A Survey on the Susceptibility of PUFs to Invasive, Semi-Invasive, and Non-Invasive Attacks: Challenges and Opportunities for Future Directions

Mohd Syafiq Mispan[1], *Basel Halak*[2], *Mark Zwolinski*[2]

[1] Micro & Nano Electronics, Centre for Telecommunication Research & Innovation (CeTRI), Fakulti Teknologi Kejuruteraan Elektrik dan Elektronik, Universiti Teknikal Malaysia Melaka, Malaysia
[2] Electronics & Computer Science, University of Southampton, University Road, Southampton, SO17 1BJ, United Kingdom
*

Abstract: Physical Unclonable Functions (PUFs) are considered to be a promising technology that provides a hardware root-of-trust for integrated circuit (IC) applications. PUFs exploit the intrinsic process variations that map a set of challenges to a set of responses. The intrinsic process variations are caused by uncontrollable deviations in the IC manufacturing process, which are unique and random from die to die and wafer to wafer. As the PUF output is device-specific, PUFs can, therefore, be used in IC identification and authentication, and cryptographic key generation. Nevertheless, many different successful attack techniques have already revealed vulnerabilities in certain PUFs, including invasive, semi-invasive, and non-invasive attacks. In this work, we survey some of the known attacks on PUFs. We also survey the countermeasures to these types of attack presented in recent literature, and finally, discuss the future challenges. Through this survey, the susceptibility of PUFs to attacks is highlighted and this information may be used to improve the quality of future PUF-based application designs.

1 Introduction

Implementation of current security solutions relies on the secret keys or unique identifiers stored in the on-chip non-volatile memory (NVM) or battery-backed static random-access memory (SRAM) [1]. As the secret keys are always present using these security technologies, they can be tampered with using invasive or semi-invasive attacks. Furthermore, the secret key needs to be programmed by a trusted party (e.g. the IC manufacturer) and so it may be compromised within the product supply chain. Besides that, the cost of using NVM or battery-backed SRAM to store the secret key is prohibitively expensive, especially for resource-constrained devices [1, 2]. For example, NVM such as electrically erasable programmable read-only memory (EEPROM) requires floating gate transistors, resulting in additional masks and processing steps which further increase the fabrication cost.

A Physical Unclonable Function (PUF) is an emerging technology that offers a promising solution to the aforementioned issues. The input-output behaviour of PUFs, which also known as challenge and response pairs (CRPs), is determined by IC manufacturing variations. Due to the random nature of these variations, the identifier or the secret key (i.e. in the form of binary strings) that are generated through the mapping of CRPs is unique and device-specific. The secret key can only be generated during the power-on state and is wiped-out in the power-off state. Furthermore, the complex and random nature of the manufacturing process variations makes a PUF practically and physically impossible to clone at the atomic level [3]. A PUF can be implemented using a standard CMOS circuit design technique which requires no special fabrication process, such as floating gate transistors in NVM.

All of the above shows that the keys generated from PUFs require no programming and they are tamper-resistant as the keys are not present during the power-off state. Moreover, it is impossible to physically clone the PUF through the fabrication process. Therefore, a PUF is considered a robust hardware-based intrinsic security device. Nevertheless, as the impact of process variations remains static in PUFs, the functionality of the PUF and the relationship of its CRPs can be modelled or software-cloned using invasive, semi-invasive and non-invasive techniques [4].

In this paper, we survey some of the known attacks on PUFs using invasive, semi-invasive and non-invasive techniques from the literature. We also highlight the methods used in the aforementioned techniques as summarised in Table 1. Moreover, we survey some of the countermeasures to these types of attacks that are proposed in recent literature. Finally, future challenges and opportunities are discussed. The aim of this survey is to lead to a better understanding of the practical implementations of PUFs and to improve the quality of future PUF-based application design. The survey also gives insight into the security requirements of a PUF-based application with respect to a certain type of attack which is always a trade-off between the implementation cost and the security level.

The remainder of this survey is organized as follows. Section 2 provides an overview of the types of PUFs and main applications of PUFs. Section 3 describes the known attacks to PUFs using invasive, semi-invasive and non-invasive techniques. Section 4 explores the countermeasures to existing attacks. Section 5 discusses the opportunities and future challenges of PUFs. Finally, Section 6 summaries the literature surveyed in this paper.

2 Preliminaries

This section provides a preliminary background on the notion of a PUF, its types, and its main applications.

2.1 Physical Unclonable Function: Definitions and Types

A PUF is defined as a function that maps inputs to outputs and that function is embodied by the physical material of the device. In the context of CMOS devices, this refers to silicon material [6]. The aggressive scaling of CMOS technology has led to a drastic increase in process variations. One of the fundamental process variations is random dopant fluctuations (RDF). RDF refers to the randomness in the amount and position of dopants during dopant implantation,

$$
\begin{array}{lcl}
\underline{\textbf{Verifier}} & \longrightarrow & \underline{\textbf{Prover } j} \\
\langle \mathbf{c}_{ij}, \mathbf{r}_{ij} \rangle \text{ with } \mathbf{c}_{ij} \leftarrow TRNG() & \longleftrightarrow & \mathbf{r}_{ij} \leftarrow PUF(\mathbf{c}_{ij}) \quad \left.\right\} \text{1x Enrolment} \\
d_j \leftarrow d & & \\
 & & \\
\langle \mathbf{c}, \mathbf{r} \rangle \leftarrow \langle \mathbf{c}_{ij}, \mathbf{r}_{ij} \rangle \text{ with } i \leftarrow d_j & & \\
d_j \leftarrow d_j - 1 & \xrightarrow{\mathbf{c}} & \tilde{\mathbf{r}} \leftarrow PUF(\mathbf{c}) \quad \left.\right\} d\text{x Authentication} \\
\text{Abort if } \mathbf{HD}(\mathbf{r},\tilde{\mathbf{r}}) > \epsilon & \xleftarrow{\tilde{\mathbf{r}}} &
\end{array}
$$

Fig. 1: Strong PUF-based identification and authentication using challenge-and-response protocol [2, 5].

which varies the electrical behaviour of geometrically similar MOSFETs. The k-bit input and n-bit output of a PUF are referred to as a challenge C and response R, respectively. Further, the set of CRPs for a PUF can be defined as (C_i,R_i), i = 1, ...N. Challenges are used to control the behaviour of a PUF and based on the challenges applied, corresponding responses are generated. When a challenge is applied to two different PUFs (PUF A and PUF B), the respective responses are produced where Response A $\neq$ Response B.

PUFs are often subdivided into two classes, according to the number of CRPs. Strong PUFs are PUFs with a number of CRPs that grows exponentially as the number of bit challenges increases. The challenge space for Strong PUF is given as 2^k, where k is the total number of the challenge bits. The response space, for most of the Strong PUF architectures, is n=1. In practice, a k-bit challenge is sent to the PUF-based devices and m k-bit sub-challenges are generated using a linear feedback shift register (LFSR) within the device before these sub-challenges are applied to a Strong PUF to generate a $(m * n)$-bit response [7, 8]. An Arbiter-PUF is a Strong-PUF and it was the first PUF fabricated on real silicon using a 180-nm CMOS technology [9]. Since the first idea of a silicon PUF [9], an enormous number of PUF techniques have been proposed in the past decade.

Weak PUFs are PUFs with a very small number of CRPs, and in the extreme case with just a single challenge [5, 10]. An SRAM-PUF [10] is considered as a Weak PUF since it has only a single challenge. The start-up values (SUVs) of SRAM-PUF (i.e., PUF responses) are generated during the SRAM power-up process. One might argue that addressing the SRAM bit cell array provides a challenge-response mechanism, however, it is only the process of reading the bit cell values. Other memory-based PUFs such as Butterfly PUF [11], Buskeeper PUF [12], etc. are also considered to be Weak PUFs.

2.2 *Applications of PUF*

Two categories of applications naturally emerge, linked to functional discrepancies between so-called Strong and Weak PUFs, which are IC identification and authentication, and cryptographic key generation [2, 13]. Strong PUFs are the types of PUFs that can be used in IC identification and authentication application. Although many Strong PUF protocols have been proposed [2], in this paper, we refer the reader to the challenge-response protocol as it presents the basic authentication protocol of using Strong PUFs. Figure 1 illustrates the PUF-based authentication process which consists of two phases; 1) enrolment phase, and 2) authentication phase. Given an authentic device j (i.e. prover) that is embedded with a Strong PUF, during an enrolment phase, a verifier applies randomly chosen challenges to obtain unpredictable responses and stores these CRPs in a database, d_j for future identification and authentication. In the field, when device j is requested for authentication, a verifier selects a challenge from d_j and obtains the PUF response, r̃ from the device j. The device j passes the authentication process if the response matches or is less than the hamming distance (HD) threshold, ϵ, as compared to the stored value in the database.

The HD threshold is determined based on the probability of rejection and the probability of misidentification. The probability of rejection is the probability of the occurrence of a valid PUF deviates significantly from its initial state stored in the database, given as p_{reject} and can be computed using, [14]:

$$
p_{reject} = 1 - \sum_{i=0}^{\epsilon} \binom{n}{i} p_{intra}^{i} (1 - p_{intra})^{n-i} \tag{1}
$$

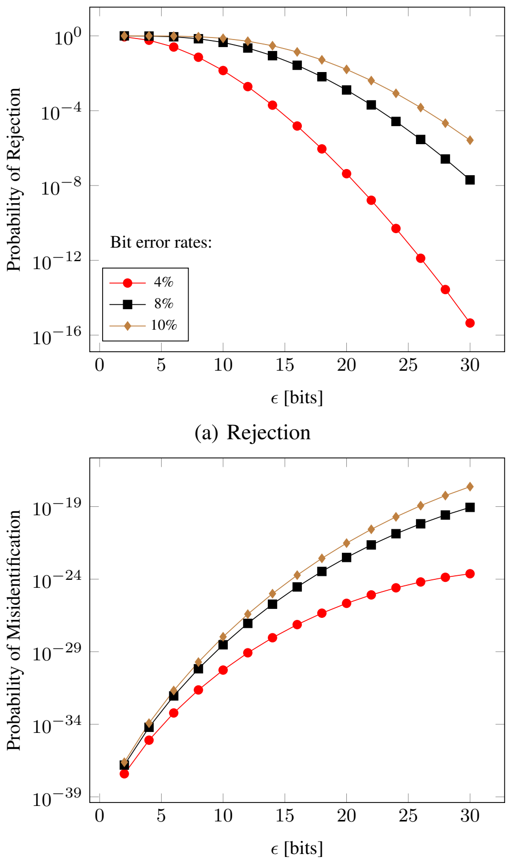
where p_{intra} denotes the bit error probability (i.e., reliability of PUF response), n is the total number of bits in the response, and ϵ is the HD threshold. The probability of misidentification is the probability of the occurrence when a wrong PUF is issued to a server, and the server authenticates it by mistake, given as p_{mis} and can be computed as [15]:

$$
\begin{aligned}
p_{mis} = & \sum_{i=0}^{2\cdot\epsilon} \binom{n}{i} p_{inter}^{i} (1 - p_{inter})^{n-i} \\
& \cdot \left[1 - \sum_{i=0}^{\epsilon} \binom{2\cdot\epsilon}{i} p_{intra}^{i} (1 - p_{intra})^{2\cdot\epsilon - i} \right]
\end{aligned} \tag{2}
$$

where p_{inter} denotes the bit unique probability (i.e., the probability of response PUF A $\neq$ response PUF B), n is the total number of bits in the response, and ϵ is the HD threshold.

Based on Eq. (1) and (2), Figure 2 depicts the probability of rejection and misidentification at different bit error rates, p_{intra} for a 128-bit identifier and the ideal uniqueness of 50% ($p_{inter} = 0.5$), under variations of ϵ. As can be seen from Figure 2, if ϵ is set too low, the probability of authentic PUFs being rejected increases, while setting ϵ too high increases the probability of misidentification. Apparently, from Figure 2, it is always desirable to achieve reliable PUF responses (i.e. low bit error rates) to reduce the probability of rejection and misidentification.

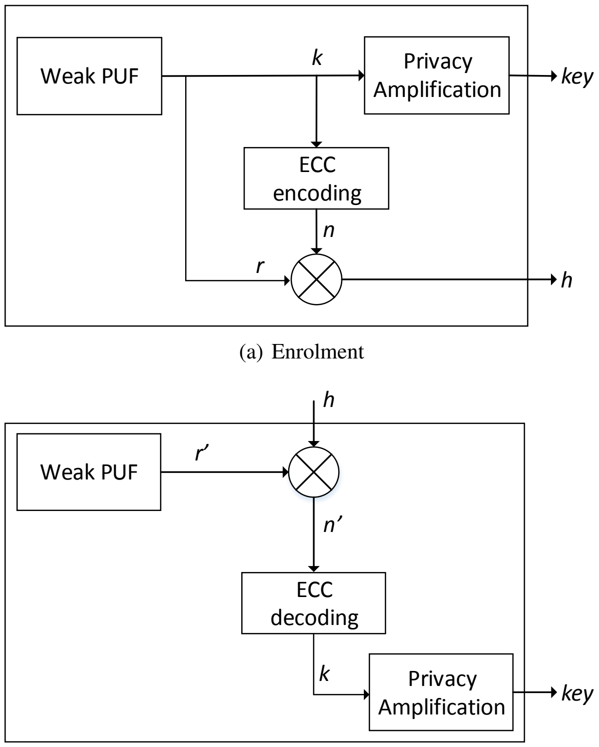
The Weak PUFs are the type of PUFs that are suitable for use as a cryptographic key generator [16]. Nevertheless, the direct use of the secret key for cryptographic primitives is not feasible as the PUF responses are known to be noisy due to environmental variations and ageing [17]. To generate an error-free cryptographic key from PUF responses, an error correction code (ECC) is required. Figure 3 shows the procedure of cryptographic key generation based on Weak PUFs. The procedure consists of two phases: 1) enrolment phase, and 2) reconstruction phase. The helper data, h, is generated during the enrolment phase and it is computed as $h = r \oplus n$, where r is the PUF response and n is the encoded codeword. The helper data is stored in the NVM. k is a subset of r. During the reconstruction phase, the helper data, h is recalled to correct the noisy PUF response, r' and finally, the secret key is regenerated. The secret key is only present during the power-on state and wiped-out in the power-off state.



(a) Rejection

(b) Misidentification

Fig. 2: Probability of rejection and misidentification at different bit error rates, p_{intra} and ϵ for n=128-bit and p_{inter}=0.5 [5].



(a) Enrolment

(b) Reconstruction

Fig. 3: The procedure of cryptographic key generation based on SRAM-PUF [12].

It is worth mentioning that the use of Weak-PUFs is not, however, limited to cryptographic key generation; they can be used for IC identification and authentication. In a recent study, Guin *et al.* [18] proposed a secure SRAM-based PUF protocol using matching and repetition schemes for device authentication of edge devices in the Internet of Things (IoT) infrastructure. Elsewhere, a lightweight authentication scheme for embedded systems utilizing SRAM and DRAM is proposed in [19].

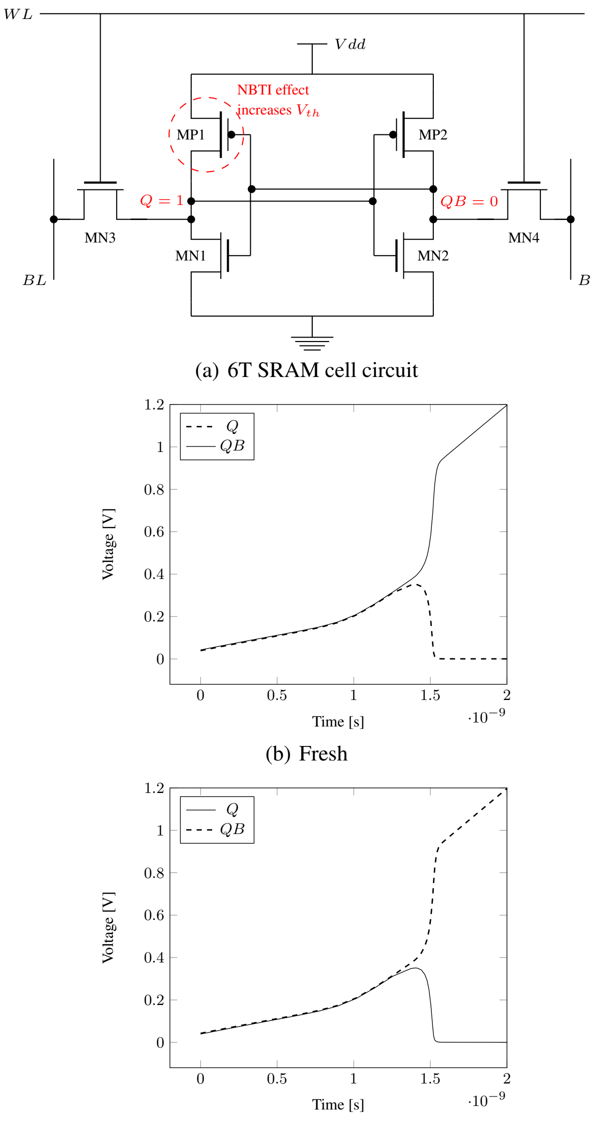
3 Types of Attack

The types of attacks on PUFs are categorised into three groups which are invasive, semi-invasive and non-invasive attacks. Invasive attacks refer to attacks on physical devices where the physical properties of the chip are irreversibly modified. Failure analysis techniques such as micro-probing, Scanning Electron Microscope (SEM), and Focus Ion Beam (FIB), are the common techniques used for invasive attacks, which require a sample preparation such as decapsulation and depassivation. Non-invasive attacks, however, require no sample preparation. Non-invasive attacks only exploit the available information externally such as input and output values, running time, power consumption, etc. For semi-invasive attacks, partial or complete removal of the device packaging is necessary. Unlike invasive attacks, no destructive modifications are required in semi-invasive attacks.

3.1 Invasive Attacks

Helfmeier *et al.*, [20] performed an invasive attack and successfully cloned an SRAM-PUF using photon emission analysis (PEA), and FIB circuit edit (FIB CE) techniques. The cloning process consists of two stages; 1) SUVs characterisation using PEA technique, and 2) circuit modification using FIB CE, which has been performed on SRAM memory in the ATmega328P microcontroller, with a feature size of approximately 600nm. The device went through the sample preparation where the package and excess bulk silicon of the device backside were removed. Subsequently, the PEA was deployed through the backside of the IC to capture extremely weak photon emissions from switching transistors during the reading process. This analysis is only targeted for NMOS transistor, MN1 or MN2 (depending on the SUVs of nodes Q and QB), see Figure 4(a), as the amount of emission for NMOS transistor is greater than PMOS transistor. Based on the captured emission image, an FIB circuit edit is performed to alter the transistor characteristics according to the fingerprint of the target device. Although this shows that an SRAM-PUF could be cloned, producing a physical clone with these techniques remains economically infeasible for devices with limited financial value. Furthermore, efficient photon emission detection for modern IC with small feature sizes requires complex and expensive PEA techniques [21].

Elsewhere, an invasive attack on an SRAM-PUF through device ageing has been analysed and evaluated, using 8kB AS6C6264 commercial SRAM IC [22]. The effect of negative bias temperature instability (NBTI) is exploited to change the strength of the PMOS transistors, either MP1 or MP2, see Figure 4(a). At the time $t = 0$ (fresh), because of the random process variations, transistor MP2 has a slightly higher threshold voltage (V_{th}) compared to that of MP1. After power-up process, the nodes Q and QB resolving to '1' and '0', respectively as illustrated in Figures 4(a) and 4(b). When the transistor MP1 is subjected to NBTI stress ($V_{gs} = -V_{dd}$) over the prolonged time, its V_{th} increases significantly, whereas the V_{th} of MP2 remains the same (i.e., asymmetric degradation or stress). As a consequence, after the power-up process, the node Q is less likely to power-up to a '1' than it was before the NBTI effect, as shown in Figure 4(c). The fingerprint generated from an SRAM-PUF could be erased through a device ageing process as discussed above, hence making it susceptible to a denial of service (DoS) attack.



(a) 6T SRAM cell circuit

(b) Fresh

(c) Ageing

Fig. 4: Bi-stable SRAM cell SUVs before and after ageing impact is taken place.

3.2 *Semi-invasive Attacks*

Nedospasov *et al.*, [23] proposed a semi-invasive attack wherein a laser stimulation (LS) technique is used to read-out the SUVs of SRAM-PUFs. The experiment has been conducted on SRAM memories in AtMega328P and ATXMEga128A1 micro-controllers, with feature sizes of around 600nm and 300nm, respectively. Prior to the LS process, the package and excess bulk silicon of the device backside were removed. An LS setup consists of a microscope incorporating a laser light source with scanning capabilities and an electrical setup to operate the device under test (DUT) [23]. As illustrated in Figure 5, the laser light is applied to one of the transistor's drain and source contacts, node Q. Then, the heat is absorbed by the bulk silicon and the metallisation layers which causes a temperature gradient that generates a voltage, known as the Seebeck voltage. The generated voltage alters the gate voltage of the inverter formed by MP2 and MN2 and directly change the conductance path between V_{dd} to GND. The laser image is captured by scanning with the laser over the entire SRAM array in which the SUVs in the SRAM cells are read out by measuring the current change at the supply terminals. Subsequently, the captured image is analysed and the fingerprint or the key is recovered. The cloning of the PUF can be continued further as in [20] using an FIB CE method, however, only extraction using the LS technique is discussed. The LS technique has been proven to be successful for devices with technology down to 180nm [23]. An additional technique is required, such as solid immersion lenses (SIL), to improve the resolution for a smaller technology down to 60nm technology node.

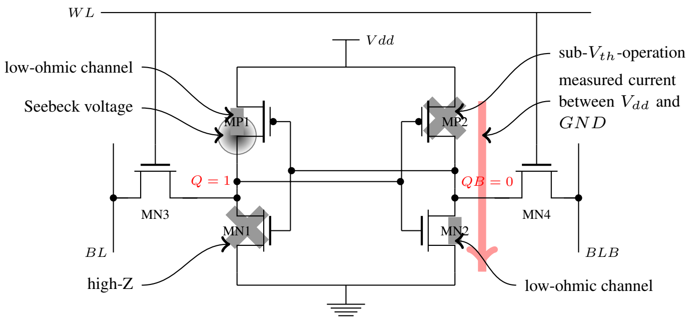


Fig. 5: Seebeck stimulation applied on 6-T SRAM cell [23].

Elsewhere, Tajik *et al.*, [24] physically characterise an Arbiter-PUF and extract its delay parameters by using the PEA technique. The 8-bit Arbiter-PUF (k=8 stages) was implemented on a Complex Programmable Logic Device (CPLD) manufactured in a 180nm technology node. The device was decapsulated and the bulk silicon material of the device was thinned down prior to the PEA process. An Arbiter-PUF consists of two parallel paths, namely the upper and the lower path. Consider the upper path and denote its k unknown delays by $\delta_1, \cdots, \delta_k$. The total propagation time through all k stages is denoted by t_i for the i^{th} measurement which measured between enabling the PUF and photon emission at the output of the last stage. Hence, the delay parameters of an Arbiter-PUF can be represented as a linear system, given below:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_{k-1} \\ \delta_k \\ \delta_{k+1} \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_{k-1} \\ t_k \\ t_{k+1} \end{pmatrix} \tag{3}$$

In Eq. (3), setting all k challenge bits to 0 is used as a reference measurement and the delay is represented as δ_{k+1}. A total of $k + 1$ challenges are measured for the upper path which includes k challenges with HD=1 as the reference measurement. Therefore, the delay parameter for stage i can be computed by subtracting the delay of the challenge with HD=1 (which differ at the challenge bit position i) and the delay of the reference measurement, given as $i = t_i - t_{k+1}$. The lower path can be computed using a similar measurement procedure. As the overall delay at the outputs of the last stage is the sum of the delay in each stage, the measured delay parameters further can be used to compute the responses for arbitrary challenges. A total of $2 * (k + 2)$ "full path" measurements is required to completely characterise an Arbiter-PUF with k stages.

Another semi-invasive attack technique is proposed in [25] whereby a semi-invasive near-field electromagnetic (EM) source has been used to attack RO-PUFs. The RO-PUF is implemented on a Field Programmable Gate Array (FPGA) with 9 ring oscillators ($m = 9$), each built out of 7 inverters. After 4096 cycles of oscillation, the ROs were compared and $(m - 1)$ PUF response bits were generated. The attack is performed by decapsulating the backside of the FPGA chip until the die's backside surface is exposed. The *Langer ICR HH 150* probe is used to measure the near-field EM emanation in which the probe is placed so close that is almost touched the die's backside surface. The probe is connected to the oscilloscope to capture the frequency spectrum of the RO-PUF. Based on the captured frequency spectrum, the tracing steps are performed to identify the frequency range, the area of main RO frequencies signal leakage, distinguish the RO frequencies and building the RO-PUF model. The

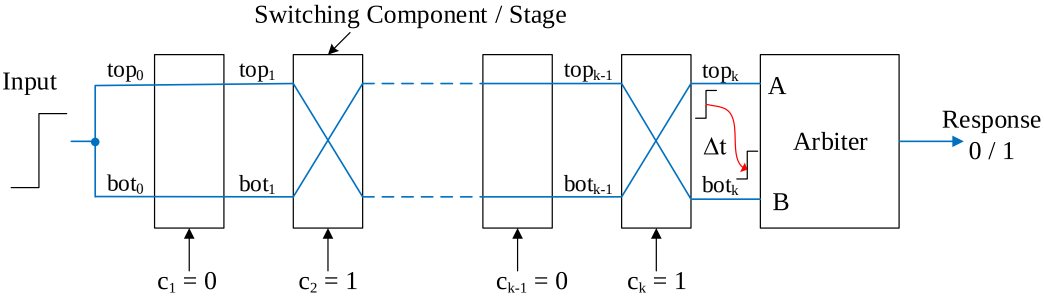


Fig. 6: k-bit Arbiter-PUF [7].

8-bit response of RO-PUF is recovered once the RO-PUF model is successfully built.

3.3 *Non-invasive Attacks*

Another attack model on a PUF is the non-invasive attacks. According to Gassend *et al.*, [6], the non-invasive attack is believed to be the most plausible attack as it is financially inexpensive for an adversary to perform an attack. Generally, non-invasive attacks on PUFs can be divided into three: ML, side-channel and hybrid attacks, which are discussed next.

3.3.1 ML Attack: The ML-attack comprehends that the attacker has access to the primary interface of the device. Therefore, an attacker is restricted to non-invasive CRPs measurement and can apply a polynomial number of challenges to the device to collect the corresponding responses. With the measured CRPs of a particular PUF in hand, the adversary tries to build a numerical model of the PUF using ML algorithms. Hence, a non-invasive attack is also known as a model-building attack. According to Rührmair *et al.* [27], an ML-attack is most applicable to Strong PUFs.

The first ML-attack was demonstrated in [28] to predict the response of an Arbiter-PUF whereby the functionality of the Arbiter-PUF was described using an additive linear model. Figure 6 illustrates the k-bit Arbiter-PUF which consists of k stages or switching components and SR-latch as an Arbiter. The total delays of both parallel paths (i.e. represented as *top* and *bot*) are modelled as the sum of the delays in each stage depending on the challenge C=$\{c_1, c_2...c_k\}$. The final delay difference Δt between the two paths in a k-bit Arbiter-PUF can be expressed as:

$$\mathit{\Delta} t = \vec{w}^T \vec{\Phi} \tag{4}$$

where parameter $\vec{w}$ is the delay-determined vector and $\vec{\Phi}$ is the feature vector. Both parameters are the functions of the applied k-bit challenge with dimension $k+1$. As described in [29], $\delta_i^{1/0}$ is denoted as the delay in stage i for the crossed $(c_i = 1)$ and uncrossed $(c_i = 0)$, respectively. Hence, δ_i^1 is the delay of stage i when $c_i = 1$, while δ_i^0 is the delay of stage i when $c_i = 0$. Then

$$\vec{w} = (w^1, w^2, ...w^k, w^{k+1})^T \tag{5}$$

where $w^1 = \frac{\delta_1^0 - \delta_1^1}{2}$, $w^i = \frac{\delta_{i-1}^0 + \delta_{i-1}^1 + \delta_i^0 - \delta_i^1}{2}$ for all $i = 2, ..., k$, and $w^{k+1} = \frac{\delta_k^0 + \delta_k^1}{2}$. Furthermore,

$$\vec{\Phi}(C) = (\Phi^1(C), ..., \Phi^k(C), 1)^T \tag{6}$$

where $\vec{\Phi}^j(C) = \prod_{i=j}^{k}(1 - 2c_i)$ for $j = 1, ..., k$. From (5), the vector $\vec{w}$ encodes the delay in each stage of the Arbiter-PUF and via $\vec{w}^T \vec{\Phi} = 0$ determines the separating hyperplane in the space of all feature vectors, $\vec{\Phi}$. The delay difference, Δt, is the inner product of $\vec{w}$ and $\vec{\Phi}$. If $\Delta t > 0$, the response bit is '1', otherwise, the response bit is '0'.

Based on the additive linear model as discussed above, Lim *et al.* [28] applied an ML algorithm known as a Support Vector Machine (SVM), which successfully modelled the separating hyperplane of the Arbiter-PUF. As a result, the 64-bit Arbiter-PUF, which was fabricated using 180nm CMOS technology, can be predicted with $\approx 97\%$ accuracy [28]. One might argue that the transformation of the challenges to feature vectors, as in Eq. (6), was in fact the primary factor which helps to improve the prediction accuracy significantly. Hence, the attacker must know the type of PUF in the first place which questions the feasibility of an ML-attack. Nevertheless, the use of a particular PUF is not necessarily confidential and it might be publicly known, as revealed by NXP Semiconductors on the use of an SRAM-PUF which is embedded in their upcoming hardware security devices [30, 31]. Only the exact configuration of the CRPs generation or key generation remains secret.

Countermeasures to reduce the susceptibility of Arbiter-PUF to ML-attack have been proposed by introducing a non-linearity

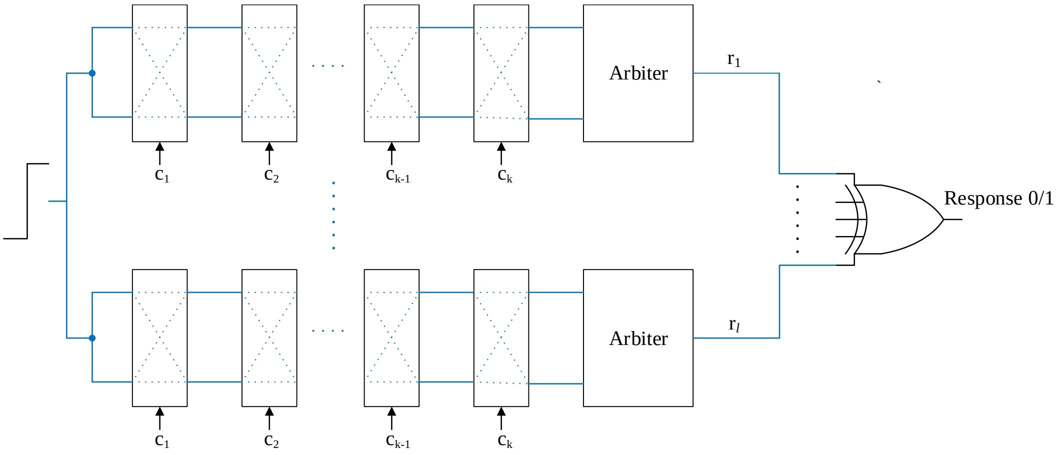


Fig. 7: l-XOR Arbiter-PUF [26].

into the Arbiter-PUF. These include the Feed-forward Arbiter-PUF [14], XOR Arbiter-PUF [26], and Lightweight-PUFs [32] The basic form of these PUFs is an Arbiter-PUF with some additional pre-processing and/or post-processing techniques. For example, the l-XOR Arbiter-PUF consists of l parallel Arbiter-PUFs in which their outputs are XORed, see Figure 7. A comprehensive ML-attack using SVM, Logistic Regression (LR), and Evolution Strategy (ES) has been performed on the aforementioned PUFs [27]. The PUFs have been modelled in simulation (the delay values of an Arbiter-PUF were chosen pseudo-randomly according to a standard normal distribution), built on an FPGA, and fabricated on Application Specific Integrated Circuit (ASIC). Based on the CRPs measurement taken from the simulation, an FPGA, and an ASIC, the results show that the ML techniques were able to model the Feed-Forward Arbiter-PUF, 5-XOR Arbiter-PUF, and 5-XOR Lightweight-PUF with $\approx$ 99% prediction accuracy [27]. Nevertheless, one might disable the ML-attack by implementing the XOR Arbiter-PUF and Lightweight-PUF with $l \geq 6$ (i.e, the number of XORs in output network) and $k > 128$ (i.e. challenge bit-length of Arbiter-PUF, see Figure 6) [27, 33, 34].

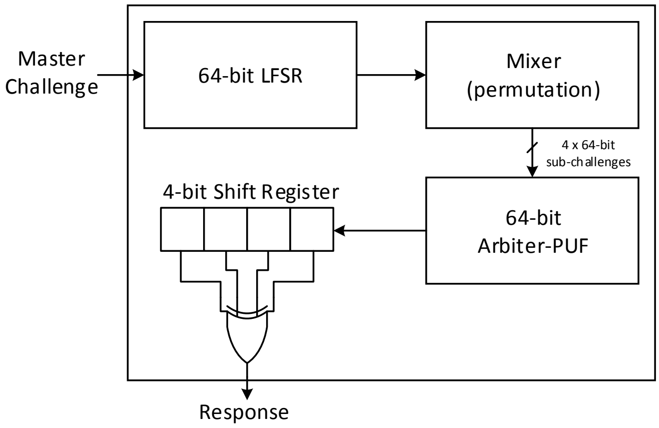


Fig. 8: PUF-based RFID tag [35].

Becker [35] performed an ML-attack on a commercial PUF-based Radio Frequency Identification (RFID) tag and demonstrated the first software cloning attack. The PUF-based RFID tag had an Arbiter-PUF that was used to generate a device-specific identifier. The internal structure of the PUF tag is illustrated in Figure 8. The PUF-based RFID tag can be authenticated based on the challenge-and-response protocol as depicted in Figure 1. An attacker who is in possession (i.e. has access to the primary inputs) of a PUF-based RFID tag can collect CRPs through a non-invasive measurement. A single protocol execution consists of sending a 64-bit challenge and receiving a 256-bit response. A total of 4 executions is executed in which 1024 CRPs are collected. Based on the collected CRPs, an ML-attack is performed using an LR technique. On average, 85.8% prediction accuracy is achieved. Meanwhile, the average reliability of the PUF-based RFID tag in Figure 8 is 87.5%. As discussed in Section 2.2, the challenge-and-response protocol requires that the HD threshold, ϵ is set such that $\frac{\epsilon}{n} > \left(1 - \frac{reliability}{100}\right)$, where n is the bit-length of the response. Based on the value of ϵ, the attacker only needs to achieve $\left(1 - \frac{\epsilon}{n}\right) \times 100\%$ prediction accuracy. As the value of the achieved model accuracy is very close to the average reliability, the ML-attack is considered a successful attack. Further, the parameters derived by the ML-attack was built into the software on a programmable RFID smart-card emulator and used to successfully clone a PUF tag.

Elsewhere, the ML-attack performance of Artificial Neural Network (ANN) and SVM on an Arbiter-PUF and an XOR Arbiter-PUF is compared and the results show that ANN outperforms SVM [36]. Vijayakumar *et al.* [37] explored the learn-ability of ML techniques such as SVM and LR as the non-linearity in a PUF increases. "Bagging" and "Boosting" were also used for the ML-attack analysis [37] since these techniques have the potential to generate a strong classifier through the combination of several classifiers prediction. Based on previous ML-attack analyses [14, 27, 35–37], ANN, LR, ES, and Boosting are the most favourable to solve the non-linearity problem.

All of the above focused on using conventional ML techniques to attack PUFs. Recently, Khalafalla *et al.* [38] explored an advanced technique of ML-attack by using a deep learning (DL) technique to attack the double arbiter PUFs (DA-PUFs). $l - 1$ DA-PUF consists of l identical Arbiter-PUFs and the final response is generated by XORing all PUF responses, as illustrated in Figure 9 which shows a 3-1 DA-PUF. A previous study, [39] shows that 3-1 DA-PUF was attack-resistant against conventional ML techniques which achieved the predictability of about 57% by using SVM. Nevertheless, Khalafalla *et al.* [38] showed that by using the Deep Neural Network (DNN) technique, 3-1 DA-PUF can be modelled with 85% of prediction accuracy. Using DL could improve the prediction accuracy and requires no feature extraction as described in Eq. (6). Nevertheless, DL requires huge computational resources as compared to conventional ML techniques to solve its complex neural network. A study in [38] executed DL on an Nvidia GeForce GTX 1080 Ti GPU card with 11GB of RAM which cost about $800.

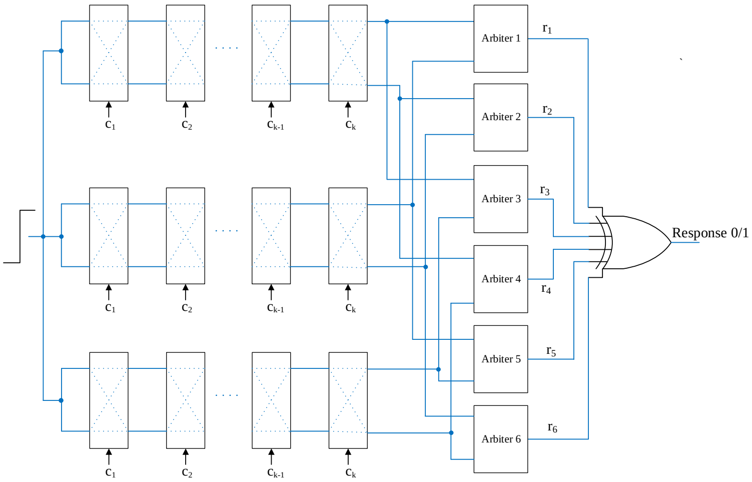


Fig. 9: 3-1 DA-PUF [38].

In another study, Pranesh *et al.* [40] employed a novel machine learning based modeling technique, a combination of the Tensor Regression Network (TRN) with an efficient version of the CANDECOMP/PARAFAC tensor decomposition (CP-decomposition) technique. The proposed method aims to reduce the computational resource requirement of model building attacks on XOR Arbiter PUF by developing a model that is driven by the structure of the data, instead of focus on developing a model aimed at solely matching its predictions with the observed data. The proposed technique uses parity-vectors derived from the challenges in the training set for constructing the tensor input. The simulation was executed on a Linux workstation with 64GB of main memory and single-core, 3.3 GHz processor. The proposed technique achieves the prediction accuracy of 93.02% with 2460 training CRP for 8-XOR 64-bit Arbiter PUF. Meanwhile, the prediction accuracy of 92.78% is achieved when the technique is applied on 7-XOR 128-bit Arbiter-PUF with 2400 training CRP set. From the above, reasonable high prediction accuracy can be achieved with efficient CP-TRN technique, with fewer data set requirement, and at low computational overhead.

In another perspective, Ganji *et al.*, [41] argued that the used of ML to attack the PUFs is purely based on trial and error estimates or heuristic approaches. Therefore, the probability of obtaining a useful model with high confidence, or the sufficient number of CRPs, or the probability of correct prediction (accuracy) is not guaranteed. Ganji *et al.*, [41] proposed a probably approximately correct (PAC) learning algorithm to address the above concerns and successfully applied on Arbiter-PUF. The proposed PAC was based on the deterministic finite automata (DFA) and regular language. The Arbiter-PUF can be represented using DFA by performing the crucial process of discretization and mapping from the real values of the multiplexer delays (i.e., switching component - see Figure 6) to a set of integer values. Subsequently, the collapsed DFA is constructed based on the integer delay values, which has the polynomial-size of stages and maximum variation of delay values. Further, the PAC-learning algorithm is used to model the Arbiter-PUF. The results show that the maximum number of CPRs and the time complexity are polynomial in the number of stages, the maximum deviation of delay values, and levels of accuracy and confidence.

3.3.2 Side-channel Attack:

A side-channel attack is a type of non-invasive attack based on the information gained from the implementation of a PUF. Information such as power consumption, CMOS device noise, charging/discharging time, running time, etc. can be exploited to perform side-channel attacks. In a study, Delvaux *et al.* [42] exploited repeatability imperfections in Arbiter-PUF responses due to CMOS device and interconnect noise as side-channel information to perform a model-building attack. To be precise, the repeatability refers to the short-term reliability of a PUF as affected by CMOS noise sources. The key insight is that repeatability measurements provide direct timing information about switching components in an Arbiter-PUF. If the delay difference, ∆t, for a given challenge is very large, it is unlikely that the noise changes the sign of ∆t, see Figure 6. In contrast, if ∆t is close to zero (i.e., entering the metastability state), the response of the Arbiter-PUF will be influenced by the noise, resulting in the changes of ∆t sign. Therefore, based on the repeatability measurements, the attacker relatively knows the delay for all switching components. Although the timing information is all relative, Delvaux *et al.* [42] successfully built the model of Arbiter-PUF with $>85\%$ prediction accuracy.

Elsewhere, Zeitouni *et al.* [43] exploited remanence decay in volatile memory as side-channel information to attack SRAM-PUFs. The SRAM-PUFs were implemented in 65-nm CMOS technology node. When the data is stored in an SRAM cell, $Q = 1$ and $QB = 0$, see Figure 4(a), after power dump happens, Q decreases slowly and QB maintains a low level. The discharging of the parasitic capacitance at node Q is known as remanence decay or data remanence. If the power is up while the parasitic capacitance still holding some charges, the SRAM cell can recover the previously stored value. By exploiting the remanence decay phenomenon, the approach in this attack is to recover the PUF response in a device after overwriting the SRAM-PUF with some data that are known to the adversary. In this study [43], the assumption that the adversary can initialise the memory with known values and that the adversary knows that the targeted PUF-based system uses a message authentication code (MAC) were made. With these assumptions, the adversary can recover the PUF state, based on the encryption observed in a series of data remanence experiments. The secret key is successfully recovered but this attack is likely to be impractical in terms of the timescale as it takes approximately two CPU-months. Besides, this attack is limited by the precision of the equipment to control the remanence decay in the SRAM.

As mentioned in Section 2.2, a PUF can be used in cryptographic key generation whereby the helper data is required for noise cancellation during the reconstruction phase. The helper data can be stored publicly in NVM as their information leakage about the responses r is sufficiently small. Nevertheless, the helper data could be manipulated by altering the connection lines between external memory and the PUF device. One study in [44] exploits the helper data manipulation in which a side-channel attack to capture the power traces using near-field EM technique to recover the response of RO-PUF is proposed. The RO-PUF design is implemented on a Xilinx Spartan 3E FPGA. The *Langer ICR HH 150* probe is used to measure the near-field EM which connected to the oscilloscope to capture the power traces of the FPGA chip during the key reconstruction phase. The attack successfully recovered 100% of the corrected PUF response (i.e. after the error correction block) by capturing and analysing 10,000 power traces.

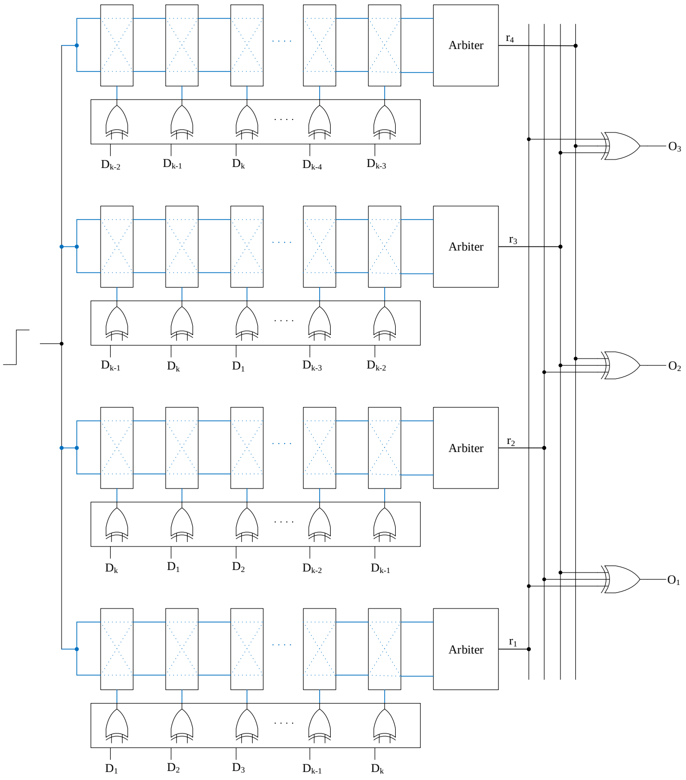


Fig. 10: Lightweight-PUF [32].

3.3.3 Hybrid Attack:

As discussed in Section 3.3.1, an ML-attack has a limitation to break the relationship of XOR-based Arbiter-PUF and Lightweight-PUF with more than five single parallel Arbiter-PUFs, and with challenge bit-length longer than 128. Using solely the ML techniques, the training time increases exponentially as the number of XORs increases. Therefore, a hybrid attack is proposed to attack XOR-based Arbiter-PUFs (see Figure 7) and Lightweight-PUFs (see Figure 10) in an FGPA implementation in which side-channel and ML techniques are combined [34]. Rührmair *et al.*, [34] exploited the power and timing traces of an XOR Arbiter-PUF and a Lightweight-PUF as side-channel information to overcome the limitations in ML techniques. The power side channel is to trace the power based on measuring the amount of current drawn from the supply voltage during the latches transitions (i.e. the arbiters) from zero to one, sub-responses before they are XOR'ed together. Besides, the timing side channel is proposed in [34] to provide information about the individual response bits (i.e. PUF response). For example, l-XOR Arbiter-PUF has l sub-response bits $\{r_1, \cdots, r_l\}$ and they're XOR'ed together to form a single PUF response bit. An l-input XOR will consist of several stages of smaller XOR gates. By measuring the delay of the overall PUF path (after XOR'ing), the delay length of different patterns of l sub-response bits is characterized. Hence, it is possible to estimate the number of flipped XOR inputs with a good probability, i.e. the cumulative number of zeros and ones among the single Arbiter PUF responses $r1, ..., rl$. Based on the power and timing side-channels information, the LR technique is adapted to minimize the squared error between a side-channel model and the actual PUF response. The adapted ML technique successfully attacks the XOR Arbiter-PUFs and Lightweight PUFs for up to 16 XORs and for a bit-length of up to 512 (timing side-channel) and 128 (power side-channel) with a minimum accuracy achieved of 97%.

Elsewhere, Liu *et al.*, [46, 48] proposed a novel optimization-theoretic attacking approach to attack Arbiter-PUF, XOR Arbiter-PUF and Memristor-PUF. This approach is composed of two parts. The first part is the weight vector estimation based on linear programming to reduce the uncertainty associated with the initial k CRPs that is known to the attacker. The goal is to minimize the uncertainty of the k CRPs and to achieve a prediction rate higher

Table 1 Summary of known attacks to PUFs.

Attack Type	PUF Type	Platform	Technology Node	Method
Invasive	SRAM-PUF [20]	ATmega328P	600-nm	PEA and FIB
	SRAM-PUF [22]	AS6C6264 SRAM IC	200-nm	Burn-in
Semi-invasive	SRAM-PUF [23]	AtMega328P	600-nm	LS
		ATXMEga128A1	300-nm	
	Arbiter-PUF [24]	Altera Max V CPLD	180-nm	PEA
	RO-PUF [25]	Xilinx Spartan-3 FPGA	90-nm	Near-Field EM
Non-invasive (*ML/DL Attack*)	Arbiter-PUF [28]	ASIC	180-nm	SVM
	5-XOR Arbiter-PUF [27]	ASIC	45-nm	LR
	Lightweight-PUF (5 XORs) [27]	Simulation	NR	LR
	Feed-Forward Arbiter-PUF [27]	Simulation	NR	ES
	Arbiter-PUF [35]	PUF-based RFID tag	NR	LR
	3-1 DA-PUF [38]	Mojo V3	45-nm	DNN
	8-XOR Arbiter-PUF [40]	Simulation	NR	CP-TRN
	Arbiter-PUF [41]	Simulation	NR	PAC
Non-invasive (*Side-channel Attack*)	Arbiter-PUF [42]	ASIC	65-nm	Repeatability imperfections (CMOS device and interconnect noise)
	SRAM-PUF [43]	ASIC	65-nm	Remanence decay
	RO-PUF [44]	Xilinx Spartan-3E FPGA	90-nm	Near-Field EM
Non-invasive (*Hybrid Attack*)	16-XOR Arbiter-PUF [34]	Xilinx Spartan-6 FPGA	45-nm	Power and timing side-channel, and adapted LR
	Lightweight-PUF (16 XORs) [34]	Xilinx Spartan-6 FPGA	45-nm	Power and timing side-channel, and adapted LR
	4-XOR 128-bit Arbiter-PUF [45]	Simulation	130-nm	Power side-channel and CNN
	XOR Arbiter-PUF [46]	Simulation	NA	Power and timing side-channel, and optimization-theoretic
	VR-PUF [47]	Simulation	130-nm	Optimized CPMA and ANN
	5-XOR VR-PUF [47]	Simulation	130-nm	Optimized CPMA and ANN

NR=not reported.

than a predetermined value. To further maximize the reduction in the uncertainty associated with k CRPs, the second part of the approach is to generate a new set of j CRPs by using the cutting-plane method. The newly generated of j CRPs are combined with the initial set of k CRPs to re-estimate the weight vector. Subsequently, the prediction rate is computed and compared against the predetermined value. If the expected prediction rate is not met, the new CRPs are generated and these processes are iterated until the expected prediction rate is achieved. The above approach can be used to attack the Arbiter-PUF and Memristor-PUF. However, a combination of the above approach with the side-channel attack as proposed in [34] is needed to attack the XOR Arbiter-PUF. As compared to a hybrid attack proposed in [34], the optimization-theoretic approach needs 66% fewer known CRPs and 79.8% less computational time. This technique also shows a promising accuracy when an attack is performed in noisy conditions in which the average attacking time overhead is 35%, 55%, and 91% for noise levels of 1%, 3%, and 5%, respectively.

Yu *et al.*, [45] proposed an efficient way to perform a hybrid attack on a 4-XOR 128-bit Arbiter-PUF (l=4 and k=128, see Figure 7) using one of the DL architectures, namely a convolutional neural network (CNN). The XOR Arbiter-PUF has been implemented in Cadence using a 130nm CMOS technology process. The uncorrelated input challenge of XOR Arbiter-PUF C=$\{c_1, c_2 ... c_k\}$ is converted into another correlated input challenge C^*=$\{c_1^*, c_2^* ... c_k^*\}$ using $C^* = C \oplus \alpha$ where $\alpha = \{\alpha_1, \alpha_2 ... \alpha_k\}$ is the correlation coefficient which derived using power-side channel information. If n number of different challenge values are applied to the XOR Arbiter-PUF, the corresponding n number of different P_d values are obtained. To increase the efficiency of finding the correlation coefficient, α, the input challenge, C and α are divided into m groups. The Hamming Weight (HW) model is used to study the correlation between $C \oplus \alpha$ and P_d in which the optimum α value is acquired. By applying the power side-channel analysis to add correlation for the input challenge of an XOR arbiter PUF, 98% prediction accuracy of the hybrid attack is achieved.

A recent work, [47] studied the vulnerability of voltage regulator PUF (VR-PUF) against hybrid attack. An optimized combined power and modeling attacks (CPMA) with Lagrange multiplier are used to increase the efficiency of ANN in attacking the VR-PUF. The optimization starts by establishing the approximate polynomial model based on the input challenge and the output response of VR-PUF. The approximate polynomial model can be used to determine which input factors drive responses and in what direction. Based on the approximate polynomial model of VR-PUF, an objective function is developed to optimize the relevant parameters that contribute to the accuracy of predicting the response of VR-PUF. Furthermore, the transient power information of VR-PUF is extracted and is used as a critical constraint for the objective function. After obtaining the objective function and constraints for the optimization problem associated with the VR-PUF, the Lagrange multiplier is applied to transform the original optimization problem into the Lagrangian (refer to [47] for the details of mathematical equations). Solving the Lagrangian function gives the optimized number of CRPs, number of hidden layers, and the number of neurons in each hidden layer for ANN architecture. The architecture of ANN which is built based on an optimized CPMA achieves 98.22% and ≈90% prediction accuracy, respectively for VR-PUF and 5-XOR VR-PUF. In contrast to the architecture of ANN which was built based on regular CPMA (i.e., without optimization - use stochastic hidden layers), the prediction accuracy of VR-PUF and 5-XOR VR-PUF is 67.81% and ≈50%, respectively.

All of the above known attacks on PUFs which include invasive, semi-invasive, and non-invasive attacks are summarised in Table 1. Though this is not a comprehensive list of attacks on PUFs, it gives an overview of PUF attack methods in general.

4 Countermeasures

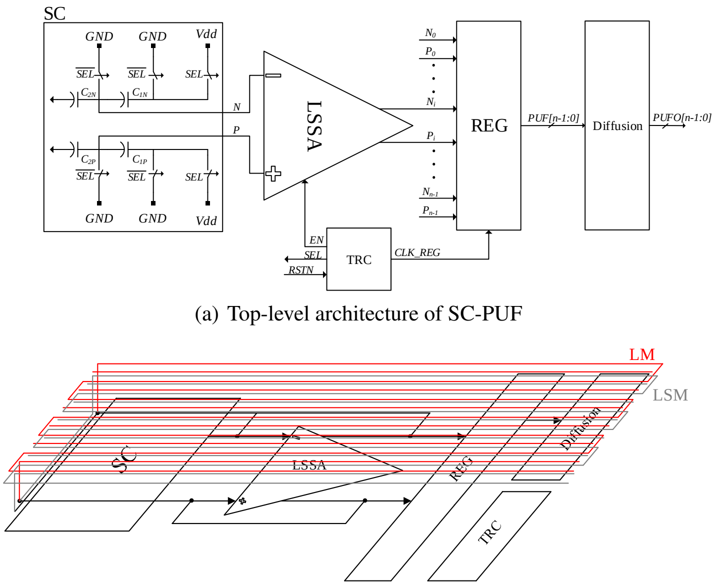
Various attack methods as discussed in Section 3 reveal the weaknesses of a particular PUF. Therefore, countermeasures to overcome these weaknesses are important to improve the quality of a PUF before its deployment in the field.

4.1 Invasive Attack Countermeasures

Some of the potential countermeasures to resist a cloning attack on SRAM-PUFs are discussed in [20], such as memory scrambling and building PUFs with synthesized logic. However, the resistance of these countermeasures against invasive attacks has not been investigated. In one study [49], an invasive-attack-resistant PUF, namely, switched-capacitor PUF (SC-PUF) was proposed which exploits the mismatch of capacitor ratios that is sampled by using a switched-capacitor (SC) circuit. Figure 11(a) depicts the top-level architecture of SC-PUF which consists of the SC circuit, latch-styled sense amplifier (LSSA), registers (REG), true random clock (TRC), and diffusion block. The circuit works when SEL goes to high, the up plates C_{1P} and C_{1N} are in the charge state as both capacitors are connected to the power supply (VDD). The voltage difference between P and N, denoted as ΔV_{PN}, depend on the mismatch ratio of $\frac{C_{1P}}{C_{2P}}$ and $\frac{C_{1N}}{C_{2N}}$. Subsequently, an LSSA amplifies the voltage difference and store the SC-PUF response in the REG. TRC acts as a control unit to enable/disable the LSSA and SEL signals assertion.

In order to resist an invasive attack, the sampling capacitors are connected to the transmission lines (i.e. metallisation) such that these capacitors become capacity sensitive. Figure 11(b) illustrates the mesh metallisation which consists of last metal (LM) and last-second metal (LSM). The proposed SC-PUF is fabricated using a 180nm technology node. Further, an invasive-attack-resistant test is conducted using a probe tip with 0.1 μm point radius and 3.3 mm length to measure the capacitance. The direct probing and destruction attacks (i.e., removing the metallisation to create a hole for probing) were conducted on SC-PUF and shown not to be successful. Nevertheless, the added metallisation increases the manufacturing cost but this is has not been discussed [49, 50].

A recent study, [51] proposed a tamper resistant spin transfer torque-magnetic random-access memory (STT-MRAM) by exploiting its switching properties. The magnetic tunnel junction (MTJ) is a basic element used in STT technology as depicted in Figure 12(a). An MTJ cell is composed of a barrier oxide layer in between two ferromagnetic layers and one access transistor. The magnetic layer with a fixed magnetic orientation is known as the reference layer (RL), whereas the other layer with freely rotating magnetization is known as the free layer (FL). MTJ cell is in the high-resistance state when the FL and the RL magnetizations are antiparallel (AP). In contrast, the MTJ cell is in a low-resistance state when the FL and the RL are parallel (P). For PUF usage, the MTJ cell is set in the AP state as it can offer a wider resistance distribution than the P state. Figure 12(b) illustrates the schematic of a single-bit PUF. When both of the MTJ cells are in AP, the PUF response will be determined by the randomly distributed resistance of the MTJ cells and intrinsic mismatches between two branches.



(a) Top-level architecture of SC-PUF

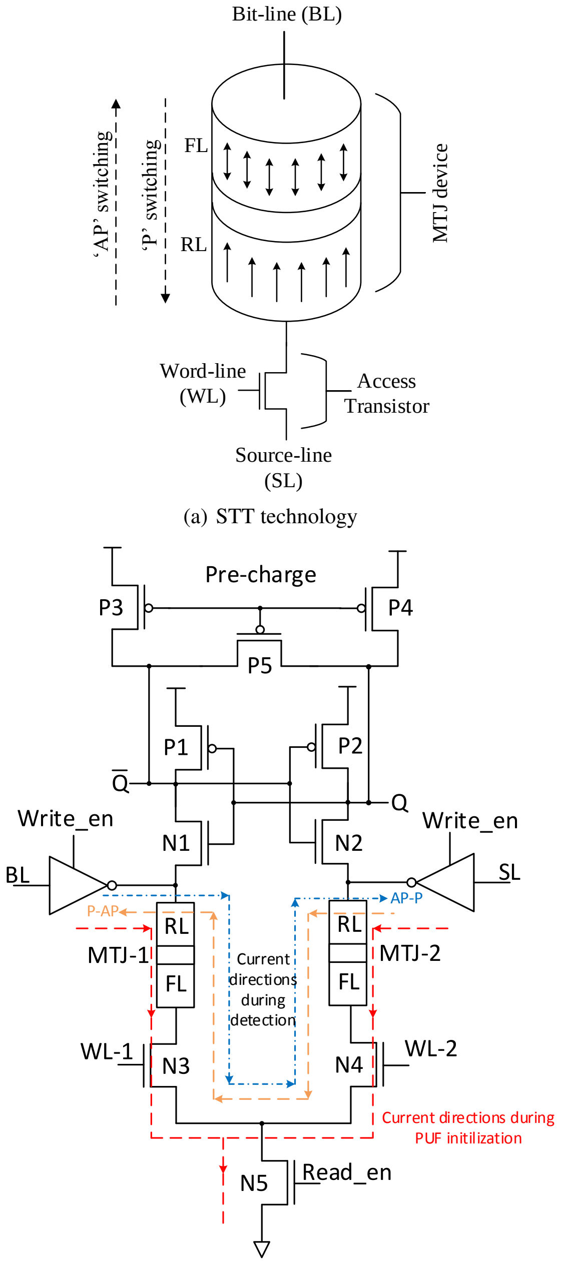
(b) SC-PUF with mesh metallization

Fig. 11: Proposed design of SC-PUF, an invasive attack resistant PUF [49].

As discussed in Section 3.1, the invasive attack proposed in [20] can also be performed on an STT-MRAM. The access transistor in Figure 12(a) is directly exposed from the back-side and can be tampered with to alter the value of PUF response. The output of single bit cell is deterministic when the MTJ cell is in the AP-P or P-AP state. This characteristic can be exploited detect a potential tampering attack. For example, the attacker tampered with the transistor N3 to obtain $Q = 1$. To detect the tampering, set the MTJ cells in AP-P. The branch with the MTJ in the AP state will be "1" and the branch with the MTJ in the P state will settle to "0". Hence, the tampered value, $Q = 1$ can be detected as the expected output should be $Q = 0$. The detection errors happen when the tampering attack effect is strong enough to modify the PUF bit value but too weak to modify the MTJ state comparisons.

4.2 Semi-invasive Attack Countermeasures

Merli *et al.* [25] discussed two countermeasures to overcome the semi-invasive attack on an RO-PUF using near-field EM. The first method is a combination of "non-overlapping" and "parallel comparison" techniques. As discussed in Section 3.2, m oscillators are compared to generate $(m - 1)$ PUF response bits [25]. To avoid the overlapping comparison, $\left(\frac{m}{2}\right)$ response bits can be extracted from m ROs. Furthermore, comparing all m oscillators in parallel reduces the information (frequencies) leakage and increasing the complexity of an EM attack. Nevertheless, the parallel comparison of all m oscillators causes immense hardware consumption since every pair of ROs needs a dedicated counter and comparator. Merli *et al.*, [25] suggested to measure a small number of n in parallel to extract $n - 1$ bits to keep the hardware overhead low. If the required number of total response bits is p, therefore, the number of RO groups is $\left(\frac{p}{n-1}\right)$. The second method is focusing on reducing the counter's information leakage by using an asynchronous ripple counter whereby only the first flip-flop is clocked by the RO signal and all others follow asynchronously.



(a) STT technology

(b) A single bit of STT-MRAM PUF

Fig. 12: Proposed design of STT-MRAM PUF, an invasive attack resistant PUF [51].

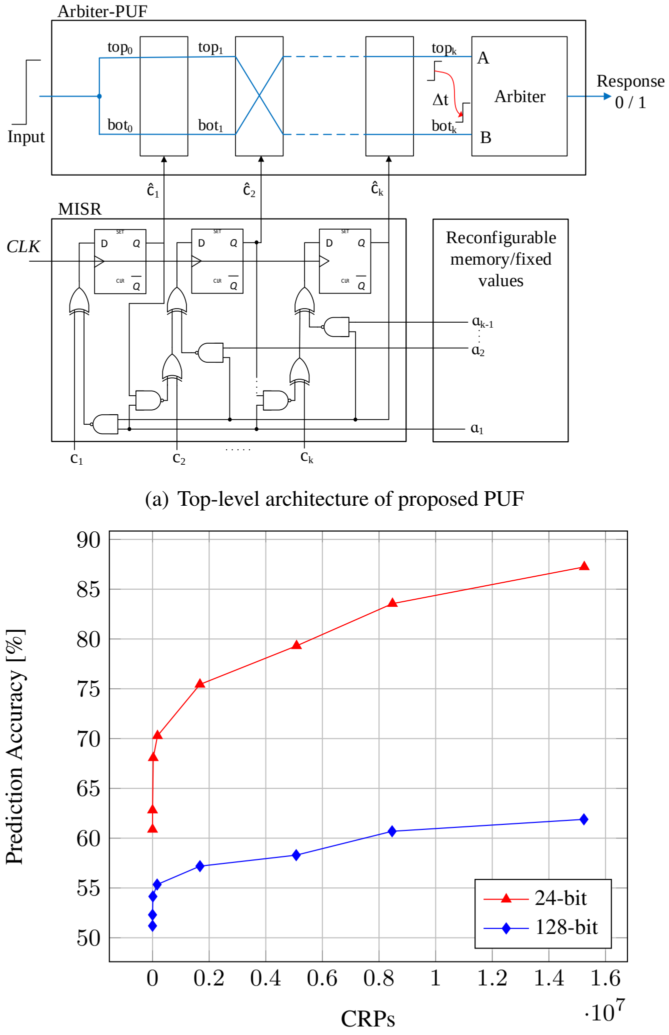
Another study focusing on enhancing the resistance of RO-PUF against EM attack is PUF based on transient effect RO (TERO) that is insensitive to the locking phenomenon by using the average number of oscillations as entropy extractor [52]. A locking phenomenon is described as an interaction between two oscillatory systems operating at close frequencies and spatially close to each other, hence make them dependent. The proposed TERO-PUF is exploiting the oscillatory meta-stability of SR latch which is composed of two AND gates and even number of inverters. Typically, the number of inverters is two but the loop can be extended by using more inverters to extend the oscillations. Both inputs of SR latch, S and R are connected to the *Ctrl* signal. To generate the random response, both outputs of SR latch, Q and $\bar{Q}$ are forced to '1' for some time. Subsequently, a rising pulse is applied to the *Ctrl* signal. Due to the intrinsic asymmetry of the cross-coupled circuits, the oscillatory meta-stability occurs for a short time. The counter, accumulator, and shift register are used to measure the mean value of the number of oscillations. The fact that TERO-PUF uses the average number of oscillations as an entropy extractor, hence it is not sensitive to the locking phenomenon. This study also claims that the proposed TERO-PUF is robust against EM attack as no information of oscillation frequency can be discovered from the number of oscillations.

4.3 Non-invasive Attack Countermeasures

4.3.1 Countermeasures to ML-attack: As for the non-invasive attacks, previous works in the literature focus on the countermeasures to increase the resilience of an Arbiter-PUF against an ML-attack. Generally, several techniques have been proposed in the literature. One of the techniques is randomising the PUF challenges. Ye *et al.* proposed an obfuscation logic-based PUF (OPUF) [53] and randomised challenge PUF (RPUF) [54] to increase the resilience of an Arbiter- PUF against an ML-attack. However, an OPUF suffers from reliability issues since it uses bi-stable (i.e. back-to-back inverter) cells to obfuscate the challenges. Mispan *et al.*, [7] proposed a challenge permutation that can be implemented by routing obfuscation at the input stage of Arbiter-PUFs. Elsewhere, a PUF protocol countermeasure technique by only exposing a subset of either challenges or responses is proposed to reduce the susceptibility of Arbiter-PUFs to an ML-attack [29, 55]. Gao *et al.* [29] proposed an Obfuscated-PUF (OB-PUF) in which a partial challenge is sent by the verifier to the OB-PUF (i.e., the prover). Subsequently, within an OB-PUF, a partial challenge is padded with a random pattern generated by a random number generator (RNG) to make up a full-length challenge. Earlier, Rostami *et al.* [55], proposed a sub-string matching technique in which only a subset of PUF response strings is sent to the verifier during authentication. Another technique is to introduce mixed-signal PUFs that adapt the architecture of Arbiter-PUF such as the Current Mirror-PUF, [56] and the VTC-PUF, [57].

A recent countermeasure to increase the robustness of Arbiter-PUF against ML-attack was proposed in [58]. Zalivaka *et al.* [58] proposed an Arbiter-PUF with challenge obfuscation and trinary digit (trit) quadruple responses. The notion of trit quadruple response is to enhance the reliability of PUF due to the metastability of the flip-flop. The challenge obfuscation is implemented using multiple input shift register (MISR) with reconfigurable seed and feedback polynomial coefficients to resist ML-attack. Figure 13(a) depicts the proposed Arbiter-PUF with MISR for obfuscation of input challenge. The covariance matrix adaptation evolution strategies (CMA-ES) has been deployed to perform ML-attack on 24-bit and 128-bit of the proposed PUF using 24 core Intel Xeon CPU server with 32 GB of RAM. The ML-attack results in Figure 13(b) show that for 128-bit PUF, the prediction accuracy is about 60% using 16 million CRPs. Zalivaka *et al.* [58] also proposed an authentication protocol consisting of 3 phases, which are enrolment, authentication, and update. It was argued that to collect 16 million CRPs using the proposed authentication protocol requires about 1875 years. Therefore, it was claimed that the proposed PUF is robust against ML-attack.

As mentioned in Section 3.3.1, one might disable the ML-attack by increasing the size of XOR Arbiter-PUF, with $l \geq 6$ (i.e, the number of XORs in output network) and $k > 128$ (i.e. challenge bit-length of Arbiter-PUF, see Figure 6). However, as l increases, the reliability of XOR Arbiter-PUF degrades severely [5]. In a study, Wisiol *et al.* [59] proposed a majority vote technique to enable the large and reliable design of XOR Arbiter-PUF, hence increase the resistant against ML-attack. A theorem to compute the upper bound of majority voting has been also developed (refer to [59] for the details). The resilience of Majority Vote XOR Arbiter-PUF against ML-attack has been evaluated using Becker's CMA-ES reliability attack technique [35]. As compared to the results reported in [35], the required number of CRPs for Majority Vote XOR Arbiter-PUF to achieve similar prediction accuracy as classic XOR Arbiter-PUFs is exponentially increased, as illustrated in Figure 14. In contrast, classic XOR Arbiter PUFs can be attacked with linear increase only and are limited by reliability decrease.



(a) Top-level architecture of proposed PUF

(b) Prediction accuracy of proposed PUF against ML-attack

Fig. 13: Proposed Arbiter-PUF using MISR with reconfigurable seed and feedback polynomial coefficients [58].

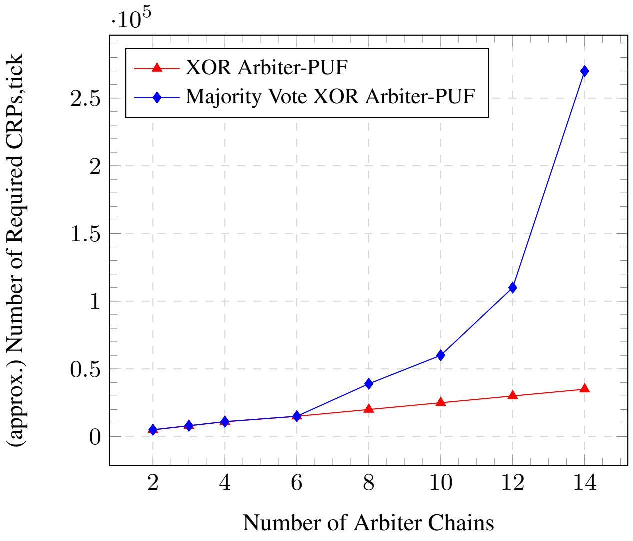


Fig. 14: Comparison of required number of CRPs between XOR Arbiter-PUF and Majority Vote XOR Arbiter-PUF for k=32.

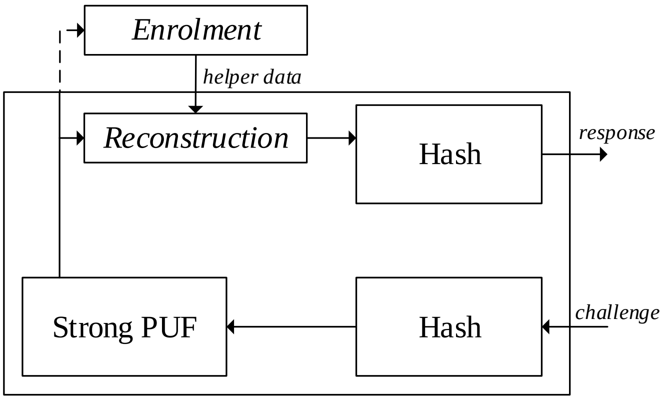


Fig. 15: The concept of Controlled PUF [62].

Herder *et al.*, [60, 61] proposed the PUF based challenge response protocol that cryptographically secure based on Learning Parity with Noise (LPN) problem. The computational security of the proposed protocol is reduced to the hardness of LPN. LPN problem is given as following. Let $\mathbf{s} \in \{0,1\}^n$ be chosen uniformly at random, generated using True Random Number Generator (TRNG). Let matrix $\mathbf{A} \in \{0,1\}^{m\times n}$ selected at random by the manufacturer, $m \geq n$. Let a noise vector $\mathbf{e} \in \{0,1\}^m$ be chosen from a distribution χ, generated using Physically Obfuscated Key (POK). Finally, the challenge $\mathbf{b} \in \{0,1\}^m$ and it is defined as:

$$\mathbf{b} = \mathbf{A} \cdot \mathbf{s} \oplus \mathbf{e} \tag{7}$$

The problem is to learn $\mathbf{s}$ given only the values of $\mathbf{b}$ and $\mathbf{A}$, when the noise vector $\mathbf{e}$ is distributed according to probability distribution χ. Once $\mathbf{s}$ is recovered, it is input to the hash function to generate the response. The RO-PUF is used as a source to generate the POK. The proposed protocol has internal error correction to generated error-free POK by using the concept of ‘confidence information’. Confidence information represents which bits of the output that have a higher/lower probability of error. In the study [60], the confidence information is the large difference in counts between the two ring oscillators which implies the higher confidence that environmental changes are unlikely to cause the output bit to flip when measured at a later time. The confidence information is generated in real-time during the measurement of POK, hence it so-called ‘stateless’ - requires no NVM for data storage. This approach eliminates the possibility of physical tampering such that the adversary has no ability to recover the confidence information. The proposed LPN-based PUF is provably secure as its security is reduced to the hardness of the LPN problem and claimed to be resistant against ML-attack.

A natural way to reduce the susceptibility of Strong PUFs to ML-attacks is to use hash functions as a logic processing unit to obfuscate the challenge and response mapping of a Strong PUF [62] (see Figure 15). One-way hash functions such as SHA-256 and SHA-3 consume large area and they are power-hungry, although they improve the unpredictability of Strong PUFs [7]. An increase in the area can be seen as a disadvantage for Strong PUF-based authentication using the challenge-and-response protocol as this protocol is intended for resource-constraint PUF-based RFID security devices [35]. Recently, lightweight hash functions have been proposed such as Hash-One [64], QUARK [65], and PHOTON [66]. Although the number of gates is significantly lower than the heavyweight hash functions (e.g., SHA-256), the lightweight hash functions still consume considerably large area for resource-constraint pervasive devices. As reported in [67], RFID devices typically require fewer than 1000 gate equivalents. Therefore, the requirements of small footprints and low power consumption for these resource-constraint pervasive devices introduce a significant challenge in providing fundamental security services, such as authentication and identification.

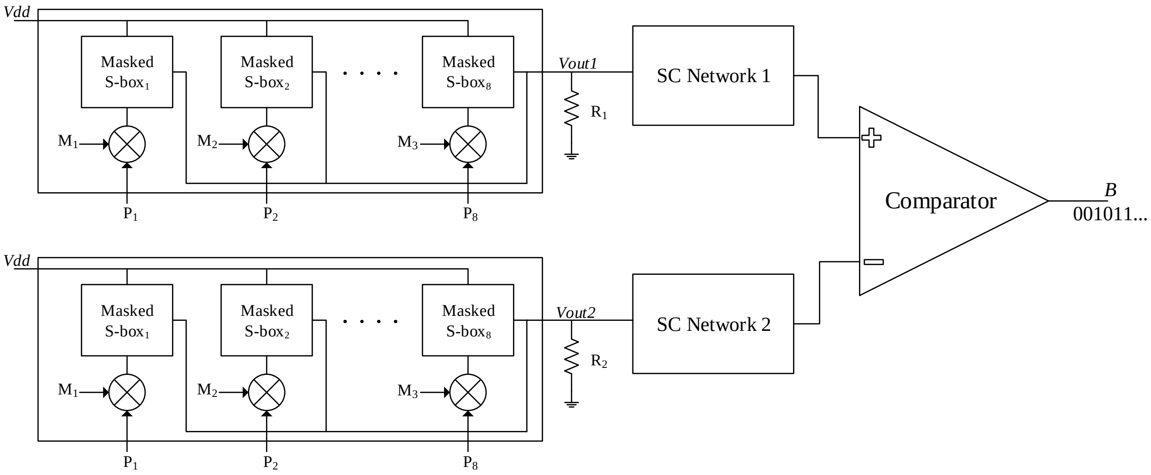


Fig. 16: Top-level architecture of Mask AES-PUF [63].

Elsewhere, a non-linear photonic PUF based on ultrafast non-linear optical interactions in chaotic silicon micro-cavities was proposed to resist the susceptibility to DL-attacks [68]. The adversary attack has been evaluated using DNN at three different input optical power levels, 32mW, 65mW, and 150mW. The non-linear silicon photonic PUF shows a promising resistance against DL-attack in which it achieves prediction accuracy of about 55% to 70% at three different power levels. Furthermore, the non-linear silicon photonic PUF offers repeatability and ease of integration with CMOS devices.

4.3.2 Countermeasures to Hybrid Attacks:

In a study, Yu *et al.*, [63] proposed a masked advanced encryption standard (AES) PUF, a combination of AES substitution-boxes (S-box) and switching capacitor networks to resist a hybrid attack. The analysis of the hybrid attack focuses on using DNN and power analysis to attack the proposed PUF. Figure 16 shows the top-level architecture of the proposed design in [63]. The input challenge, P is masked with M before being input to the S-box to enhance the security against DL-attack. The switching capacitor acts as a loading capacitance for the masked S-box in which the random mismatches in the capacitors due to the process variations are exploited for generating critical-authentication data against SCA-attack.

For a hybrid attack evaluation, the power leakage of the masked AES-PUF is assumed to be used by the adversary to assist the DL-attack. The measured input power dissipation for each applied challenge, C to generate a response, R is given as P_{in}. Therefore, the CRPs of the DNN model are described as $\{C_i, R_i\}$ where $C_i = \{P_1, P_2, \cdots, P_8, P_{in}\}$ and $R_i = B$. Figure 17 shows the prediction accuracy of masked AES PUF against DL and a hybrid attack. The result of the hybrid attack shows a low prediction rate of about 55.2% at 100,000 training CRPs. Nevertheless, the findings in [63] are not conclusive as the DL technique requires an enormous amount of CRP data to train the massive DNN [38].

Elsewhere, key-updating (KU) AES-PUF is proposed by embedding an AES between two 128-bit Arbiter-PUFs to encrypt their CRPs against ML-attack and the secret key of AES is updated for every cycle of response generation to increase the resistance against side-channel attacks [69]. Figure 18 shows the proposed KU AES-PUF. CNN has been deployed to perform an attack. The proposed PUF and 128-bit Arbiter-PUF have been implemented using a 130nm IBM CMOS technology in the Cadence environment. Table 2 shows the susceptibility comparison between the proposed PUF and 128-bit Arbiter-PUF using the CNN technique. As opposed to the classical Arbiter-PUF which achieved high prediction accuracy of about 92.7% with only 100,000 CRPs, the KU AES-PUF is hardly predictable even if 1 million CRPs are used to train the network model. The CRPs encryption using the AES block helps to increase the resistance against ML-attack. Meanwhile, the resistance against side-channel attack has been evaluated using power attack analysis. By combining the input challenge, C and the hypothesized secret key, the power attack analysis is performed to estimate the secret key. With the key-updating implementation, it greatly weakens the correlation between the power dissipation of the processed data against the power attack.

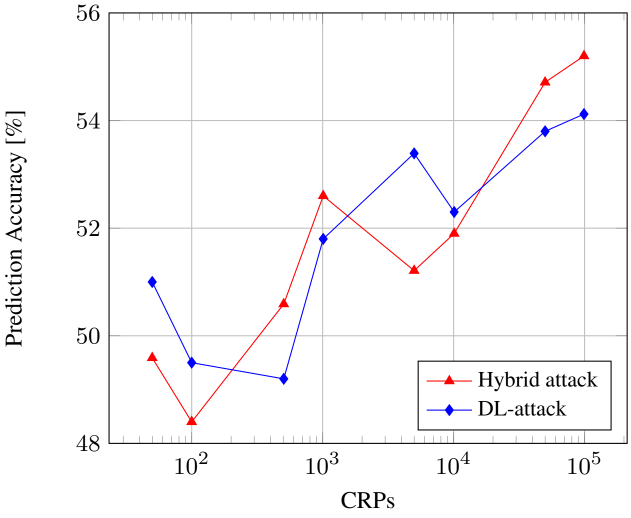


Fig. 17: Prediction accuracy of masked AES PUF against DL and hybrid attack.

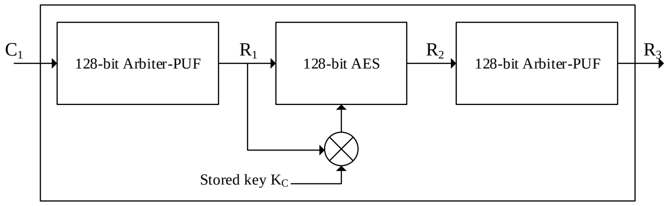


Fig. 18: Top level architecture of KU AES-PUF [69].

5 Opportunities and Future Challenges

Despite being termed "unclonable functions", there have been some successful cloning and modelling attacks against PUFs. Based on our survey, as described in Section 3, most of the attacks focus on

Table 2 CNN attack comparison.

CRPs	128-bit KU AES-PUF		128-bit Arbiter-PUF	
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
100,000	0.512	0.509	0.934	0.927
500,000	0.527	0.521	NR	NR
1,000,000	0.516	0.518	NR	NR

using non-invasive techniques. An invasive attack seems a promising way to clone a PUF but it requires complex IC failure analysis instruments which can be very expensive, assuming that the adversary has no access to this equipment. Conversely, the non-invasive technique requires software/hardware to perform an ML/DL attack which costs less than $1000, as described in Section 3.3.1. Additionally, an SCA attack can be performed using basic electronic equipment to measure the power traces of the targeted devices.

The recently reported attack on PUFs was using the DL technique which is expected to be more powerful than conventional ML technique. DL requires no feature extraction as compared to conventional ML technique, hence PUF system can be treated totally as a black box. The hybrid attack which combined SCA and DL techniques seems very promising in attacking high complexity CRPs and open a new challenge to overcome this type of attack. Although most of the authors have pointed out potential countermeasures as described in Section 4, most of these countermeasures incur high hardware overhead by embedding AES and/or HASH functions into the PUF system. One of the opportunities to be further explored is the PUF based on non-linear photonic to increase the non-linearity against DL-attack. Nevertheless, the proposed method must not sacrifice the reliability of the PUF response as previously experienced by using the XOR technique.

Another area to be explored is the evaluation method to identify the existence of side-channel leakages. The method should be able to describe statistically the potential of side-channel leakages without the need for actual modeling attacks. The development of this method could be very useful to quantify the performance of PUF against all possible SCA attacks. Therefore, an efficient countermeasure can be implemented to resist the SCA attack. Elsewhere, the continuous development of quantum computing and its application has been seen as a major threat to the cryptographic algorithms. The cryptographic algorithms such as Rivest-Shamir-Adleman (RSA), elliptic-curve cryptography (ECC), and LED-like block chipers are becoming insecure with the development of quantum computers [70, 71]. The term post-quantum cryptography has been established to represents the cryptographic algorithms that are secure against an attack by a quantum computer. Quantum computing can be exploited to improve the ML-attack efficiency of PUFs. Thorough research is needed to ensure that PUFs are robust and secure against quantum computing, and stay relevant in the cryptographic field.

6 Conclusion

Various attack methods such as invasive, semi-invasive and non-invasive attacks have been proposed to attack Strong PUFs and Weak PUFs. In this survey, some of these known attacks to PUFs are discussed which reveal the weaknesses of a particular PUF. Countermeasures to the aforementioned types of attacks are also surveyed. Despite the known attacks on PUFs, a few advantages still hold for a PUF such as requires no key programming, device-specific key, easy key management, low cost and scalability. Shortly, intense competition is expected between PUF-designers and PUF-breakers in the area of Strong PUFs and Weak PUFs. More fundamental research is required, aiming to create functionally unclonable Strong PUFs and Weak PUFs with great cryptographic properties. Furthermore, a lightweight implementation of PUF is necessary to promote the widespread adoption of PUFs. This survey can facilitate future PUF research.

7 References

1 V. V. D. Leest, R. Maes, G.-J. S. Pim, and P. Tuyls, "Hardware intrinsic security to protect value in the mobile market," in *Information Security Solutions Europe*, 2014, pp. 188–198.

2 J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede, "A survey on lightweight entity authentication with strong PUFs," *ACM Computing Surveys*, vol. 48, no. 2, pp. 26:1–26:42, 2015.

3 Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "Emerging physical unclonable functions with nanotechnology," *IEEE Access*, vol. 4, pp. 61–80, 2016.

4 B. Halak, "Security attacks on physically unclonable functions and possible countermeasures," in *Physically Unclonable Functions : From Basic Design Principles to Advanced Hardware Security Applications*. Springer International Publishing, 2018, pp. 131–182.

5 M. S. Mispan, "Towards Reliable and Secure Physical Unclonable Functions," Ph.D. Thesis, University of Southampton, 2018.

6 B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *ACM Conference on Computer and Communications Security*, 2002, pp. 148–160.

7 M. S. Mispan, H. Su, M. Zwolinski, and B. Halak, "Cost-Efficient Designs for Modeling Attacks Resistant PUFs," in *Design, Automation & Test in Europe Conference & Exhibition*, 2018.

8 M. S. Mispan, B. Halak, and M. Zwolinski, "Lightweight Obfuscation Techniques for Modeling Attacks Resistant PUFs," in *IEEE International Verification and Security Workshop*, 2017, pp. 19–24.

9 J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *Symposium on VLSI Circuits Digest of Technical Papers*, 2004, pp. 176–179.

10 J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *International Conference on Cryptographic Hardware and Embedded Systems*, 2007, pp. 63–80.

11 S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract : The Butterfly PUF protecting IP on every FPGA," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 67–70.

12 P. Simons, E. Van Der Sluis, and V. Van Der Leest, "Buskeeper PUFs, a promising alternative to D Flip-Flop PUFs," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2012, pp. 7–12.

13 B. Halak, "A primer on cryptographic primitives and security attacks," in *Physically Unclonable Functions : From Basic Design Principles to Advanced Hardware Security Applications*. Springer International Publishing, 2018, pp. 1–15.

14 D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. V. Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1200–1205, 2005.

15 Y. Su, J. Holleman, S. Member, B. P. Otis, and A. Abstract, "A digital 1.6 pJ/bit chip identification circuit using process variations," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 69–77, 2008.

16 M. S. Mispan, S. Duan, B. Halak, and M. Zwolinski, "A reliable PUF in a dual function SRAM," *Integration*, vol. 68, pp. 12–21, 2019.

17 M. S. Mispan, M. Zwolinski, and B. Halak, "Ageing mitigation techniques for SRAM memories," in *Ageing of Integrated Circuits*, B. Halak, Ed. Springer, Cham, 2020, pp. 91–111.

18 U. Guin, A. Singh, M. Alam, J. Canedo, and A. Skjellum, "A secure low-cost edge device authentication scheme for the internet of things," in *International Conference on VLSI Design*, 2018, pp. 85–90.

19 S. Sutar, S. Member, and A. Raha, "Memory-based combination PUFs for device authentication in embedded systems," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 793–810, 2018.

20 C. Helfmeier, C. Boit, D. Nedospasov, and J. P. Seifert, "Cloning physically unclonable functions," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2013, pp. 1–6.

21 A. Schlösser, D. Nedospasov, J. Krämer, S. Orlic, and J.-P. Seifert, "Simple Photonic Emission Analysis of AES," in *Cryptographic Hardware and Embedded Systems - CHES 2012*, E. Prouff and P. Schaumont, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 41–57.

22 A. Roelke and M. R. Stan, "Attacking an SRAM-based PUF through Wearout," in *IEEE Computer Society Annual Symposium on VLSI*, 2016, pp. 206–211.

23 D. Nedospasov, J. P. Seifert, C. Helfmeier, and C. Boit, "Invasive PUF analysis," in *Fault Diagnosis and Tolerance in Cryptography*, 2013, pp. 30–38.

24 S. Tajik, E. Dietz, S. Frohmann, J.-P. Seifert, D. Nedospasov, C. Helfmeier, C. Boit, and H. Dittrich, "Physical characterization of Arbiter PUFs," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2014, pp. 493–509.

25 D. Merli, D. Schuster, F. Stumpf, and G. Sigl, "Semi-invasive EM attack on FPGA RO PUFs and countermeasures," in *ACM Workshop on Embedded Systems Security*, 2011, pp. 1–9.

26 G. E. Suh and S. Devadas, "Physical Unclonable Functions for device authentication and secret key generation," in *ACM/IEEE Design Automation Conference*, 2007, pp. 9–14.

27 U. Ruhrmair and J. Solter, "PUF modeling attacks: An introduction and overview," in *Design, Automation & Test in Europe Conference & Exhibition*, 2014, pp. 1–6.

28 D. Lim, "Extracting secret keys from integrated circuits," MSc. Thesis, Massachusetts Institute of Technology, 2004.

29 Y. Gao, G. Li, H. Ma, S. F. Al-Sarawi, O. Kavehei, D. Abbott, and D. C. Ranasinghe, "Obfuscated challenge-response: A secure lightweight authentication mechanism for PUF-based pervasive devices," in *IEEE International Conference on Pervasive Computing and Communication Workshops*, 2016, pp. 1–6.

30 NXP Semiconductors N.V., "PUF - Physical Unclonable Functions: Protecting next-generation smart card ics with sram-based pufs," 2013. [Online]. Available: http://www.nxp.com/documents/other/75017366.pdf

31 ——, "Step up security and innovation with next generation SmartMX2 products," 2016. [Online]. Available: https://cache.nxp.com/docs/en/brochure/75017695.pdf

32 M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," in *IEEE/ACM International Conference on Computer-Aided Design*, 2008, pp. 670–673.

33 U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF modeling attacks on simulated and silicon data," *IEEE Transactions on Information Forensic and Security*, vol. 8, pp. 1876–1891, 2013.

34 U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, and W. Burleson, "Efficient power and timing side channels for physical unclonable functions," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2014, pp. 476–492.

35 G. T. Becker, "The gap between promise and reality: On the insecurity of XOR Arbiter PUFs," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2015, pp. 535–555.

36 G. Hospodar, R. Maes, and I. Verbauwhede, "Machine learning attacks on 65nm Arbiter PUFs: Accurate modeling poses strict bounds on usability," in *IEEE International Workshop on Information Forensics and Security*, 2012, pp. 37–42.

37 A. Vijayakumar, V. C. Patil, C. B. Prado, and S. Kundu, "Machine learning resistant strong PUF : Possible or a pipe dream?" in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2016, pp. 19–24.

38 M. Khalafalla and C. Gebotys, "PUFs Deep Attacks: Enhanced modeling attacks using deep learning techniques to break the security of double arbiter PUFs," in *Design, Automation and Test in Europe Conference and Exhibition*, 2019, pp. 204–209.

39 T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, "Implementation of Double Arbiter PUF and Its Performance Evaluation," in *Asia and South Pacific Design Automation Conference*, 2015, pp. 6–7.

40 P. Santikellur, Lakshya, S. R. Prakash, and R. S. Chakraborty, "A computationally efficient tensor regression network based modeling attack on XOR APUF," in *IEEE Asian Hardware Oriented Security and Trust Symposium*, 2019, pp. 1–6.

41 F. Ganji, "On the Physical Security of Physically Unclonable Functions," Ph.D. Thesis, Technical University of Berlin, 2017.

42 J. Delvaux and I. Verbauwhede, "Fault injection modeling attacks on 65 nm arbiter and RO Sum PUFs via environmental changes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 6, pp. 1701–1713, 2014.

43 S. Zeitouni, Y. Oren, C. Wachsmann, P. Koeberl, and A.-r. Sadeghi, "Remanence decay side-channel: The PUF case," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1106–1116, 2016.

44 D. Merli, D. Schuster, F. Stumpf, and G. Sigl, "Side-channel analysis of PUFs and fuzzy extractors," in *International Conference on Trust and Trustworthy Computing*, 2011, pp. 1–15.

45 W. Yu and Y. Wen, "Efficient hybrid side-channel/machine learning attack on XOR PUFs," *Electronics Letters*, vol. 55, no. 20, pp. 1080–1082, 2019.

46 Y. Liu, Y. Xie, C. Bao, and A. Srivastava, "A combined optimization-theoretic and side-channel approach for attacking strong physical unclonable functions," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 1, pp. 73–81, 2018.

47 W. Yu, "Optimization of Combined Power and Modeling Attacks on VR PUFs with Lagrange Multipliers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. PP, no. PP, pp. 1–5, 2020.

48 R. Liu, H. Wu, Y. Pang, H. Qian, and S. Yu, "A highly reliable and tamper-resistant RRAM PUF: Design and experimental validation," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2016, pp. 13–18.

49 M. Wan, Z. He, S. Han, K. Dai, and X. Zou, "An invasive-attack-resistant PUF based on switched-capacitor circuit," *IEEE Transaction on Circuits and Systems I: Regular Papers*, vol. 62, no. 8, pp. 2024–2034, 2015.

50 Z. He, M. Wan, J. Deng, C. Bai, and K. Dai, "A reliable strong PUF based on switched-capacitor circuit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 6, pp. 1073–1083, 2018.

51 S. B. Dodo, R. Bishnoi, S. M. Nair, and M. B. Tahoori, "A spintronics memory puf for resilience against cloning counterfeit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1–12, 2019.

52 L. Bossuet, X. T. Ngo, Z. Cherif, and V. Fischer, "A PUF based on a transient effect ring oscillator and insensitive to locking phenomenon," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 1, pp. 30–36, 2014.

53 J. Ye, Y. Hu, and X. Li, "OPUF: Obfuscation logic based physical unclonable function," in *IEEE International On-Line Testing Symposium*, 2015, pp. 156–161.

54 ——, "RPUF: Physical unclonable function with randomized challenge to resist modeling attack," in *IEEE Asian Hardware Oriented Security and Trust Symposium*, 2016, pp. 1–6.

55 M. Rostami, M. Majzoobi, F. Koushanfar, D. Wallach, and S. Devadas, "Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, pp. 37–49, 2014.

56 R. Kumar and W. Burleson, "On design of a highly secure PUF based on non-linear current mirrors," in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2014, pp. 38–43.

57 A. Vijayakumar and S. Kundu, "A novel modeling attack resistant PUF design based on non-linear voltage transfer characteristics," in *Design, Automation & Test in Europe Conference & Exhibition*, 2015, pp. 653–658.

58 S. S. Zalivaka, A. A. Ivaniuk, and C.-H. Chang, "Reliable and modeling attack resistant authentication of arbiter PUF in FPGA implementation with trinary quadruple response," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1109–1123, 2019.

59 N. Wisiol and M. Margraf, "Why attackers lose: design and security analysis of arbitrarily large XOR arbiter PUFs," *Journal of Cryptographic Engineering*, vol. 9, no. 3, pp. 221–230, 2019.

60 C. Herder, L. Ren, M. Van Dijk, M. D. Yu, and S. Devadas, "Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 65–82, 2017.

61 C. Jin, C. Herder, L. Ren, P. H. Nguyen, B. Fuller, S. Devadas, and M. van Dijk, "FPGA implementation of a cryptographically-secure PUF based on learning parity with noise," *Cryptography*, vol. 1, no. 3, pp. 1–20, 2017.

62 B. Gassend, M. van Dijk, D. Clarke, E. Torlak, and S. Devadas, "Controlled physical random functions and applications," *ACM Transactions on Information and System Security*, vol. 10, no. 4, pp. 15:1 –15:22, 2008.

63 W. Yu and J. Chen, "Masked AES PUF: A new PUF against hybrid SCA/MLAs," *Electronics Letters*, vol. 54, no. 10, pp. 618–620, 2018.

64 P. Mukundan, S. Manayankath, C. Srinivasan, and M. Sethumadhavan, "Hash-One: A lightweight cryptographic hash function," *IET Information Security*, vol. 10, no. 5, pp. 225–231, 2016.

65 J.-P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia, "Quark: A lightweight hash," *Journal of Cryptology*, vol. 26, no. 2, pp. 313–339, 2013.

66 J. Guo, T. Peyrin, and A. Poschmann, "The PHOTON family of lightweight hash functions," in *Advances in Cryptology-CRYPTO*, 2011, pp. 222–239.

67 E. Oztürk, G. Hammouri, and B. Sunar, "Towards robust low cost authentication for pervasive devices," in *IEEE International Conference on Pervasive Computing and Communications*, 2008, pp. 170–178.

68 I. Atakhodjaev, B. T. Bosworth, B. C. Grubel, M. R. Kossey, J. Villalba, A. B. Cooper, N. Dehak, A. C. Foster, and M. A. Foster, "Investigation of deep learning attacks on nonlinear silicon photonic PUFs," in *Conference on Lasers and Electro-Optics*, 2018, pp. 1–2.

69 Y. Wen, S. F. Ahamed, and W. Yu, "A novel PUF architecture against non-invasive attacks," in *ACM/IEEE International Workshop on System Level Interconnect Prediction*, 2019, pp. 1–5.

70 L. Xu, J. Guo, J. Cui, and M. Li, "Key-recovery attacks on LED-like block ciphers," *Tsinghua Science and Technology*, vol. 24, no. 5, pp. 585–595, 2019.

71 Y. Wang, X. Xi, and M. Orshansky, "Lattice PUF: A strong physical unclonable function provably secure against machine learning attacks," in *Available online: https://arxiv.org/pdf/1909.13441v2.pdf (accessed on 1 September 2020).*