Efficient Attacks on Robust Ring Oscillator PUF with Enhanced Challenge-Response Set

Phuong Ha Nguyen*, Durga Prasad Sahoo†, Rajat Subhra Chakraborty‡, and Debdeep Mukhopadhyay§
SEAL/CSE, Indian Institute of Technology Kharagpur, INDIA - 721302
Email: phuongha.ntu@gmail.com* & {dpsahoo†,rschakraborty‡,debdeep§}@cse.iitkgp.ernet.in

***Abstract*—Physically Unclonable Function (PUF) circuits are an important class of hardware security primitives that promise a paradigm shift in applied cryptography. Ring Oscillator PUF (ROPUF) is an important PUF variant, but it suffers from hardware overhead limitations, which in turn restricts the size of its challenge space. To overcome this fundamental shortcoming, improved ROPUF variants based on the subset selection concept have been proposed, which significantly "expand" the challenge space of a ROPUF at acceptable hardware overhead. In this paper, we develop cryptanalytic attacks on a previously proposed low–overhead and robust ROPUF variant. The proposed attacks are practical as they have quadratic time and data complexities in the worst case. We demonstrate the effectiveness of the proposed attack by successfully attacking a public domain dataset acquired from FPGA implementations.**

***Index Terms*—Cryptanalysis, hardware–intrinsic security, physically unclonable function (PUF), ring oscillator PUF (ROPUF).**

I. Introduction

Physically Unclonable Function (PUF) [1], [2] on silicon is a circuit (analog/digital) that is sensitive to process variation due to uncontrollable imperfection in CMOS process. PUF circuit is used to extract these existing randomness in the embedding devices. It is a physical one-way function which typically exhibits a unique and instance-specific input-output (a.k.a. challenge-response) behavior: response to a given challenge is defined using the intrinsic and random physical properties of the embedding device.

In recent years, researchers have proposed versatile security solutions using PUFs as an alternative root-of-trust for conventional cryptographic solution using black-box model. For example, PUFs are used in device identification and authentication [3], binding software to hardware platforms [4], secure storage of cryptographic secrets [5], and secure protocol designs [6]–[8].

A silicon PUF circuit [2], [9], to be acceptable, must have a set of desirable properties [2], the most important among them are: *uniqueness*, *uniformity* and *reliability*. These are well–defined quantitative performance metrics which estimate the suitability of a given PUF implementation for security applications. However, probably the most important property of a given PUF that determines its acceptability in security applications is its physical and computational *unclonability* [2]. Physical unclonablity refers to the impossibility of physically manufacturing a PUF with pre-specified challenge-response behavior. A more interesting property is mathematical unclonability, whereby given a small subset of its complete challenge-response pair (CRP) set, it is computationally infeasible for an adversary to build a mathematical model of the PUF, that can be used to predict the correct response to an yet unseen challenge with a high probability of success. It has been shown that certain computational *model building attack* techniques, most based on advanced machine learning algorithms, are extremely successful in modeling certain types of PUF circuits [2], [10].

Ring Oscillator PUF (ROPUF) [11], [12] is considered to be one of the most common PUF variants. A ROPUF instance, as shown in Fig. 1, consists of m ring oscillators with a set of oscillation frequencies $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$. The CRPs are generated by applying a $\log_2 m$–bit challenge, selecting (depending on the applied challenge) a unique pair of ROs from the available set, and comparing the oscillation frequencies of the two ROs in the pair. It has been shown that the upper bound of the CRP space is $\log_2(m!)$ [11]. While relatively resistant against machine learning based modeling attacks, ROPUFs are severely limited by the exponential dependence of the number of ROs required on the challenge bit-length. This means that only a small challenge space is feasible, because of practical restrictions on hardware overhead, which makes an adversary's task of modeling a given ROPUF instance trivial, just by exhaustive characterization. In addition, ROPUFs are not the best from the *reliability* perspective. This implies that the response of the same ROPUF instance to the same challenge might vary over time, which is a serious drawback for PUFs to be used as a hardware security primitive.

To overcome the above mentioned shortcomings of ROPUFs, an improved ROPUF variant was proposed in [13], which incorporated two main ideas:

1) Instead of comparing the oscillation frequencies of a pair of ROs, response to a given challenge is defined by selecting a subset of the m oscillation frequencies $\{f_1, \ldots, f_m\}$ of the ring oscillators (details in Section II). This improves the hardware overhead of the ROPUF significantly.
2) Using the method of *Shielding Function* [14], and auxiliary data termed as *helper data* (W), the noisy response of the PUF is corrected.

The authors of [13] performed very detailed analysis and quantitative experimental evaluation of the hardware overhead, statistical properties and the security properties, including its resistance to various forms of attacks such as machine learn-

978-3-9815370-4-8/DATE15/©2015 EDAA

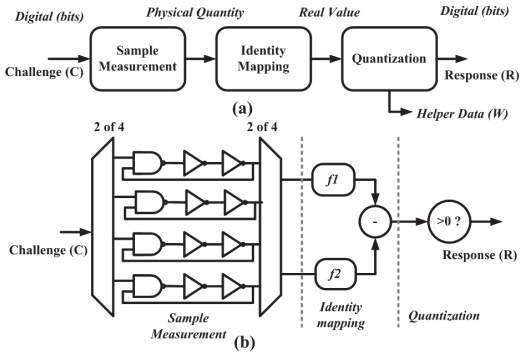

Fig. 1. (a) Generic PUF design framework, (b) ROPUF as an example [13].

ing based model building attack, differential attack, reverse engineering attack. However, in this paper we demonstrate that regardless of these desirable properties, the PUF proposed in [13] is vulnerable to a cryptanalytic attack. The main insight that enables the attack is the fact that the responses to different challenges are not independent of each other, and if this fact is utilized in conjunction with suitably chosen challenges, responses to certain other challenges can be known with a success probability greater than $\frac{1}{2}$ for each response bit (note that a success probability of $\frac{1}{2}$ implies completely random prediction by the adversary's attack algorithm, with "coin tossing" levels of success). Furthermore, by utilizing the *helper data* intelligently, the adversary can predict the response with probability of success equal to 1.

The rest of the paper is organized as follows. In Section II, we analyze the mathematical specification of the robust ROPUF circuits as in [13], and the notion of security in the context of a cryptanalytic attack on it. In Section III, we present the actual cryptanalysis of the enhanced ROPUF. We present experimental results in Section IV. Finally, we conclude the papers in Section V.

II. Mathematical Specification of Enhanced ROPUF and Security Notion

A. Mathematical Specification

Let $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ be the set of frequencies of the m ROs and $\mathcal{I} = \{1, 2, \ldots, m\}$ be the set of indices of m ROs. The following (positive) real mapping, $Q : 2^{\mathcal{I}} \setminus \mathcal{A} \rightarrow \mathbb{R}^+$, $2^{\mathcal{I}}$ being the power set of $\mathcal{I}$ and $\mathcal{A} = \{\phi, \{1\}, \{2\}, \ldots, \{m\}\}$, will assign a positive real number Q to each subset $\mathcal{S} = \{i_1, \ldots, i_t\} \in 2^{\mathcal{I}} \setminus \mathcal{A}$, using the following relationship:

$$Q(i_1, \ldots, i_t) = \sum_{u=1}^{t-1} \sum_{v=u+1}^{t} w_{i_u i_v} \left| f_{i_u} - f_{i_v} \right|^e, \qquad (1)$$

where f_{i_u},$f_{i_v} \in \mathcal{F}$, $1 \leq i_1, \ldots, i_t \leq m, i_1 \neq i_2 \neq \ldots \neq i_t$, $2 \leq t \leq m$, the weights are defined as $w_{i_u i_v} = |i_u - i_v|$, and the exponent e is a real number other than 1.

Without loss of generality, assume that the m-bit challenge $c = (c_1, c_2, \ldots, c_m)$ will select a subset $\mathcal{S} \subseteq 2^{\mathcal{I}} \setminus \mathcal{A}$ of index of ROs in the following way: $i \in \mathcal{S}$ if $c_i = 1$, $i = 1, \ldots, m$. The corresponding Q-value of the selected set $\mathcal{S}$ is then computed by applying (1). To overcome the issue of unreliability of ROPUFs, the 1-bit response r corresponding to the challenge c is generated following a two-phase process, as described below.

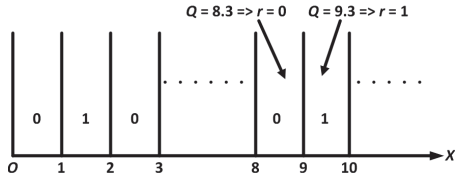

Fig. 2. The labeling scheme to determine the reference response r from the Q value calculated.

1) *Enrollment Phase:* During the enrollment phase, the range of Q is divided into equal intervals, each with width q. The intervals are alternatively assigned labels 0 and 1. For a given challenge c, the corresponding Q is calculated after characterizing the ROPUF at normal operating conditions, and the label of the interval to which Q belongs to is considered to be the corresponding *reference* response r_{ref}. This r_{ref} is stored in a secure database. For example, if $q = 1$, and $Q = 8.3$ (or $Q = 9.3$) then $r_{ref} = 0$ (or $r_{ref} = 1$), for the labeling scheme shown in Fig. 2. According to [13], Q should be measured several times during the enrollment phase to define the reference response, and let its average be denoted by $\overline{Q} = \mu$. For the subsequent discussion, we assume $Q = \overline{Q} = \mu$. A helper data W is generated by using the Shielding function defined as follows:

$$W = \begin{cases} (2\bar{n} + \frac{1}{2})q - \mu, & \text{if } r = 1 \\ (2\bar{n} - \frac{1}{2})q - \mu, & \text{if } r = 0 \end{cases} \qquad (2)$$

where $\bar{n}$ is the maximum positive integer such that $W < q$.

2) *Evaluation Phase:* Since PUF is not reliable, the response of a given challenge c may be noisy, i.e., applying c twice we would have two different responses. During the evaluation phase, the helper data W is used to generate the response r which is similar to the reference response r_{ref}. Let Q' be the noisy Q-value of challenge c. Using W, the response r is computed as follows:

$$r = \begin{cases} 1, & \text{if } 2\hat{n}q \leq Q' + W \leq (2\hat{n}+1)q \\ 0, & \text{if } (2\hat{n}-1)q \leq Q' + W \leq 2\hat{n}q, \end{cases} \qquad (3)$$

where $\hat{n}$ is an integer. To elucidate the enrollment and evaluation phases, we provide four different examples in Table I, for $q = 1$.

Let $Q = n + \delta$, where $n = \lfloor Q \rfloor$ is an integer and $0 < \delta < 1$. For example, if $Q = 8.3$, then $n = 8$, $\delta = 0.3$ and $r = r_{ref} = 0$. We have the following observation:

Observation 1. *Let $Q = n + \delta$. Based on the given helper data W and the response r of a given arbitrary challenge c,*

TABLE I
EXAMPLE OF ENROLLMENT AND EVALUATION PHASE COMPUTATIONS
($q = 1$ AND $Q' = Q$)

r_{ref}	0	0	1	1
Q	8.3	8.7	9.3	9.7
$\bar{n}$	4	5	4	5
W	-0.8	0.8	-0.8	0.8
$Q' + W$	7.5	9.5	8.5	10.5
r	0	0	1	1

the parity[1] *of n and value of δ can be inferred as follows:*

1) n is even if $r = 0$, and vice versa. Otherwise, n is odd.
2) $\delta = |0.5 + W|$ if $W < 0$. Otherwise, $\delta = 1 - |W - 0.5|$.

The relationship between W and δ is formalized in Algorithm–1.

Algorithm 1 Compute_Delta_From_W

Input: W
Output: δ
1: **if** $W < 0$ **then**
2: $\quad \delta \leftarrow |0.5 + W|$
3: **else**
4: $\quad \delta \leftarrow 1 - |W - 0.5|$.

B. Notion of Security in the Context of ROPUF Cryptanalysis

As mentioned previously, one of the most important PUF properties is its *mathematical unclonability*. The essence of the unclonability of a PUF lies in the property that the challenge-response pairs (CRPs) should not be predictable by an adversary. We formally state the following notion of security in the context of PUF modeling:

Definition 1. ***[Security Notion]*** *[1] Let $P_{m,k}$ denote a PUF instance P with m-bit challenge and k-bit response. A PUF $P_{m,k}$ is considered to be secure if and only if there is no algorithm which can predict, for a given challenge c, the corresponding response r, under the following conditions:*

1) the accuracy of the prediction is greater than $\frac{1}{2^k}$,
2) the time complexity is less than $\mathcal{O}(2^m)$, and,
3) the data complexity, estimated in terms of the number of challenge-response pairs (CRP) needed for the attack, is less than $\mathcal{O}(2^m)$.

In the next section, we demonstrate that the above notion of security is violated by a cryptanalytic attack on the enhanced ROPUF.

III. SECURITY ANALYSIS

The main insight behind the proposed attack is the fact that the mathematical form of the Q-value (see (1)) suggests that each of the terms in the sum can be individually recovered by choosing a proper challenge. Without loss of generality, we assume that the interval length $q = 1$ in the rest of the paper. Let $Q_{i,j} = w_{ij} |f_i - f_j|^e, 1 \leq i, j \leq m, i \neq j$. The set of all values $Q_{i,j}$ can be considered to be a *basis* of the Q-space. Let $n_{i,j} = \lfloor Q_{i,j} \rfloor$, and $\delta_{i,j}$ be a real number, with $0 < \delta_{i,j} < q$ and $Q_{i,j} = n_{i,j}q + \delta_{i,j}$. Let $c_{i,j}$ denote the particular challenge $c = (0, 0, \ldots, 0, c_i = 1, 0, \ldots, 0, c_j = 1, 0, \ldots, 0)$, with $Q(i, j) = Q_{i,j}$ being its corresponding Q-value, and $r_{i,j}$ the corresponding response r. If $n_{i,j}$ is odd, then $r_{i,j} = 1$; otherwise $r_{i,j} = 0$.

For the sake of explanation, but without loss of generality, we consider the following challenge: $c = (c_1, \ldots, c_t, c_{t+1}, \ldots, c_m)$ where $c_1 = c_2 = \cdots = c_t = 1$, $3 \leq t \leq m$, and the remaining challenge bits being zeros. Thus,

$$Q(1, \ldots, t) = \sum_{i=1}^{t-1} \sum_{j=i+1}^{t} Q_{i,j} = \sum_{i=1}^{t-1} \sum_{j=i+1}^{t} (n_{i,j} + \delta_{i,j}) \quad (4)$$

For an arbitrary challenge c, not necessarily of the special form $c_{i,j}$, the relevant pieces of information are Q, W and r. We show that:

1) **if all $W_{i,j}$ and all $r_{i,j}$ values corresponding to of all challenges $c_{i,j}$s are available to the adversary**, then an algorithm can be developed to predict the corresponding response r of an arbitrary challenge c, with probability of success 1, and,
2) **if only the $r_{i,j}$ values for all challenges $c_{i,j}$ are available to the adversary**, then an algorithm can be developed to predict the corresponding response r of an arbitrary c, with probability of success greater than $\frac{1}{2}$.

A. Case I: Helper Data Available

By Observation–1, we know that the response r for a given challenge c is dependent only on the parity of n. Furthermore, the value of δ can be computed if W is known, by Algorithm–1.

Let $\delta = \sum_{i=1}^{t-1} \sum_{j=i+1}^{t} \delta_{i,j}$. In (4), since $\delta_{i,j}$ can be computed based on the helper data $W_{i,j}$, the adversary can calculate δ, and then decompose it in its integral and fractional parts as: $\delta = n_\delta + \delta'$, with $n_\delta = \lfloor \delta \rfloor$ and $0 \leq \delta' < 1$. Once we know n_δ, we know its parity. We define $n = \sum_{i=1}^{t-1} \sum_{j=i+1}^{t} n_{i,j} + n_\delta$, and then we have $Q(1, \ldots, t) = n + \delta'$. As mentioned above, in order to determine the corresponding r–value for Q, we only need to know the parity of n. Since the parity of $n_{i,j}$ can be derived from $r_{i,j}$ and the parity of n_δ is already known, the adversary can compute the parity of n, and then predict the response r based on the computed parity following the simple rule of Observation–1. The algorithm to predict the response r of a given challenge c of the form $c = (c_1, \ldots, c_t, c_{t+1}, \ldots, c_m)$ where $c_1 = c_2 = \cdots = c_t = 1, 3 \leq t \leq m$ is summarized in Algorithm–2. Note that this algorithm can be easily modified to predict the response for any arbitrary challenge c, with bits at arbitrary positions (not necessarily at the beginning of the challenge string and not necessarily contiguous) being 1, i.e. $c_{i_1} = c_{i_2} = \cdots = c_{i_t} = 1, 3 \leq t \leq m$, where $i_1, i_2, \ldots i_t$ are arbitrary indices.

[1]Parity indicates whether a number is even or odd.

Algorithm 2 Calculate_Response_W_Available

Input: c, $W_{i,j}$ values, $r_{i,j}$ values with $1 \leq i < j \leq t$. Here, challenge $c = (c_1, \ldots, c_t, c_{t+1}, \ldots, c_m)$ where $c_1 = c_2 = \cdots = c_t = 1, 3 \leq t \leq m$.

Output: r corresponding to c

```
for i = 1 to t − 1 do
    for j = i + 1 to t do
        δ_{i,j} ← Compute_Delta_From_W
        δ ← δ + δ_{i,j}
Decompose δ = n_δ + δ' where n_δ = ⌊δ⌋ and 0 ≤ δ' < 1
if n_δ ≡ 0 mod 2 then
    parity ← 0
else
    parity ← 1
for i = 1 to t − 1 do
    for j = i + 1 to t do
        parity ← parity ⊕ r_{i,j}
if parity == 0 then
    r ← 0
else
    r ← 1
```

Algorithm 3 Calculate_Response_W_Unavailable

Input: Challenge c where only $c_{i_1} = c_{i_2} = c_{i_3} = 1$, the responses $r_{i_1,i_2}, r_{i_1,i_3}, r_{i_2,i_3}$ of $c_{i_1,i_2}, c_{i_1,i_3}, c_{i_2,i_3}$, respectively.

Output: Response r

```
if r_{i1,i2} ⊕ r_{i1,i3} ⊕ r_{i2,i3} == 0 then
    r ← 0
else
    r ← 1
```

Complexity Analysis: For a given challenge, the *data complexity* of the proposed cryptanalytic attack is determined by the additional information (e.g. helper data and response for other challenges) required. So, it is evident that Algorithm–2 has data complexity $\mathcal{O}(m^2)$, and has time complexity $\mathcal{O}(t^2)$.

B. Case-II: Helper Data Not Available

The cryptanalysis is considerably more challenging if the adversary cannot access the helper data database. In this section, we develop a cryptanalytic attack such that based on a given set of CRPs, the adversary would be able to predict the corresponding response r of a given challenge c, with a success probability greater than $\frac{1}{2}$. The main insight for this attack is the following: given a challenge c, the parity of the integer part of the corresponding Q-value, along with the sum of the fractional parts of the constituent $Q_{i,j}$ terms, leak information about the value of the corresponding response r. It is essentially a *divide-and-conquer* strategy.

For the sake of explanation, we develop an algorithm which predicts the response r of c where $c_1 = c_2 = c_3 = 1$ and remaining challenge bits are zero or $t = 3$. The corresponding Q-value:

$$\begin{aligned} Q(1,2,3) &= Q_{1,2} + Q_{1,3} + Q_{2,3} \\ &= n_{1,2} + n_{1,3} + n_{2,3} + \delta_{1,2} + \delta_{1,3} + \delta_{2,3}, \end{aligned} \tag{5}$$

where $n_{i,j} = \lfloor Q_{i,j} \rfloor$ are the integer values and $\delta_{i,j} = Q_{i,j} - n_{i,j}$ are the fractional values. Define $n = n_{1,2} + n_{1,3} + n_{2,3}$ and $\delta = \delta_{1,2} + \delta_{1,3} + \delta_{2,3}$, and assume that the parities of $n_{1,2}, n_{1,3}$ and $n_{2,3}$ are derived from the responses $r_{1,2}, r_{1,3}$, and $r_{2,3}$ (by Observation-1). Then, the parity of n can be computed based on the parities of $n_{1,2}, n_{1,3}$ and $n_{2,3}$. Without loss of generality, assume that n is an even number. Since $\delta = \delta_{1,2} + \delta_{1,3} + \delta_{2,3}$ and $0 \leq \delta_{1,2}, \delta_{1,3}, \delta_{2,3} < 1$, $0 \leq \delta < 3$.

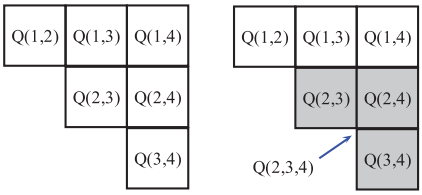

Fig. 3. Two possible decompositions of $Q(1,2,3,4)$

The attack depends on the fact that although we do not know for certain which among these three possible intervals (of size $q = 1$) δ lies in, since we know for certain the parity of n, we can predict one of the two possible values of r with probability greater than $\frac{1}{2}$. For example, in the case above if n is known to be an even number, both $0 \leq \delta < 1$ or $2 \leq \delta < 3$ would result in $r = 0$, and $1 \leq \delta < 2$ would imply $r = 1$. In other words, the adversary has a scheme by which she can predict the corresponding response r for a given challenge c, with probability of success $\frac{2}{3}$, which is greater than $\frac{1}{2}$. Hence, by the notion of security described in Section II-B, it is a successful cryptanalysis. This cryptanalysis scheme is formally generalized in Algorithm–3 for any challenge having three non-zero challenge bits.

It is worth mentioning that adversary cannot develop any such scheme as mentioned in Algorithm–3 for $t = 4$ due to the absence of a decomposition of $Q(1,2,3,4)$ into odd number of components.

The two possible decompositions of $Q(1,2,3,4)$ are shown in Fig. 3 and Observation–2:

Observation 2. *When $t = 4$, there is no decomposition of $Q(1,2,3,4)$ having odd number of components. Two possible decomposition are as follows:*

1) $Q(1,2,3,4) = \sum_{i=1}^{3} \sum_{j=i+1}^{4} Q_{i,j}$, *and*

2) $Q(1,2,3,4) = \sum_{j=2}^{4} Q_{1,j} + Q(2,3,4)$.

Note that none of these two decompositions have an odd number of components. It implies that the adversary has no theoretical advantage.

On the other hand, if $t > 4$, the adversary can develop an attack scheme to predict the response r of a given challenge c with prediction accuracy greater than $\frac{1}{2}$ based on Observation–3 and Observation–4 given below:

Observation 3. *If the parity of t is odd with $t > 4$, then $Q(1,\ldots,t)$ can be decomposed into an odd number of components as follows:*

$$\begin{aligned} Q(1,\ldots,t) &= \sum_{j=2}^{t} Q_{1,j} + Q(2,\ldots,t) \\ &= \sum_{j=2}^{t} (n_{1,j} + \delta_{1,j}) + Q(2,\ldots,t). \end{aligned} \tag{6}$$

Let us define $Q(2,\ldots,t) = n_{t-1} + \delta_{t-1}$, then (6) can be rewritten as:

$$\begin{aligned} Q(1,\ldots,t) &= \sum_{j=2}^{t} (n_{1,j} + \delta_{1,j}) + n_{t-1} + \delta_{t-1} \\ &= \sum_{j=2}^{t} n_{1,j} + n_{t-1} + \sum_{j=2}^{t} \delta_{1,j} + \delta_{t-1} = n_t + \delta_t, \end{aligned} \tag{7}$$

where $n_t = \sum_{j=2}^{t} n_{1,j} + n_{t-1}$ and $\delta_t = \sum_{j=2}^{t} \delta_{1,j} + \delta_{t-1}$.

Observation 4. *If the parity of t is even with $t > 4$, then $Q(1,\ldots,t)$ can be decomposed into an odd number of components as follows:*

$$\begin{aligned} Q(1,\cdots,t) &= \sum_{j=2}^{t} Q_{1,j} + \sum_{j=3}^{t} Q_{2,j} + \sum_{j=4}^{t} Q_{3,j} \\ &\quad + Q(4,\ldots,t) \\ &= \sum_{j=2}^{t} (n_{1,j} + \delta_{1,j}) + \sum_{j=3}^{t} (n_{2,j} + \delta_{2,j}) \\ &\quad + \sum_{j=4}^{t} (n_{3,j} + \delta_{3,j}) + Q(4,\ldots,t). \end{aligned} \tag{8}$$

Let us define $Q(4,\ldots,t) = n_{t-3} + \delta_{t-3}$, then (8) can be rewritten as:

$$\begin{aligned} Q(1,\ldots,t) &= \sum_{j=2}^{t} (n_{1,j} + \delta_{1,j}) + \sum_{j=3}^{t} (n_{2,j} + \delta_{2,j}) \\ &\quad + \sum_{j=4}^{t} (n_{3,j} + \delta_{3,j}) + n_{t-3} + \delta_{t-3} \\ &= n_t + \delta_t, \end{aligned} \tag{9}$$

where $n_t = \sum_{j=2}^{t} n_{1,j} + \sum_{j=3}^{t} n_{2,j} + \sum_{j=4}^{t} n_{3,j} + n_{t-3}$ and $\delta_t = \sum_{j=2}^{t} \delta_{1,j} + \sum_{j=3}^{t} \delta_{2,j} + \sum_{j=4}^{t} \delta_{3,j} + \delta_{t-3}$.

In summary, the adversary predicts the response of challenge c with only $c_i = 1, 1 \le i \le t$ by employing (7) and Algorithm–4 when parity of t is odd, and (9) and Algorithm–5 when t has even parity. The theoretical prediction success rate for response r according to Algorithm–4 and Algorithm–5 are $\frac{1}{2} + \frac{1}{2t}$ and $\frac{1}{2} + \frac{1}{2(3t-5)}$, respectively. Algorithm–4 and Algorithm–5 can be generalized for any challenge c, where only challenge bits $c_{i_1} = \cdots = c_{i_t} = 1$ as the Algorithm–3, $1 \le i_1 \ne \cdots \ne i_t \le m, 5 \le t \le m$.

Algorithm 4 Predict_Response_Odd_t (c)

Input: Challenge c where only t bits $c_1 = c_2 = \ldots = c_t = 1$, the responses $r_{1,2}, r_{1,3}, \ldots, r_{1,t}$ corresponding to $c_{1,2}, c_{1,3}, \ldots, c_{1,t}$, respectively, and r_{t-1} for challenge $(0, c_2, \ldots, c_t, 0, \ldots, 0)$ which has value $Q(2,\ldots,t)$.
Output: Response r corresponding to c

```
if r_{1,2} ⊕ r_{1,3} ⊕ ··· ⊕ r_{1,t} ⊕ r_{t-1} == 0 then
    r ← 0
else
    r ← 1
```

Algorithm 5 Predict_Response_Even_t (c)

Input: Challenge c where only t bits $c_1 = c_2 = \cdots = c_t = 1$; responses $\{r_{1,2}, r_{1,3}, \ldots, r_{1,t}\}$, $\{r_{2,3}, r_{2,4}, \ldots, r_{2,t}\}$, $\{r_{3,4}, r_{3,5}, \ldots, r_{3,t}\}$ corresponding to $\{c_{1,2}, c_{1,3}, \ldots, c_{1,t}\}$, $\{c_{2,3}, c_{2,4}, \ldots, c_{2,t}\}$, $\{c_{3,4}, c_{3,5}, \ldots, c_{3,t}\}$, respectively, and r_{t-3} for challenge $(0,0,0,c_4 = 1,\ldots,c_t = 1,0,\ldots,0)$ which has value $Q(4,\ldots,t)$.
Output: Response r corresponding to challenge c

```
if r_{1,2} ⊕ ··· ⊕ r_{1,t} ⊕ r_{2,3} ⊕ ··· ⊕ r_{2,t} ⊕ r_{3,4} ⊕ ··· ⊕ r_{3,t} ⊕ r_{t-3} == 0 then
    r ← 0
else
    r ← 1
```

Complexity Analysis: Table II summarizes the complexity and prediction accuracy of Algorithm–4 and Algorithm–5. Here, time complexity is the number of XOR operations required. Hence, both Algorithm–4 and Algorithm–5 have linear data and time complexities.

TABLE II
COMPLEXITY DETAILS

Algorithm	Complexity		Theoretical Prediction Accuracy
	Time	Data	
Algorithm–4	$t-1$	t	$\frac{1}{2} + \frac{1}{2t} > 1/2$
Algorithm–5	$3t-6$	$3t-5$	$\frac{(3t-6)/2+1}{3t-5} = \frac{1}{2} + \frac{1}{2(3t-5)}$

IV. EXPERIMENTAL VALIDATION

In this section, we discuss the experimental validation of the cryptanalysis scheme of robust ROPUF without helper data, as described in Section III-B. It was observed that the sum of the δ values of all components of a $Q(i_1,\ldots,i_t)$ decomposition (see Equations (7) and (9)) lies in one of the possibly odd number intervals, except when $t = 4$. Experimentally, we show that there is indeed a bias when δ values are distributed over odd number of intervals, and the bias is comparable with the theoretical value. In our experiment we have used the publicly

TABLE III
DISTRIBUTION OF δ OVER UNIT INTERVALS WITH EVEN INDICES [20 DIFFERENT DATASETS] (%)

Subset Size (t)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**10**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
3	67.078	67.03	67.082	66.969	66.966	67.021	67.013	67.064	66.944	66.996	66.959	66.912	66.847	66.918	66.968	67.099	67.073	66.998	66.926	67.019
4	50.004	50.054	50.055	50.09	50.032	50.047	50.023	50.091	50.004	50.031	50.127	50.079	50.028	50.023	50.078	50.093	50.011	50.058	50.111	50.021
5	56.688	56.736	56.756	56.794	56.784	56.799	56.856	56.859	56.775	56.691	56.819	56.871	56.693	56.646	56.864	56.818	56.84	56.8	56.737	56.671
6	50.165	50.094	50.236	50.084	50.182	50.307	50.171	50.099	50.225	50.149	50.315	50.21	50.311	50.248	50.199	50.038	50.115	50.193	50.272	50.168

TABLE IV
THEORETICAL BIAS VS. AVERAGE OBSERVED BIAS

t	**Theoretical Bias (%)**	**Average Observed Bias(%)**
3	(2/3)*100 = 66.66	66.99
4	(2/4)*100 = 50.00	50.05
5	(3/5)*100 = 60.00	56.77
6	(7/13)*100 = 53.84	50.18

available RO-frequency dataset [15], [16] for Xilinx Spartan (XC3S500E) FPGAs. It consists of 100 frequency samples for each of 512 ROs on 193 FPGAs, but for our experiment we have considered one frequency sample of frequency dataset for FPGA board with chip id "D059546" (first column of the file "D059546.csv"). The Q–value is computed as: $Q_{i,j} = w_{i,j}|f_i - f_j|^e$, where $w_{i,j}$ is the Euclidean distance between CLB-coordinates of i-th and j-th ROs and $e = 0.5$.

The distribution of δ over even indexed intervals of unit length is reported in Table–III, for subset sizes t=3, . . . , 6. The Q–values $Q(1,2,3), Q(1,2,3,4), Q(1,2,3,4,5), Q(1,2,3,4,5,6)$ for challenges $(1,1,1,0,\ldots,0)$, $(1,1,1,1,0,\ldots,0)$, $(1,1,1,1,1,0,\ldots,0)$, $(1,1,1,1,1,1,0,\ldots,0)$, respectively, were computed according to the following decompositions:

$$\begin{aligned}
Q(1,2,3) &= Q(1,2) + Q(1,3) + Q(2,3)\\
Q(1,2,3,4) &= Q(1,2) + Q(1,3) + Q(1,4) + Q(2,3,4)\\
Q(1,2,3,4,5) &= Q(1,2) + Q(1,3) + Q(1,4) + Q(1,5) + Q(2,3,4,5)\\
Q(1,2,3,4,5,6) &= Q(1,2) + Q(1,3) + Q(1,4) + Q(1,5) + Q(1,6) + Q(2,3)\\
&\quad + Q(2,4) + Q(2,5) + Q(2,6) + Q(3,4) + Q(3,5) + Q(3,6)\\
&\quad + Q(4,5,6). \qquad (10)
\end{aligned}$$

We have used an in-house *Matlab* script to compute the δ_t values for all Q–values in (10) using (7) and (9). For each t, we executed 20 different tests, and in each test the δ_t is computed 400000 times using different subsets of size t. From Table–III it is evident that there is a bias in the distribution of δ values obtained from experimental dataset, over the range $2n < \delta_t < 2n + 1$, $n \in \mathbb{N}_0$. Table–IV shows a comparison between theoretical and average empirical bias, and empirical bias can be observed to be close to theoretical bias. Also, from Table III, it is evident that the empirical bias for δ decreases as t increases, and the bias is higher for odd values of t compared to even values of t. Both these observations are in agreement with the predicted trends.

V. CONCLUSION

In this paper, we have shown that the robust ROPUF with enhanced CRP set proposed in [13] is not secure with respect to certain cryptanalytic attacks. If the adversary has access to some specific CRPs with the corresponding helper data, she can efficiently predict the response r for an arbitrary challenge c, with probability of success 1. We have also shown that responses to unseen challenges can be predicted with probability of success better than random guess, even if the helper data is not available to the adversary. We conclude that the two most important aspects of the design proposed in [13], i.e, the subset selection and helper data to improve challenge space size and robustness of ROPUF, turn out to be weaknesses from a cryptanalytic perspective.

REFERENCES

[1] R. S. Pappu, "Physical one-way functions," Ph.D. dissertation, Massachusetts Institute of Technology, March 2001.

[2] D. Lim, "Extracting Secret Keys from Integrated Circuits," Master's thesis, MIT, USA, 2004.

[3] M. Majzoobi and F. Koushanfar, "Time-Bounded Authentication of FPGAs," in *IEEE TIFS*, vol. 6, no. 3, 2011, pp. 1123–1135.

[4] S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *Proc. of IEEE Int. Symposium on HOST* , June 2008, pp. 67–70.

[5] M.-D. M. Yu, D. M'Raïhi, R. Sowell, and S. Devadas, "Lightweight and Secure PUF Key Storage Using Limits of Machine Learning," in *Proc. of CHES* , vol. 6917. Springer Berlin / Heidelberg, 2011, pp. 358–373.

[6] C. Brzuska, M. Fischlin, H. Schrder, and S. Katzenbeisser, "Physically Uncloneable Functions in the Universal Composition Framework," in *Advances in Cryptology (CRYPTO)*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6841, pp. 51–70.

[7] F. Armknecht, R. Maes, A.-R. Sadeghi, B. Sunar, and P. Tulys, "Memory Leakage-Resilient Encryption Based on Physically Unclonable Functions," in *Proc. of ASIACRYPT*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 685–702.

[8] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *Proc. of Information Hiding*, ser. Lecture Notes in Computer Science, 2009, vol. 5806, pp. 206–220.

[9] J. Rajendran, G. S. Rose, R. Karri, and M. Potkonjak, "Nano-ppuf: A memristor-based security primitive," in *IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, 2012, pp. 84–87.

[10] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF Modeling Attacks on Simulated and Silicon Data," *IEEE Transactions on Information Forensics and Security*, vol. 8, pp. 1876–1891, 2013.

[11] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Design Automation Conference*. New York, NY, USA: ACM Press, 2007, pp. 9–14.

[12] A. Maiti and P. Schaumont, "Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive," *Journal of Cryptology* , vol. 24, pp. 375–397, 2011.

[13] A. Maiti, I. Kim, and P. Schaumont, "A Robust Physical Unclonable Function With Enhanced Challenge-Response Set," *IEEE TIFS*, vol. 7, no. 1, pp. 333 –345, feb. 2012.

[14] J. paul Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates," in *In AVBPA 2003*, 2003, pp. 393–402.

[15] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of RO-PUF," in *Proc. of IEEE HOST*. IEEE Computer Society, 13-14 June 2010, pp. 94 –99.

[16] Ring oscillator dataset. [Online]. Available: http://rijndael.ece.vt.edu/puf/download.html