New Configurable Galois/Inverter Ring Oscillator (GIRO) Physically Unclonable Functions: Design, Analysis and Authentication Algorithms

Fayez Gebali
Department of Electrical and Computer Engineering,
University of Victoria, Victoria BC, Canada.
fayez@uvic.ca

Abstract: Ring oscillator physically unclonable function (RO-PUF) is a strong PUF that is simple to operate and simple to fabricate for authentication and secure key exchange for IoT edge devices. However, the unique device identity provided by RO-PUFs suffer from inevitable noise which leads to the conflicting requirements of using a large num- ber of bits to establish a unique identity and reducing the number of erroneous bits. In this work we develop a statistical model for RO-PUF and identify the main parameters affecting the performance. We also propose three CRP selection algorithms for selecting the challenge to control the number of response bits in error. Numerical results indicate that an RO-PUF must use word sizes of at least 32 bits and reducing or eliminating the number of bits in error can be established through the use of the proposed algorithms.

Keywords: Physically unclonable function, Ring oscillator, Fibonacci ring oscillator, Galois ring oscillator, PUF modeling, Authentication algorithm.

مذبذب حلقة جالوا / العاكس الجديد القابل للتكوين (GIRO) وظائف غير قابلة للاستنساخ فعليًا: خوارزميات التصميم والتحليل والمصادقة

الملخص: تم وظيفة مذبذب الحلقة غير القابلة للنسخ جسديًا (RO-PUF) عبارة عن PUF قوي سهل التشغيل وبسيط في التصنيع للمصادقة وتبادل المفاتيح الآمن لأجهزة حافة إنترنت الأشياء IoT edge. ومع ذلك، فإن هوية الجهاز الفريدة التي توفرها RO-PUFs تعاني من ضوضاء لا مفر منها مما يؤدي إلى المتطلبات المتضاربة لاستخدام عدد كبير من البتات لإنشاء هوية فريدة وتقليل عدد البتات الخاطئة. في هذا العمل قمنا بتطوير نموذج إحصائي لـ RO-PUF وتحديد المعلمات الرئيسية التي تؤثر على الأداء. نقترح أيضًا ثلاث خوارزميات اختيار CRP لاختيار التحدي للتحكم في عدد بتات الاستجابة الخاطئة. تشير النتائج العددية إلى أن RO-PUF يجب أن يستخدم أحجام كلمات لا تقل عن ٣٢ بتًا ويمكن تحديد أو تقليل عدد البتات الخاطئة من خلال استخدام الخوارزميات المقترحة.

1.Introduction

Internet of things (IoT) devices are becoming an essential part in many applications especially in telehealth in the current situation of pandemic crisis. Securing this critical infrastructure is essential and relies on authentication and secure key exchange. These two goals are satisfied through silicon physically unclonable functions (PUF) that add unique, unclonable identities to IoT devices. This is equivalent to biometrics in humans such as iris, retina, voice, facial or fingerprint. PUFs not only help to authenticate IoT devices, but also aid in storing secret keys in the way a PUF is constructed. Instead of storing secret keys using nonvolatile memories (NVM), PUFs use their structure to build the secret key on demand. An attacker can not gain access to the secret key since any attempt at reverse engineering disturbs the PUF and irrevocably alter the secret key. Thereby rendering the IoT device useless.

Authentication and key exchange using PUF is based on a challenge-response pair (CRP) where a set of challenges and their associated unique response is established by the device manufacturer. There are several criteria for CRP construction:

1) Several CRP must be generated and a single CRP should be used once only to prevent forging a valid response by observing past CRP activities.
2) Choosing the number of bits of the IoT response to a challenge must be “sufficient” to be able to distinguish between genuine and fake IoT devices. At the same time the number of response bits in error must be kept ”small enough” to correct but not to accept and correct responses from fake devices.
3) Algorithms must be provided to extract a high-entropy stable session secret keys from noisy low-entropy response.

1.1 Main Contributions:

The main contributions of this work are summarized as follows:

1) A statistical model for the operation of RO PUF is developed taking into account random process variations (RPV) and the three sources of noice in CMOS technology.
2) Algorithms for measuring the statistical parameters of the RO PUF by the IoT device manufacturer are proposed.
3) A new Galois RO PUF is proposed to exponentially increase the CRP available space.
4) Three RO PUF based authentication algorithms are proposed. One of the proposed algorithms uses the statistical distribution of the oscillators to ensure that the device response is noise-free without using the helper data algorithm.

1.2 Organization

The organization of this paper is as follows. Section II provides a review of works related to PUFs in general and ring oscillators (RO) in particular. Section III discusses the main issues related to multifactor and context- ware authentication. Section IV provides a discussion on the construction of RO-based PUFs and how the response bits are constructed. Section V show the derivation of a RO statistical model and the sources of long-term and dynamic noise. The main parameters of the model are also identified. Section VI describes how the main statistical parameters of the RO system are experimentally obtained by the device manufacturer and how the golden response is obtained for sharing with a registration authority (RA). Section VII reviews the scheme for mutual authentication and secure key exchange using PUFs.

Section VIII summarizes the main attacks on hardware and specifically IoT devices. Section IX proposes a new Galois ring oscillator structure that results in a large set of possible challenge response pairs. Section X shows three algorithms for device authentication and secure key exchange. Algorithm #1 is the standard algorithms often discussed in the literature and its limitations and vulnerabilities are highlighted. Two new algorithms are proposed viz. Algorithm #2 and Algorithm #3. The latter manages to eliminate the noise in the response and obviates the need to use the helper data algorithm. Thus machine learning attack is countered.

2. Related Work

Physically unclonable functions (PUFs) were first discussed in the innovative works of Gassend et al. [1], [2] introduced the concept of silicon PUF as a means to identify and authenticate individual integrated circuits (ICs). Later McGrath et al. [3] provided a taxonomy of PUFs and reviewed several important PUF concepts such as weak and strong PUFs, implicit and explicit randomness, intrinsic and extrinsic evaluation.

Suh and Devadas [4] proposed building eight RO PUFs and selecting the fastest and slowest among them to generate one bit of the response. This complexity hindered the adoption of RO PUFs for practical use for authentication.

Like any new innovation in security, attacks emerged to utilize the PUF in hardware. Bou-Harb and Neshenko [5] authored a book about IoT vulnerabilities and how to provide remediations for such IoT attacks through situation awareness, generating and sharing IoT-centric cyber threat intelligence.

A very useful resource for authentication and key establishment was provided by Boyed et al. [?]. In this book, the authors discuss protocol architectures for session key generation, authentication and data integrity. Adversary capabilities and types of attacks are also discussed. A discussion is provided about not only identity-based protocols but also to pairing-based protocols. A very timely discussion is also provided on group key establishment, which is applicable to inter communication between IoT devices. The main factors impacting group key establishment were identified as application type, group size and dynamics, scalability, and finally, trust model. All these factors apply very well to telehealth applications as a crucial application for IoT technology.

Babaei and Schiele [6] provided an excellent review of the state of the art and challenges for using PUF in securing IoT devices. The challenges for using IoT devices are being lightweight with limited processing and energy capabilities. Also, performing remote updates over unprotected ports opens the gates for various types of attacks. The security issues therefore revolve around authentication and secret keys. The review also included design considerations for PUF implementation in ASIC and FPGA technologies.

Protocols to establish authentication and secure session keys were first proposed by Delvaux et al. [7]–[9], Dodis et al. [10], [11] and Maes et al. [12]–[14]. Further discussion of secure key exchange is discussed in Section VII.

References [15]–[17] addressed the high complexity of RO PUF through proposed a configurable design having a multiplexer in each delay stage. The multiplexers' selection bits are used to obtain a richer set of responses from the system.

A low-cost configurable RO PUF was implemented in FPGA in [18]. The configurable RO PUF was meant to circumvent the limited use of these PUFs due to their high cost of implementation. Obtaining a large number of bit responses is accomplished using only an inverter and a multiplexer in each delay stage.

The authors in [19] constructed an RO PUF using Galois ring oscillators that were originally proposed in [20]. The inverter in each delay stage is now replaced with an XOR gate that can either pass the input signal or its complement. Instead of comparing frequencies in traditional ROs, the authors compared statistical biases of pairs of oscillators implemented in different locations of the FPGA. Choosing the characteristic polynomial coefficients in Fibonacci or Galois ROs allowed for developing a rich set of responses using the same hardware.

The authors in reference [21] provided a performance analysis of XOR-Inverter based ring oscillator PUF. A three-stage RO PUF in their design consists of one NAND gate, one XOR gate and one inverter. A five-stage RO PUF would have one NAND gate, two XOR gates and two inverters. The authors quantized the performance of RO PUFs through defining equations for several system parameters such as reliability, uniqueness, uniformity, randomness, and bit aliasing.

The authors in reference [22] extended the designs proposed in [21]. The RO system was implemented in Xilinx Artix-7 FPGA using hard macros at the same location in each device using constant routing. The authors found that a 3-stage PUF had better frequency distribution compared to 5- and 7-stage designs.

3.The problem of authentication and secret key exchange

Multifactor authentication (MFA) is now commonly used to further protect systems and users form attacks. A new type of authentication is context-aware authentication. Usually MFA is based on two or more criteria [23]:

1) What a user knows: e.g. secret password
2) What a user have: e.g. access card RFID card, or a cell phone that can be used to establish one-time password (OTP)
3) Something unique about a user are: e.g. fingerprint, retina print, etc.
4) User's context of present location or past history such as location history or transaction history.

These same criteria can be used to provide security not only to individuals but also to IoT devices especially in telehealth. Endowing IoT devices with a secret password and fingerprint can be accomplished with the use of a PUF.

4.Ring Oscillator PUF Circuit

Figure 1 shows a ring oscillator system used as a PUF. The system consists of W ring oscillators and each ring oscillator contains S inverters connected back-to-back, where S is the number of inverter stages and must be an odd integer.

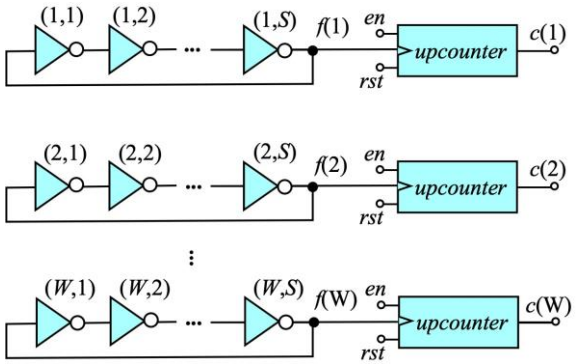

Fig. 1. Ring oscillator array used as a PUF (RO-PUF).

Each ring oscillator circuit consists of S inverters, where S is an odd number. To conserve power, an enable signal en is used to activate the RO system when a response is required during authentication. The oscillation frequencies f (w), with $1 \leq w \leq W$, are measured through up counters that are are triggered by the rising edges of the connected RO pulses.

The RO-based PUFs is delay based and as such establishing a time base must ensure immunity from environmental effects

and lack of synchronization between the server (authenticator) and the client (IoT edge device).

Assuming τ (w, s) represents the rise or fall time of the inverter at row w and column s, the frequency of oscillation of the ring oscillator in row w is given by:

$$c(w) = \lfloor T_{obs} f(w) \rfloor = \left\lfloor \frac{T_{obs}}{2T(w)} \right\rfloor \quad (1)$$

$$f(w) = \frac{1}{2T(w)} \text{ Hz} \quad (2)$$

$$T(w) = \sum_{s=1}^{S} \tau(w, s) \quad (3)$$

where Tobs is the observation time given to allow the upcounters to count serveral RO cycles, f (w) is the oscillation frequency of row w and T (w) is the total delay through the S oscillators

in row w. Random process variations (RPV) and CMOS noise ensure τ (w, s) is unique to each inverter in a given IoT device and across all the devices.

We can make several conclusions about RO-PUF from the above equation:

1) It is impossible to determine the values of τ (w, s) for any inverter in row w since all we can observe is the aggregate sum T (w).
2) If we attempt to increase the value of S we start to lose the random features of each row in the PUF system.
3) Increasing the observation time *Tobs* allows the RO response to cancel out the dynamic CMOS noise effects and enhances RPV effects and get a more stable value for $c(w)$. This value is unique for each row in Fig. 1.

We conclude therefore that obtaining unique identity, i.e. $c(w)$ value, for each row requires increasing the value of *Tobs* and reducing the value of S.

Fig. 2 shows how a common time base is established in the RO-based PUF system that allows the server (authenticator) and client (IoT device) to agree on a time base in the presence of different environmental conditions such as temperature, aging and lack of synchornization. S inverters are connected in series to form a ring oscillator. The output of this system is fed to an upcounter just like the RO PUF system. However, the upcounter is connected to a comparator to detect when the observation count *cobs* has been reached. The start signal initiates operation of the PUF and as long as the reference counter

value $c(r) < cobs$, the ring oscillators are operating since the comparator asserts the enable signal en. When $c(r) = cobs$ the comparator lowers en and the system starts reading the counter responses $c(w)$ with $1 \leq w \leq W$.

A challenge for such a PUF consists of selecting a $B + 1$ address vector **a** from among the W addresses of the RO system:

$$\mathbf{a} = \{a_i | \quad 0 \leq i \leq B\} \tag{4}$$

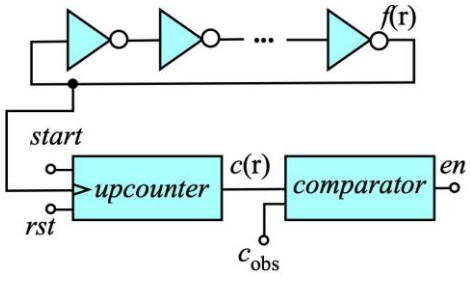

Fig. 2. Establishing an adaptive time base for the RO system.

where a0 corresponds to the address of the reference RO c0 and ai, $1 \leq i \leq B$ is i-th bit of the response word r. Bit ri of the response corresponds to comparing corresponding counter ci with the reference counter c0 according to Eq. (5).

The standard coding algorithm for composing the response vector of the device to be authenticated is to select the reference RO counter corresponding to the address specified by r0 and compare that value with the counter values ci ($1 \leq i \leq B$) will generate bit ri of a B-bit response vector r according to the following encoding rule:

$$r_i = \begin{cases} 0 & \text{when} \quad c(r_0) < c(r_i) \\ 1 & \text{when} \quad c(r_0) \geq c(r_i) \end{cases} \qquad (5)$$

We see that each bit of the response requires a ring oscillator system, which corresponds to on of the rows in Fig. 1. As we shall see in the sequel, this algorithm is not very effective for RO-based PUFs and a better algorithm is urgently needed.

5. Ring Oscillator PUF Statistical Model

The operation of ring oscillator PUF (RO-PUF) relies to two random physical phenomena: static or long-term random processing variations (RPV) and dynamic or short-term random CMOS noise. These phenomena control the oscillation frequency of the RO system. Random process variation is static for a given device and facilitates creation of the device ”biometric” or unique fingerprint. Dynamic random CMOS noise, on the other hand, is dynamic and introduces noise to the device identity (ID). There are several sources of CMOS noise:

1) Thermal noise represented as an additive white Gaussian noise (AWGN) showing flat spectral distribution
2) Shot noise due to charge carrier flow across semiconductor junctions showing flat spectral distribution
3) Flicker noise due to charge trapping centers in the semiconductor bulk showing 1/f spectral distribution These noise sources introduce variations in the rise and fall times of CMOS inverter transitions.

The random variable we choose to model should be amenable to study under mass production setting by the device manufacturer. In the context of using an RO-PUF, an appropriate random variable is delay of the inverter τ . Figure 3 shows the different types of distributions due to the

random processes involved in determining the inverter delay. There are two independent and additive random processes or physical phenomenal involved in determining τ and we can write

$$\tau = \tau p + \tau n \quad (6)$$

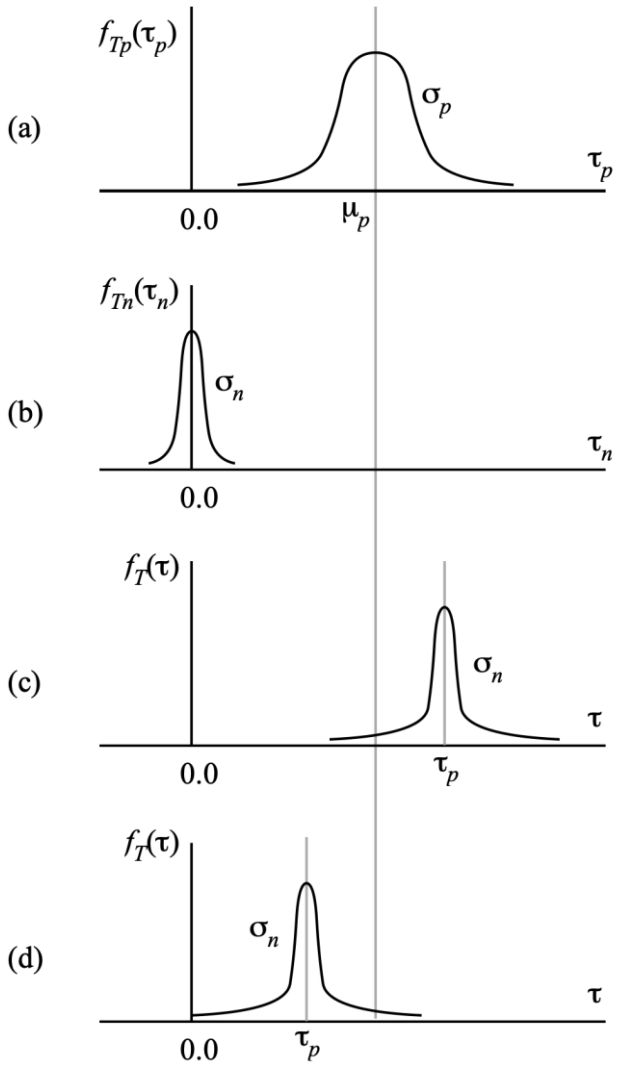

Fig. 3. The pdf distributions of inverter delay τ due to the different physical phenomena. (a) pdf of τp due to random process variations (RPV). (b) pdf of n due to dynamic random CMOS noise. (c) pdf of τ due to the combined effects of RPV and random dynamic random CMOS noise when τp > μp. (d) pdf of a due to the combined effects of RPV and random dynamic random CMOS noise when τp < μp.

where τp is the delay component due to the slowly-varying RPV and τn is the delay component due to the dynamic CMOS electronic noise.

Figure 3(a) shows the pdf of the random variable τp due to RPV which is a biased Gaussian process with mean μp and variance σ2. The parameters for RPV change slowly due to device aging and, more importantly, due to environmental effects such as the chip temperature. τp follows the biased Gaussian distribution whose pdf is given by

$$f_{T_p}(\tau_p) = \frac{1}{\sigma_p\sqrt{2\pi}} e^{-(\tau_p-\mu_p)^2/2\sigma_p^2} \qquad (7)$$

The value of τp as given by

$$\tau p \ = \ G(\mu p, \sigma p) \qquad (8)$$

where G(μp, σp) is a Gaussian random process with mean μp and variance σ2.

We should note that μp and σp are identical for all inverters within a device or among different devices. Furthermore, these values are static, or slowly-varying, and do not change with time once the IC is fabricated. For simplicity, we ignore IC aging effects

It is very important to note that μp and σp values are very much temperature dependent and they vary with ambient operating conditions of the IC. However, all inverters in the RO-PUF vary in unison and by speeding up or slowing down. Hence response encoding of Eq. (5) error effects due to ambient conditions cancel for the derivation of the response bits.

Figure 3(b) shows the pdf of the combined CMOS noise sources n which is given by the zero-mean Gaussian process

$$f_{T_n}(\tau_n) = \frac{1}{\sigma_n\sqrt{2\pi}} e^{-\tau_n^2/2\sigma_n^2} \qquad (9)$$

where σ2 is the variance of the dynamic random CMOS noise process. The value of τn is given by

$$\tau n \ = \ G(0, \sigma 2\ ) \qquad (10)$$

The combined effects of RPV and CMOS noise generate a pdf given by

$$f_T(\tau) = \frac{1}{\sigma_n\sqrt{2\pi}} e^{-(\tau-\tau_p)^2/2\sigma_n^2} \qquad (11)$$

where τp is the contribution of RPV and σn is the contribution of dynamic random CMOS noise.

Figure 3(c) shows the pdf of the transition probability τ when both RPV and dynamic random CMOS noise are present and the mean value τp > μp. Figure 3(d) shows the pdf of the transition probability a when both static RPV and dynamic random CMOS noise are present and the mean value τp < μp. For either case, the value of delay time τ is given by:

$$\tau \ = \ G(\tau p, \sigma n) \qquad (12)$$

Assuming S inverters in row w of the RO system, reference [24] indicates that the distribution of the inverter delay Tw

follows the Gaussian distribution tt(Vw , σ2) where the mean Tw and variance σ2

are given by:

$$T_w = \sum_{s=1}^{S} \tau(w,s) \approx \sum_{s=1}^{S} \tau_p(w,s) \quad (13)$$

$$\sigma_w^2 = \sum_{s=1}^{S} \left[\sigma_p^2 + \sigma_n^2(w,s)\right] = S\left[\sigma_p^2 + \sigma_n^2(w,s)\right] \quad (14)$$

In the limit, as S assumes large values, we can write

$$T_w = S\mu_p \quad (15)$$

$$\sigma_w^2 = S\left[\sigma_p^2 + \sigma_n^2(w,s)\right] \quad (16)$$

and we should expect the CMOS random noise to dominate the behaviour of the RO PUF. At this limit, the unique ID of the PUF is almost completely eliminated. This proves that it is advantageous to have small values of S to impart unique identities to the RO-based PUFs and also to reduce the amount of dynamic random noise introduced to the responses.

5.1 A. Signal-to-Noise Ratio (SNR) for RO PUF

By “signal” in the context of this work we refer to the inverter delay component due to the static or long-term RPV, which gives each inverter its unique identity and does not vary over time except perhaps for aging effects. The noise in this work refers the RO delay component due to the dynamic or short-term random CMOS noise.

The SNR for the production system can be described as:

$$SNR = 10\log\left(\frac{\mu_p^2 + \sigma_p^2}{\sigma_n^2}\right) \quad (17)$$

where the nominator represents the “energy” associated with the signal due to RPV and the denominator represents the “energy” associated with the noise due to CMOS noise.

6. Obtaining Ro Statistical Parameters And Golden CRP

Ring oscillators are used by IC fabricators as a standard means of obtaining basic inverter delay as a means of characterizing the performance of the CMOS devices being fabricated. RO circuits are distributed at different places in the IC to average out the inevitable RPV.

Assume the manufacturer constructs W ROs distributed over the surface of the IC being manufactured and assume also that each RO consists of S inverters. The total delay of each RO was given in Eq. (3). We can write the average inverter delay as

$$\begin{aligned} \tau_{avg} &= < T(w) > \\ &= \frac{1}{WS} \sum_{w=1}^{W} T(w) \\ &= \frac{1}{WS} \sum_{w=1}^{W} \sum_{s=1}^{S} \tau(w,s) \\ &= \frac{1}{WS} \sum_{w=1}^{W} \sum_{s=1}^{S} (\tau_p(w,s) + \tau_n(w,s)) \\ &= \frac{1}{WS} \sum_{w=1}^{W} \sum_{s=1}^{S} \tau_p(w,s) \\ &= \mu_p \end{aligned} \quad (18)$$

Therefore the device manufacturer is able to obtain the average inverter delay after taking into account RPV and CMOS noise. The variance of τ (w, s) is measured as follows:

$$\begin{aligned} \sigma^2 &= < (\tau(w,s) - \mu_p)^2 > \\ &= \frac{1}{WB} \sum_{w=0}^{W-1} \sum_{s=0}^{S-1} [\tau(w,s) - \mu_p]^2 \\ &= \frac{1}{WB} \sum_{w=0}^{W-1} \sum_{s=0}^{S-1} [\tau^2(w,s) - 2\mu_p \tau(w,s) + \mu_p^2] \\ &= \frac{1}{WB} \sum_{w=0}^{W-1} \sum_{s=0}^{S-1} [\tau^2(w,s) - \mu_p^2] \end{aligned} \quad (19)$$

However, we know that τ (w, s) = τp(w, s) + τn(w, s). Substituting this in the above equation, we can write:

$$\begin{aligned} \sigma^2 &= \frac{1}{WB} \sum_{w=0}^{W-1} \sum_{s=0}^{S-1} [\tau_p^2(w,s) + \tau_n^2(w,s) - \mu_p^2] \\ &= \sigma_p^2 + \sigma_n^2 \end{aligned} \quad (20)$$

We see that the device manufacturer is only able to measure μp and the sum $\sigma_p{}^2 + \sigma_n{}^2$.

6.1 Obtaining the Golden CRP by Device Fabricator

The main goal for obtaining the golden CRP is to eliminate the short-term random CMOS noise effects. To that end, the manufacturer chooses an appropriate value for c0 and the number of tests to perform N . A W × N matrix C of observed counter values is constructed as follows:

$$\mathbf{C} = \begin{bmatrix} c(1,1) & c(1,2) & \cdots & c(1,N) \\ c(2,1) & c(2,2) & \cdots & c(2,N) \\ \vdots & \vdots & \ddots & \vdots \\ c(W,1) & c(W,2) & \cdots & c(W,N) \end{bmatrix} \quad (21)$$

where c(w, n) is the value of the counter in row w at iteration n.

To obtain a golden response, we define the sample mean of row w count values by the following equation which is a slightly modified version from the one given in [25]:

$$c_{avg}(w) = \frac{1}{N}\sum_{n=1}^{N} c(w,n) \quad (22)$$

The above equation gives the average counter value at row w that removes the effects of CMOS noise. The golden response of the RO system is given by the W -bit vector cg:

$$\boldsymbol{c_g} = \left[c_g(1)c_g(2)\cdots c_g(W)\right] \quad (23)$$

The fabricator now prepares a database of the different address vectors a according to Eq. (4) and their associated responses cg which are obtained using the coding rules summarized in Eq. (5).

Now the device fabricator is able to generate a CRP dataset, associated with a given IoT device, by choosing the parameters associated with each CRP as shown in Table I.

Table 1: Parameters associated with each CRP in a particular IoT device

Symbol	Comment	Reference
$\mathbf{a}$	Length $B+1$ challenge address vector	Eq. (4)
$\mathbf{r}$	Length $B+1$ response vector	Eq. (5)
N	Number of iterations to average counters	Eq. (22)
$\mathbf{c}_g$	Length W golden counter values	Eq. (23)

7.MUTUAL AUTHENTICATION AND SECURE KEY EXCHANGE

A very crucial role for using PUFs in unsecured IoT edge devices is to establish mutual authentication and secure key exchange. The unique device ID is the basis for establishing mutual authentication through the CRP mechanism and some hashing operation. The generation of a secure session key is done through forward error correction codes by what is know as the helper data algorithm [6], [7], [8], [11], [12], [25], [26], [27], [28], [29].

Figure 4 explains the helper data algorithm to establish secure key exchange between the server (any device communicating with IoT device) and the client (the IoT device itself). The server uses the challenge c to get the reference/golden response r.

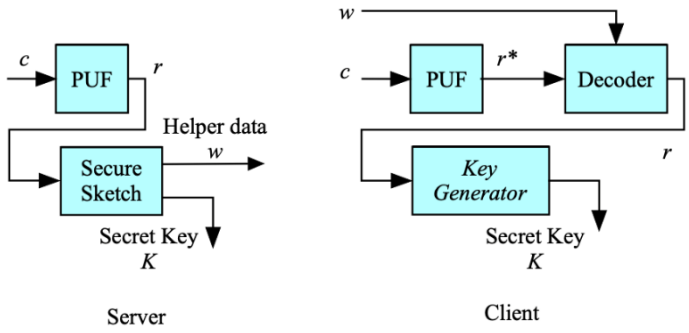

Figure 4: Modified ring oscillator array used as a PUF (RO-PUF).

This response is fed to a secure sketch block which generates two outputs: the secret key K and helper data w. The helper data is generated as the redundant bits generated through block error correcting codes [26]. W can be sent to the client device on a public unsecured challenge.

The client now receives the challenge c and helper data w. From the challenge, the client uses the resident PUF to generate the noisy response r∗. The noise is removed using both r∗ and w to generate the correct copy r at the server. The key generator uses r to generate the secret session key K.

Table 2: compares the Client/Server model for PUF-based authentication with the Receiver/Sender model for communication over a noisy channel.

Telecommunication System	**Authentication System**
Sender	**Server**
$m_1 = \text{Encode}(m)$	$r = \text{PUF}(c)$ $w = \text{Encode}(r)$ h = Hash(r)
Channel	**Channel**
$m_2 = \text{Xmt}(m_1, n)$	$m_2 = \text{Xmt}(\{c\|\|w\}, 0)$
Receiver	**Client**
$m = \text{Decode}(m_2)$	$r^* = \text{PUF}(c)$ $r = \text{Decode}(r^*, w)$ $h^* = \text{Hash}(r)$
Channel	**Channel**
ACK/NAK	h^*

The salient features of the comparison between telecommunications (Telecom) and IoT PUF-based authentication (IoT) are:

1) In Telecom the sender encodes a message while in IoT the server calculates response due to a challenge and prepares helper data based on response.
2) In Telecom the channel is noisy and introduces noise on the transmitted message. In IoT the channel is assumed noiseless or forward error correcting coding could be employed.
3) In Telecom the receiver decodes the received corrupted message to recover the original message. In IoT the client generates a noisy response and uses the helper data to recover the noise-free response.

8.ATTACKS ON PUFS

There are many attacks on hardware, such as IoT devices, such as [27]:

1) Cloning
2) Counterfeiting
3) Overbuilding in gray markets
4) Intellectual property theft and piracy
5) Reverse engineering
6) Tampering
7) Side-channel
8) Machine learning

All the attacks, with the exception of the last, can be defeated or mitigated with the use of PUFs since the ID of the PUF will be altered. The last attack can defeat PUF action through machine learning to try to mimic the PUF response. This is a very destructive attack since the action of the PUF can be predicted with a small number of response bits in error.

As we shall see in Section X-C, this attack can be defeated using Algorithm #3 we propose here.

9.PROPOSED CONFIGURABLE STRONG RO PUF

A standard RO PUF offers a limited number of possible CRP options based on two parameters: the number of RO rows *W* in Fig. 1 and the number of inverter stages *S*. The number of possible challenges is estimated as

$$\#CRP(\text{standard}) = \binom{W-1}{B} \qquad (24)$$

In order to increase the CRP space and hence improve security we propose a modified RO derived form a Galois linear feedback shift register (LFSR). A similar design can be found in [19]. In the design of [19] the inverters and XOR gates are connected in series and are always in the path of the system. Our proposed design uses the inverters and XNOR gates that are connected in parallel as shown in Fig. 5.

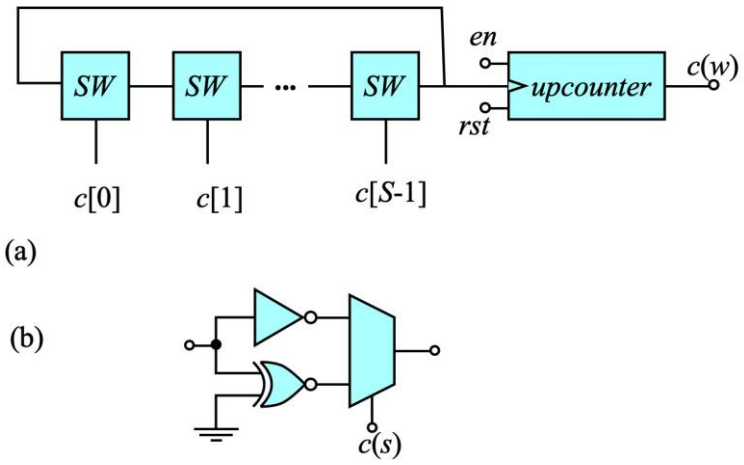

Fig. 5. Highly configurable RO array used based on Fig.**1**.
(a) Introduction of configurable switch boxes *SW* between the inverters in all RO rows. (b) The different configurations of the switch boxes. Each switch can be in the *open* position (far left), *feedback* position (middle), and *through* position (far right).

Fig. 5(a) shows the proposed RO system where we have *S*-stages and each stage is a selectable design based on the selection word *c*(*s*).

Fig. 5(b) shows the details of each SW stage which selects between the delay of an inverter or an XNOR gate.

The number of CRP pairs can now be expressed as

$$\#CRP \text{ (proposed)} = B \times 2^S \qquad (25)$$

where it was assumed that we only had $B + 1$ rows. If we had W rows, the number of CRP is increased to

$$\#CRP(\text{proposed}) = 2^S \times \binom{W-1}{B} \qquad (26)$$

The following CRP generation strategy is adopted to generate a strong PUF out of the RO PUF that is immune to thermal noise and environmental variations:

1) Thermal noise is removed by using a long observation time which translates to a large observed upcounter value $cobs$.
2) Since establishing a common time base is difficult in the face of no global synchronization, we replace the time base with an observation counter value $cobs$ as shown by the blue blocks at the top of Fig.2. The upcounter is clocked by its own ring oscillator system and generates the reference upcounter value $c(r)$. The comparator compares this value with the observation counter value $cobs$ supplied by the server/authenticator. As long as $c(r) < cobs$ the enable output en is asserted to allow the RO system to operate. When $c(r) = cobs$ the enable output en is 0 to stop the RO system. This indicates the response of the PUF is ready for measurement.
3) An address vector **a** is used to select B counters to generate a B-bit response vector **r**. The elements of **a** are randomly selected from among W oscillators. This gives $\binom{W}{B}$ CRP choices.
4) Environmental variations are overcome by basing the RO response on a comparative evaluation of counters in the same IoT device. This is done by comparing the chosen reference counter $c(0)$ with the other B counters to remove the effect of environmental variations.

10. ALGORITHMS FOR AUTHENTICATING RO PUF

The next three subsections present three algorithms for authenticating an IoT device

10.1 Algorithm 1: Single-Challenge

In Algorithm 1, the server, or the device fabricator, selects the challenge in the form of an an address vector **a** for the RO counters to be interrogated. To establish a time base, the authenticator also selects an upcounter reference value c_{obs}.

Algorithm 1 Single challenge algorithm

Server/fabricator side

Require: c_{obs}, **a**
1(a) $c = \text{Challenge}(\mathbf{a})$, $r = \text{PUF}(c)$
2(a) $w = \text{Encode}(r)$, $h = \text{Hash}(r)$, $K = \text{Generate\_key}(c, r)$
3(a) **return** r, w, h & K

Client/IoT device side

Require: c_{obs}, **a**, w
1(b) $c = \text{Challenge}(\mathbf{a})$, $r^* = \text{PUF}(c)$
2(b) $r = \text{Decode}(r^*, w)$, $h^* = \text{Hash}(r)$
3(b) $K = \text{Generate\_key}(r)$
4(b) **return** r, h^*, K

The details for the server/fabricator in Algorithm 1 are:

L1(a): Server selects parameters for CRP: c_{obs} and **a** to obtain the device response *r* after fabrication.

L2(a): Server generates helper data *w*, hash for authentication *h*, and session secret key *K*. The has value and secret key could depend on several parameters to ensure context-aware authentication or adaptive authentication.

L3(a): Client creates a PUF database that includes c_{obs}, **a**, *r* and *w* The details of the client/IoT device operation in the field for Algorithm 1 are:

L1(b): Client prepares the challenge *c* based on the received address vector **a** and applies it to the PUF to obtain the noisy response *r**

L2(b): Client uses the helper data *w* to remove the noise from *r** and obtain noise-free response *r*. Using *r*, the client obtains the hash value *h* to be used for authentication.

L3(b): Using the estimated *r*, the client obtains the session key *K* to be used for coding and decoding of data.

The performance results for Algorithm #1 are summarized in Table 3.

Table 3: Algorithm #1 normalized intra and inter Hamming distances.
Case when $S = 3$ inverters, $\mu_p = 1$, $\sigma_p = 0.3$, $SNR_{max} = 30\ dB$.

B (bits)	16	32	64	128
Normalized Intra Hamming Distance	0.04	0.02	0.07	0.05
Normalized Inter Hamming Distance	0.44	0.44	0.34	0.48

We observe that the values of normalized intra Hamming distance are close to the ideal value of 0. The values of normalized inter Hamming distance are close to the ideal value of 0.5. Similar results are obtained when the value of S is increased to 5, 7 and 9. The number of bits in error is 5 when B = 128. This estimate determines the error correcting capabilities of the helper data algorithm.

10.2 Algorithm #2: Repeated Challenge

The basic idea behind Algorithm #2 is to eliminate dynamic random CMOS noise by repeating the steps used by the manufacturer to obtain the golden reference RO response according to the discussion in Sec. 6.

Algorithm 2 Repeated challenge

Server/fabricator side

Require: c_{obs}, $\mathbf{a}$, N
1(a) $c = \text{Challenge}(a)$
2(a) Generate $\mathbf{C} = \text{PUF}(c_{obs}, \mathbf{a}, N)$
3(a) Calculate $\mathbf{r}_g$, w, h, K
4(a) **return** $\mathbf{r}_g$, w, h, K

Client/IoT device side

Require: c_{obs}, $\mathbf{a}$, N, w
1(b) Generate $\mathbf{C}^* = \text{PUF}(c_{obs}, \mathbf{a}, N)$
2(b) Calculate $\mathbf{r}_g^*$
3(b) Calculate $\mathbf{r}_g = \text{Decode}(\mathbf{r}_g^*, w)$, $h = \text{Hash}(\mathbf{r}_g)$
4(b) $K = \text{Generate\_key}(\mathbf{r}_g)$
5(b) **return** $\mathbf{r}_g$, h^*, K

At the server after device fabrication, the following steps are performed:

L1(a): Server generates the challenge word based on address vector **a**

L2(a): Server generates the $(B + 1) \times N$ matrix **C**

L3(a): Server calculates golden response $\mathbf{r}g$, as well as w, h and K

L4(a): Server prepares the authentication database consisting of $\mathbf{r}g$, At the client side in the field, the following operations are performed:

L1(b): Client calculates the counter values matrix $\mathbf{C}^{*_g}$

L2(b): Client calculates the average response $\mathbf{r}^*$

L3(b): Client calculates corrected averaged response $\mathbf{r}g$ and corresponding hash value h

L4(b): Client calculates the session secret key k

Table 4 shows the performance of Algorithm #2.

Table 4: Algorithm #2 normalized intra and inter Hamming distances.
Case when $S = 3$ inverters, $\mu_p = 1$, $\sigma_p = 0.3$, $SNR_{max} = 30\ dB$.

B (bits)	16	32	64	128
Normalized Intra Hamming Distance	0.00	00.00	0.00	0.01
Normalized Inter Hamming Distance	0.34	0.38	0.53	0.46

From the table we observe that the number of bits in error in the response is very small and in the range of 0 to one bit only. The error correcting code requirements for Algorithm #2 are very modest compared to Algorithm #1.

10.3 Algorithm # 3: Repeated Challenge with Bit Selection

Algorithm #3 is derived from Algorithm #2. The main idea of this algorithm is to consider or select the response values that have high SNR in a further attempt to reduce effects of CMOS noise. This selection is based on the statistical properties of the individual RO PUF modules in the system. The RO rows to be eliminated are those that have low SNR. The criterion to select a response bit to be part of the filtered response is based on the difference in counter the RO values.

Given a collection of B counters c used to construct the response r according to the procedure in Eq. (??). The algorithm for selecting a counter to generate the reduced response is shown in Algorithm 3.

Algorithm 3 Repeated challenge with bit selection.

Server/fabricator side

Require: c_{obs}, a, N
1(a) Generate $\mathbf{C} = \text{PUF}(c_{obs}, \mathrm{a}, N)$
2(a) Calculate $\mathbf{c}_g$, σ_c
3(a) Initialize $\mathrm{a}_{red} = [\,]$, $\mathbf{c}_{red} = [\,]$, $\mathrm{r}_{red} = [\,]$ % Empty arrays

4(a) Select c_{ref} % Reference counter
5(a) Calculate r_g, σ_c, w, h, K
6(a) **for** $b = 1 : B$ **do**
7(a) **if** $|\mathbf{c}_g(b) - c_{ref}| > \sigma_c$ **then**
8(a) $\mathrm{a}_{red} = [\mathrm{a}_{red} \quad b]$; % Augment address vector
9(a) $\mathbf{c}_{red} = [\mathbf{c}_{red} \quad \mathbf{c}(b)]$; % Augment counters vector
10(a) $\mathrm{r}_{red} = [\mathrm{r}_{red} \quad \mathrm{r}(b)]$; % Augment response vector
11(a) **end if**
12(a) **end for**
13(a) Calculate h, K
14(a) **return** a_{red}, r_{red}, h, K

Client/IoT device side

Require: a_{red}
1(b) Generate $\mathbf{C} = \text{PUF}(c_{obs}, \mathrm{a}, N)$
2(b) Calculate $\mathbf{c}_g$, σ_c
3(b) a_{red}, c_{red}, $r_{red} = [\,]$;
4(b) **for** $b = 1 : B$ **do**
5(b) **if** $|c(b) - c_i| > \sigma_c$ **then**
6(b) $\mathrm{a}_{red} = [\mathrm{a}_{red} \quad b]$; % Augment address vector
7(b) $\mathbf{c}_{red} = [\mathbf{c}_{red} \quad \mathbf{c}(b)]$; % Augment counters vector
8(b) $\mathrm{r}_{red} = [\mathrm{r}_{red} \quad \mathrm{r}(b)]$; % Augment counters vector
9(b) **end if**
10(b) **end for**
11(b) Calculate h, K
12(b) **return** r_{red}, h, K

L1(a): Server exercises the device to generate the counter matrix **C** for all iterations *N*

L2(a): Server

L3(a): Server prepares empty arrays to represent the reduced address vector **a**, counters *c*, and response *r*

L4(a): Server selects the averaged reference counter value *cref* based on the address vector **a**

L5(a): Server calculates average golden response **r***g* , standard deviation *σc*, helper data *w*, hash value *h* and secret session key *K*

L6(a)–12(a): Server scans all the B counters used to generate the response and select the counters that satisfy the condition in Line 7. Reduced address bits ($\mathbf{a}red$), counter values ($\mathbf{c}red$), and response bits ($\mathbf{r}red$) are extracted.

The details of the client/IoT device operations in the field for Algorithm 3 are as follows:

L1(b): Client exercises the PUF for N times and measures the counters values as matrix $\mathbf{C}$

L2(b): Client averages the counters values as the vector $\mathbf{c}_g$ and calculates their standard deviation σ_c

L5(b)–L9(b): Client selects the counters to be used to generate the response by constructing the reduced vectors $\mathbf{a}red$, $\mathbf{c}red$, and $\mathbf{r}red$.

L11(b): Client calculates hash value h to be used for authenticating the device and the session secret key K

Table 5 shows the performance of Algorithm #3.

Table 5: Algorithm #3 maximum intra Hamming distance and minimum separation between inter and intra Hamming distances for the. Case when $S = 3$ inverters, $\mu_p = 1, \sigma_p = 0.3$, $SNR_{max} = 30\ dB$.

B (bits)	16	32	64	128
# Selected bits B_{red}	9	17	19	42
Normalized Intra Hamming Distance	0.00	0.00	0.00	0.00
Normalized Inter-Intra Hamming Distance	0.22	0.41	0.27	0.24

From the table we observe that the number of bits in error in the response is zero and there is no need for any error correcting coding such as using helper data algorithm or fuzzy extractors. This gives a clear advantage of Algorithms #1 and Algorithm #2. Therefore Algorithm #3 is very much suited for use in IoT devices that have limited compute and energy resource

11. Conclusions and Future Work

This paper proposed a novel strong ring oscillator PUF (ROPUF) structure that significantly increase the number of possible challenge/response pairs. In addition, the paper proposed three algorithms for selecting the challenge/response pairs. One of these algorithms has the property of eliminating the noise from the PUF response by selecting certain bits of the response based on its statistical properties. Which bits to be selected is to be independently determined by the server and client. This further increases the security of the system.

Further work in the area of RO-PUF should target reducing the observation period and speed up the CRP response time. One method to achieve this is to reduce the number of stages in each ring oscillator system. This results in faster oscillations but the effect on stability of the response must be studied.

12. Acknowledgement

This research was supported by a grant from the National Research Council of Canada (NRC) through the Collaborative R&D Initiative.

13. References

[1] B. Gassend, D. Clarke, M. V. Dijk, and S. Devadas, "Silicon physi- cal random functions," in Proceedings of the 9th ACM Conference on Computer and Communications Security, 2002, pp. 148–160.

[2] B. Gassend, D. Lim, D. Clarke, M. van Dijk, and S. Devadas, "Iden- tification and authentication of integrated circuits," Concurrency and Computation: Practice & Experience, vol. 6, no. 11, pp. 1077– 1098, 2004.

[3] T. McGrath, I. E. Bagci, Z. Wang, U. Roedig, and R. Young, "A PUF taxonomy," Applied Physics Reviews, vol. 6, no. doi: 10.1063/1.5079407, 2019.

[4] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in Design Au- tomation Conference, 2007, pp. 9–14.

[5] E. Bou-Harb and N. Neshenko, Cyber Threat Intelligence for the Internet of Things. Springer, 2020.

[6] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper data algorithms for PUF-based key generation: Overview and analysis," IEEE Transactions on Computers, vol. 34, no. 6, pp. 889– 902, 2014.

[7] J. Delvaux, "Security analysis of PUF-based key generation and entity authentication," Ph.D. dissertation, University of KU Leu- ven and ShangHai Jiao Tong University, 2017.

[8] ——, "Machine learning attacks on PolyPUF, OB-PUF, RPUF, and PUF–FSM," in IACR Cryptology, 2017.

[9] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," SIAM Journal on Computing, vol. 38, no. 1, pp. 97–139, 2008.

[10] 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 volume 3027 of Lecture Notes in Computer Science, C. Cachin and J. L. Camenisch, Eds., 2004, pp. 523–540.

[11] R. Maes, Physically Unclonable Functions: Constructions, Properties and Applications. Springer, 2013.

[12] R. Maes, A. van Herrewege, and I. Verbauwhede, "PUFKY: A fully functional PUF-based cryptographic key generator," in Cryp- tographic Hardware and Embedded Systems (CHES), 2012.

[13] R. Maes, P. Tuyls, and I. Verbauwhede, "Low-overhead implemen- tation of a soft decision helper data algorithm for SRAM PUFs," in Cryptographic Hardware and Embedded Systems (CHES), C. Clavier and K. Gaj, Eds. Springer, 2009, pp. 332–347.

[14] X. Xin, J. Kaps, and K. Gaj, "A configurable ring-oscillator-based PUF for xilinx FPGAs," in IEEE Euromicro Conference Digital System Design, 2011, pp. 651–657.

[15] A. Maiti and P. Schaumont, "Improved ring oscillator PUF: an FPGA-friendly secure primitive," Journal of Cryptology, vol. 24, pp. 375–397, 2011.

[16] Z. Cherif, J. Danger, and S. Guilley, "An easy-to-design PUF based on a single oscillator: the loop PUF," in IEEE Euromicro Conf. Digital System Design, 2012, pp. 156–162.

[17] Y. Cui, C. Wang, W. Liu, Y. Yu, M. O'Neill, and F. Lombardi, "Low- cost configurable ring oscillator PUF with improved uniqueness," in International Symposium on Circucits and Systems (ISCAS), 2016, pp. 558–561.

[18] M. Garcia-Bosque, G. Diez-Senorans, C. Sanchez-Azqueta, and S. Celma, "Proposal and analysis of a novel class of PUFs based on galois ring oscillators," IEEE Access, vol. 8, 2020.

[19] J. D. J. Golic, "New methods for digital generation and postpro- cessing of random data," IEEE Trans. Comput., vol. 55, no. 10, pp. 1217–1229, Oct. 2006.

[20] N. A. Hazari, F. Alsulami, A. Oun, and M. Niamat, "Performance analysis of XOR-inverter based ring oscillator PUF for hardware security," in IEEE National Aerospace and Electronics Conference (NAECON), 2019.

[21] F. Alsulami and M. Niamat, "Performance study of FPGA based AND-inverter ring oscillator PUFs," in IEEE International Confer- ence on Electro Information Technology (EIT), 2020.

[22] B. Maciej, E. F. Imed, and M. Kurkowski, "Multifactor authenti- cation protocol in a mobile environment," IEEE Access, vol. 7, pp. 157 185 – 157 199, 2019.

[23] C. E. Rasmussen and C. K. I. Willams, Gaussian Processes for Machine Learning. MIT Press, 2005.

[24] A. Gersho and R. M. Gray, Vector Quantization and Signal Compres- sion. Boston, USA: Kluwer Academic Publishers, 1992.

[25] G.-J. Schrijen, "SRAM PUF: A closer look at the most reliable and most secure PUF," https://www.design- reuse.com/articles/47782/sram-puf-a-closer-look-at-the-most- reliable-and-most-secure-puf.html, 2020.

[26] Y. Gao, H. Ma, S. F. Al-Sarawi, D. Abbott, and D. C. Ranas- inghe, "PUF-FSM: A controlled strong PUF," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 5, pp. 1104 – 1108, May 2018.

[27] M. Hiller, "Key derivation with physical unclonable functions," Ph.D. dissertation, Universitat Munchen, 2016.

[28] V. van der Leest, B. Preneel, and E. van der Sluis, "Soft decision error correction for compact memory-based pufs using a sin- gle enrollment," in Cryptographic Hardware and Embedded Systems (CHES), E. Prouff and P. Schaumont, Eds., 2012, pp. 268–282.

[29] C. Bo˙ sch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," in Cryptographic Hardware and Embedded Systems (CHES), E. Oswald and P. Rohatgi, Eds. Springer, 2008, vol. 5154, pp. 181–197.

[30] J. Baylis, Error-Correcting Codes: A Mathematical Introduction. Chapman & Hall, 2018.

[31] R. G. Dutta, X. Guo, and Y. Jin, "IP trust: The problem and design/validation-based solution," in Fundamentals of IP and SoC Security: Design, Verification, and Debug, S. Bhunia, S. Ray, and S. Sur-Kokay, Eds. Springer, 2017.