optimization algorithm BFGS has been combined with the global optimization algorithm "Basin-hopping" in order to reduce sensitivity to initial conditions and converge faster to an optimal solution.

5. Results

5.1. Identifiability analysis

To estimate identifiability, the average intra-*HD* (μ^{intra}) has been calculated for each FPGA and the average inter-*HD* (μ^{inter}) has been obtained for all FPGAs for both training and validation sets. Then, the *FAR* and *FRR* curves have been derived. Finally, the *EER* has been obtained using (3) and PUF identifiability improvement has been analyzed using EF factor.

In Table 4, the PUF properties obtained using the training and validation datasets respectively have been calculated. This way, the average intra-*HD* (μ^{intra}), the average inter-*HD* (μ^{inter}) and the *EER* has been obtained for each of the PUFs constructed using the four mentioned databases. Furthermore, the enhancement factor (*EF*) has been calculated using (14). In those cases where a mask has not been applied, it is evident that $EF = 1$. As mentioned in the previous section, the non-overlapping comparison topology has been used to construct the PUF response. Using this approach, if we have two oscillators a and b, adding different values to both oscillators is equivalent to keeping the parameter of oscillator a unchanged while adding a different value to oscillator b. For this reason, no transformation has been applied to odd oscillators. Therefore, for each pair of oscillators (a, b), there is only one parameter to optimize. The table also shows the results obtained using masks with different numbers of parameters n.

In general, it is observed that, in all cases, the identifiability of the PUF can be increased, reducing the *EER* by up to two orders of magnitude for certain types of PUFs and masks. Another observation is that, in general, the *EF* obtained with the training sets tends to be higher than those obtained with the validation sets. This aligns with expectations, as the optimization and search for the optimal parameters are carried out with the first set, with the aim of generalizing the results to datasets with a larger number of FPGAs. In any case, it must be pointed out that significant improvements are obtained in the validation sets, proving that the optimization is generalizable. Finally, it is also observed that some masks tend to increase the reliability of the PUF while worsening the uniqueness, whereas in other cases, the uniqueness improves while the reliability decreases, as previously explained. The results obtained with each of the four datasets are analyzed below.

Regarding the HOST10 database, in the training set, the first observation is that as the number of mask parameters n increases, the *EF* factor also increases. However, in the validation set, it is also observed that for $n = 1$, $n = 2$ and $n = 4$, a similar factor is obtained, $EF \sim 2.01$. This indicates that for this database, using masks with two parameters ($n = 2$) would suffice, as increasing the number of parameters does not lead to a lower *EER*. This is partly due to the fact that the optimization problem becomes more complex as the number of parameters increases.

Table 4
Average intra-*HD* (μ^{intra}) in %, average inter-*HD* (μ^{inter}) in %, equal error rate (*EER*) and enhancement factor (*EF*) for each type of PUF constructed using the four databases: HOST10, FPL17, HOST18, SNS23. The parameter n refers to the number of parameters used in the optimization mask.

PUF type	Property	Training					Validation				
		No mask	n=1	n=2	n=3	n=4	No mask	n=1	n=2	n=3	n=4
HOST10											
RO-PUF	μ^{intra}	1.11	1.07	1.19	0.97	1.01	1.29	1.23	1.15	1.19	1.20
	μ^{inter}	45.94	45.65	46.74	45.89	45.94	46.79	46.79	46.79	46.62	46.79
	EER	$8.22 \cdot 10^{-15}$	$4.95 \cdot 10^{-15}$	$4.22 \cdot 10^{-15}$	$3.29 \cdot 10^{-15}$	$3.02 \cdot 10^{-15}$	$7.77 \cdot 10^{-15}$	$3.87 \cdot 10^{-15}$	$3.87 \cdot 10^{-15}$	$5.16 \cdot 10^{-15}$	$3.87 \cdot 10^{-15}$
	EF	1.00	1.66	1.95	2.50	2.72	1.00	**2.01**	**2.01**	**1.51**	**2.01**
FPL17											
RO-PUF	μ^{intra}	1.39	1.39	1.39	1.39	1.86	1.52	1.58	1.55	1.64	1.65
	μ^{inter}	44.12	46.60	47.21	46.37	47.02	44.75	46.39	46.52	46.31	46.31
	EER	$3.03 \cdot 10^{-13}$	$2.70 \cdot 10^{-14}$	$9.96 \cdot 10^{-15}$	$2.70 \cdot 10^{-14}$	$6.72 \cdot 10^{-14}$	$1.29 \cdot 10^{-13}$	$3.87 \cdot 10^{-14}$	$3.11 \cdot 10^{-14}$	$4.45 \cdot 10^{-14}$	$4.87 \cdot 10^{-14}$
	EF	1.00	11.23	30.44	11.23	4.51	1.00	**3.33**	**4.15**	**2.90**	**2.65**
TERO-PUF	μ^{intra}	3.05	2.53	2.71	2.73	2.73	2.84	2.84	2.84	2.84	2.84
	μ^{inter}	46.50	46.50	46.50	46.50	46.50	48.28	48.75	48.80	48.78	48.72
	EER	$8.51 \cdot 10^{-12}$	$1.66 \cdot 10^{-12}$	$3.02 \cdot 10^{-12}$	$3.02 \cdot 10^{-12}$	$3.02 \cdot 10^{-12}$	$8.19 \cdot 10^{-13}$	$3.90 \cdot 10^{-13}$	$3.61 \cdot 10^{-13}$	$3.72 \cdot 10^{-13}$	$4.09 \cdot 10^{-13}$
	EF	1.00	5.13	2.81	2.81	2.81	1.00	**2.10**	**2.27**	**2.20**	**2.00**
Loop-Latch-PUF	μ^{intra}	2.88	2.12	2.88	2.37	2.55	2.70	2.60	3.00	2.55	2.50
	μ^{inter}	24.70	23.24	30.95	23.89	24.22	24.28	23.49	32.16	23.85	23.91
	EER	$2.74 \cdot 10^{-05}$	$1.97 \cdot 10^{-05}$	$9.25 \cdot 10^{-07}$	$2.11 \cdot 10^{-05}$	$2.67 \cdot 10^{-05}$	$4.17 \cdot 10^{-05}$	$3.29 \cdot 10^{-05}$	$3.32 \cdot 10^{-07}$	$2.66 \cdot 10^{-05}$	$2.16 \cdot 10^{-05}$
	EF	1.00	1.39	29.64	1.30	1.03	1.00	**1.27**	**125.60**	**1.57**	**1.93**
Loop-Wire-PUF	μ^{intra}	4.06	4.06	7.62	4.10	4.10	4.32	4.05	7.12	4.10	4.08
	μ^{inter}	25.53	25.72	40.69	25.69	25.97	25.46	25.46	40.07	25.67	25.88
	EER	$9.16 \cdot 10^{-05}$	$7.60 \cdot 10^{-05}$	$1.14 \cdot 10^{-06}$	$7.79 \cdot 10^{-05}$	$7.59 \cdot 10^{-05}$	$1.36 \cdot 10^{-04}$	$9.76 \cdot 10^{-05}$	$1.08 \cdot 10^{-06}$	$9.76 \cdot 10^{-05}$	$9.76 \cdot 10^{-05}$
	EF	1.00	1.21	80.35	1.18	1.21	1.00	**1.39**	**125.93**	**1.39**	**1.39**
Loop-PDL-PUF	μ^{intra}	6.21	7.58	7.73	7.11	7.77	6.45	7.49	7.83	7.31	7.67
	μ^{inter}	38.14	43.68	47.87	42.83	46.45	37.46	42.91	47.85	42.86	46.53
	EER	$1.14 \cdot 10^{-06}$	$2.04 \cdot 10^{-07}$	$1.08 \cdot 10^{-08}$	$1.95 \cdot 10^{-07}$	$3.93 \cdot 10^{-08}$	$2.17 \cdot 10^{-06}$	$2.38 \cdot 10^{-07}$	$1.23 \cdot 10^{-08}$	$1.89 \cdot 10^{-07}$	$2.89 \cdot 10^{-08}$
	EF	1.00	5.59	105.56	5.85	29.01	1.00	**9.12**	**176.42**	**11.48**	**75.09**
HOST18											
RO-PUF (LL)	μ^{intra}	1.03	0.83	0.71	1.03	0.74	0.98	0.89	0.95	0.91	0.95
	μ^{inter}	45.07	45.07	45.04	47.34	45.05	46.80	46.79	46.75	47.38	46.77
	EER	$2.18 \cdot 10^{-15}$	$3.33 \cdot 10^{-16}$	$8.10 \cdot 10^{-17}$	$1.52 \cdot 10^{-15}$	$4.26 \cdot 10^{-16}$	$6.14 \cdot 10^{-16}$	$9.89 \cdot 10^{-17}$	$3.28 \cdot 10^{-16}$	$2.50 \cdot 10^{-16}$	$2.12 \cdot 10^{-16}$
	EF	1.00	6.55	26.91	1.44	5.12	1.00	**6.21**	**1.87**	**2.46**	**2.90**
RO-PUF (LU)	μ^{intra}	0.98	0.98	1.04	0.80	0.89	1.04	0.94	0.96	1.01	0.97
	μ^{inter}	46.20	48.64	48.69	46.02	46.20	49.00	49.00	49.02	48.36	48.36
	EER	$6.63 \cdot 10^{-16}$	$1.55 \cdot 10^{-16}$	$1.43 \cdot 10^{-16}$	$2.07 \cdot 10^{-16}$	$3.14 \cdot 10^{-16}$	$1.11 \cdot 10^{-16}$	$8.24 \cdot 10^{-17}$	$7.94 \cdot 10^{-17}$	$5.02 \cdot 10^{-17}$	$3.53 \cdot 10^{-17}$
	EF	1.00	4.28	4.64	3.20	2.11	1.00	**1.35**	**1.40**	**2.21**	**3.14**
RO-PUF (RL)	μ^{intra}	1.14	1.14	1.14	1.20	0.96	0.87	0.90	0.88	0.90	0.91
	μ^{inter}	46.81	46.86	46.92	46.81	46.81	47.32	47.33	47.34	47.32	47.35
	EER	$3.75 \cdot 10^{-15}$	$3.45 \cdot 10^{-15}$	$3.08 \cdot 10^{-15}$	$3.62 \cdot 10^{-15}$	$6.80 \cdot 10^{-16}$	$2.79 \cdot 10^{-16}$	$2.75 \cdot 10^{-16}$	$2.71 \cdot 10^{-16}$	$2.79 \cdot 10^{-16}$	$2.67 \cdot 10^{-16}$
	EF	1.00	1.09	1.22	1.04	5.51	1.00	**1.01**	**1.03**	**1.00**	**1.04**
RO-PUF (RU)	μ^{intra}	0.96	0.74	0.76	0.69	0.77	0.96	0.95	0.96	0.97	0.93
	μ^{inter}	47.19	47.08	47.04	47.12	47.11	48.28	48.26	48.28	48.28	48.15
	EER	$3.53 \cdot 10^{-16}$	$8.52 \cdot 10^{-17}$	$1.01 \cdot 10^{-16}$	$2.01 \cdot 10^{-17}$	$9.66 \cdot 10^{-17}$	$3.51 \cdot 10^{-17}$	$2.59 \cdot 10^{-17}$	$2.85 \cdot 10^{-17}$	$2.41 \cdot 10^{-17}$	$1.20 \cdot 10^{-17}$
	EF	1.00	4.14	3.50	17.56	3.65	1.00	**1.36**	**1.23**	**1.46**	**2.93**
SNS23											
RO-PUF (3LUT)	μ^{intra}	1.29	1.52	1.37	1.38	1.50	1.60	1.43	1.55	1.51	1.24
	μ^{inter}	47.71	48.48	48.57	47.90	48.00	47.77	47.77	47.77	47.77	47.43
	EER	$4.27 \cdot 10^{-12}$	$3.37 \cdot 10^{-12}$	$1.73 \cdot 10^{-12}$	$4.23 \cdot 10^{-12}$	$3.73 \cdot 10^{-12}$	$6.95 \cdot 10^{-12}$	$3.78 \cdot 10^{-12}$	$4.86 \cdot 10^{-12}$	$5.06 \cdot 10^{-12}$	$2.42 \cdot 10^{-12}$
	EF	1.00	1.27	2.47	1.01	1.14	1.00	**1.84**	**1.43**	**1.37**	**2.87**
RO-PUF (11LUT)	μ^{intra}	0.70	0.70	0.70	0.63	0.70	0.70	0.70	0.71	0.53	0.73
	μ^{inter}	42.11	42.11	42.46	42.11	42.57	41.69	41.95	42.10	41.69	42.03
	EER	$9.35 \cdot 10^{-12}$	$9.35 \cdot 10^{-12}$	$8.23 \cdot 10^{-12}$	$9.35 \cdot 10^{-12}$	$8.23 \cdot 10^{-12}$	$1.59 \cdot 10^{-11}$	$1.14 \cdot 10^{-11}$	$9.98 \cdot 10^{-12}$	$7.93 \cdot 10^{-12}$	$1.34 \cdot 10^{-11}$
	EF	1.00	1.00	1.14	1.00	1.14	1.00	**1.39**	**1.59**	**2.01**	**1.19**

Regarding the FPL17 database, in the training set, it is observed that applying a mask with $n = 1$ to the RO-PUF is enough to reduce the *EER* by two orders of magnitude. However, the highest EF with the RO-PUF is obtained for $n = 2$, with an $EF = 4.15$ in the validation set. For the TERO-PUF, the *EER* was reduced in all cases. Finally, regarding the Loop-based PUFs, the highest EF values in the training set were obtained for $n = 2$, reducing the *EER* by two orders of magnitude. Moreover, the parameters obtained for these types of PUFs and masks were also generalizable, achieving $EF = 125.60$ for the Loop-Latch PUF, $EF = 125.93$ for the Loop-Wire PUF, and $EF = 176.42$ for the Loop-PDL PUF in the validation set. Once again, it can be concluded that the optimal mask to apply would be the one with $n = 2$, as it provides the highest EF in the validation set.

Regarding the HOST18 database, it is noted that in most cases the applied masks have contributed more to improving the uniqueness of the PUF than to increasing its reliability. For example, in the validation set, it has been possible to reduce the *EER* in some cases by up to two orders of magnitude (e.g., RO-PUF Left-Lower $EER = 2.18 \cdot 10^{-15}$ without a mask and $EER = 8.10 \cdot 10^{-17}$ with a mask using $n = 2$ parameters). In the validation set, it has also been possible to improve the *EER* by up to one order of magnitude, e.g., the RO-PUF Left-Lower from $EER = 6.14 \cdot 10^{-16}$ without a mask to $EER = 9.89 \cdot 10^{-17}$ with $n = 1$.

Regarding the SNS23 database, similar conclusions are drawn. In the case of the 3-LUT RO-PUF, although the highest EF in the training set was achieved with $n = 2$ parameters, this result has proven to be less generalizable than the $n = 4$ case. Finally, for the 11-LUT RO-PUF,

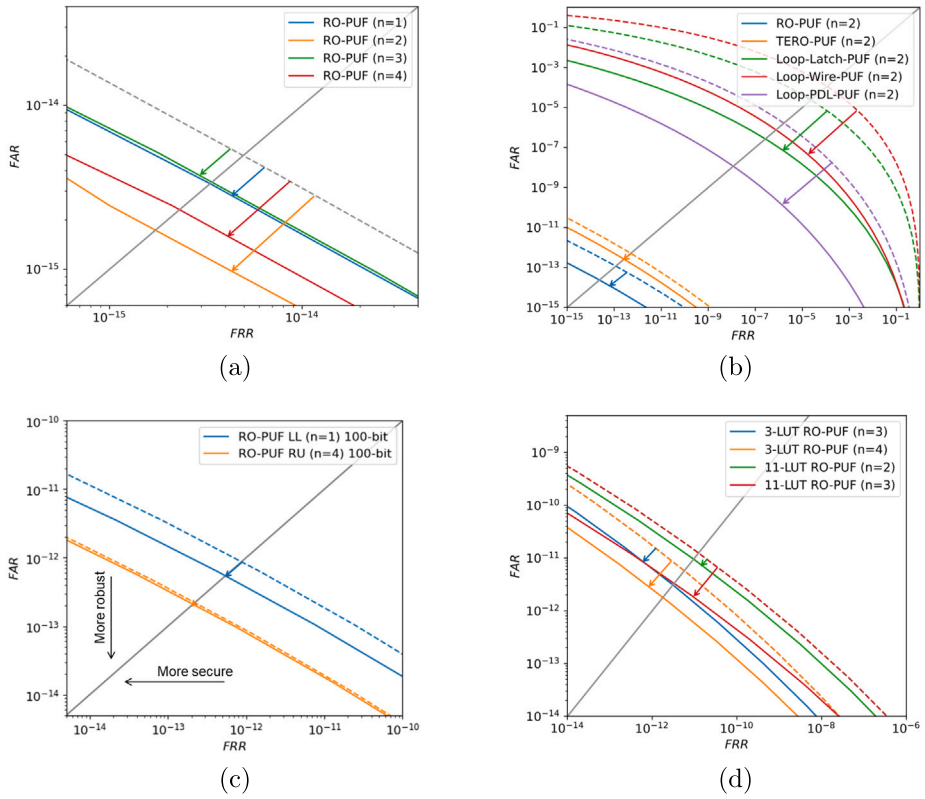

Fig. 6. Some identifiability curves obtained for (a) HOST10, (b) FPL17, (c) HOST18 and (d) SNS23 databases. Gray line corresponds to the *EER*. Dashed lines correspond to identifiability curves without applying any mask.

a similar situation occurs where the highest EF in the training set was achieved with $n = 4$, but it turned out to be less generalizable than $n = 3$. This may be due to the fact that, in some cases, the obtained parameters fit the training data very well, but later prove to be less generalizable to other datasets, which is known as over-fitting.

It is important to note that this strategy is particularly interesting for implementation in IoT devices, as it can significantly improve the identifiability of the authentication scheme with minimal additional resources. It is especially notable that with $n = 1$, identifiability can be improved by an order of magnitude in some cases. This strategy is based on initializing one of the counters and no extra resources are required.

Fig. 6 shows some of the identifiability curves, well known as Receiver Operating Characteristic (ROC) curves, obtained with each of the four databases. On the horizontal axis, the False Acceptance Rate (*FAR*) is represented, and on the vertical axis, the False Rejection Rate (*FRR*). The gray line corresponds to the *EER*. Shifting the ROC curve downward on the graph implies improving the robustness of the PUF, while shifting it to the left indicates an increase in security. this way, curves that are closer to the lower left corner will demonstrate greater robustness and security. Dashed lines correspond to identifiability curves without applying any mask.

5.2. Modeling-attack implications

In this section, it is analyzed whether any bias is introduced in the PUF responses by applying the optimization scheme, which could be exploitable and thus reduce their resistance to potential modeling attacks. For this purpose, the responses of the 12 PUFs listed in Table 4 have been examined before and after applying the optimal transformation to the validation dataset.

Firstly, an analysis of the response uniformity has been conducted. The average uniformity of the PUFs across the different devices has been calculated both before ($\bar{U}_{\text{before}}$) and after ($\bar{U}_{\text{after}}$) applying the transformation. Then, for each case, the deviation of these values from the ideal value has been measured. In Fig. 7, the deviations $\sigma_{U_{\text{before}}} = |\bar{U}_{\text{before}} - 0.5|$ (blue) and $\sigma_{U_{\text{after}}} = |\bar{U}_{\text{after}} - 0.5|$ (orange) are shown. Additionally, for each PUF, the difference:

$$\Delta_{\text{unif}} = \sigma_{U_{\text{before}}} - \sigma_{U_{\text{after}}}, \tag{15}$$

has been calculated. A value of $\Delta_{\text{unif}} > 0$ indicates that the response uniformity has been improved after the transformation, whereas $\Delta_{\text{unif}} < 0$ indicates that the uniformity has been deteriorated. The first observation is that, in the case of the RO-PUF, Loop-Latch-PUF, Loop-Wire-PUF, and Loop-PDL-PUF from the FPL17 database, the uniformity is significantly improved. In these cases, applying the mask led to a considerable increase in the PUF inter-*HD*, for example from 25.46% to 40.47% ($n = 2$) for the Loop-Wire-PUF, or from 37.46% to 47.85% ($n = 2$) with the validation set. This improvement is due to the existence of a relationship between the uniqueness and uniformity properties. Similarly, for the other implementations, no significant difference in uniformity is observed: in some cases, it improves by up to 2%, while in the worst case it does not deteriorate by more than 1%.

Secondly, an attempt was made to model the PUF responses using various algorithms. As input to the models, the indices of the corresponding entropy sources were used, and as output, the resulting bit from the comparison between both sources was considered. The goal of each model is to determine whether, given that pair of indices, the output bit will be a 1 or a 0. Ideally, for a response that cannot be modeled, an accuracy of 50% would be obtained. The algorithms used for this purpose were: Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and a Multilayer Perceptron (MLP), which are commonly employed in this type of PUF modeling analysis.

To evaluate the impact of the optimization scheme on identifiability, the deviation of the accuracy value without applying the transformation was calculated: $\sigma_{A_{\text{before}}} = |\bar{A}_{\text{before}} - 0.5|$, and after applying the

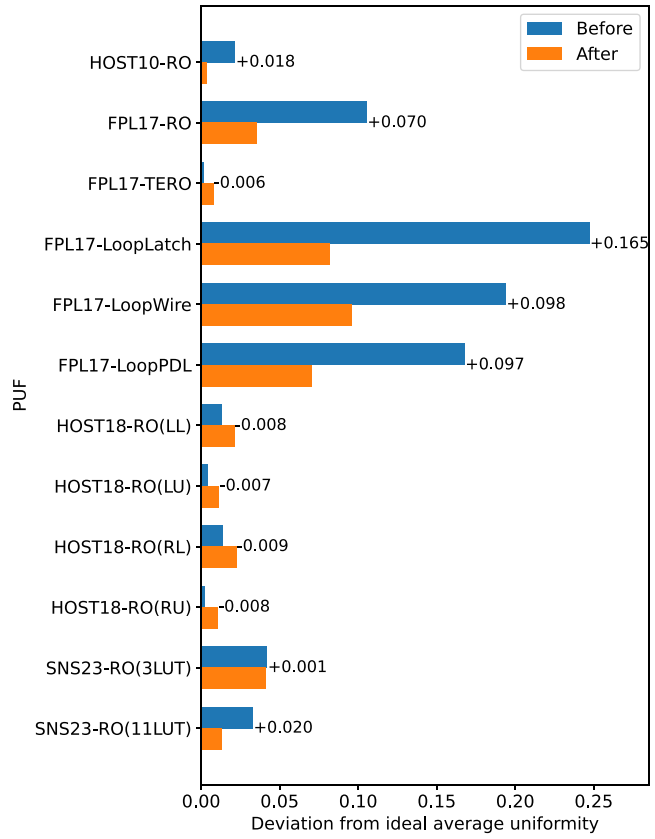

Fig. 7. PUF uniformity analysis before (blue, $\sigma_{U_{\text{before}}}$) and after (orange, $\sigma_{U_{\text{after}}}$) applying the optimization scheme.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

optimal transformation: $\sigma_{A_{\text{after}}} = |\bar{A}_{\text{after}} - 0.5|$, for the same PUFs as in the previous case. Subsequently, the difference was calculated:

$$\Delta_{\text{acc}} = \sigma_{A_{\text{before}}} - \sigma_{A_{\text{after}}}, \tag{16}$$

so that a value of $\Delta_{\text{acc}} > 0$ indicates that the modeling resistance has been improved after the transformation, whereas $\Delta_{\text{acc}} < 0$ indicates that it has been deteriorated. The values of Δ_{acc} for each PUF and algorithm are shown in Fig. 8. As can be observed, for the vast majority of PUFs and models, $\Delta_{\text{acc}} > 0$, indicating that, in most cases, applying the mask improves the PUF resistance to modeling attacks; whereas in the worst case, the degradation does not exceed 0.03.

5.3. Case study: PUF prototype implementation in FPGA

To provide evidence of the practical implementation of this PUF optimization system, we have implemented two fully functional prototypes of 3-stage 200-oscillator RO-PUFs on a PYNQ-Z2 board including an Artix-7 FPGA, producing 100-bit responses. This corresponds to one of the two architectures from the SNS23 database used previously, upon which this optimization process has been applied. As shown in Table 4, using the validation dataset and masks with $n = 4$, the identifiability of the PUF can be improved, achieving an $EF = 2.87$.

This way, two functional PUF designs have been implemented, both including the proposed RO-PUF along with other auxiliary circuitry (such as test signals and some IP blocks to establish PC-FPGA communication): one design implements the RO-PUF without the optimization scheme, and the other implements the system with the corresponding optimization. Moreover, in this design, each ring oscillator has its own counter, so implementing this optimization technique, even with masks of size $n = 4$, is expected to incur almost no resource cost on the FPGA. This is a common practice, employed to parallelize the measurement process and thereby minimize the response time of the PUF. In Fig. 9, the implementation of both designs on the FPGA is shown. As can be seen, there is an area where the oscillators have been implemented, while the rest of the design modules — including debug modules and auxiliary circuitry for PC communication — are highlighted in red. These were generated by Vivado after the synthesis and implementation process.

Table 5
FPGA resource utilization comparison between the baseline RO-PUF implementation and the design including the proposed optimization technique.

Resource	Before Opt.	After Opt.	Increase	Available
LUT	3466 (6.52%)	3590 (6.75%)	124	53 200
LUTRAM	62 (0.36%)	62 (0.36%)	0	17 400
FF	7665 (7.20%)	7665 (7.20%)	0	106 400
IO	1 (0.80%)	1 (0.80%)	0	125
BUFG	1 (3.13%)	1 (3.13%)	0	32

Fig. 10 shows the on-chip power estimation reported by Vivado for both implementations: the baseline RO-PUF and the optimized design including the proposed optimization block. As can be observed, the total power consumption remains practically unchanged in both cases (approximately 1.43 W). Moreover, regarding dynamic power consumption, it should be noted that the TOP module of the design consumes only 0.022 W, while the remaining power is mainly due to the FPGA–PC communication interface. The distribution between dynamic and static power components is also nearly identical. This result confirms that integrating the optimization mechanism does not introduce any noticeable power overhead. This behavior is expected because the design already includes dedicated counters for each ring oscillator, meaning that the optimization logic can reuse the existing measurement infrastructure without requiring additional switching activity or complex control structures.

Table 5 compares the FPGA resource utilization for both implementations. The results show that the inclusion of the optimization block leads to only a negligible increase in the number of LUTs, while the usage of other resources such as LUTRAM, FF, I/O pins, and clock buffers remains unchanged. Therefore, the proposed optimization technique can be integrated into the RO-PUF architecture without introducing a meaningful hardware overhead, confirming that the improvement in identifiability can be achieved without increasing the resource footprint of the system.

5.4. State-of-the-art comparison

As can be seen, this work addresses these specific "bottleneck" issues that these earlier strategies (Table 1) could not resolve:

- Eliminating Hardware Overhead: Unlike previous approaches based on nonlinear transformations or physical cell modifications, such as custom SRAM designs, this work uses a weight mask technique that can be implemented simply by initializing counters to specific values. This results in a low additional hardware cost, which is critical for resource-constrained IoT environments.
- Solving the Generalizability Problem: A common flaw in previous selection or tuning methods was the need for per-device characterization, where every single chip had to be extensively tested. In this work, we demonstrate that the optimization can be performed on a small subset of devices (20%) and then successfully generalized to the entire population (80%).
- Identifiability Optimization: By mathematically applying weight masks to parameters before comparison, the work achieves a reduction in the Equal Error Rate (*EER*) of up to two orders of magnitude. This allows the PUF to reach high security and robustness levels that previously required much more complex hardware-based error correction (ECC) or active reconfigurations.

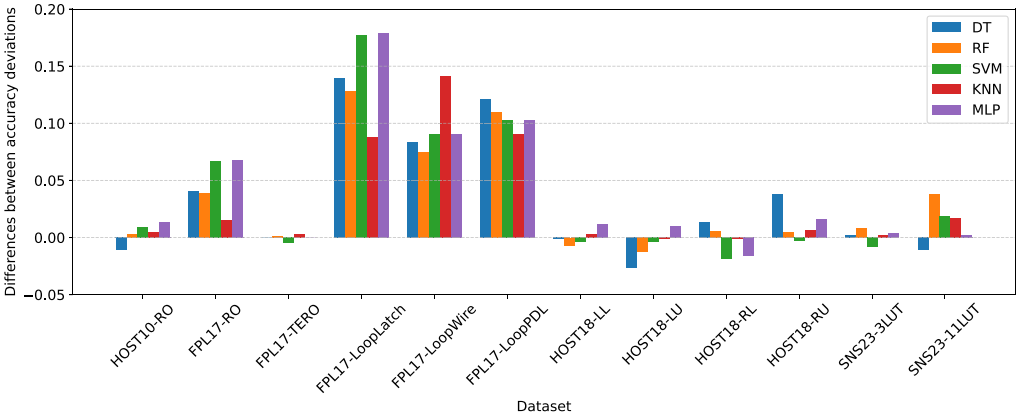

Fig. 8. Δ_{acc} for different PUFs and algorithms.

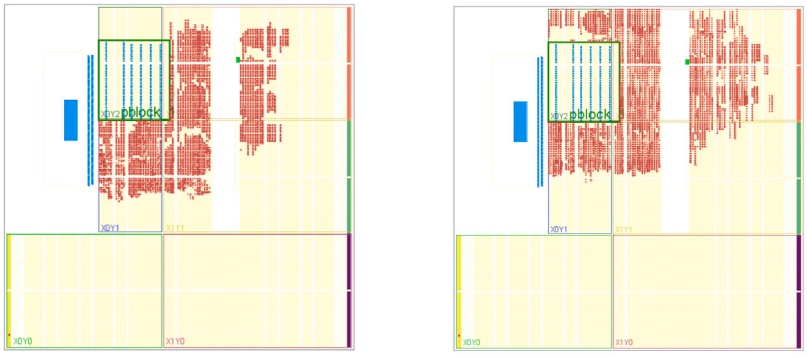

Fig. 9. Layout of the PUF prototype implementation on FPGA including auxiliary circuitry: (a) without and (b) with optimizer block.

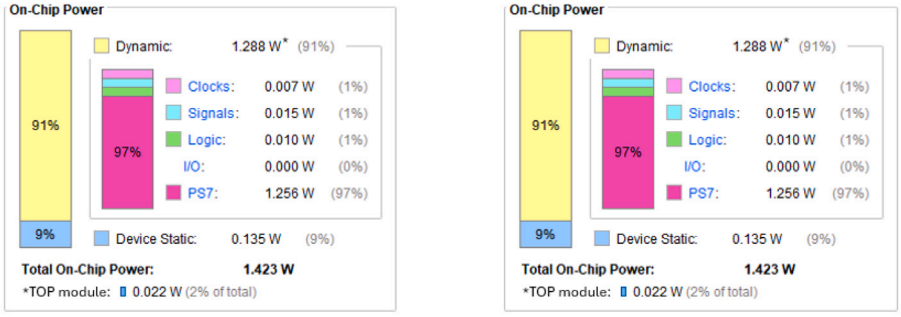

Fig. 10. On-chip power estimation obtained from Vivado for the FPGA implementation: (a) baseline RO-PUF design without the optimization scheme and (b) RO-PUF design including the proposed optimizer.

6. Conclusions

In this work, a novel strategy to improve the identifiability of compensated measurement PUFs has been proposed for IoT device authentication. This technique consists of initializing the PUF counters with specific values. Furthermore, to test its efficiency, the proposed technique has been applied to different types of PUFs constructed using public databases.

Firstly, this technique avoids the need to use extra logic to improve the efficiency of the PUF unlike other state-of-the-art strategies, as it only relies on initializing the PUF counters with specific values. This is crucial for PUF-based authentication schemes in IoT devices, where resource consumption is critical. Secondly, the optimal parameters for improving PUF efficiency that are obtained from a training set are also generalizable. This means that a PUF vendor could use a small number of FPGAs to determine the optimal parameters, initialize the counters, and then provide a larger set of IoT devices with the integrated PUF. Thirdly, using a weight mask with two parameters is an effective strategy to enhance the identifiability of the PUF. This way, this approach can improve the *EER* up to two orders of magnitude without requiring a large number of parameters, as that complicates the optimization process and considerably increases the time needed

to obtain the optimal weight mask values, thus improving security in IoT authentication schemes.

Future research directions include exploring new optimization strategies, using weight masks of different sizes and types such as convolutional masks.

CRediT authorship contribution statement

Raúl Aparicio-Téllez: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Miguel Garcia-Bosque:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Conceptualization. **Guillermo Díez-Señorans:** Writing – review & editing, Visualization, Validation, Supervision, Methodology, Investigation, Conceptualization. **Concepcion Aldea:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition. **Santiago Celma:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Raul Aparicio-Tellez reports financial support was provided by Diputación de Aragón (DGA). All authors reports financial support was provided by Spanish State Research Agency, Centro Universitario de la Defensa and University of Zaragoza. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been partially supported by Spanish State Research Agency (PID2020-114110RA-I00, PID2023-150244OB-I00, PID2024-157204OA-I00), University of Zaragoza, Spain (UZ2024-IyA-01), Centro Universitario de la Defensa (CUD-2026_07) and Diputación de Aragón (DGA) fellowship to Raúl Aparicio-Téllez.

Data availability

Data will be made available on request.

References

Anandakumar, N.N., Hashmi, M.S., Chaudhary, M.A., 2022. Implementation of Efficient XOR Arbiter PUF on FPGA With Enhanced Uniqueness and Security. IEEE Access 10, 129832–129842. http://dx.doi.org/10.1109/ACCESS.2022.3228635.

Aparicio-Téllez, R., Garcia-Bosque, M., Díez-Señorans, G., Celma, S., 2023a. Oscillator Selection Strategies to Optimize a Physically Unclonable Function for IoT Systems Security. Sensors 23 (9), http://dx.doi.org/10.3390/s23094410, URL https://www.mdpi.com/1424-8220/23/9/4410.

Aparicio-Téllez, R., Garcia-Bosque, M., Díez-Señorans, G., Celma, S., 2024. Enhancing Identifiability of PUFs with Built-in Compensation through Nonlinear Transformations. In: 2024 IEEE International Symposium on Circuits and Systems. ISCAS, pp. 1–5. http://dx.doi.org/10.1109/ISCAS58744.2024.10558553.

Aparicio-Téllez, R., Garcia-Bosque, M., Díez-Señorans, G., Sánchez-Azqueta, C., Celma, S., 2023b. Design Strategies to Select the Best Locations in a Ring Oscillator PUF. In: 2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications To Circuit Design. SMACD, pp. 1–4. http://dx.doi.org/10.1109/SMACD58065.2023.10192224.

Baturone, I., Román, R., Corbacho, Á., 2023. A Unified Multibit PUF and TRNG Based on Ring Oscillators for Secure IoT Devices. IEEE Internet Things J. 10 (7), 6182–6192. http://dx.doi.org/10.1109/JIOT.2022.3224298.

Cook, H., Tripp, Z., Hutchings, B., Goeders, J., 2023. Improving the Reliability of FPGA CRO PUFs. In: 2023 33rd International Conference on Field-Programmable Logic and Applications. FPL, pp. 311–316. http://dx.doi.org/10.1109/FPL60245.2023.00053.

Della Sala, R., Bellizia, D., Scotti, G., 2022. A Novel Ultra-Compact FPGA-Compatible TRNG Architecture Exploiting Latched Ring Oscillators. IEEE Trans. Circuits Syst. II: Express Briefs 69 (3), 1672–1676. http://dx.doi.org/10.1109/TCSII.2021.3121537.

Diez-Senorans, G., Garcia-Bosque, M., Sánchez-Azqueta, C., Celma, S., 2021. Digitization Algorithms in Ring Oscillator Physically Unclonable Functions as a Main Factor Achieving Hardware Security. IEEE Access 9, 147343–147356. http://dx.doi.org/10.1109/ACCESS.2021.3123867.

Farha, F., Ning, H., Ali, K., Chen, L., Nugent, C., 2021. SRAM-PUF-Based Entities Authentication Scheme for Resource-Constrained IoT Devices. IEEE Internet Things J. 8 (7), 5904–5913. http://dx.doi.org/10.1109/JIOT.2020.3032518.

Fletcher, R., 2000. Practical Methods of Optimization. John Wiley & Sons.

Gassend, B., Clarke, D., van Dijk, M., Devadas, S., 2002. Silicon physical random functions. In: Proceedings of the 9th ACM Conference on Computer and Communications Security. CCS '02, Association for Computing Machinery, New York, NY, USA, pp. 148–160. http://dx.doi.org/10.1145/586110.586132.

Hemavathy, S., Bhaaskaran, V.S.K., 2023. Arbiter PUF—A Review of Design, Composition, and Security Aspects. IEEE Access 11, 33979–34004. http://dx.doi.org/10.1109/ACCESS.2023.3264016.

Hesselbarth, R., Wilde, F., Gu, C., Hanley, N., 2018. Large scale RO PUF analysis over slice type, evaluation time and temperature on 28nm Xilinx FPGAs. In: 2018 IEEE International Symposium on Hardware Oriented Security and Trust. HOST, pp. 126–133. http://dx.doi.org/10.1109/HST.2018.8383900.

Idriss, T.A., Idriss, H.A., Bayoumi, M.A., 2021. A Lightweight PUF-Based Authentication Protocol Using Secret Pattern Recognition for Constrained IoT Devices. IEEE Access 9, 80546–80558. http://dx.doi.org/10.1109/ACCESS.2021.3084903.

Kalam, S., Keshri, A.K., 2025. Advancing IoMT security: A two-factor authentication model employing PUF and Fuzzy logic techniques. Comput. Secur. 148, 104138. http://dx.doi.org/10.1016/j.cose.2024.104138, URL https://www.sciencedirect.com/science/article/pii/S0167404824004437.

Li, J., Li, L., Yang, J., He, Y., Zhou, W., Yuan, S., 2020. An efficient and stable composed entropy extraction method for FPGA-based RO PUF. IEICE Electron. Express 17 (24), 20200350. http://dx.doi.org/10.1587/elex.17.20200350.

Lin, H.-T., Liang, Y.-Y., 2021. A PUF-based secure wake-up scheme for Internet of Things. Comput. Secur. 110, 102415. http://dx.doi.org/10.1016/j.cose.2021.102415, URL https://www.sciencedirect.com/science/article/pii/S016740482100239X.

Maes, R., 2013. Physically Unclonable Functions : Constructions, Properties and Applications, 2013 Springer Berlin Heidelberg, Berlin, Heidelberg, http://dx.doi.org/10.1007/978-3-642-41395-7, URL https://library.biblioboard.com/viewer/ae726c7b-bd42-11ea-a10e-0a28bb48d135.

Maiti, A., Casarona, J., McHale, L., Schaumont, P., 2010. A large scale characterization of RO-PUF. In: 2010 IEEE International Symposium on Hardware-Oriented Security and Trust. HOST, pp. 94–99. http://dx.doi.org/10.1109/HST.2010.5513108.

Maiti, A., Schaumont, P., 2011. Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive. J. Cryptology 24 (2), 375–397. http://dx.doi.org/10.1007/s00145-010-9088-4.

Mall, P., Amin, R., Das, A.K., Leung, M.T., Choo, K.-K.R., 2022. PUF-Based Authentication and Key Agreement Protocols for IoT, WSNs, and Smart Grids: A Comprehensive Survey. IEEE Internet Things J. 9 (11), 8205–8228. http://dx.doi.org/10.1109/JIOT.2022.3142084.

Marchand, C., Bossuet, L., Mureddu, U., Bochard, N., Cherkaoui, A., Fischer, V., 2018. Implementation and Characterization of a Physical Unclonable Function for IoT: A Case Study With the TERO-PUF. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 37 (1), 97–109. http://dx.doi.org/10.1109/TCAD.2017.2702607.

McGrath, T., Bagci, I.E., Wang, Z.M., Roedig, U., Young, R.J., 2019. A PUF taxonomy. Appl. Phys. Rev. 6 (1).

Miah, A., Hossain, F.S., 2025. CRO-PUF: Resilience to machine learning and differential power attacks. Comput. Secur. 151, 104313. http://dx.doi.org/10.1016/j.cose.2025.104313, URL https://www.sciencedirect.com/science/article/pii/S0167404825000021.

Modarres, A.M.A., Sarbishaei, G., 2023. Systematic Cryptanalysis of PUF-Based Authentication Protocols for IoT, A Case Study. IEEE Netw. Lett. 5 (4), 304–308. http://dx.doi.org/10.1109/LNET.2023.3298775.

Ni, L., Wang, P., Zhang, Y., Li, X., Li, G., Ding, L., Zhang, J., 2024. SI PUF: An SRAM and Inverter-Based PUF With a Bit Error Rate of 0.0053% and 0.073/0.042 pJ/bit. IEEE Trans. Circuits Syst. II: Express Briefs 71 (4), 2339–2343. http://dx.doi.org/10.1109/TCSII.2023.3339296.

Ning, H., Farha, F., Ullah, A., Mao, L., 2020. Physical unclonable function: Architectures, applications and challenges for dependable security. IET Circuits, Devices & Syst. 14 (4), 407–424.

Qureshi, M.A., Munir, A., 2022. PUF-RAKE: A PUF-Based Robust and Lightweight Authentication and Key Establishment Protocol. IEEE Trans. Dependable Secur. Comput. 19 (4), 2457–2475. http://dx.doi.org/10.1109/TDSC.2021.3059454.

Rahman, M.T., Rahman, F., Forte, D., Tehranipoor, M., 2016. An aging-resistant RO-PUF for reliable key generation. IEEE Trans. Emerg. Top. Comput. 4 (3), 335–348. http://dx.doi.org/10.1109/TETC.2015.2474741.

Santana-Andreo, A., Saraza-Canflanca, P., Castro-Lopez, R., Roca, E., Fernandez, F., 2024. Reliability improvement of SRAM PUFs based on a detailed experimental study into the stochastic effects of aging. AEU - Int. J. Electron. Commun. 176, 155147. http://dx.doi.org/10.1016/j.aeue.2024.155147, URL https://www.sciencedirect.com/science/article/pii/S1434841124000323.

Santos-Prieto, F.d.l., Rubio-Barbero, F.J., Castro-Lopez, R., Roca, E., Fernandez, F.V., 2024. A Comprehensive Approach to Improving the Thermal Reliability of RTN-Based PUFs. IEEE Trans. Circuits Syst. I. Regul. Pap. 72 (2), 661–670. http://dx.doi.org/10.1109/TCSI.2024.3458057.

SciPy, 2024. SciPy documentation — Scipy v1.14.1 Manual. [Online]. Available at https://docs.scipy.org/doc/scipy/index.html. (Accessed October 15 2024).

Shamsoshoara, A., Korenda, A., Afghah, F., Zeadally, S., 2020. A survey on physical unclonable function (PUF)-based security solutions for Internet of Things. Comput. Netw. 183, 107593. http://dx.doi.org/10.1016/j.comnet.2020.107593, URL https://www.sciencedirect.com/science/article/pii/S1389128620312275.

Shifman, Y., Miller, A., Weizman, Y., Fish, A., Shor, J., 2019. An SRAM PUF with 2 Independent Bits/Cell in 65nm. In: 2019 IEEE International Symposium on Circuits and Systems. ISCAS, pp. 1–5. http://dx.doi.org/10.1109/ISCAS.2019.8702612.

Su, Z., Li, B., Zhang, W., Gao, J., Su, X., Zhang, G., Ren, H., Lu, P., Liu, F., Zhao, F., Li, S., 2022. Reliability Improvement on SRAM Physical Unclonable Function (PUF) Using an 8T Cell in 28 nm FDSOI. IEEE Trans. Nucl. Sci. 69 (3), 333–339. http://dx.doi.org/10.1109/TNS.2021.3126587.

Wang, Z., Zhang, Y., Ma, Y., Zhang, M., He, Z., Wan, M., 2024. Enhancing the Reliability of SC PUF Through Optimal Capacitor Configuration. IEEE Trans. Circuits Syst. I. Regul. Pap. 71 (1), 85–98. http://dx.doi.org/10.1109/TCSI.2023.3327250.

Wild, A., Becker, G.T., Güneysu, T., 2017. A fair and comprehensive large-scale analysis of oscillation-based PUFs for FPGAs. In: 2017 27th International Conference on Field Programmable Logic and Applications. FPL, pp. 1–7. http://dx.doi.org/10.23919/FPL.2017.8056795.

Yao, L., Liang, H., Huang, Z., Jiang, C., Yi, M., Lu, Y., 2021. A Lightweight Configurable XOR RO-PUF Design Based on Xilinx FPGA. In: 2021 IEEE 4th International Conference on Electronics Technology. ICET, pp. 83–88. http://dx.doi.org/10.1109/ICET51757.2021.9451016.

Zerrouki, F., Ouchani, S., Bouarfa, H., 2022. A survey on silicon PUFs. J. Syst. Archit. 127, 102514. http://dx.doi.org/10.1016/j.sysarc.2022.102514, URL https://www.sciencedirect.com/science/article/pii/S1383762122000832.

Zhang, J., Shen, C., Guo, Z., Wu, Q., Chang, W., 2022. CT PUF: Configurable Tristate PUF Against Machine Learning Attacks for IoT Security. IEEE Internet Things J. 9 (16), 14452–14462. http://dx.doi.org/10.1109/JIOT.2021.3090475.