Normalization and Multi-Valued Symbol Extraction From RO-PUFs for Enhanced Uniform Probability Distributions

Holger Mandry, Andreas Herkle, Sven Müelich, Joachim Becker, Robert F. H. Fischer, and Maurits Ortmanns

***Abstract*—Physical Unclonable Functions (PUFs) offer the possibility for on-chip generation of unique fingerprints for integrated circuits. Ring-oscillator (RO) PUFs are small and easy to configure on Field Programmable Gate Arrays (FPGAs) and thus received great attention over the years. In the state-of-the-art two neighboring ROs are compared and mapped to only a single bit of information. Few publications aims to extract more bits out of one PUF-cell, but struggle with non-uniform distributions. In this brief multi-valued symbol extraction is presented as a method to extract more bits of information out of each individual RO. A new post-processing approach is introduced to produce close-to-ideal uniformly distributed responses independent of the underlying physically probability distribution. To eliminate bias, caused by placement inequalities, multiple methods of normalization are utilized and analyzed by means of area and complexity. Based on metrics for symbol transmission, the Euclidean-distance and entropy are used as metrics to evaluate the uniqueness and reliability of multi-valued PUFs. This new approach allows to increase the amount of extracted information to 3 bits per RO.**

***Index Terms*—Physical unclonable function (PUF), FPGA, multi-valued PUF, ring-oscillator (RO), cryptography keys.**

I. Introduction

INTERNET-OF-THINGS (IoT) devices become more important in daily life and bring along the need for secure communication and authentication and consequently, strong cryptographic keys are needed. Especially for low-power Internet-of-Things (IoT) devices, Physical Unclonable Functions (PUFs) are a great alternative to dedicated power-hungry memory. PUFs are electrical circuits that make use of small manufacturing variations in order to produce unpredictable but repeatable fingerprints. These fingerprints are derived from the physical structure of the circuit itself, thus no secure memory is needed. For Field Programmable Gate Arrays (FPGAs) it has been shown that ring-oscillator (RO) PUFs are the most preferable implementation choice [1]. They make use of small time differences of identical routed delay lines and measure these differences by counting the edges [2] within a fixed evaluation time. The measured differences are usually quantized to a single bit output.

This brief analyzes RO PUFs on FPGAs and presents a method to extract multi-valued responses in order to extract more than one bit per RO and thus reduce the required area for key generation.

This brief is organized as follows. Section II describes how normalizing post-processing steps are used to reject the influence of different logic types on the ROs responses. Two different methods of multi-valued quantization are presented in Section III and analyzed in Section IV. The uniqueness and reliability of the new quantization methods are explored in Section V. Section VI concludes this brief.

Manuscript received January 31, 2020; accepted March 6, 2020. Date of publication March 13, 2020; date of current version November 24, 2020. This work was supported by the German National Science Foundation DFG under Grant FI 982/15-1. This brief was recommended by Associate Editor M. Small. *(Corresponding author: Holger Mandry.)*

Holger Mandry, Andreas Herkle, Joachim Becker, and Maurits Ortmanns are with the Institute of Microelectronics, University of Ulm, 89081 Ulm, Germany (e-mail: holger.mandry@uni-ulm.de).

Sven Müelich and Robert F. H. Fischer are with the Institute of Communications Engineering, University of Ulm, 89081 Ulm, Germany.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2020.2980748

II. Normalization of RO Distributions

As explained in [3] and shown in Fig. 1, the frequency distributions of ROs on FPGAs depend not only on the deviations from manufacturing, but also on the slice type in which the ROs are placed. E.g., slice types with memory capability (type M) have lower frequencies than pure logic slices (type L). Type L slices can be further separated into slices with even and odd X coordinates, which introduces additional differences in the mean oscillation frequencies due to routing differences. Additionally, knowing the slice type makes it easier for attackers to guess the response of an individual RO. Thus, normalization is inevitable in order to develop a common processing method. This section examines different ways of normalization and evaluates them in terms of hardware cost and complexity. All ROs placed in the same slice type are combined as a subgroup g. These subgroups are normalized separately. The used data sets come from 22 Zybo boards with a Xilinx ZYNQ XC7Z010 FPGA, each containing a total of 3800 ROs. For each RO, $R = 1000$ readouts r with an evaluation time of 10 μs were taken into account. The extraction was done in parallel mode at room temperature with the framework described in [3].

For the first method of normalization, which will be called *mean normalization*, the mean frequency $\bar{f}$ of each RO

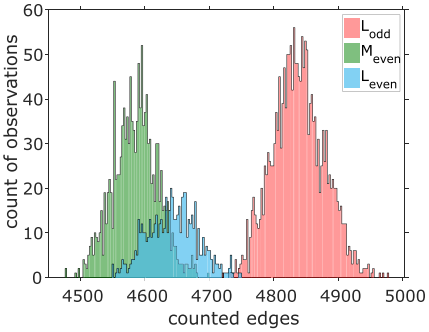
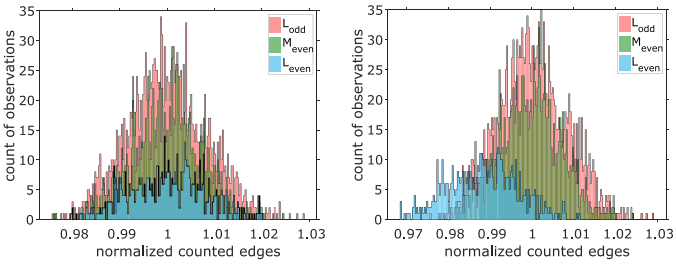


Fig. 1. RO frequency distributions of selected individual board with no normalization for three different slice types/subgroups.



(a) mean normalization (b) group normalization

Fig. 2. RO frequency distributions of selected individual board after mean and group normalization with division.

subgroup g is chosen as the reference value:

$$f_{\text{ref,g}} = \bar{f}_g = \text{mean}_{\forall ro|g}(\text{mean}_{\forall r}(f_{ro,r})) \tag{1}$$

The second normalization method, called *group normalization*, uses the frequency of a selected reference RO of the same RO subgroup as the reference value:

$$f_{\text{ref,g}} = f_{ro_1}, \quad ro_1 \in g \tag{2}$$

For calculating the normalized RO frequencies, two different methods are compared. The first method in equation (3) divides each RO frequency by the reference value. The second method in equation (4) removes offsets by subtracting the reference value from the RO frequencies.

$$\tilde{f}_{ro} = \frac{f_{ro}}{f_{\text{ref},g|ro\in g}} \tag{3}$$

$$\tilde{f}_{ro} = f_{ro} - f_{\text{ref},g|ro\in g} \tag{4}$$

Fig. 2 compares (a) the *mean* and (b) the *group* normalization calculated with the division method as explained above. The subtraction method in comparison has similar distributions but differs in the frequency data range and has a mean value of 0.

The *Kolmogorov–Smirnov Test* [4] was used to test if the frequency distributions are still Gaussian shaped after normalization. A Gaussian probability density function (PDF) with mean and standard deviation of the tested data set was used as null hypothesis. The resulting p value is the probability that the given test statistic will be drawn from the assumed distribution. Therefore small p values cast doubt on the assumed distribution. If the p value is smaller than the significance level $\alpha = 0.05$ the null hypothesis is rejected. The p values were only changed in a range of $\delta p = 10^{-4}$ by normalization and the average over all boards vary from $\bar{p} = 0.36$ to 0.44 for the different subgroups. All distributions that fail to reject the null hypothesis before normalization also fail to reject it afterwards. Board 4 and 17 rejected the null hypothesis in all cases. In conclusion the normalizations do not influence the probability distributions of the RO frequencies, but one can doubt that some boards follow the assumed Gaussian PDF in the beginning. This might cause worse results in post-precessing steps that rely on Gaussian distributions.

Regarding accuracy, e.g., the mean frequencies $\mu_{L_{even}}$ of subgroup L_{even} of board 16 after *group* normalization are 0.9883 for division and -54.58 for subtraction and have a noticeable difference to the optimum value of 1 for division and 0 for subtraction. Compared to the data range, these values correspond to a deviation of $\delta\mu = 19\%$. Due to the fact that the reference RO is always the first one of the subgroup it might be badly chosen as its frequency is not close to the mean value of the subgroup. The *mean* normalization is far more accurate for all evaluated boards.

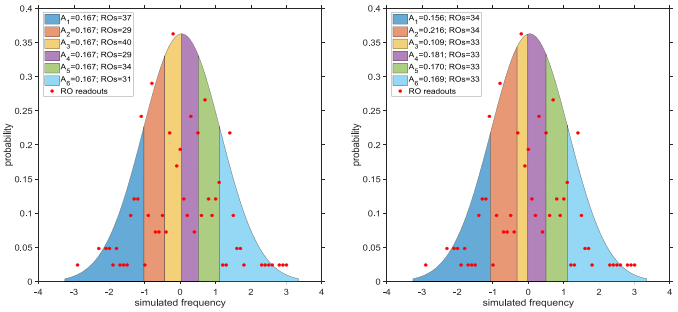
Regarding hardware consumption and complexity the *mean* normalization requires an adder and a divider to compute the reference value. The calculation needs several clock cycles, increasing with the amount of ROs. Additionally the reference has to be recalculated periodically to counteract global noise sources like temperature changes or aging. In comparison, the *group* normalization requires one additional RO as reference and its counter for each subgroup. Another advantage of a dedicated reference RO is that it will also be affected by the same global disturbances and therefore can minimize them. The possibility of removing noise with a reference RO was already shown in [3]. Comparing *division* with *subtraction*, the *subtraction* requires less area since a divider consists of more look-up tables and additional flip flops in comparison to an adder. Furthermore an area efficient division takes several clock cycles, depending on the accuracy, whereas an addition can be computed in one clock cycle.

In summary the *group* normalization with subtraction has the lowest hardware requirements and complexity while the *mean* normalization creates more accurate results for both division and subtraction.

III. Symbol Extraction

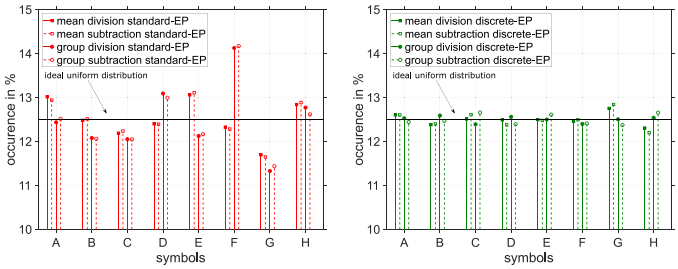
To extract PUF responses, each RO frequency has to be mapped to a value. In former publications [5], [6], this is done by comparing the RO frequency with a reference frequency and returning either a bit value of 0 (lower than reference) or 1 (higher than reference). The reference value can either be a static value (first order quantization) or another RO frequency (second order quantization) [5]. Although these methods are easy to implement and provide good statistics, their output is limited to just one bit while requiring up to two ROs.

In this brief higher order symbols are used to extend the response space of RO-PUFs and generate responses of multiple bits per RO. The PDF of the dataset is divided into k frequency intervals to get k different symbols. In [7], two different methods are described to divide the PDF into intervals. The *equi-distant* method, as used in [8], arranges all intervals of



(a) equi-probable (b) discrete equi-probable

Fig. 3. Comparison of both methods to determine intervals.



(a) equi-probable (b) discrete equi-probable

Fig. 4. Exemplary distribution of eight extracted symbols ($S = 8$) for selected individual board. The black lines mark the ideal uniform distribution.

the Gaussian PDF such that they have the same width but different probabilities. The *equi-probable* method arranges all intervals such that the individual areas under the Gaussian PDF have the same probability. In order to reduce the predictability of individual symbols and thus enhance security of the PUF, equal distribution of all resulting symbols is required. Therefore only the *equi-probable* method is further considered and will be called *standard equi-probable (standard-EP)*.

As the Gaussian PDF is a statistical description of the real RO frequency distribution, it can happen that one symbol interval corresponds to more normalized RO frequencies than others, which will lead to a non-uniform distribution of symbols. To counteract this problem we introduce a *discrete equi-probable (discrete-EP)* method. With this method the amount of ROs assigned to one symbol are almost the same, thus the *discrete* method can be expected to result in a better uniform distribution of symbols for real RO readouts. To achieve this the RO frequencies are sorted in ascending order and then grouped to symbols, in such a way that each group contain $n = \lfloor R/S \rfloor$ and S measured RO frequencies. The first $m = R \ mod \ S$ intervals cover one RO frequency more to avoid unassigned ROs.

One further advantage of this method is the independence from the underlying PDF so it can also be applied to other PUF sources.

A comparison of both methods is illustrated in Fig. 3. In this example 200 normally distributed RO readouts, indicated as red dots, were divided into six intervals using both methods. They differ in two aspects. For the *discrete-EP* method in Fig. 3(b), the interval widths are not symmetric and the areas under the Gaussian PDF, A_1 to A_6 in the legend, are not of the same size. In contrast for the *standard-EP* method in Fig. 3(a) all symbols have the same probability as the areas are of the same size. The amount of ROs assigned to one symbol, given as *ROs* in the legend, makes the second difference between both methods. For the *standard-EP* method the amounts differ widely regarding the individual symbols, whereas in the *discrete-EP*, the amount of ROs are almost the same. With an infinite number of ROs both methods will result in the same intervals, but as in real applications the number of ROs is limited the *discrete-EP* method is more accurate.

IV. Comparison of Symbol Distributions

In this section, the distribution of the extracted symbols from the different normalization and extraction methods are discussed. Each data set is analyzed individually but for plotting the results are averaged. In order to compare the different distributions, the magnitude of the deviation was calculated with the *chi-squared test* [9]:

$$\chi^2 = \sum_{j=1}^{S} \frac{(p(s_j) - P_j)^2}{P_j} \tag{5}$$

$P_j = S^{-1} \forall j$ is the expected probability of symbol j, where S is the amount of symbols and $p(s_j)$ the measured probability of symbol j. The higher the chi-square value the more the dataset differs from the expected uniform distribution.

In order to compare all different normalizations, the RO mean frequencies of each data set from the individual boards were normalized according to Section II. In the next step frequency intervals for different amounts of symbols were calculated with the methods described in Section III. The amount of symbols was swept from two to twelve and each single readout was labeled with its respective symbol. Therefore the intervals have to be stored in memory for field applications. Leakage analysis of this stored data is not in the scope of this brief and will be investigated in future publications.

Fig. 4 displays the exemplary number of occurrence of eight different symbols for board number 16. The black lines indicate the optimal uniform distribution. Fig. 4(a) illustrates the distributions of *standard-EP* (see Fig. 3(a)) symbol extraction. The individual symbols differ up to 1.5% from an uniform distribution. The chi-square values are about $1.2 \cdot 10^{-3}$ for *mean* and $3.8 \cdot 10^{-3}$ for *group* normalization. In comparison the distributions for the proposed *discrete-EP* symbol extraction (see Fig. 3(b)) are shown in Fig. 4(b). Here the maximum difference to the optimum of 12.5% is around 0.4%. The chi-square values vary from $0.2 \cdot 10^{-3}$ to $0.03 \cdot 10^{-3}$. It can concluded that the *discrete-EP* method results in much better uniform distribution and thus provides a more secure PUF response due to a lower predictability.

In Fig. 5 the chi-square values averaged, over all boards, are illustrated for all normalization and extraction types. With increasing amount of symbols extracted from a single PUF source, the chi-square values increase linearly which indicates a worse representation of a uniform distribution, but the different methods yield significantly different results.

Subtraction vs. division: The solid *division* and the dashed *subtraction* normalization differ only with a maximum of $5 \cdot 10^{-4}$ which indicates that they perform equally.

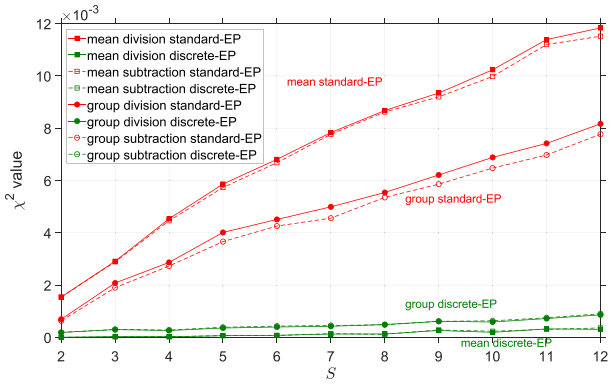


Fig. 5. Chi-square values for different amount of symbols, averaged over all boards.

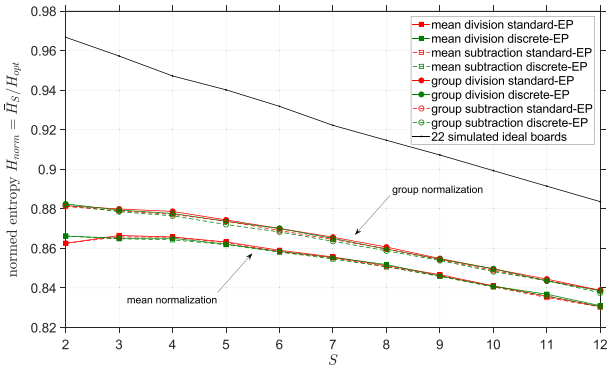


Fig. 6. Averaged entropy $\bar{H}_S$ normalized by optimum entropy $H_{\text{opt}} = \log_2(S)$ of uniform distribution.

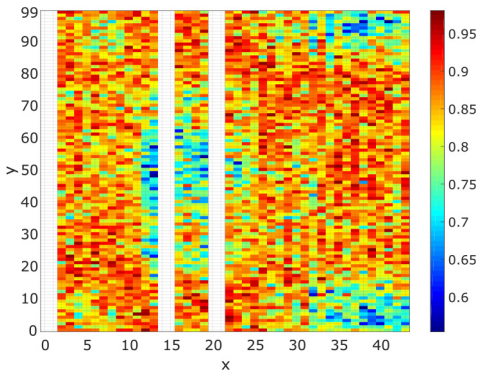


Fig. 7. Averaged entropy normed by optimum entropy of uniform distribution, 8 symbols, discrete-EP extraction, group normalization.

Standard vs. discrete EP: The *discrete-EP* extraction shown in green always outperforms *standard-EP* extraction shown in red as the chi-square values are a factor of 100 smaller. For *standard-EP* extraction the slope is bigger than for *discrete* extraction. This can be explained by the fact that *discrete-EP* extraction tries to ensure that every symbol corresponds to an equal amount of ROs and therefore distributes them in a more uniform way than the *standard-EP* method, which in contrast calculates with statistical characteristics as already explained in Section III. The more symbols were taken into account the less ROs correspond with one symbol. Therefore the impact of unequal amounts of ROs per symbol gets higher, which results in a higher chi-square value.

Group vs. mean normalization: Comparing the *group* normed *discrete-EP* method with the *mean* normed *discrete-EP* method the difference between both is a maximum of $6 \cdot 10^{-4}$. In contrast to the *standard-EP* method this difference can be neglected. Therefore the performance of *mean* and *group* normalization can be regarded as equal related to chi-square values and uniform distribution of symbols.

V. PUF Characteristics

In this section the uniqueness and reliability of the RO's corresponding symbols are analyzed. Each RO mean frequency is labeled with its respective reference symbol and examined for cross-board distribution and noise.

A. Uniqueness

The most important characteristic of PUFs is their uniqueness [2], which was evaluated by the inter-Hamming distance in previous work. As Hamming distances are calculated from binary values, they are not suitable for multi-valued symbol extraction.

Instead the uniqueness is analyzed by counting the amount of reference symbol occurrences for each RO and calculate the individual information entropy [10] along all boards by:

$$H_{ro,S} = -\sum_{j=1}^{S} p(s_{j,ro}) \cdot \log_2 p(s_{j,ro}) \tag{6}$$

In Fig. 6 the averaged entropy of all ROs is plotted against the chosen number of symbols. All values are normalized with the optimal entropy of a uniform distribution $H_{\text{opt}} = \log_2(S)$. The normalized entropy for *group* normalization is around 1 to 2% higher than for *mean* normalization. The averaged non-normalized entropy $\bar{H}_S$ ranges from 0.8819 bit for two symbols to 3.0024 bit for twelve symbols. The normalized entropy decreases with increasing number of symbols, especially because the number of test boards is limited. The more symbols are available for labeling, the higher the probability that individual symbols never occur. To check how reasonable these results are, randomly extracted symbols of a uniform distribution were analyzed as comparison. The results for 22 simulated board readouts are plotted as a black line. When increasing the amount of symbols the entropy also shrinks as with the real data. With 1000 simulated boards the resulting entropies were very close to the optimum of 1. This proves that the small amount of boards is one reason for non optimal entropy.

The remaining differences between measured and simulated data are about 10% for two symbols and 5% for twelve symbols. One reason for this difference is the averaging of the values. Regarding the entropy for each individual RO in Fig. 7, where the spatial distribution of RO is shown on the used FPGA, one can identify some ROs with a very low entropy. The four regions in the corners and the one located left to the middle of the chip were previously identified to have a relative frequency shift to the average [3]. To increase the overall entropy, one can exclude these regions or treat them as individual subgroups and normalize them separately.

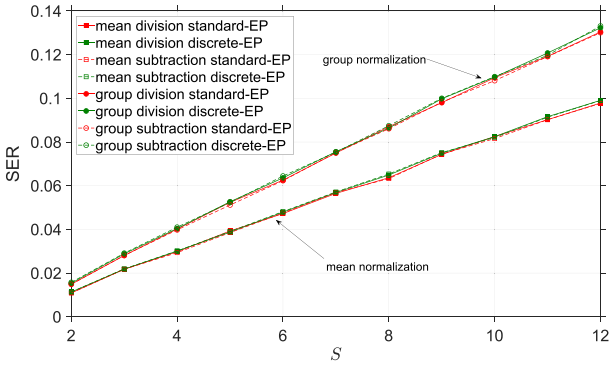


Fig. 8. Averaged SER of all ROs for different amount of symbols.

B. Reliability

Reliability of PUFs is a further evaluation criterion, which is determined by the intra-Hamming distance in the state-of-the-art. For multi-valued PUFs the use of the Euclidean distance as metric is more appropriate as it is independent of the binary representation. The distance d between two different symbols $s_1 = (s_{1,1}, \ldots, s_{1,n})$ and $s_2 = (s_{2,1}, \ldots, s_{2,n})$ can be calculated by:

$$d(s_1, s_2) = \sqrt{\sum_{j=1}^{n} (s_{1,j} - s_{2,j})^2} \tag{7}$$

With the Euclidean distance d it is possible to calculate the symbol error rate (SER) for each RO readout r, with a total of R readouts and a reference symbol s_{ref} using

$$SER_{ro} = \frac{1}{R} \sum_{r=1}^{R} (d(s_r, s_{ref}) \neq 0) \tag{8}$$

In Fig. 8 the averaged SER of all ROs are shown for the different number of symbols. With increasing symbol space the SER also increases as the frequency interval widths decrease. As the noise margin of a single RO stays constant with variable amount of symbols it becomes more likely that the RO frequency is mapped to a different symbol which results in a higher SER.

The SER of *group* normalization is a factor of 0.14 to 1.34 times higher than for *mean* normalization. The *group* normalization uses a noisy RO as reference value. This noise interferes with the noise of the RO that is normalized. With this interference both noise sources might sum up, which results in an increased SER.

For comparison all extraction methods were applied as well to the data sets presented in [11], resulting in similar results with slightly higher entropy of 3.6 bits for twelve symbols.

VI. Conclusion

In this brief we presented several methods of RO frequency normalization. Normalized RO frequencies can be post processed to become independent of delay offsets caused by slice types, which makes it harder to predict the RO response. The *group* normalization, which normalizes the ROs of one slice type by a reference RO, requires no recalculation phases and less hardware than the *mean* normalization, which normalizes by the mean frequency of all ROs of the same slice type. To compute the normalizations, a division and a subtraction method have been compared. In following post-processing both showed similar results. A huge difference can be observed in the amount of required hardware components. The divider utilizes more components than the subtracter and has a longer computation time. Therefore a subtraction *group* normalization seems to be the best choice.

Additionally two techniques to arrange the normed frequency spectrum into intervals are compared with the goal of shaping the intervals such that the corresponding ROs are distributed uniformly. Regarding the uniform distribution and entropy the newly introduced *discrete-EP* method outperforms the *standard-EP* [7] method. The chi-square values are up to 24 times smaller and in comparison to the *standard-EP* method nearly constant for variable amount of symbols-intervals.

The Euclidean distance was introduced as metric for multi-valued PUFs to calculate the SER. Furthermore the information entropy is used to quantify the uniqueness. The *group* normalization had a slightly higher SER than the *mean* normalization due to the noisy reference. On the other hand the extracted entropy was higher with *group* normalization than with *mean* normalization. With twelve different symbols it is possible to extract an entropy of 3 bit from one RO, which is three times more than state-of-the-art quantization of RO PUFs.

References

[1] S. Morozov, A. Maiti, and P. Schaumont, "An analysis of delay based PUF implementations on FPGA," in *Proc. Int. Symp. Appl. Reconfigurable Comput.*, Heidelberg, Germany, 2010, pp. 382–387.

[2] B. Gassend, D. Clarke, M. V. Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. 9th ACM Conf. Comput. Commun. Security*, 2002, pp. 148–160.

[3] A. Herkle, H. Mandry, J. Becker, and M. Ortmanns, "In-depth analysis and enhancements of RO-PUFs with a partial reconfiguration framework on Xilinx Zynq-7000 SoC FPGAs," in *Proc. Int. Symp. Hardw. Orient. Security Trust (HOST)*, McLean, VA, USA, 2019, pp. 238–247.

[4] J. Frank and Jr. Massey "The Kolmogorov–Smirnov test for goodness of fit," *J. Amer. Stat. Assoc.*, vol. 46, no. 253, pp. 68–78, 1951.

[5] V. Immler, M. Hiller, J. Obermaier, and G. Sigl, "Take a moment and have some t: Hypothesis testing on raw PUF data," in *Proc. Int. Symp. Hardw. Orient. Security Trust (HOST)*, McLean, VA, USA, 2017, pp. 128–129.

[6] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th Annu. Des. Autom. Conf.*, San Diego, CA, USA, 2007, pp. 9–14.

[7] V. Immler, M. Hennig, L. Kürzinger, and G. Sigl, "Practical aspects of quantization and tamper-sensitivity for physically obfuscated keys," in *Proc. Workshop Cryptograph. Security Comput. Syst. (CS2)*, 2016, pp. 13–18.

[8] V. Immler, M. Hiller, Q. Liu, A. Lenz, and A. Wachter-Zeh, "Variable-length bit mapping and error-correcting codes for higher-order alphabet PUFs," in *Proc. Int. Conf. Security Privacy Appl. Cryptograph. Eng.*, 2017, pp. 190–209.

[9] S. Baker and R. D Cousins, "Clarification of the use of chi-square and likelihood functions in fits to histograms," *Nucl. Instrum. Methods Phys. Res.*, vol. 221, no. 2, pp. 437–442, 1984.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Somerset, U.K.: Wiley, 2012.

[11] R. Hesselbarth, F. Wilde, C. Gu, and N. Hanley, "Large scale RO PUF analysis over slice type, evaluation time and temperature on 28nm Xilinx FPGAs," in *Proc. Int. Symp. Hardw. Orient. Security Trust (HOST)*, Washington, DC, USA, 2018, pp. 126–133.