An Efficient Grouping Method and Error Probability Analysis for RO-PUFs

Giray Kömürcü
National Research Institute of Electronics and Cryptology,
TÜBİTAK, 41470, Kocaeli, Turkey. Email: giray.komurcu@tubitak.gov.tr
Ali Emre Pusane, Günhan Dündar
Bogazici University, Dept. of Electrical and Electronics Eng.
34342 Bebek, Istanbul, Turkey. Email: {ali.pusane, dundar}@boun.edu.tr

Abstract

Physical Unclonable Functions (PUFs) are primitives that have wide usage areas in information security. Ordering based Ring Oscillator (RO)-PUFs have been introduced recently to overcome the robustness and area efficiency issues related to PUF implementations. With this approach, 100% robust outputs are generated, providing a solution for cryptographic key generation. High entropy extraction with relatively few ROs is also achieved, resulting in high area utilization of the PUF circuit. Frequency threshold determination is the most critical step in ordering based RO-PUFs, and determines a trade-off between area efficiency and robustness. In this work, we overview an efficient grouping method for RO-PUFs and analyze the error vulnerability of PUFs based on the frequency threshold determination. Next, we analyze the length of groups used in such PUF circuits and determine the symbol error probability. In addition to these, we demonstrate the relationship between the symbol error probability and bit error probability. We also investigate the bit error probability based on the wrong determination of the frequency threshold in ordering based RO-PUFs. Finally, a trade-off between area usage and robustness is presented for identification applications.

Keywords

PUF, Physical Unclonable Functions, Reliability, Robustness, Ring Oscillator, FPGA

I. Introduction

Physical unclonable functions (PUFs) are powerful techniques for addressing security problems. They have a wide range of applications, including cryptographic key generation and storage, authentication, ID generation, and IP protection. PUFs offer new, cheap, and highly secure solutions in these areas with their ability to generate chip specific outputs on the fly. Another advantage of PUF circuits is their adaptability to FPGAs. Due to widespread use of FPGAs, we focus on PUF structures that are suitable for FPGA implementation.

PUF notion was introduced by Pappu $et\ al.$ in 2001 [1]. The first PUF structure used the unique reflection of light from a bubble filled epoxy spread over an integrated circuit and is called Optical PUF [1], [2]. Next, a similar structure was proposed in the name of Coating PUF [3]. Due to impractical usage and expensive equipment requirements of these PUF structures, silicon PUFs became more popular with their low cost and ease of integration. Unique intrinsic physical properties of ICs, such as oxide thickness, threshold voltage, and doping concentration provide the capability of generating chip specific signatures on silicon devices.

Ring Oscillator (RO) PUFs, Arbiter PUFs, SRAM PUFs, Butterfly PUFs, and Glitch PUFs [4]–[9], are common silicon PUF types developed within the last decade. Among these, RO-PUFs are the most convenient type for FPGA implementation and work more reliably under changing environmental conditions [10]–[12]. The output generation mechanism of RO-PUFs mainly depends on the oscillation frequency comparison of ROs. In conventional systems, frequencies of two ROs are compared and one bit output is generated depending on the comparison outcome [13]. Even though this approach is very easy to implement, it does not effectively use the entropy present in the system, resulting in an area costly solution. In addition to this, the generated outputs are noisy, and robustness can be achieved up to a certain level, especially under changing environmental conditions. In order to extract the maximum entropy from the system and generate 100% robust outputs, an ordering based RO-PUF technique was introduced in [14]. In this method, ROs, whose frequencies are adequately apart from each other, are grouped together and PUF output is generated based on frequency ordering of ROs, which can generate up to $\lfloor \log_2(N!) \rfloor$ bits using N ROs [13]. Choosing ROs with their frequencies far enough from each other maintains reliability. To determine the groups from the whole RO set, two methods are proposed. In the first method, Longest Increasing Subsequence based Algorithm (LISA) is developed and each RO is measured under extreme conditions, such as the highest and lowest operating temperatures and a parameter called frequency threshold (f_{th}) is determined to overcome the noise present in the system [14]. In the second method, Dynamic Programming (DP) is employed for grouping and a more conservative f_{th} is determined (called pre-determined frequency threshold, f_{thp}) and frequencies of ROs only measured at normal operating conditions are used [15]. In this method, f_{thp} is basically the minimum frequency distance of ROs within each group to protect the system from changes in ordering, due to environmental variations and noise. The advantage of the second method over the first method is its lower computational complexity and elimination of the need for RO frequency measurements at extreme temperatures, which simplifies the registration phase significantly. Even though the cited works present the ordering based RO-PUFs well and emphasize their importance, robustness of the structures are not analyzed for the case of incorrect f_{thp} determination.

In this work, our main aim is to determine the error probability of ordering based RO-PUFs. For this purpose, we first analyze the error vulnerability of the ordering based RO-PUF system due to f_{thp} determination issues in Section II. DP approach, which was originally proposed in [15], is briefly reviewed in Section III. Next, the

distribution of groups formed, based on their lengths for various values of f_{thp} is analyzed in Section IV. In addition to this, symbol error probability, which is the probability of erroneous ordering with respect to the reference, is calculated analytically and validated experimentally. Then, the relation between the symbol error probability and the output bit error probability is investigated and bit error probability is calculated analytically in the same section. Area usage vs. robustness is presented in Section V for identification applications based on PUF responses. Finally, Section VI concludes the paper.

II. Error vulnerability analysis based on f_{thp} determination issues

f_{thp} determination is the most critical step in ordering based systems. An ideal f_{thp} value should guarantee the robustness of the PUF outputs by grouping the ROs, whose frequencies are adequately apart from each other, while maintaining the largest group sizes to extract the maximum entropy present in the system. Determining the f_{thp} value smaller than the ideal value increases the group sizes, but lowers the robustness of the system. If the distance of RO frequencies in the same group is smaller than the amount of noise and fluctuations in the environmental conditions, such as process variations, temperature, and supply voltage, robustness of the system may be compromised. This situation creates so-called problematic RO pairs, which have the potential for generating errors in the output. On the other hand, determining the f_{thp} value higher than the required value lowers the entropy extraction from the system by forming smaller groups, whereas guaranteeing higher reliability of the outputs. A detailed description of how the f_{thp} value is determined is presented in [15] and reviewed in Section III.

The main idea of the method presented in [15] is to measure a subset of all circuits, for instance 50 samples, at extreme conditions and detect the maximum frequency deviation within the RO pairs in each sample. In order to guarantee robustness, the f_{thp} value should be chosen higher than the maximum frequency deviation, maintaining the validity of the ordering within the group even under extreme conditions. This is illustrated in Example 1.

Example 1: 180 ROs are implemented on a Xilinx 3S5000 FPGA board and their oscillation frequencies are measured at 20^{o}C and 100^{o}C. As seen from Figure 1, all ROs become slower when the IC temperature increases, whereas the RO frequency deviations with temperature are shown in Figure 2. For instance, RO_{63} slows down by 8.35 MHz, and RO_{115} slows down by 9.335 MHz when the temperature increases from 20^{o}C to 100^{o}C. Since these ROs are the ones with the highest and lowest frequency deviations, the f_{thp} value should be determined by considering the difference between the frequency deviations of these ROs. This experiment is repeated for 5 different FPGAs to determine a valid f_{thp} value. In this case, adding a certain safety margin to the measured maximum frequency deviation to compensate for the noise in the system will result in an f_{thp} value of approximately 1 MHz to guarantee robustness.

Since the f_{thp} value determines the level of robustness, an error vulnerability analysis is required for varying f_{thp} values, which is presented in Section V.

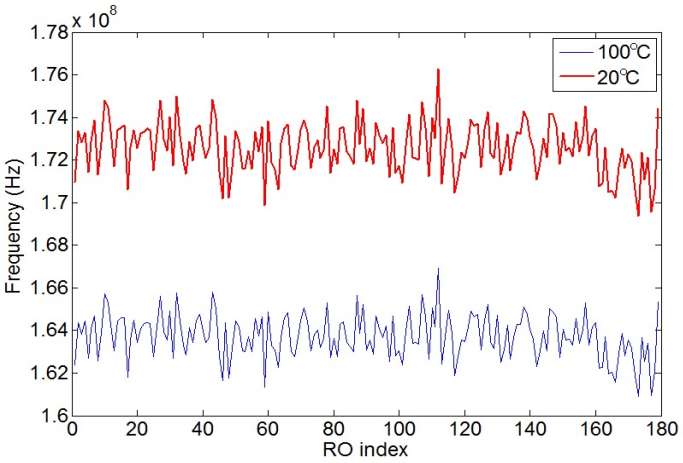

Fig. 1. Frequency of ROs measured at 20^{o}C and 100^{o}C

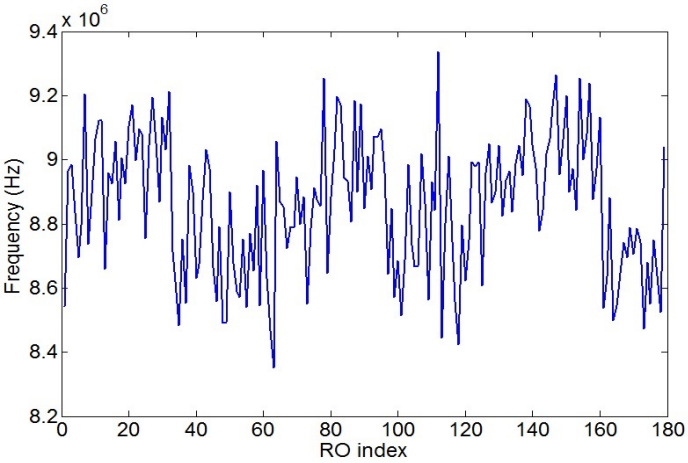

Fig. 2. Frequency deviation of ROs

III. Dynamic Programming Based Grouping Algorithm

DP is employed to solve the problem of grouping in ordering based RO-PUFs with guaranteed reliability in [15]. With this approach, it is possible to form the largest groups to obtain the highest amount of entropy from a certain number of ROs, with minimum computational complexity. In this method, the frequencies of ROs, measured under normal operating conditions, and f_{thp} are the inputs of the algorithm. The output of the algorithm is the list of ROs in each group, which will then be used to generate the output bit stream.

In the first step of DP, a sorted list of ROs, $Fsorted[n]$, is created according to their frequencies, where n is the number of ROs implemented in the system. Then, the nearest RO with a frequency of at least f_{thp} higher is found for each RO and a linked list, $list[n]$, is created. In the third step, groups are formed according to the requirements of maximum entropy and 100% robustness using $list[n]$. The algorithm groups RO_1 with RO_j, which is the one that $list[1]$ points to. Then, the algorithm jumps to the position j in the $list$ vector and groups the one that $list[j]$ shows. This ends when the last position is reached in the linked list. The first group is formed with the first run and

the group members are locked against further groupings. The procedure is repeated until all ROs are grouped. If an RO that the list shows is already grouped, the nearest ungrouped RO towards the end of list is grouped instead. The pseudo code of the DP approach is presented in Algorithm 1 and explained in Example 2.

Example 2: An RO set of 12 ROs is created as the first step and their frequencies are placed into an array, $FreqRO[n]$. In the second step, a sorted list of RO frequencies is created, $Fsorted[n]$. Then, a linked list, $list[n]$, is created using an f_{thp} value of 1.5 MHz, which shows the first avaliable RO to be placed in the same group as the nth RO. As the last step, groups are formed one by one, and ROs placed in a group are removed from $list[n]$. This step is repeated until all ROs are grouped. After the algorithm is applied, 3 distinct groups are formed, satisfying the reliable PUF output generation conditions with maximum entropy extraction. The first group is composed of six ROs, which are $RO_1, RO_3, RO_6, RO_7, RO_{10},$ and RO_{12}. In this group, a total of 720 different orderings may occur and hence 9 bits of output can be generated. The second group is composed of four ROs, which are RO_2, RO_4, RO_8, and RO_{11}. In this group, a total of 24 different orderings may occur and hence 4 bits of output can be generated. The third group is composed of two ROs, which are $RO_5, andRO_9$. In this group, a total of 2 different orderings may occur and hence 1 bit of output can be generated. With such a system of 12 ROs, 15 bits of output can be generated with the proposed approach. This process is illustrated in Figure 3.

Data:
1. A linked list of ROs with their frequencies measured under normal operating conditions, $FreqRO[n]$.
2. f_{thp} for robustness
Result: Groups of ROs.
Sort $FreqRO[n]$ by frequency in increasing order: $Fsorted[n]$
for $i \leftarrow 1$ **to** $n-1$ **do**
  find the nearest element $Fsorted[j]$ that is
  ($Fsorted[i] < Fsorted[j]$-f_{thp}) and link i to j in $list[n]$
end
$i = 1$
while *ungrouped RO exists* **do**
  if ROi *is ungrouped* **then**
    Add ROj to the group of ROi
    Jump to $ROj(i = j)$
  end
  if ROi *is grouped* **then**
    Increment i until ROi is ungrouped
  end
  if *i=n and still ungrouped RO exists* **then**
    $i = 1$
  end
end

Algorithm 1: Dynamic Programming approach in pseudo code

An advantage of DP over LISA is its low computational complexity. The reduced complexity of the proposed DP approach is a result of avoiding the redundant RO search done by LISA as shown in [15]. Even though our

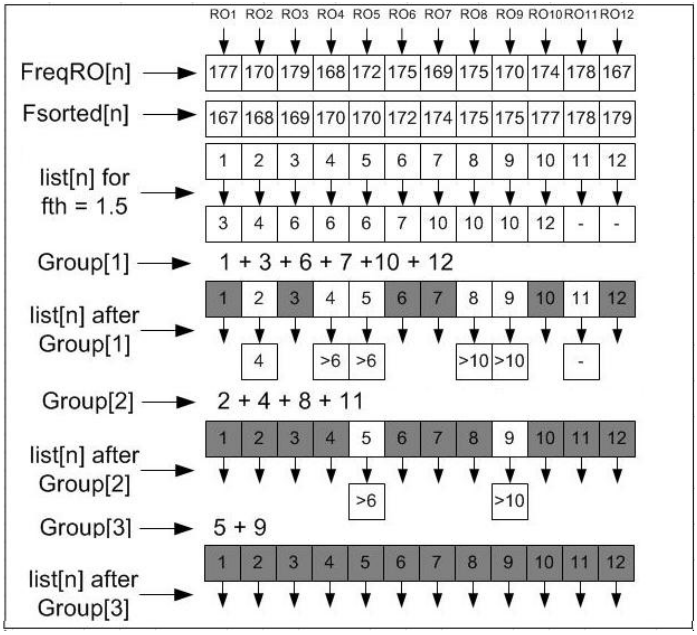

Fig. 3. DP sample execution for 12 elements

error probability analysis is mainly based on the method presented in [15], it can be directly applied to the method explained in [14].

In both methods, grouping step is applied once for each PUF sample in the registration phase and grouping information is stored either on an NVM in the circuit, or on a server. If the information is stored on a server and sent to the circuit as a challenge during the use of the PUF, information leakage may occur, which threatens the security of the device. This situation is analyzed and different solutions are proposed in [16].

Even though the ordering based RO-PUFs generate 100% robust outputs using the f_{thp} parameter, their uniqueness property is still questionable. For this purpose, RO frequencies from 25 FPGA samples are collected and outputs are generated based on the DP approach. In the ideal case, Hamming distance (HD) of the outputs should be 0.5 for uniqueness, since a random distribution is required. Then, HD of the outputs is calculated as 0.498, which is very close to the ideal value, proving the uniqueness quality of the PUF responses.

IV. Systematic Analysis of Bit Error Probability

Even though f_{thp} detection, presented in Section II, gives an idea about the error vulnerability of the system, it does not fully express the bit error probability that will result in the output. Since erroneous outputs are acceptable upto a certain degree in some applications that utilize PUF circuits, a bit error probability analysis is required. The analysis of bit errors due to problematic RO pairs being placed in the same group is complicated, since it is closely related to group lengths, symbol error probability, and symbol error to bit error conversion. In the following subsections, these issues are addressed.

A. Group length analysis

In ordering based RO-PUFs, ROs are grouped according to the selected f_{thp} value and each group generates a certain number of output bits according to the conversion method selected. If the ordering in a group changes due to one of the problematic RO pairs, a symbol error occurs and some of the output bits become erroneous. Since the length of the group that generates the symbol error affects the number of erroneous bits, an analysis on group lengths is required for a complete bit error probability estimation. One way of determining the group lengths is to measure a large number of samples and calculate the average number of groups per size. But, since it is impractical to build and measure many samples, a large synthetic data set can be created using mean and standard deviation (STD) of RO frequencies measured from a sample system. Then, DP is applied to the data set and group lengths are analyzed according to the selected f_{thp} values. This is explained in Example 3.

Example 3: 25 sample RO-PUFs of 160 ROs each are implemented, and mean and STD of RO frequencies are calculated. Then, 10,000 sets of 160 RO frequencies are generated by MATLAB based on the calculated mean and STD. The reason for using 160 ROs for each RO-PUF is that they can generate appoximately 128 bits of 100% robust output by using ordering based methods, which is adequate for many applications, such as AES encryption. DP approach is used to determine the lengths of groups formed with respect to the selected f_{thp} value. A range of 0.5 MHz to 1.1 MHz is used for the f_{thp} value, since 1 MHz seems to be the optimum value for this case. A smaller value of f_{thp} allows ROs with frequencies closer to each other to be placed in the same group, hence larger groups are more likely to be formed. On the other hand, larger f_{thp} values limit the sizes of groups, allowing only those ROs with frequencies far apart from each other to be in the same group. As can be seen from Figure 4, among 10,000 sets, the largest group formed with an f_{thp} value of 1.1 MHz involves 11 ROs (This group size appears in 11 sets among 10,000), whereas the largest group formed with an f_{thp} value of 0.5 MHz involves up to 19 ROs (This group size appears in 4 sets among 10,000). An interesting result observed from this analysis is the number of groups with a single RO, which do not contribute to output generation by default. With an f_{thp} value of 1.1 MHz, approximately 10 ROs could not be grouped with any other ROs on the average. This number becomes smaller than 4, when the f_{thp} value is 0.5 MHz.

B. Symbol error probability and validation

As discussed in Section II, if the selected f_{thp} value is less than the ideal f_{thp} value, problematic RO pairs arise, which have the possibility to create symbol errors. The conditions for a problematic RO pair to create a symbol error are stated as follows:

1. Both ROs of a problematic RO pair should be in the same group and next to each other in frequency ordering.

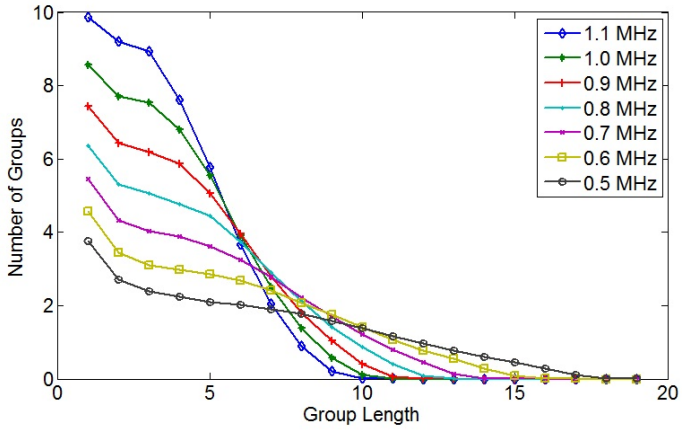

Fig. 4. Number of groups per group length for different f_{thp} values

[1]

2. ROs of a problematic RO pair should be in the correct order. [2]

3. Environmental changes and noise should be large enough to create the symbol error.

ROs in the same group affect the output bit generation. If the problematic ROs are distributed to different groups or placed into the same group but are not next to each other in the frequency ordering, ordering in any group does not change and a symbol error is not created. The symbol error probability is closely related to the number of groups formed as well as the group lengths. A few groups with higher number of ROs tend to have a higher probability of resulting symbol error from problematic RO pairs. In the extreme case, if the RO-PUF has a single group with all the ROs included, a symbol error will definitely occur from a problematic RO pair, when certain environmental conditions occur. The probability of an RO pair to be placed next to each other in a group of length s, when the total number of ROs in the PUF is M, can be stated as

$$p_s = 2 * (s-1)/(M * (M-1)). \tag{1}$$

Based on this formula, the probability of an RO pair to be placed in a group of size 2, in a system of 160 ROs, can be calculated as 7.86×10^{-5}. This probability increases up to 1.1×10^{-3} if the group size is 15, as shown in Figure 5. Since the distribution and length of groups depends on the f_{thp} value, this value determines the probability of placing the problematic RO pair in the same group. Let k_s be the average number of groups with size s and m be the size of the largest group. Then, this probability can be calculated as

[1]Assuming that the selected f_{thp} value is greater than half of the ideal f_{thp} value. Otherwise, ROs that are not next to each other may create symbol errors as well. This case is disregarded for the sake of simplicity.

[2]If the frequency of RO_1 is less than the frequency of RO_2, but RO_1 becomes faster than RO_2, a symbol error occurs. Then, this situation is called the correct order for symbol error creation. On the other hand, if RO_1 slows down more than RO_2, the frequency order does not change even if the frequency deviation is less than the f_{thp} value, and a symbol error is not created.

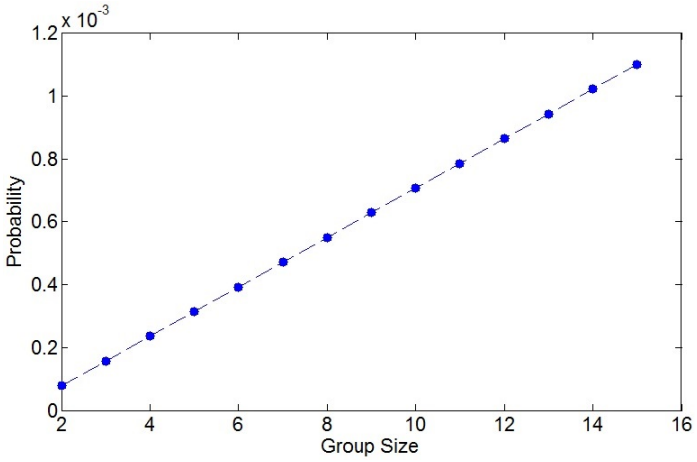

Fig. 5. Probability of two ROs to be placed in the same group

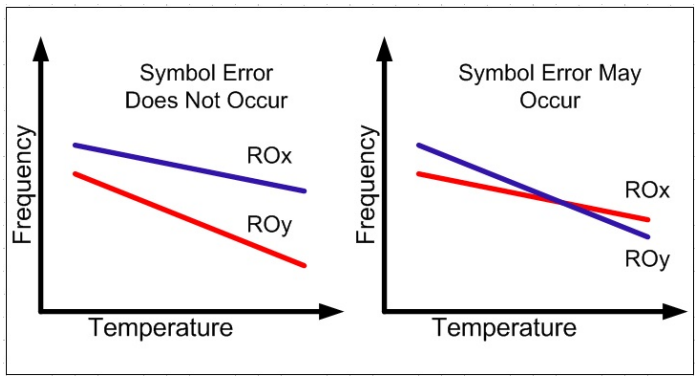

Fig. 6. Effect of ordering on symbol error [11]

$$p_{av} = \sum_{s=2}^{m} k_s * p_s. \tag{2}$$

Even though the problematic RO pair is placed in a group, this does not mean that a symbol error will occur when the required environmental conditions are realized. The RO pair should be in the correct order for a symbol error creation. As explained in Section II, frequency deviation of ROs differ with changing temperature or supply voltage. Considering an RO pair with frequency difference smaller than the ideal f_{thp} value, if the slower RO slows down more than the faster RO when the environmental fluctuations occur, a symbol error does not occur, since the ordering does not change. This is illustrated in Figure 6. Since two different orderings may occur within the RO pair with equal probability, probability of the correct order is 0.5.

When the problematic RO pair is placed in a group with the correct order, a symbol error is likely to occur due to extreme environmental conditions. Since we cannot determine the probability of these conditions and these depend on the working conditions of the device, probability of environmental fluctuations to create a symbol error is assumed to be 1 (worst-case scenario). Resulting probability of symbol error is the product of probabilities of

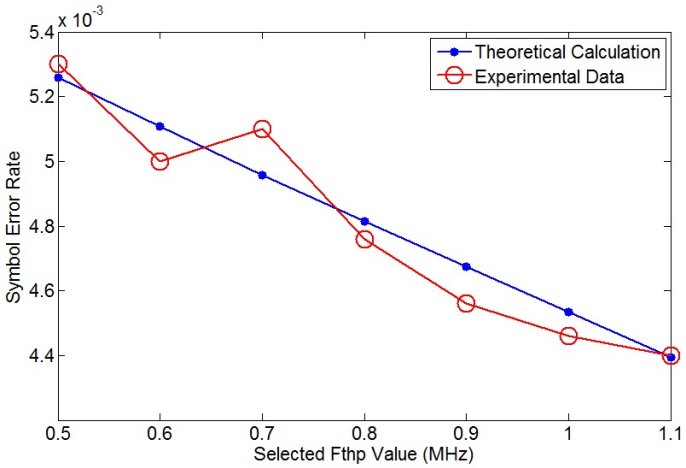

Fig. 7. Symbol error rate

conditions stated above and presented as

$$p = \sum_{s=2}^{m} k_s * (s-1)/(M*(M-1)). \tag{3}$$

Assuming that a problematic RO pair is somehow present in the system, the symbol error rate is calculated based on the data presented in Section II, Section IV-A and equations (1) and (3) for f_{thp} values in the range of 0.5 MHz to 1.1 MHz. As seen from Figure 7, theoretically calculated symbol error rate varies linearly between 5.26×10^{-3} and 4.40×10^{-3}.

Validation of the symbol error probability analysis is done via creating random sets of RO frequencies in MATLAB environment. For this purpose, 30,000 sets of 160 RO frequencies with Gaussian distribution are created. The mean and standard deviation of the distribution is obtained from real data collected from FPGA implementation. Then, DP algorithm is applied to each set of 160 ROs using an f_{thp} value of 0.5 MHz. Next, a problematic RO pair is defined by choosing two ROs randomly from RO_1 to RO_{160}. Then, the RO sets that include the selected RO pair next to each other and in the correct order are counted to determine the symbol error rate. This process is repeated for different f_{thp} values in the range of 0.5 MHz to 1.1 MHz. The results are presented in Figure 7. As seen from the figure, even though an outlier is present at an f_{thp} value of 0.7, which may be a result of randomly generated RO sets, experimental data are compatible with the theoretical calculations validating the analysis presented above.

C. Symbol error to bit error conversion and bit error probability

When the ordering in a group changes, a symbol error occurs and the output becomes erroneous. The number of bits affected by a symbol error depends on the output generation scheme. In order to analyze the bit error probability, two simple and efficient output generation schemes are selected, direct mapping and Gray encoding. With these schemes, each ordering in a group of length s is mapped to a bit stream with $\lceil \log_2(s!) \rceil$ bits. An example

TABLE I
OUTPUT GENERATION MAPPING

Frequency Ordering	Output Bits by Direct Mapping	Output Bits by Gray Encoding
RO1>RO2>RO3	000	000
RO1>RO3>RO2	001	001
RO2>RO1>RO3	010	011
RO2>RO3>RO1	011	010
RO3>RO1>RO2	100	110
RO3>RO2>RO1	101	111

mapping for a group size 3 is presented in Table I.

By using the mapping schemes, a symbol error results in a certain number of changes in the output bit stream. Since the number of erroneous bits depends on the group size and the pre-error and post-error orderings, a complete analysis is required to determine the bit error probability. The number of symbol error cases, E_c, for a group of size s, can be calculated using the number of possible pre-error orderings, $s!$, and possible post-error orderings, $s-1$, given as

$$E_c = s! * (s-1). \tag{4}$$

For group sizes of up to 25 ROs, the number of bit errors is calculated for E_c and the average number of bit errors for a group size of s, $bepg_s$, is presented in Figure 8. [3] Next, bit error probabilities per generated bit are calculated for both mapping schemes and presented in Figure 9. As can be seen from the figure, the bit error probability is lower in larger groups. Finally, by using the symbol error rate and $bepg_s$, the bit error probability per problematic RO pair, bep can be calculated as

$$bep = \sum_{s=2}^{m} (k_s * bepg_s) * (s-1)/(M * (M-1)). \tag{5}$$

Bit error probability for fth_p values in the range of 0.5 MHz to 1.1 MHz is presented in Figure 10. Interestingly, both mappings result in similar bit error probabilities. For both mappings, bit error probabilities are as low as 10^{-2} for an f_{thp} value of 1.1 MHz, whereas, for an f_{thp} value of 0.5 MHz, they are slightly higher than 2.4×10^{-2}.

D. Worst case bit errors per problematic RO Pair

As discussed above, the number of erroneous bits due to a symbol error differs due to group size and pre-error and post-error ordering of RO frequencies. Even though the average bit error rate gives a clear idea about the

[3]Bit errors are calculated for all E_c up to group sizes of 10. For group sizes larger than 10, Monte Carlo method is applied and bit errors are calculated via 500,000 random trials.

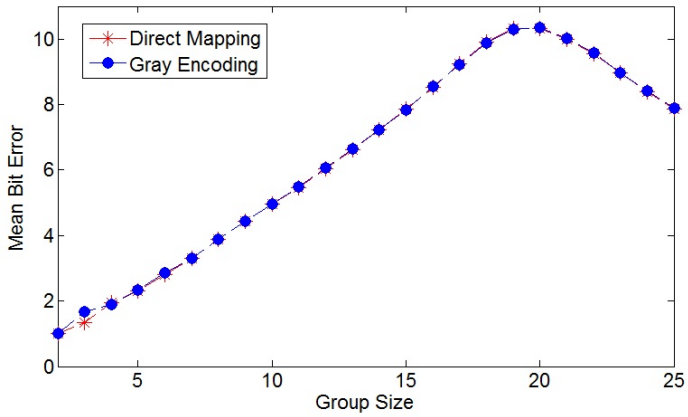

Fig. 8. Number of bit errors per group

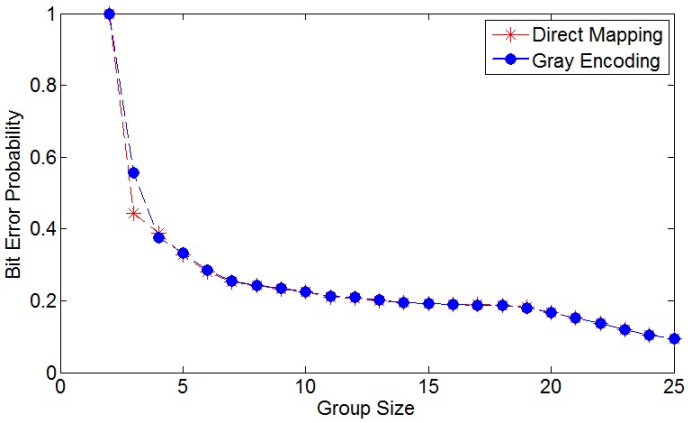

Fig. 9. Bit error probability per generated bit

erroneous output, the worst case data will also be helpful for some applications such as authentication, where the system performance heavily relies on the maximum number of errors in a key. In this sense, based on group sizes, all pre-error and post-error orderings are analyzed to determine the maximum number of output bits that can flip due to a symbol error. For both direct mapping and Gray encoding, the maximum number of bit errors at the output is shown in Figure 11 as a function of the group size. As can be seen from the figure, using larger groups has a disadvantage in terms of the worst case bit error despite their area efficiency. In addition to this, Gray encoding has a significant advantage over direct mapping for almost all group sizes.

Even though the number of erroneous bits at the worst case is a very critical information, their occurrence probability is also important. For this purpose, the occurrences of worst symbol errors among all possible errors are counted and their probabilities are presented in Figure 12 for group sizes of up to 20. [4] As can seen from the figure, the worst case symbol error probabilities, $wcsep_s$, decrease swiftly with increasing group sizes and approaches 10^{-6} for a group size of 20.

[4]Bit errors are calculated for all E_c up to group sizes of 10. For group sizes larger than 10, Monte Carlo method is applied and bit errors are calculated via 500,000 random trials.

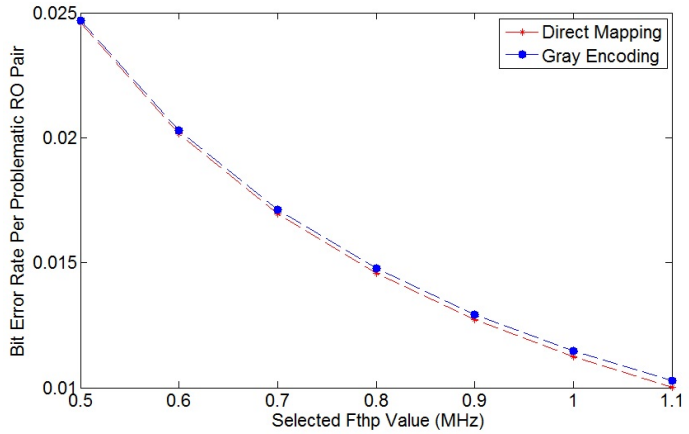

Fig. 10. Bit error probability per problematic RO pair

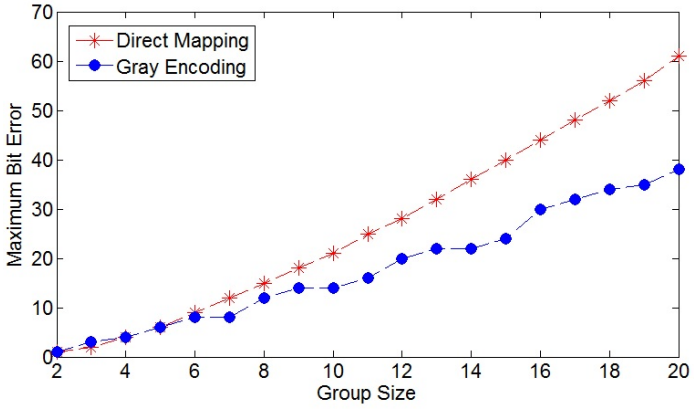

Fig. 11. Maximum error vs. group size

V. Area usage vs. robustness in identification systems

Unlike the cryptographic key generation application, erroneous outputs are acceptable up to a certain degree in some other applications, such as identification [13], [17]. The degree of the acceptable error rate depends on the false rejection rate (FRR) and false acceptance rate (FAR) requirements. Due to the errors at the output, a circuit may be authenticated as another circuit, which contributes to the FAR. Similarly, a circuit may unnecessarily fail to be authenticated, which contributes to the FRR [13]. [18] specifies the relation between the number of erroneous bits in the output and the false acceptance and rejection rates for identification. For instance, if 10 bits out of a total output length of 128 bits are allowed to be erroneous, the FAR can be calculated as 2.1×10^{-21} and the FRR can be calculated as less than 5×10^{-11}.

The immunity of these applications to erroneous outputs up to a degree enables increasing the area efficiency of the underlying PUF circuit. As described in Section II, choosing an f_{thp} value smaller than the optimum value results in forming larger groups by the DP algorithm, and, hence, increasing the entropy extraction from the system. This enables an implementation consisting of a smaller number of ROs to achieve a target number of output bits.

In order to quantify the effects of non-ideal f_{thp} values on the number of erroneous bits at the output, an analysis

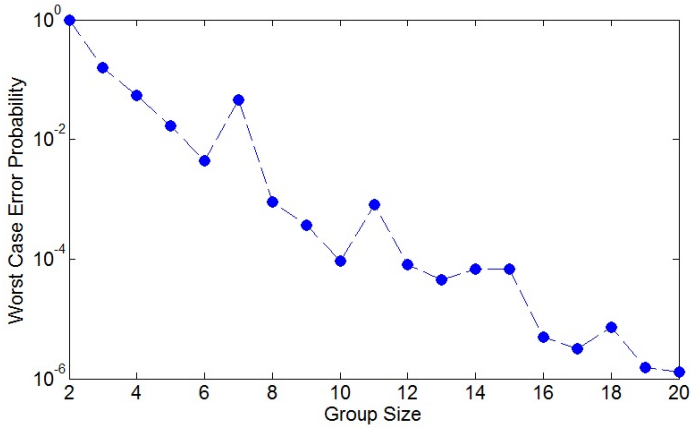

Fig. 12. Worst case symbol error probability vs. group size

TABLE II
NUMBER OF PROBLEMATIC RO PAIRS

Selected f_{thp} (MHz)	**1**	**0.95**	**0.9**	**0.85**
Num. of Problematic Pairs	0	2	5	15
Num. of Required ROs	114	110	108	105

is performed to determine the number of RO pairs that may change their ordering under extreme conditions. For this purpose, the RO frequency data, collected from the sample implementation described in Example 2.1, is used and an f_{thp} value of 1 MHz is assumed as ideal. Then, the RO pairs with frequency deviation more than the so-called non-ideal f_{thp} values in the range of 0.85 MHz to 1 MHz is counted to determine the maximum number of problematic RO pairs. This results in an upper bound on the error vulnerability with respect to the non-ideal f_{thp} values. Here, the lowest f_{thp} value is selected as 0.85 MHz, since lower values will result in a very large number of errors, and the highest f_{thp} value is selected as 1 MHz, since higher values will not result in erroneous bits and hence the analysis will not be applicable. It can be seen from Table II that, as the non-ideal f_{thp} values get smaller and smaller, the number of problematic RO pairs increases significantly, as expected. On the other hand, due to the increase in the group sizes, the entropy extraction is increased and so is the area efficiency. In order to analyze the improvement in the area consumption of the system, the number of required ROs for a fixed 128 bits of output is analyzed on a sample system using the DP algorithm with non-ideal f_{thp} values. As shown in Table II, the required number of ROs decreases 9% with a reduction of 150kHz in the fth_p value.

Since each problematic RO pair contributes to the generation of erroneous symbols, the bit error probability of the system will increase directly proportional to the number of problematic RO pairs. Based on the bit error probability per problematic RO pair analysis presented in Figure 10, and the analysis on the number of problematic RO pairs given in Table II, the bit error probability of the sample system with respect to the selected f_{thp} value can also be calculated. As can be seen from Figure 13, the bit error probability increases significantly as the selected f_{thp} value decreases and a 150kHz reduction causes over 20% of the output bits to be erroneous.

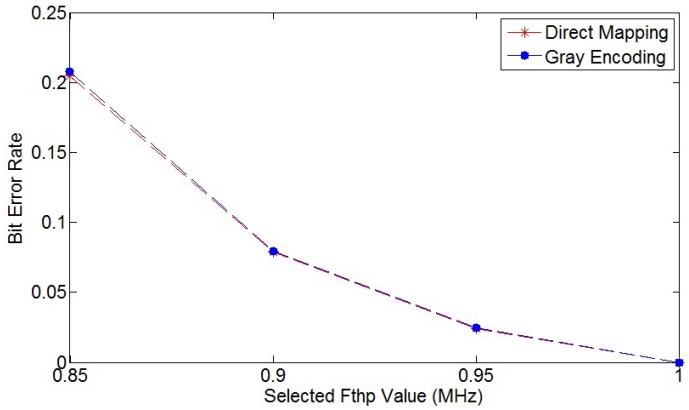

Fig. 13. Bit error probability vs. f_{thp} chosen

TABLE III
SIMULTANEOUS SYMBOL ERROR PROBABILITIES FOR NON-IDEAL $fthp$ VALUES

Selected f_{thp} (MHz)	**1 error**	**2 error**	**3 error**	**4 error**	**5 error**
0.95	0.0092	2.1178e-005			
0.90	0.0229	2.15e-004	1.01e-006	2.37e-009	2.22e-012
0.85	0.0666	0.0022	4.58e-005	6.56e-007	6.88e-009

The worst case bit error amount depends on the number of symbol errors that occur at the same time, n. Based on the number of problematic RO pairs k, the symbol error probability per problematic RO pair p, and the probability of simultaneous n errors, $p_{n,k}$ can be calculated as

$$p_{n,k} = p^n * (1-p)^{k-n} * \binom{k}{n}. \tag{6}$$

Based on this relation, the symbol error rates for different numbers of errors at a time are calculated and presented in Table III. As can be seen from these results, the probability of multiple errors occurring simultaneously decreases exponentially as n increases. Using this data, the maximum error amount per group size, and the expected largest group size, a designer can choose the f_{thp} value maintaining the system requirements. The probability of worst case error situation can be calculated as

$$wcep_n = p_{n,k} * wcsep_s^n, \tag{7}$$

assuming that all errors will take place within the largest groups. For instance, if the probability of three or more errors at a time can be disregarded and the largest group size is limited by 10, then the worst case bit error amount for an f_{thp} value of 0.9 MHz will be 28 with Gray encoding and the probability of error caused by this will be 1.83×10^{-12}.

VI. Conclusion

In this work, an efficient grouping method for RO-PUFs is presented and the error vulnerability of PUFs based on frequency threshold determination is analyzed. Next, the length of groups used in such PUF circuits is analyzed and the symbol error probability is obtained analytically. In addition to these, the symbol error to bit error conversion and the bit error probability calculations based on the wrong determination of the frequency threshold are shown. Finally, a trade-off between the area usage and robustness is presented for identification systems. Theoretical calculations are validated via experimental measurements using Xilinx FPGAs. This analysis can be used by the system designer to determine the trade-off between the area efficiency and error immunity of the system. Depending on the particular application, the average and maximum number of erroneous bits that the PUF circuit can tolerate are first determined and these performance parameters allow the designer to choose the f_{thp} value. This selection determines the distribution of the group lengths and allow for obtaining a good balance between error probability, and area and power efficiency.

References

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

[2] R. S. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 6, pp. 2026–2030, 2002.

[3] P. Tuyls, G. J. Shrijen, B. Skoric, J. V. Geloven, N. Verhaegh, and R. Walters, "Read proof hardware from protective coatings," in *18th Annual Computer Security Applications Conference (CHES)*, vol. 4249, 2006, pp. 369–383.

[4] D. Lim, J. Lee, B. Gasend, G.E.Suh, M. V. Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Transactions on VLSI Systems*, vol. 13, no. 10, pp. 1200–1205, 2005.

[5] B. Gassend, D. Clarke, M. V. Dijk, and S. Devadas, "Delay-based circuit authentication and applications," in *ACM Symposium on Applied Computing*, 2003, pp. 294–301.

[6] B. Gassend, "Physical random functions," M.S. Thesis, Massachusetts Institute of Technology, Massachusetts, 2003.

[7] J. Guajardo, S. Kumar, G. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *18th Annual Computer Security Applications Conference (CHES)*, vol. 4727, 2007, pp. 63–80.

[8] J. Guajardo, S. Kumar, R. Maes, G. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *Hardware-Oriented Security and Trust (HOST)*, 2008, pp. 67–70.

[9] D. Suzuki and K. Shimizu, "The glitch PUF: A new delay-PUF architecture exploiting glitch shapes," in *Cryptographic Hardware and Embedded Systems (CHES)*, 2010, pp. 366–382.

[10] 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.

[11] C. Yin and G. Qu, "Temperature aware cooperative ring oscillator PUF," in *IEEE International Workshop on Hardware Oriented Security and Trust (HOST)*, 2009, pp. 36–42.

[12] S. Katzenbeisser, U. Kocabas, V. Rozic, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "Pufs: Myth, fact or busted? a security evaluation of physically unclonable functions (PUFs) cast in silicon," in *Cryptographic Hardware and Embedded Systems (CHES)*, 2012, pp. 283–301.

[13] G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *Design Automation Conference (DAC)*, 2007, pp. 9–14.

[14] C. Yin and G. Qu, “LISA: Maximizing RO-PUF’s secret extraction,” in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2010, pp. 100–105.

[15] G. Komurcu, A. E. Pusane, and G. Dundar, “Dynamic programming based grouping method for RO-PUFs,” in *9th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME)*, 2013, pp. 329–332.

[16] G. Komurcu, A. E. Pusane, and G. Dundar, “Enhanced challenge-response set and secure usage scenarios for ordering based RO-PUFs,” *Accepted for publication in IET-Circuits, Devices, and Systems, (IET-CDS)*. [Online]. Available: http://giraykomurcu.net/Akademik/PUF_C5_giray_komurcu.pdf

[17] S. Devadas, E. Suh, S. Paral, R. R. Sowell, T. Ziola, and V. Khandelwal, “Design and implementation of PUF-Based ”unclonable” RFID ICs for anti-counterfeiting and security applications,” in *IEEE International Conference on RFID*, 2008, pp. 58–64.

[18] C. Bohm and M. Hofer, *Physical Unclonable Functions in Theory and Practice*. Springer, 2013.