Exploiting PUF Unreliability to Secure Wireless Sensing

Yansong Gao*, Hua Ma*, Said F. Al-Sarawi, Derek Abbott, and Damith C. Ranasinghe.

***Abstract*—Wireless sensors attract increased attention from both academia and industry owing to emerging applications such as Internet of Things (IoT), smart homes, e-health, etc. It is widely accepted that security assessment to this super large distributed ubiquitous devices and privacy of collected data are ultimate important, Sensor security that relies on traditional cryptography is vulnerable to various attacks and usually does not lend itself to low-cost and lightweight applications. To overcome it, this paper proposes an alternative secure wireless sensing approach that is suitable for those resource-restricted IoT devices. In particular, we exploit the unreliability of a physical unclonable function (PUF) that is sensitive to ambient environmental variations to guarantee the veracity of the sensed value. In this case, the PUF itself acts as a sensor or is integrated with a sensor, called a *PUF sensor*. Thus, for a PUF sensor, the processes of cryptography and sensing are inseparable. In our security analyses, it is assumed that i) the PUF sensor is located in a hostile environment, ii) the communication channel is insecure, and iii) no pricey crypto module relying on stored secret keys is involved. Even under such cases, the PUF sensor still provides high level security at low-cost. In addition, the PUF sensor is inherently unclonable. We validate such an alternative wireless sensing approach based on an proof-of-concept experimental implementation of the proposed PUF sensor.**

***Index Terms*—PUF sensor, wireless sensing, hardware security, physical unclonable function, modeling attacks.**

I. Introduction

Wireless ireless sensors are widely used in applications such as monitoring wildfires, traffic, building security, or a hospital patient's movement. There are emerging applications such as building smart homes, smart cities, and Internet of Things that depend on the wireless sensor networks. Veracity of the sensor measurement is a security issue in the aforementioned applications. If the measurement sent to the user is spoofed, it may lead to incorrect decisions, and consequently may threaten personal safety. The security of a sensor, traditionally, relies on a separate crypto module encrypting measurement values obtained from analog sensors—however, this is usually not suitable for low-cost and lightweight application scenarios. Moreover, security of cryptographic algorithms rely on digital secret keys stored in non-volatile memory (NVM) that is assumed untouchable or unbreakable. It appears such an assumption cannot hold nowadays as digital secret keys within NVM can be extracted via various types of attacks [1].

The emerging hardware security primitive—physical unclonable function (PUF)—provides a promising lightweight security solution for resource-constrained devices [2], [3]. A PUF is a small hardware device exploiting the imperfections or uncertainties of its fabrication process, which cannot be physically cloned and very hard, if not impossible, to be physically attacked. The PUF maps an input (*challenge*) to an output (*response*) through a complex physical function that is mathematically analogous to the one way function. The physical function is derived from the inherent static randomness resulting from uncontrollable process variations during manufacturing. Therefore, responses differ significantly from different PUF instances given the same queried challenge, even if these PUF instances have identical design and consequently fabricated by the same manufacturer.

The PUF is expected to regenerate the same response when it is queried by the same challenge. However, in practice, it is prone to the changes in the environment. In typical PUF-based applications, for instance, cryptographic key generation requiring high stable response regeneration [4], it is imperative to improve PUF reliability for the sake of easing the error correction implementation overhead to decrease area and power consumption. In PUF-based authentication applications [3], [5], it is preferable to maximize reliability to increase the complexity of modeling attacks by an adversary [6], [7], [8].

By contrast, we exploit this unavoidable unreliability to provide a high degree of assurance of sensed data, where the PUF itself is a sensor. Specifically, the unreliable response bit may flip from one state '0'/'1' to its opposite state '1'/'0' if an environmental parameter varies. However, note that such a response bit will nevertheless reproduced consistently for a given environmental condition. Unreliable response bits are not desired in typical PUF applications since they result in errors in key generation and authentication applications. However, these otherwise unreliable response bits can track environmental parameter changes in a repeatable manner. From such a perspective, the PUF itself can be used as a sensor or integrated with a sensor by exploiting response bits that are inherently sensitive to the environmental changes.

The concept of employing the PUF as a sensor to sense a particular physical quantity (PQ)—an environmental parameter—has been recently presented by Rosenfeld *et al.* [9]. Here, the PUF takes not only the challenge but also a PQ—specifically light used in [9]—as its inputs. Hence, the response is mapped from two inputs instead of one as shown in Fig. 1. The motivation is to merge sensing with cryptography.

Contributions are equal as co-first authors.

Y. Gao, S F. Al-Sarawi, D. Abbott are with the School of Electrical and Electronic Engineering, The University of Adelaide, SA 5005, Australia. e-mail: {yansong.gao, said.alsarawi, derek.abbott}@adelaide.edu.au.

H. Ma, D. C. Ranasinghe are with the Auto-ID Labs, School of Computer Science, The University of Adelaide, SA 5005, Australia. e-mail: {mary.ma, damith.ranasinghe}@adelaide.edu.au.

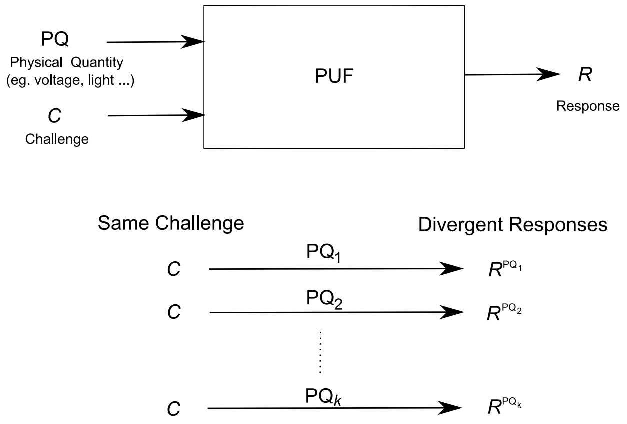

Figure 1. PUF as a sensor. The response is determined by both the PQ and the queried challenge.

More specifically, in this case, the PUF itself acts as a sensor to avoid a traditional crypto module, which is vulnerable to attacks once the underling stored key is extracted. Note the optical coating based PUF is not experimentally validated [9]. Its performance is evaluated through simulation to demonstrate the feasibility of sensing a PQ—light. Ruhrmair *et al.* [10] select temperature as a PQ to experimentally demonstrate their new security concept, *virtual proof of reality*—a complementary security concept of physical zero-knowledge protocols [11]—that assures the proof of a physical statement over an untrusted digital communication channel between two parties (a "prover" and a "verifier") without cryptographic algorithms.

In this paper, in contrast to previous studies, we i) in comparison with [9], firstly, use experimental data rather than simulation. Secondly, we harness the inherent unreliable responses—generally, an undesirable performance of a PUF—induced by an electrical signal fluctuations for secure sensing applications. ii) In comparison with [10], it is observed that the response sensitivity to temperature variation is not high—also reported in [12], which will result in greatly increased bit length of the response in order to successfully sense the temperature in which environment the PUF is placed [10]. The bit length of the response is significantly decreased owing to the method that we proposed to select unreliable response bits, which are highly prone to voltage variations. iii) Compared to both previous studies [9], [10], instead of only sensing a specific PQ, eg. temperature [10] or light [9]. Sensing electrical signals—eg. voltage—seems more attractive, because the chosen electrical parameter is versatile, various types of PQs such as temperature, humidity, sound energy, can eventually be converted into electrical signals that can be used to influence the reliability of the PUF. Therefore, these PQs are able to be securely sensed indirectly through the corresponding electrical signal. Our contributions in this paper are threefold:

1) We *extend* the conventional PUF to be a sensor or part of a sensor by exploiting its unreliability to secure wireless sensing—in particular, sensing change in voltage or other PQs that can be converted into electrical signals—without implementing a separate traditional crypto module. While the extended sensing function has no influence on the PUF's performance while it is still serving as a trust anchor bounded to a hardware device.
2) We validate the feasibility of our proposed wireless sensing methodology using PUF sensor through empirical data collected from five ring oscillator PUFs (ROPUFs) implemented in five FPGA boards [3], [13], [14]. In addition, the security of the PUF sensor is analyzed in details.
3) We present an approach to speed up the selection of unreliable response bits that are highly sensitive to voltage variations to greatly decrease the bit length of the response for sensing.

The rest of the paper is organized as follows. Related work is introduced in Section II. Section III presents a sensor implementation that exploits the PUF unreliability for sensing. Then experimental validation is presented in Section IV. In-depth discussions on the security and feasibility are carried out in Section V followed by a conclusion.

II. Related Work

A. Physical Unclonable Functions

The PUF primitive is first proposed by Pappu *et al.* in 2001 [15], [16]. The implementation is an optical PUF, initially dubbed a physical one-way function. The response (speckle pattern) is dependent on the input laser location/polarization (challenge) when the laser irradiates a stationary scattering medium. The optical PUF is limited by its need to be integrated with electronic hardware, leading to increased implementation cost. Following this prototype PUF, a practical implementation of a microelectronic circuit based PUF initially called a Physical Random Function, later termed the Arbiter PUF (APUF), was proposed by Gassend *et al.* [17]. The APUF exploits manufacturing variability in gate and wire delays as the source of unclonable randomness. The response is generated based on the time delay difference between two signal propagation paths consisting of serially connected electronic cells, eg., multiplexers. The path through each electronic cell is determined by a selection bit in a challenge (input bit vector) [18]. This structure is simple and capable of generating an exponential number of challenge response pairs (CRPs). Later, more variants of APUFs were proposed such as the XOR-APUF [3], [6] and the feed forward APUF [2], [6] to increase the complexity of modeling/emulation attacks that is considered as a plausible attack on PUFs. The APUF is prone to be metastable when the delay difference between two paths are close to zero given certain challenges, because the arbiter, eg. a latch, is unable to definitely determine the winning signal path due to inability of such gates to resolve a small time difference. Moreover, the APUF requires stringent routing to guarantee two identical symmetrical paths for sake of the response being predominated by the process variations, especially, when it is implemented in FPGA platforms.

The ring oscillator PUF (ROPUF), was firstly proposed in [3] to mitigate the above issues of the APUF. The ROPUF is further improved to conquer its limited number of CRPs [14], [19] and increase resilience to modeling attacks [14]. A survey of ROPUFs can be found in [20]. A typical ROPUF structure

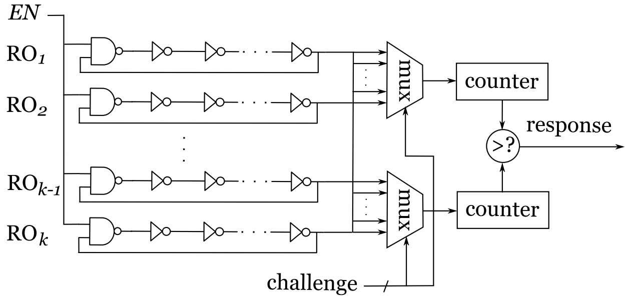

Figure 2. Typical structure of a ring oscillator PUF (ROPUF).

consists of k ring-oscillators (ROs), two k-to-1 multiplexers that select a pair of ROs, RO$_i$ and RO$_j$, two counters and a comparator, as shown in Fig. 2. All ROs are designed identically. Ideally, the frequency of each oscillator must be equal. However, because the oscillating frequency is a function of the physical device parameters, which are subject to process variations, the oscillation frequencies of different oscillators are not identical. Therefore, the oscillation frequencies of each pair are compared by counting the frequency using a digital counter. If $f_i < f_j$, where f_i and f_j are the oscillating frequencies of RO$_i$ and RO$_j$, respectively, the digital comparator output will be '0', otherwise '1'. The pairing of oscillators is controlled using two digital multiplexers, each uses a subset of the input challenge bits to select an RO.

Besides the aforementioned delay-based PUFs, there are mismatch based silicon PUFs such as the SRAM PUF [21], [22], [23], latch PUF [24], flip-flop PUF [25], [26], butterfly PUF [27], and analog PUFs based on silicon such as the current-based PUF [28] and nonlinear current mirror based PUF [29], which exploit nonlinear dynamic characterizations of current. Comprehensive reviews of conventional PUF designs can be found in [30], [31]. In recent years, emerging PUFs with nanotechnology are initially investigated aiming to build PUFs beyond the aforementioned conventional silicon PUFs by taking advantage of prevalent process variations as a consequence of scaling down to the nano region, and other unique properties offered in emerging nanoelectronics devices [32], [33], [34], [35], [36]. A survey of such nano PUFs can be found in [37].

B. PUF Sensor

The definition of PUF sensors is first given in [9]. A PUF sensor has the following features:

(a) Its response is not only a direct function of the challenge, rather has a strongly dependence on a particular PQ.
(b) Two identical PUFs cannot be forged.
(c) The response stays relatively stable given the same challenge and the same PQ value.
(d) Given one challenge-response pair for a PQ value, it is hard to predict the response for the same challenge to a different PQ value.

It is practical that a PUF can satisfy the above features. In this paper, the inherent unreliability of PUF fits feature (a). The inherent randomness in manufacturing process guarantees features (b) and (d) owing to the unpredictability of the responses. We discuss PUF resilience to modeling attacks in Section V to ensure feature (d) as well. For feature (c), the PUF is only required to be sensitive to a specific PQ (eg. voltage) but insensitive to other uninterested PQ (eg. temperature). This feature can also be easily met in practice, which will be discussed in detail in Section V.

Most importantly, the PUF sensor cannot be physically cloned, which is a major difference from the traditional separated crypto module using the stored digital keys, which can be extracted and physically cloned.

III. Secure Wireless Sensing Based on PUF Unreliability

A. Reliable Responses Based on Unreliable Responses

The reliability of a PUF is the probability of regenerating the same response given the same challenge queried to the same PUF instance [38]. It is always evaluated by its complementary performance metric—bit error rate (BER). In detail, for the same challenge applied to the same PUF, BER is the probability of two randomly regenerated responses, $\mathbf{R}$ and $\mathbf{R}'$, from the same PUF instance by applying the same challenge that are same. The BER is an average evaluation to all of responses generated by a PUF. This assumes that each bit in a response vector $\mathbf{R}$ has an equally probability of error, however, this might not be the case in reality [39]. In fact, it has been experimentally demonstrated that certain response bits are more prone to be erroneous than other bits [39], [40].

Considering the reliability of a specific 1-bit response r for a given challenge, in reality, the reliability for different responses r, is heterogeneous in nature. In other words, it is inappropriate to evaluate the reliability of a specific response bit r using the BER that is an average evaluation metric. For example, for a response bit r_1, if the probability of delivering '1' is 99% given multiple regenerations—there is only 1% probability for r_1 flipped to its unstable state, then the reliability for this specific response r_1 is 99%. It is clear that for most 1-bit responses r, the reliability is 100%. This relies on the fact that the BER is small, less than 10%, based on experimental results of most popular PUF designs, eg. APUF, ROPUF and SRAM PUF [30]. Therefore, most bits in an n-bit response vector $\mathbf{R}$ will be invariant.

Note that the unreliable response bit r generated under a specific physical quantity value, PQ$_i$, will become stable under another physical quantity value PQ$_j$—$i \neq j$. It is illustrated in Fig. 3, where the ROPUF is employed for description in this paper. The frequency of the RO has an almost linear relationship—coefficient—with its supply voltage. However, the coefficient varies from one RO to the other. For instance, in Fig. 3, the coefficient of RO$_1$ is higher than the coefficient of RO$_3$ because RO$_1$ oscillates faster than the RO$_3$ as the supply voltage increases. Here, the challenge bit c_3 selects a RO pair—RO$_1$ and RO$_3$—in order to produce a response bit r_3 according to the frequency comparison. When the response bit regenerated under the voltage just located at the crosspoint of f_1 and f_3—between V_2 and V_3, the response bit r_3 will be highly unstable due to the predominant impact from the noise.

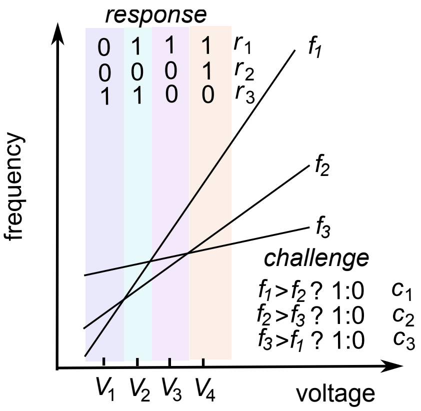

Figure 3. For these unreliable response bits in a ROPUF, it is not only dependent on the challenge, but also a function of the voltage. A challenge bit c selects a pair of ROs, and the response bit r is generated according to the comparison of frequencies of these two selected ROs.

However, if the voltage shifts to another point, the regeneration of r_3 becomes stable. For example, when r_3 is regenerated at the voltage V_1, it consistently results in '1'. Similarly, it will result in '0' when it is regenerated at the voltage of V_4.

In Fig. 3, the large frequency difference between two ROs ensures a reliable response bit. Overall, the crosspoint of two frequencies always induces an unstable response bit. Unreliable response bit becomes reliable once the voltage deviates far away from the crosspoint.

In general, the PUF sensor makes use of auxiliary physical effects to alter original challenge-response mapping relationship. More specifically, we take advantage of a new dimension of randomness which is induced from environmental conditions' change for remapping challenge response pairs. Technically, it is very similar to the concept of reconfigurable PUF where external effects are used for reconfiguring challenge-response characteristics [41], [42], [43]. In [43], Sharif *et al.* used multiple V_{DD}—supply voltage—to alter the challenge-response pair behaviors, which works as a reconfigurable PUF in order to increase the number of CRPs of a ROPUF, and the reconfigurable parameter is actually the voltage. The main difference between our work and [43] is that we reversely make use of the altered challenge-response behavior induced by the voltage change in order to recover the applied voltage for secure sensing, while the foregoing works do not consider such an application.

B. Sensing Through Unreliable Response Bits

Inspired by the foregoing observation, reliability of some response bits in a PUF strongly depend on the voltage given the same challenge. These response bits can be exploited to recover the voltage applied to the PUF. For instance, in Fig. 3, if the response $\mathbf{R}$ for the given challenge $\mathbf{C}$ is '001', then the voltage is derived as V_1. Then as well, if the response is '101', the voltage is V_2.

Employing such a sensing approach, advantages are:

(a) Removing the NVM need for storing digital secret keys and the subsequent cryptographic operations.

(b) The response sent from the PUF sensor contains no exact sensed voltage value. The user can recover the sensed value by observing the response. Also, the sensing scheme is secure as an adversary cannot spoof the user with a fraudulent voltage value.

(c) The adversary cannot send a false response to the user as the response for a given challenge is unpredictable. If the adversary does send a guessed response to the user, the user is able to reject this fraudulent response.

According to application scenarios, the PQ might be other environmental parameters rather than the electrical signal—eg. voltage. Note that other PQs, eg., temperature, light, sound, humidity can be sensed as long as the sensing parameters can be represented by an electrical signal that can influence the RO frequencies. Therefore, the fluctuation of these PQs can be converted into variations of electrical signal, and thereupon be recovered by the unreliable response bits.

The sensing based on the PUF unreliability is achieved with the help of the following authenticated sensing protocol.

C. Authenticated Sensing Protocol

The authenticated sensing protocol is performed as follows:

(step 1) In *enrollment phase*, the user prepares a PUF and measures a number of responses $\mathbf{R}_i^{\mathrm{PQ_j}}$ for the given challenge $\mathbf{C}_i$ under different PQ$_j$—different voltages, $j \in \{1, ..., p\}$. The user saves the measured CRPs in the database. Then the PUF sensor is installed in the intended (hostile) location for monitoring the PQ—eg. voltage or other PQs that can be metered by the voltage.

(step 2) In *authentication phase*, when collection of data from the PUF sensor is requested. The user randomly selects a challenge $\mathbf{C}$ and sends it to the PUF sensor. The PUF sensor is stimulated by the challenge $\mathbf{C}$ under PQ$_i$ that is the voltage the PUF sensor currently working on. Consequently, the $\mathbf{R}^{\mathrm{PQ_i}}$, $i \in \{1, ..., p\}$ is sent back to the user.

(step 3) The user compares each recorded response $\mathbf{R}^{\mathrm{PQ_j}}$, $j \in \{1, ..., p\}$—obtained under PQ$_j$ to the challenge $\mathbf{C}$—with the received response $\mathbf{R}^{\mathrm{PQ_i}}$. Only the response $\mathbf{R}^{\mathrm{PQ_j}}$, where $i = j$, stored in the database will match the received response $\mathbf{R}^{\mathrm{PQ_i}}$ given the same queried challenge $\mathbf{C}$. If the user finds that one of the saved response $\mathbf{R}^{\mathrm{PQ_j}}$ matches the received $\mathbf{R}^{\mathrm{PQ_i}}$. Then the sensed value of PQ$_i$ is discovered. Otherwise, this round of authenticated sensing is rejected.

IV. Experimental Validation

In this section, we use the public experimental data from five ROPUFs across five Spartan3E S500 FPGAs for validation of the aforementioned PUF sensor. Each FPGA consists of 512 ROs to form a ROPUF. Detailed implementation information can be found in [13]. As for the same challenge, the response is reproduced under 0.96 V, 1.08 V, 1.20 V, 1.32 V, 1.44 V respectively, while the temperature is 25°C. Each response $\mathbf{R}^{\mathrm{PQ}}$ is re-evaluated 100 times.

Before delving into comprehensive analyses of the practicability of this authenticated sensing protocol, we first provide some preliminaries that will ease following descriptions.

A. Preliminaries

Definition 1. InterPQ-distance. The interPQ-distance is a random variable describing the distance between two PUF responses $\mathbf{R}^{\mathrm{PQ}_1}, \mathbf{R}^{\mathrm{PQ}_2}$ produced under different PQs by applying the same challenge to the same PUF sensor, hence,

$$D_{\mathrm{interPQ}} = \mathrm{dist}(\mathbf{R}^{\mathrm{PQ}_1}, \mathbf{R}^{\mathrm{PQ}_2}) \tag{1}$$

where $\mathbf{R}^{\mathrm{PQ}_1}, \mathbf{R}^{\mathrm{PQ}_2}$ are two responses generated under two random and distinct PQs by applying the same challenge to the same PUF sensor.

Definition 2. IntraPQ-distance. The intraPQ-distance is a random variable describing the distance between two PUF responses $\mathbf{R}^{\mathrm{PQ}}, \mathbf{R}^{\mathrm{PQ}'}$ from the same PUF sensor and using the same challenge under the same PQ setting.

$$D_{\mathrm{intraPQ}} = \mathrm{dist}(\mathbf{R}^{\mathrm{PQ}}, \mathbf{R}^{\mathrm{PQ}'}) \tag{2}$$

where $\mathbf{R}^{\mathrm{PQ}}, \mathbf{R}^{\mathrm{PQ}'}$ are two randomly re-evaluated responses from a randomly chosen PUF sensor by using the same challenge under the same PQ setting.

The dist(.;.) can be any well-defined and appropriate distance metric over the responses. In this paper, responses are always bit vectors and the used distance metric is Hamming distance (HD) or fractional Hamming distance that are formally defined as:

Definition 3. Hamming distance. For bit vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ with the same length l, the HD between them is defined as:

$$f_{\mathrm{HD}}(\mathbf{X}_1, \mathbf{X}_2) = \sum_{i=1}^{l} \mathbf{X}_1 \oplus \mathbf{X}_2. \tag{3}$$

Definition 4. Fractional Hamming distance. Built upon Eq. (3), the fractional Hamming distance (FHD) is defined as:

$$f_{\mathrm{FHD}}(\mathbf{X}_1, \mathbf{X}_2) = \frac{f_{\mathrm{HD}}(\mathbf{X}_1, \mathbf{X}_2)}{l}. \tag{4}$$

Readers who are familiar with PUFs will notice that the definition of the interPQ-distance is similar to the inter-distance of PUFs that measures the difference between two responses from two distinct PUF instances given the same challenge. The difference is that the InterPQ-distance is evaluated across differing PQ values, still referred to the same PUF instance, but the inter-distance is evaluated across different PUF instances. Whereas the IntraPQ-distance is similar to the intra-distance of PUF responses that measures the difference between two responses reproduced by two random and distinct evaluations by applying the same challenge to the same random chosen PUF instance. The main difference is that the intra-distance does not care the source of PQs, it treats any PQ as noise source. However, we only treat the unwanted PQs—in the example, temperature is noise source but voltage is not—as noise source. Similar to the inter-distance and intra-distance distribution [30], both of the interPQ-distance and intra-distance can be assumed following the binomial distribution $B(n, p)$. The binomial probability estimator of interPQ-distance and intra-distance distributions are $\hat{p}_{\mathrm{interPQ}}$ and $\hat{p}_{\mathrm{intraPQ}}$, respectively. Similar to [30], the $\hat{p}_{\mathrm{interPQ}}$, in general, is the probability of $\mathbf{R}^{\mathrm{PQ}_1} \neq \mathbf{R}^{\mathrm{PQ}_2}$, see definition 1, the $\hat{p}_{\mathrm{intraPQ}}$ is the probability of $\mathbf{R}^{\mathrm{PQ}} \neq \mathbf{R}^{\mathrm{PQ}'}$, see definition 2.

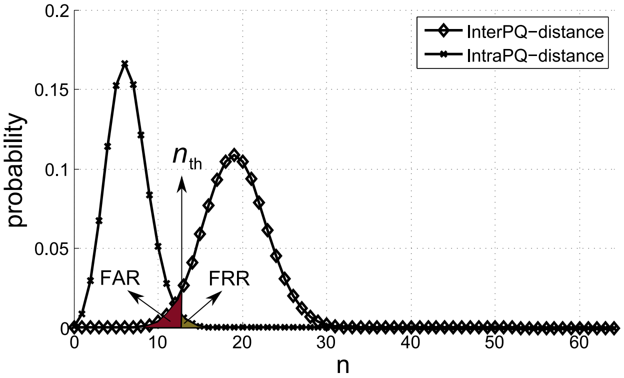

Figure 4. Distribution of interPQ-distance and intraPQ-distance for 64-bit response.

In step 3, the successful authenticated sensing relies on the fact that the intraPQ-distance is less than the interPQ-distance as visualized in Fig. 4. For example, in Fig. 3, responses under V_1, V_2, V_3, V_4 are divergent to the same challenge referring to the interPQ-distance. In contrast, the response is relatively stable when it is reproduced under the same $V_j,\ j \in \{1, 2, 3, 4\}$ referring to the intraPQ-distance. The recorded response $\mathbf{R}^{\mathrm{PQ}}$ matches the received response $\mathbf{R}^{\mathrm{PQ_j}}$ only when they are generated under the same PQ$_j$ for the same challenge queried by the same PUF sensor.

B. Unreliable Response Bit Selection

If the ROs oscillation frequencies of f_i and f_j do not intersect within a specific range, specifically, between 0.96 V to 1.44 V. Then the regeneration of response bits upon frequency comparison is always consistent and shows strong tolerance to voltage deviations. In such cases, these response bits cannot be used to sense the voltage. Because they cannot reflect voltage changes. One task is to find out the unreliable response bit based on the frequency difference Δf among ROs. If this difference is small among different ROs, response bits generated upon them will flip with high probability when the voltage changes. This is the foundation of our proposed PUF sensor. Fig. 5 presents the plot of the frequency distribution under 1.20 V, which is the nominal/reference voltage. The mean value is 197.8 MHz. We select ROs satisfying $|f - 197.8\ \mathrm{MHz}| < \Delta f$ for response generation. It is clear that the number of ROs selected is related to the setting of Δf. The number will increase as Δf becomes larger.

The reason for selecting unreliable response bits under 1.20 V is that it is the central of different voltage settings. It is desirable to increase the difference between $\hat{p}_{\mathrm{interPQ}}$—PQ is voltage in this specific experimental validation—and $\hat{p}_{\mathrm{intraPQ}}$. the larger the separation the easier the recovery of the PQ information—will be detailed and quantified in

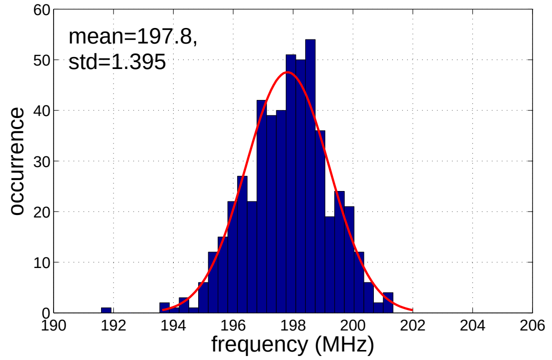

Figure 5. Frequency distribution of 512 ROs in one ROPUF.

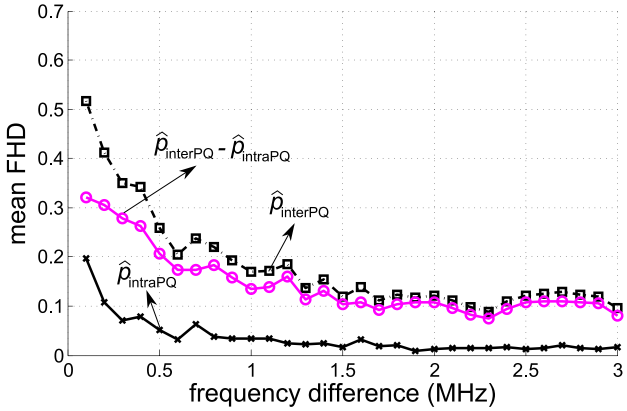

Figure 6. The $\hat{p}_{\text{interPQ}}$ and $\hat{p}_{\text{intraPQ}}$ performance for one ROPUF to different Δf—frequency difference—settings. Unreliable response bits selection is performed under the reference voltage of 1.20 V. The $\hat{p}_{\text{intraPQ}}$ is evaluated under 1.20 V

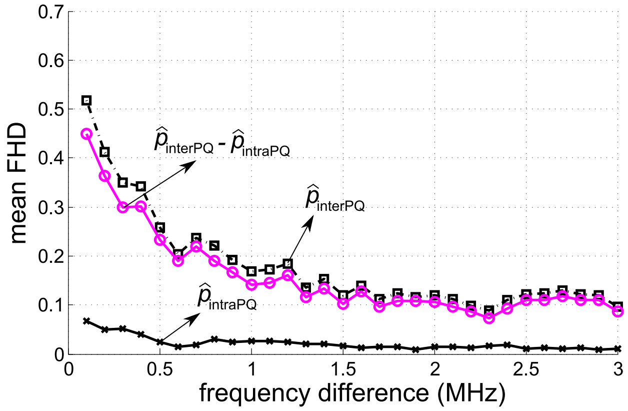

Figure 7. The $\hat{p}_{\text{interPQ}}$ and $\hat{p}_{\text{intraPQ}}$ performance for one ROPUF to different Δf—frequency difference—settings. Unreliable response bits selection is performed under the reference voltage of 1.20 V. The $\hat{p}_{\text{intraPQ}}$ is evaluated under 1.32 V.

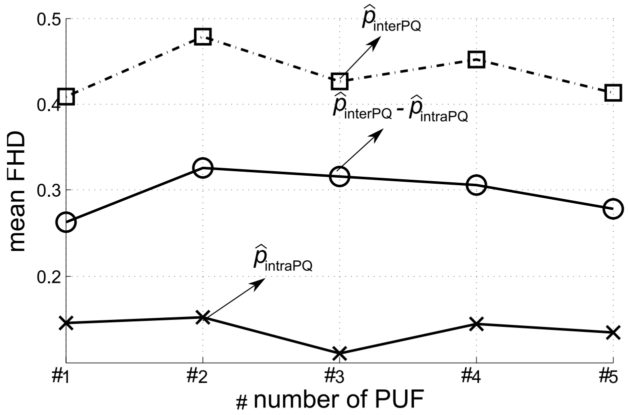

Figure 8. The $\hat{p}_{\text{interPQ}}$ and $\hat{p}_{\text{intraPQ}}$ performance across five ROPUFs. Unreliable response bits selection is performed under the reference voltage of 1.20 V. The $\hat{p}_{\text{intraPQ}}$ is evaluated under 1.32 V. The Δf is set to be 0.3 MHz.

Section IV-C. The relationship between the difference of $\hat{p}_{\text{interPQ}}$ and $\hat{p}_{\text{intraPQ}}$ and the setting of Δf is shown in Fig. 6. We can see that the difference is significantly increased from less than 10% to more than 30% when the Δf shrinks. As a consequence, the bit length of the response to successfully perform authenticated sensing compared with [10] will be significantly shorten—analyses of the required bit length of response for sensing voltage will be presented in Section IV-C.

Due to unreliable response bits being selected under the reference voltage of 1.20 V, the $\hat{p}_{\text{intraPQ}}$ under 1.20 V rapidly rises when Δf shrink, because the responses are more prone to noise, especially when the Δf is very small (less than 0.3 MHz, see Fig. 6). As a comparison, Fig. 7 shows the $\hat{p}_{\text{intraPQ}}$ under 1.32 V, where unreliable response bits are still selected based on the reference voltage of 1.20 V. The $\hat{p}_{\text{intraPQ}}$ is lower compared to Fig. 6 due to the selected unreliable response tending to tolerate more noise. Because unreliable response bits are turning into reliable when the voltage moves from 1.20 V to 1.32 V. Fig. 8 shows the $\hat{p}_{\text{intraPQ}}$ and $\hat{p}_{\text{interPQ}}$ of five different ROPUFs across five FPGA boards. We can see that the difference between $\hat{p}_{\text{intraPQ}}$ and $\hat{p}_{\text{interPQ}}$ is large enough to distinguish V_i, where $V_i \in \{0.96\text{ V}, 0.1.08\text{ V}, 1.20\text{ V}, 1.32\text{ V}, 1.44\text{ V}\}$, from the rest.

Note in practice, the user and the adversary are assumed to have different security access levels to the ROPUF. Specifically, in the enrollment phase, the user is able to obtain the RO frequency directly from the counter. This direct access is disabled/destroyed once the enrollment phase is completed. The adversary needs to decompose the layer for probing if the original RO frequency is attempted to be measured. But this operation is very likely to change or destroy the PUF behavior.

C. Sensing Capability

Clearly one single CRP is not able to correctly recover the specific PQ in the field. We need use multiple response bits or a number of CRPs to minimize the error for both mistakenly accepting a response when it is generated under a false PQ, referred as false acceptance rate (FAR), and falsely reject the authentic response when it is from a genuine PQ, referred as false rejection rate (FRR). It is imperative to ensure both FAR and FRR to be minimized for meeting requirements in practice when the PUF sensor employed in field to reversely recover the PQ based on the authenticated sensing protocol. More generally, the FAR stands for the probability of a user incorrectly taking PQ$_i$ instead of PQ$_j$, $i \neq j$. While the FRR stands for the probability of the authentic PQ$_i$ is falsely rejected.

These two undesirable errors are illustrated in Fig. 4. The right tail of the intraPQ-distance distribution indicates the FRR, while the left tail of the interPQ-distance implies the FAR. When the length of response bit or number of CRPs, n, and a threshold n_{th} used for authenticated sensing are given, and considering both interPQ-distance and intraPQ-distance following the binomial distribution, then the FAR and FRR can be formally expressed as [44], [30]:

$$\text{FRR} = 1 - \sum_{i=0}^{n_{\text{th}}} \binom{n}{i} (\hat{p}_{\text{intraPQ}})^{i} (1-\hat{p}_{\text{intraPQ}})^{(n-i)}, \quad (5)$$

$$\text{FAR} = \sum_{i=0}^{n_{\text{th}}} \binom{n}{i} (\hat{p}_{\text{interPQ}})^{i} (1-\hat{p}_{\text{interPQ}})^{(n-i)}. \quad (6)$$

From the security and practicability perspectives, the FAR expresses the security of an authenticated sensing, because a high FAR indicates a high risk of incorrectly accepting an false PQ, which could cause a security issue. The FRR expresses the robustness or usability of the authenticated sensing and indicates a misrejection of an authentic PQ.

Based on Eq (5) and (6), we can see that the FRR and FAR depend on the $\hat{p}_{\text{intraPQ}}$ and $\hat{p}_{\text{interPQ}}$, the threshold n_{th}, and the number of employed CRPs n. Suppose the n is fixed to be 64 as an example that is shown in 4, a large n_{th} benefits false rejection rate but aggravates false acceptance rate, and vice versa for a small n_{th}. We want to balance them in practice. There exists a threshold value to make both FAR and FRR equal. We refer this interested threshold value, n_{th}, as *equal error threshold* and termed as n_{EER}. Consequentially, when both error rates are equal, we refer this equal rate as *equal error rate* (EER) following Roel's work [30]. For a discrete distribution, FAR and FRR may not exactly equal for a discrete threshold n_{EER}, and in that case, n_{EER} and EER are defined as in [30]:

$$n_{\text{EER}} = \underset{n_{\text{th}}}{\operatorname{argmin}}\{\max\{\text{FAR}(n_{\text{th}}), \text{FRR}(n_{\text{th}})\}\}, \quad (7)$$

$$\text{EER} = \max\{\text{FAR}(n_{\text{EER}}), \text{FRR}(n_{\text{EER}})\}. \quad (8)$$

Given PUF sensors with binomial probability estimator $\hat{p}_{\text{interPQ}}$ and $\hat{p}_{\text{intraPQ}}$, the task is to find minimal number of CRPs, n, for ensuring an acceptable EER that meet desired requirements, eg., a value lower than 10^{-6}. In Table. I, we give quantitative evaluations of n—minimal bit length of the response to meet the EER, and n_{th} of PUF sensors under different $\hat{p}_{\text{interPQ}}$ and $\hat{p}_{\text{intraPQ}}$, both $\hat{p}_{\text{interPQ}}$ and $\hat{p}_{\text{intraPQ}}$ are influenced by Δf as shown in Fig. 6. The used PQ in this table is voltage.

As can be seen from Table. I, necessary bit length of n is decreasing as the Δf is shrinking. This indicates the efficiency of the authenticated sensing and the need to implement the proposed unreliable response bits selection.

V. Discussions

Although physical cloning of a PUF is impossible, it is possible to generate a mathematical model/copy. This mathematical copy is usually achieved by modeling attacks. In this section, we firstly analyze the security of the PUF sensor mainly from the modeling attacks perspective by the fact that this is the plausible attack on PUFs. Next we discuss mitigation of influence from the unwanted PQ, eg., temperature.

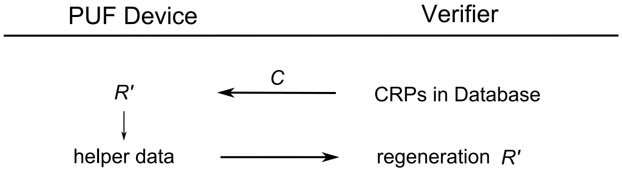

Figure 9. Reverse fuzzy extractor [47].

A. Replaying Attacks

In the authenticated sensing protocol, CRPs are exposed directly without protection and communicated between the PUF sensor and the user. An adversary may eavesdrop the CRPs and therefore exploit them for replaying attacks.

The limitation can be circumvented by never using a CRP more than once. Although the basic ROPUF structure [3] is only capable of producing limited number of CRPs that seems not applicable, the following approaches eliminate this limitation [19], [45], [46].

B. Modeling Attacks

Modeling attacks pose a major threat to the security of current PUF structures, especially those PUFs are able to generate large number of CRPs [8], [44], [48]. Here we state that although weak PUFs, eg., SRAM PUFs, are always claimed to be inherently resilient to modeling attacks, it does not imply that they are secure. Because CRPs can be exhaustively read out within a very short period once the physical access to them is possible. Therefore, the response is not directly exposed to an adversary, instead the response is usually hashed after an error correction [3].

An adversary can collect CRPs by two means: eavesdropping and physical measurement. Then the adversary may implement modeling attacks by employing powerful machine learning tools using the obtained CRPs as a training dataset. The security of the PUF is compromised if the prediction accuracy of the model is higher than the reliability of the PUF. In other words, a highly accurate model is able to fraud the user/verifier by imitating the CRP behavior of the original physical PUF during the authenticated phase.

1) Solution 1: Strong PUF: The first straightforward solution would be to employ a strong PUF, which is i) able to generate a large number of CRPs that hinder the PUF being fully characterized within a short time, eg., several days, months. and ii) shows resilience to modeling attacks, where an accurate model is hard to be obtained even by polynomially increasing the size of CRPs for training. An improved ROPUF design [45] and XOR8-APUF [8] can be exploited to stepside the modeling attacks. Notably, XOR-ing of multiple PUF response bits to gain a 1-bit response sacrifices reliability while increasing the complexity of modeling attacks. In contrast, in our authenticated sensing approach we benefit from such

Table I

QUANTITATIVE EVALUATION OF NECESSARY BIT LENGTH OF THE RESPONSE FOR AUTHENTICATED SENSING UNDER DIFFERENT $\hat{p}_{\text{interPQ}}$ AND $\hat{p}_{\text{intraPQ}}$ THAT ARE DETERMINED BY Δf.

			EER $< 10^{-2}$				EER $< 10^{-4}$				EER $< 10^{-6}$			
Δf MHz	$\hat{p}_{\text{intraPQ}}$	$\hat{p}_{\text{interPQ}}$	n	n_{EER}	FAR*	FRR*	n	n_{EER}	FAR*	FRR*	n	n_{EER}	FAR*	FRR*
3	1.62%	9.68%	146	7	−2.00	−2.02	383	18	−4.00	−4.21	623	29	−6.00	−6.27
2	1.34%	12.04%	93	5	−2.01	−2.12	235	12	−4.01	−4.14	380	19	−6.03	−6.04
1	3.48%	16.88%	98	9	−2.02	−2.12	247	22	−4.04	−4.16	397	35	−6.03	−6.10
0.5	5.21%	25.80%	63	9	−2.04	−2.28	148	20	−4.05	−4.03	244	33	−6.01	−6.21
0.3	7.16%	31.00%	41	8	−2.02	−2.11	106	20	−4.08	−4.23	167	31	−6.04	−6.02

Note: the * symbol indicates $\log_{10}(\cdot)$ of the value.

reliability deterioration, and take advantage of the response sensitivity to environmental changes.

2) Solution 2: Reverse Fuzzy Extractors: In PUF-based key generation applications, the *fuzzy extractor* is employed [49]. The fuzzy extractor consists of two parts: *secure sketch* and *randomness extractor*. Secure sketch eliminates noise from the collected noisy data. In other words, it maps the similar regenerated response into the same value. Randomness extractor guarantees the uniform distribution/randomness of derived keys based on the corrected response.

There are two steps involved into the secure sketch. The first step is to generate the *helper data* that is computed from PUF response $\mathbf{R}$ during helper data generation phase. In the second step, the helper data is employed to recover the original response $\mathbf{R}$ from the afterward regenerated response $\mathbf{R}'$, where the difference between the original response $\mathbf{R}$ and reproduced response $\mathbf{R}'$ is small. Usually, the helper data generation is performed in the user/verifier side, while the reconstruction of the original $\mathbf{R}$ is implemented in the PUF integrated device.

Note that implementing a decoding algorithm in the PUF integrated device to recover the original $\mathbf{R}$ results in higher area and power overhead, reverse fuzzy extractor is proposed [47] to overcome this issue. Fig. 9 illustrates the simplified structure of the reverse fuzzy extractor. The reverse fuzzy extractor moves the costly decoding computation to the potent verifier, while only leaves the lightweight helper data generation to the PUF integrated device. It can be seen, the PUF integrated device generates a new helper data based on the reproduced response $\mathbf{R}'$ whenever the authenticated of the PUF device is requested. The verifier carries out the computationally intensive decoding to recover the reproduced response $\mathbf{R}'$ based on the recorded response $\mathbf{R}$ and the helper data from the PUF device, then performs the authenticated.

The reverse fuzzy extractor enables unlimited usage of the same CRP without worrying about replaying attacks, and it is also secure despite the possibility of physical access to the PUF and the eavesdropping occurrence. This is because the regenerated response $\mathbf{R}'$ is invisible to the adversary—only helper data is observable. In addition, each regeneration of response $\mathbf{R}'$ gives rise to a new helper data. Even multiple executions of helper data generation leak some information, but the reverse fuzzy extractor based on the syndrome construction [50] is able to ensure a certain amount of min-entropy in the PUF response.

Therefore, the lightweight reverse fuzzy extractor can be employed to countermeasure modeling attacks as an alternative to the strong PUF solution if this is not preferred. Moreover, the reverse fuzzy extractor does not need PUFs to produce a large number of CRPs. Therefore, PUFs limited by the number of CRPs, eg. SRAM PUF, basic ROPUF [3], can also be deployed as a kernel for the PUF sensor.

C. Mitigation of Influence from the Unwanted PQ

As stated in the definition of PUF sensor in Section II-B, the response bit of PUF sensor should be stable when the uninterested PQ varies. In our study, the PQ of interest is voltage, while the reliability degradation originated from temperature is unwanted. In other words, the response bit is preferred to be stable within a wide range of temperature variations.

This concern can be easily addressed by harnessing the negative temperature coefficient of current starved inverters to compensate the positive temperature coefficient of regular inverters in order to prevent response bits flipping due to temperature fluctuations [46]. The current starved inverters and regular inverters can be combined to construct a RO. Such circuits can be designed to ensure the coefficient between the temperature and the frequency of the RO is invariant of the temperature fluctuations. Based on the experimental data given in [46], the reliability of ROPUF is nearly 100% when the temperature changes from $-20°\text{C}$ to $120°\text{C}$—the reference temperature is $27°\text{C}$ and the voltage is fixed at 1.2 V.

VI. CONCLUSION

In this paper, we proposed to treat the PUF itself as a sensor by turning its undesirable unreliability into an asset. The PUF sensor secures communication of sensed information by performing sensing and cryptography in an inseparable manner. We provided an authenticated sensing protocol applicable to PUF sensors. In this protocol, the unclonablility and unpredictability of responses of the PUF impede measurement spoofing. Moreover, we proposed a method of selecting unreliable response bits to expedite the enrollment phase and also greatly cut down the necessary bit length of the response during authenticated sensing phase. The quantitative analyses of bit length of response is carried out based on experimental data. The practicability and security analyses validate the

PUF sensor as a promising lightweight alternative for secure wireless sensing. Future work includes increasing the response sensitivity to a certain PQ to increase sensing precision and also a generic way to mitigate the unwanted PQs, eg., not only the temperature but also the aging influence.

References

[1] R. Torrance and D. James, "The state-of-the-art in IC reverse engineering," in *Cryptographic Hardware and Embedded Systems-CHES*. Springer, 2009, pp. 363–381.

[2] B. Gassend, D. Lim, D. Clarke, M. Van Dijk, and S. Devadas, "Identification and authentication of integrated circuits," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 11, pp. 1077–1098, 2004.

[3] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th Annual Design Automation Conference*. ACM, 2007, pp. 9–14.

[4] R. Maes, A. Van Herrewege, and I. Verbauwhede, "PUFKY: A fully functional PUF-based cryptographic key generator," in *Cryptographic Hardware and Embedded Systems*–CHES. Springer, 2012, pp. 302–319.

[5] 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*. IEEE, 2016, DOI: 10.1109/PERCOMW.2016.7457162.

[6] D. Lim, "Extracting secret keys from integrated circuits," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

[7] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 2010, pp. 237–249.

[8] U. Ruhrmair, J. Solter, 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 Forensics and Security*, vol. 8, no. 11, pp. 1876–1891, 2013.

[9] K. Rosenfeld, E. Gavas, and R. Karri, "Sensor physical unclonable functions," in *Proc. IEEE. Int. Symp. Hardware Oriented Hardware-Oriented Security and Trust (HOST)*, 2010, pp. 112–117.

[10] U. Rührmair, J. Martinez-Hurtado, X. Xu, C. Kraeh, C. Hilgers, D. Kononchuk, J. J. Finley, and W. P. Burleson, "Virtual proofs of reality and their physical implementation," in *36th IEEE Symposium on Security and Privacy*, 2015, pp. 70–85.

[11] B. Fisch, D. Freund, and M. Naor, "Physical zero-knowledge proofs of physical properties," in *Advances in Cryptology–CRYPTO*. Springer, 2014, pp. 313–336.

[12] R. Maes, V. Rožić, I. Verbauwhede, P. Koeberl, E. Van der Sluis, and V. Van der Leest, "Experimental evaluation of physically unclonable functions in 65 nm CMOS," in *Proceedings of the ESSCIRC*. IEEE, 2012, pp. 486–489.

[13] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of RO-PUF," in *IEEE International Symposium on Hardware-Oriented Security and Trust (*HOST),, 2010, pp. 94–99.

[14] A. Maiti and P. Schaumont, "Improved ring oscillator PUF: an FPGA-friendly secure primitive," *Journal of Cryptology*, vol. 24, no. 2, pp. 375–397, 2011.

[15] P. S. Ravikanth, "Physical one-way functions," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[16] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[17] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002, pp. 148–160.

[18] S. S. Zalivaka, A. V. Puchkov, V. P. Klybik, A. A. Ivaniuk, and C.-H. Chang, "Multi-valued arbiters for quality enhancement of PUF responses on FPGA implementation," in *21st IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 533–538.

[19] M. Gao, K. Lai, and G. Qu, "A highly flexible ring oscillator PUF," in *51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2014, DOI:10.1145/2593069.2593072.

[20] J.-L. Zhang, G. Qu, Y.-Q. Lv, and Q. Zhou, "A survey on silicon PUFs and recent advances in ring oscillator PUFs," *Journal of Computer Science and Technology*, vol. 29, no. 4, pp. 664–678, 2014.

[21] D. E. Holcomb, W. P. Burleson, and K. Fu, "Initial SRAM state as a fingerprint and source of true random numbers for RFID tags," in *Proceedings of the Conference on RFID Security*, 2007.

[22] Holcomb, Daniel E, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1198–1210, 2009.

[23] 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, 2015.

[24] Y. Su, J. Holleman, and B. P. Otis, "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.

[25] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic PUFs from flip-flops on reconfigurable devices," in *3rd Benelux Workshop on Information and System Security (*WISSec), vol. 17, 2008.

[26] V. van der Leest, G.-J. Schrijen, H. Handschuh, and P. Tuyls, "Hardware intrinsic security from D flip-flops," in Proceedings of the 5th ACM Workshop on Scalable Trusted Computing, 2010, pp. 53–62.

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

[28] M. Majzoobi, G. Ghiaasi, F. Koushanfar, and S. R. Nassif, "Ultra-low power current-based PUF," in *Proc. IEEE Int. Sym on Circuits and Systems* (ISCAS), 2011, pp. 2071–2074.

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

[30] M. Roel, "Physically unclonable functions: Constructions, properties and applications," Ph.D. dissertation, University of KU Leuven, 2012.

[31] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of IEEE*, vol. 102, pp. 1126–1141, 2014.

[32] L. Zhang, Z. H. Kong, C.-H. Chang, A. Cabrini, and G. Torelli, "Exploiting process variations and programming sensitivity of phase change memory for reconfigurable physical unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 921–932, 2014.

[33] Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "mrPUF: A novel memristive device based physical unclonable function," in *Applied Cryptography and Network Security*. Springer, 2015, pp. 595–615.

[34] Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "Memristive crypto primitive for building highly secure physical unclonable functions," *Scientific Reports*, vol. 5, art. no. 12785, 2015.

[35] L. Zhang, X. Fong, C.-H. Chang, Z. H. Kong, and K. Roy, "Highly reliable spin-transfer torque magnetic RAM based physical unclonable function with multi-response-bits per cell," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1630–1642, 2015.

[36] J. Rajendran, R. Karri, J. B. Wendt, M. Potkonjak, N. McDonald, G. S. Rose, and B. Wysocki, "Nano meets security: Exploring nanoelectronic devices for security applications," *Proceedings of the IEEE*, vol. 103, no. 5, pp. 829–849, 2015.

[37] 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.

[38] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," in *Embedded Systems Design with FPGAs*. Springer, 2013, pp. 245–267.

[39] R. Maes, "An accurate probabilistic reliability model for silicon PUFs," in *Cryptographic Hardware and Embedded Systems-CHES 2013*. Springer, 2013, pp. 73–89.

[40] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper data algorithms for PUF-based key generation: Overview and analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 889–902, 2015.

[41] K. Kursawe, A. Sadeghi, D. Schellekens, B. Skoric, and P. Tuyls, "Reconfigurable physical unclonable functions-enabling technology for tamper-resistant storage," in *IEEE International Workshop on Hardware-Oriented Security and Trust*, 2009, pp. 22–29.

[42] S. Katzenbeisser, Ü. Kocabaş, V. Van Der Leest, A.-R. Sadeghi, G.-J. Schrijen, and C. Wachsmann, "Recyclable PUFs: Logically reconfigurable PUFs," *Journal of Cryptographic Engineering*, vol. 1, no. 3, pp. 177–186, 2011.

[43] S. Sharif Mansouri and E. Dubrova, "Ring oscillator physical unclonable function with multi level supply voltages," in *30th International Conference on Computer Design*. IEEE, 2012, pp. 520–521.

[44] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van 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.

[45] A. Maiti, I. Kim, and P. Schaumont, “A robust physical unclonable function with enhanced challenge-response set,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 333–345, 2012.

[46] Y. Cao, L. Zhang, C.-H. Chang, and S. Chen, “A low-power hybrid RO PUF with improved thermal stability for lightweight applications,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 1143–1147, 2015.

[47] A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, “Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs,” in *Financial Cryptography and Data Security*. Springer, 2012, pp. 374–389.

[48] U. Rührmair, C. Jaeger, C. Hilgers, M. Algasinger, G. Csaba, and M. Stutzmann, “Security applications of diodes with unique current-voltage characteristics,” in *Financial Cryptography and Data Security*. Springer, 2010, pp. 328–335.

[49] Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” in *Advances in Cryptology-Eurocrypt 2004*. Springer, 2004, pp. 523–540.

[50] X. Boyen, “Reusable cryptographic fuzzy extractors,” in *Proceedings of the 11th ACM Conference on Computer and Communications Security*. ACM, 2004, pp. 82–91.