# Implementation and Comparison of Conventional and Ordering Based RO-PUFs for Secret Key Generation

Giray Kömürcü
*National Research Institute of Electronics and Cryptology*
*TÜBİTAK, Kocaeli, Turkey*
*Email: giray.komurcu@tubitak.gov.tr*

Ali Emre Pusane, Günhan Dündar
*Bogazici University, Dept. of Electrical and Electronics Eng.*
*Istanbul, Turkey*
*Email: {ali.pusane, dundar}@boun.edu.tr*

***Abstract*—Physical Unclonable Functions (PUFs) are security primitives that have the capability of key generation on the fly. Ordering based Ring Oscillator (RO) PUFs are one of the best performing structures in terms of robustness, since key generation requires error-free bit streams. Even though many aspects of ordering based RO-PUFs have been analyzed in considerable detail in the literature, a full implementation has not been presented yet. Hence, the total area cost of the system is still in question. In this work, we first implement a conventional RO-PUF including an Error Correction Coding (ECC) block. Then, we present a full implementation of an ordering based RO-PUF. Finally, performance of conventional and ordering based RO-PUFs are compared and their advantages and disadvantages are discussed.**

***Keywords-PUF, Physical Unclonable Functions, Reliability, Robustness, Ring Oscillator, FPGA, Key Generation***

## I. Introduction

Physical Unclonable Functions (PUFs) provide economic and secure solutions in the areas of cryptographic key generation, IP protection, authentication, and ID generation with their capability of signature generation on the fly [1]. With this property, they eliminate the need for a non-volatile memory for ID and key storage purposes. Even though Optical PUFs and Coating PUFs are the first two structures proposed in the literature, their impracticality and expensive equipment requirement prevented wide usage of these primitives [2][3]. In spite of this, Silicon PUFs, such as Arbiter PUFs, SRAM PUFs, Ring Oscillator (RO) PUFs, Butterfly PUFs, and Glitch PUFs have drawn significant attention with their ease of integration and low cost [4]-[8].

The main working principle of PUFs depends on small mismatches present in the manufacturing process, which lead to the deviation of parameters such as doping concentration, threshold voltage, and oxide thickness. These deviations are the basis for the uniqueness, robustness, unclonability, and unpredictability properties of PUF structures. Certain PUF types, such as RO-PUFs, are convenient for FPGA implementations as well, since manufacturing imperfections are also present in FPGAs [9]. Robustness is a key feature of PUF circuits, which aims at minimizing the number of unstable bits at the output [10]. Since PUF outputs are generated depending on small imperfections in the IC, any temporal variation present in the system may easily result in generating unstable outputs [11]. Almost all PUF structures, except for ordering based RO-PUFs, are vulnerable to internal and external effects and generate noisy outputs. However, certain applications, such as key generation, require 100% robust outputs for correct operation. Adding an Error Correction Coding (ECC) block is a proper but costly solution for key generation systems that utilize noisy PUF circuits.

RO-PUFs, which are the most convenient type of PUFs for FPGA implementation, work relatively reliably under changing environmental conditions and are suitable for key generation applications [9][12]. A conventional RO-PUF compares the frequencies of two identical ROs for one bit output generation. In these systems, the output bit can be set to 0, if $RO_1$ is faster than $RO_2$, and can be set to 1, otherwise. Since applications require generation of certain length bitstreams, a number of identical ROs are implemented in the circuit and different pairs are selected via multiplexers for each output bit generation. Ordering based RO-PUFs generate outputs using the frequency ordering of a group of ROs. During the grouping step, ROs whose frequencies are adequately apart from each other are grouped together in order to prevent ordering changes due to environmental variations and noise. Despite the noisy nature of conventional RO-PUFs, ordering based RO-PUFs enable 100 % robust, noise-free outputs, and avoid the need for ECC in key generation [13]. In addition to this, they have the capability of high entropy extraction, enabling higher area and power efficiency than conventional RO-PUFs [13][14]. Another advantage of ordering based RO-PUFs is their high number of CRP support that has been introduced recently [15]. Despite these advantages of ordering based RO-PUFs, a full hardware implementation including the output generation mechanisms has not been presented in the literature yet.

In this work, our main aim is to determine the area cost of ordering based RO-PUFs with all required components and compare their area efficiency with conventional RO-PUFs. For this purpose, we first present an implementation of conventional RO-PUFs with an ECC block for 100% robustness that is required for key generation in Section II.

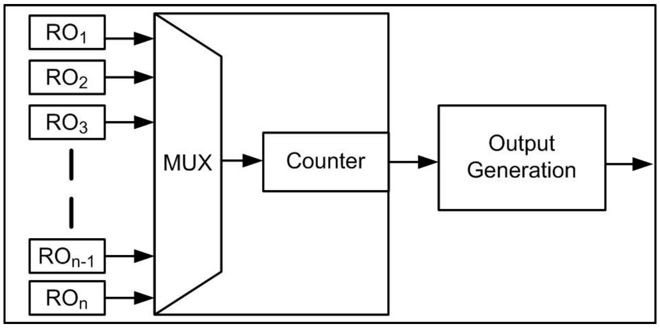


Figure 1. Block Structure of Conventional RO-PUFs.

Table I
AREA UTILIZATION OF FREQUENCY DETECTION CIRCUITRY FOR SPARTAN3 AND VIRTEX5 DEVICES.

| FPGA Type | 96 ROs | 128 ROs | 160 ROs | 192 ROs | 224 ROs | 256 ROs |
|---|---|---|---|---|---|---|
| Spartan3S | 40 | 48 | 57 | 65 | 73 | 81 |
| Virtex5 | 31 | 44 | 44 | 57 | 62 | 68 |

Next, a full implementation of ordering based RO-PUFs is presented in Section III. Performances of conventional and ordering based RO-PUFs are compared and their advantages and disadvantages are discussed in Section IV. Finally, Section V concludes the paper.

## II. IMPLEMENTATION OF CONVENTIONAL RO-PUFS AND ERROR CORRECTION CODES

Block structure of conventional RO-PUFs is presented in Figure 1. As can be seen from the figure, frequencies of implemented ROs are detected and output bits are generated depending on these frequencies. Frequency detection is a common step for both conventional and ordering based RO-PUFs and composed of a multiplexer and a counter. In this step, oscillation counts of all ROs are detected within a certain measurement time, $t_m$. With the proposed design, a multiplexer and a counter are implemented. Each RO is selected one-by-one with the multiplexer and their frequencies are detected with the counter. Six sample structures are implemented using combinatorial circuits for systems composed of 96, 128, 160, 192, 224, and 256 ROs. Area utilization results for Xilinx Spartan3 and Virtex5 FPGAs are presented in Table I. Maximum achievable frequencies for the proposed frequency detection circuit are 230 MHz for Spartan3 and 430 MHz for Virtex5 devices, which are significantly higher than the oscillation frequencies of 5-stage RO structures in both FPGA types. The output generation step is composed of a comparator to compare the oscillation counts and is implemented using 9 and 5 slices for Spartan3 and Virtex5 devices, respectively.

The last block required for 100% robust output generation using conventional RO-PUFs is ECC. The use of ECC in PUF implementations is illustrated in Figure 2. As can be seen from the figure, PUF output is applied to the ECC encoder and helper data is generated and recorded to a database during the initialization phase. Then, during the usage phase, ECC decoder removes the noise present in the PUF output by using the information stored in the helper data. Bose, Chaudhuri, and Hocquenghem (BCH) codes are convenient for data recovery in PUF circuits with their guaranteed error recovery for multiple errors. In this study, BCH codes are implemented and analyzed in terms of area and timing performance.

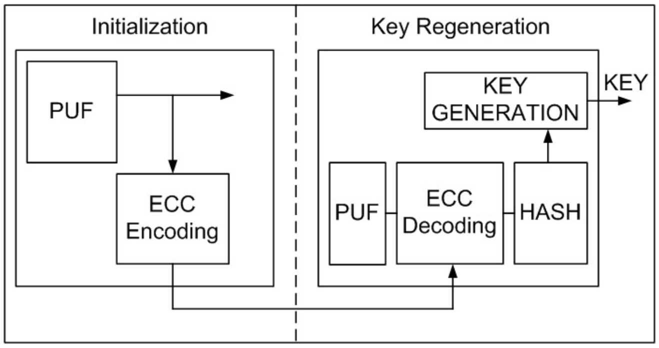


Figure 2. Key Generation Schematic with Conventional RO-PUFs.

Table II
AREA UTILIZATION OF ERROR CORRECTION CODES FOR SPARTAN3 AND VIRTEX5 DEVICES.

| Err. Cor. Capabilty | (255, 231,3) | (255, 207,6) | (255, 187,9) | (255, 163,12) | (255, 139,15) | (255, 131,18) |
|---|---|---|---|---|---|---|
| Enc. Sp. | 20 | 31 | 36 | 44 | 58 | 60 |
| Dec. Sp. | 223 | 334 | 471 | 581 | 705 | 843 |
| Enc. Vir. | 17 | 19 | 21 | 25 | 33 | 33 |
| Dec. Vir. | 148 | 178 | 272 | 288 | 363 | 427 |

The capabilities of multi-bit correcting ECC are shown with a three item notation, $(a, b, c)$. In this format, $a$ represents the total number of data and helper data bits, $b$ represents the total number of data bits, and $c$ represents the maximum number of erroneous bits that ECC can recover successfully in a noisy data. As the number of maximum number of erroneous bits that can be recovered increases, the complexity; hence, the area, time, and power consumption of both ECC encoder and decoder increase as well.

In order to determine the area overhead of ECC on PUF systems, BCH encoders and decoders for different error correction capabilities are implemented and their area usages are analyzed. In all systems considered, $a$ is selected as 255 bits. Results are presented in Table II. As can be seen from the table, area usage increases as the error correction capability increases. For instance, 3 bit correcting BCH decoder consumes 223 slices, whereas 18 bit correcting BCH decoder consumes 843 slices on Spartan3 FPGAs. Since the implemented conventional RO-PUF may result in up to 18 bits of errors, $(255, 131, 18)$ BCH encoder and decoder seems ideal for this case [10].

## III. Implementation of Ordering Based RO-PUFs

As mentioned previously, the main advantages of ordering based RO-PUFs are their 100% robust output generation capability and high entropy extraction. Even though the number of required ROs for the generation of certain length outputs is significantly reduced with ordering based RO-PUFs compared to the conventional structures, analysis of the output generation mechanisms in terms of area and speed will be beneficial for a fair comparison. For this purpose, ordering and output generation circuits are developed and implemented for different number of ROs and group lengths.

The output generation mechanism of the proposed ordering based RO-PUF is illustrated in Figure 3. According to this structure, it is assumed that grouping is done either by a PC during the initialization step and resulting groups are stored in a memory on-chip or off-chip, or done by a microprocessor present on the IC. Determining the ordering of ROs in a group and generating the output depending on this ordering are mandatory steps in ordering based RO-PUFs and are critical for the performance and cost of the system. This step can be performed using a microprocessor already present in the system, or by implementing a dedicated hardware. Assuming a microprocessor is not present in the system, dedicated hardware blocks are designed and implemented for ordering and output generation steps. Ordering of the oscillation counts is performed sequentially. RO IDs and their counts are stored in an array of registers in increasing order of the oscillation counts. Ordering of four ROs are illustrated in Figure 4. Execution time of ordering the circuits is upper-bounded by $m^2/2$ for a group of $m$ oscillators. However, since the ordering can overlap with the frequency detection of ROs, only the ordering time of the last group will reduce the speed of the operation.

Output generation of the ordering based RO-PUF is performed by mapping each ordering to a different bitstream using a sequential circuit. In this step, RO IDs and ordering information are used together. Pseudo code of the output generation is presented in Figure 5 and output generation of a group of four ROs is illustrated in Figure 6. Execution time of the ordering circuit is upper-bounded by $m$ for a group of $m$ oscillators. Similar to the ordering case, only the output generation time of the last group will reduce the speed of the operation.

## IV. Implementation Results and Comparison

Since measuring the ROs one-by-one is a good design practice to prevent the inter-locking of ROs, implementing one ordering detection and output generation circuit according to the largest group present in the system is the most convenient way for ordering based RO-PUFs. In this method, an upper-bound for the group lengths is set and the grouping step forms the groups according to this upper-bound. The proposed ordering and output generation circuits are implemented for different group lengths in the range of 3 to 10 and their area utilization results are presented in Tables III and IV. As can be seen from the tables, required resources increases immensely for ordering and output generation circuits as the group lengths increase.

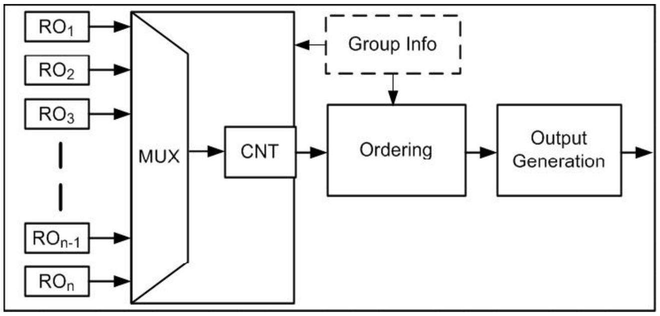
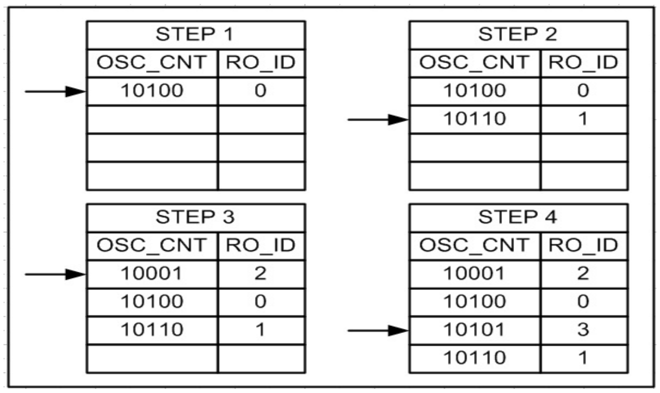


Figure 3. Block Structure of Ordering Based RO-PUFs.



| STEP 1 | |
|---|---|
| OSC_CNT | RO_ID |
| 10100 | 0 |
| | |
| | |
| | |

| STEP 2 | |
|---|---|
| OSC_CNT | RO_ID |
| 10100 | 0 |
| 10110 | 1 |
| | |
| | |

| STEP 3 | |
|---|---|
| OSC_CNT | RO_ID |
| 10001 | 2 |
| 10100 | 0 |
| 10110 | 1 |
| | |

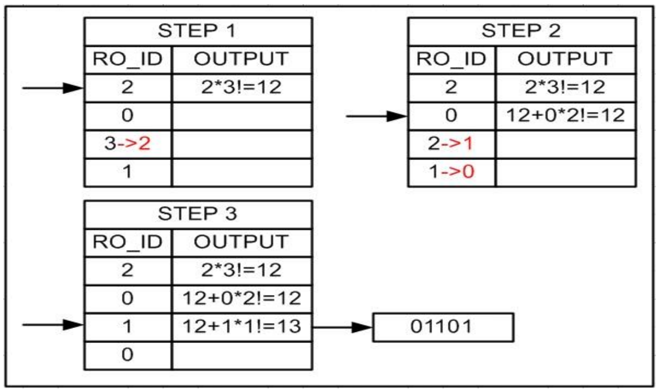
| STEP 4 | |
|---|---|
| OSC_CNT | RO_ID |
| 10001 | 2 |
| 10100 | 0 |
| 10101 | 3 |
| 10110 | 1 |

Figure 4. Ordering circuit sample execution.

Total number of slices for the generation of 128 bit outputs using conventional RO-PUFs and ordering based RO-PUFs with different maximum group lengths are presented in Tables III, IV and Figure 7. As can be seen from the tables, the required number of ROs decreases with increasing maximum allowed group lengths due to the more and more entropy extraction. These values are obtained from a Matlab analysis and rounded up for a safety margin. According to the presented results, ordering based RO-PUFs with maximum group lengths of 3 and 4 seem to be the optimum case for Spartan3 and Virtex5 devices, respectively, for the area performance of the system. Increasing the group lengths more than the indicated values does not contribute to the overall performance due to the increasing cost of ordering and output generation circuits. It should be also noted that the area performance of the conventional circuit is significantly worse than the ordering based structure due to the high cost of ECC implementation. However, this step can not be eliminated for the applications that require 100% reliable outputs.

```
Data:
List of RO IDs in a group sorted according to their frequencies, $RO[m]$.
Result: Output bitstream.
for $i \leftarrow 1$ to $m - 1$ do
    Output = Output + RO[i]*(m-i)!
    for $j \leftarrow i$ to $m - 1$ do
        if RO[i] < RO[j] then
            Decrement RO[i]
        end
    end
end
```

Figure 5. Output generation in pseudo code.



STEP 1

| RO_ID | OUTPUT |
|---|---|
| 2 | 2*3!=12 |
| 0 | |
| 3->2 | |
| 1 | |

STEP 2

| RO_ID | OUTPUT |
|---|---|
| 2 | 2*3!=12 |
| 0 | 12+0*2!=12 |
| 2->1 | |
| 1->0 | |

STEP 3

| RO_ID | OUTPUT |
|---|---|
| 2 | 2*3!=12 |
| 0 | 12+0*2!=12 |
| 1 | 12+1*1!=13 |
| 0 | |

01101

Figure 6. Output generation sample execution.

## V. Conclusion

Ordering based RO-PUFs are recently developed promising structures with their 100% robust output generation capability, high entropy extraction, and suitability to FPGA implementations. However, a full implementation has not been yet presented, preventing a fair comparison with conventional RO-PUFs. In this work, we have investigated the area cost of both conventional and ordering based RO-PUFs in detail for two different FPGA types. According to the analysis results, ordering based RO-PUFs with small group seems to be the best performing structures for generating robust outputs.

Table III
Area Utilization of RO-PUFs for Spartan3 Devices.

| PUF Type | RO Num | RO Slice | F. Det. Slice | Ord. Slice | O. Gen. Slice | ECC Slice | Total Slice |
|---|---|---|---|---|---|---|---|
| Conv. | 256 | 512 | 81 | 0 | 9 | 903 | 1505 |
| OB(3) | 195 | 390 | 73 | 28 | 10 | 0 | 501 |
| OB(4) | 185 | 370 | 65 | 57 | 16 | 0 | 508 |
| OB(5) | 175 | 350 | 65 | 97 | 36 | 0 | 548 |
| OB(6) | 170 | 340 | 65 | 128 | 57 | 0 | 590 |
| OB(7) | 165 | 330 | 65 | 163 | 82 | 0 | 640 |
| OB(8) | 160 | 320 | 57 | 213 | 98 | 0 | 688 |
| OB(9) | 155 | 310 | 48 | 260 | 175 | 0 | 793 |
| OB(10) | 150 | 300 | 48 | 336 | 210 | 0 | 894 |

Table IV
Area Utilization of RO-PUFs for Virtex5 Devices.

| PUF Type | RO Num | RO Slice | F. Det. Slice | Ord. Slice | O. Gen. Slice | ECC Slice | Total Slice |
|---|---|---|---|---|---|---|---|
| Conv. | 256 | 512 | 68 | 0 | 5 | 460 | 1045 |
| OB(3) | 195 | 390 | 62 | 10 | 7 | 0 | 469 |
| OB(4) | 185 | 370 | 57 | 26 | 10 | 0 | 463 |
| OB(5) | 175 | 350 | 57 | 49 | 15 | 0 | 471 |
| OB(6) | 170 | 340 | 57 | 54 | 23 | 0 | 474 |
| OB(7) | 165 | 330 | 57 | 71 | 45 | 0 | 503 |
| OB(8) | 160 | 320 | 44 | 114 | 56 | 0 | 534 |
| OB(9) | 155 | 310 | 44 | 117 | 98 | 0 | 569 |
| OB(10) | 150 | 300 | 44 | 181 | 123 | 0 | 648 |

## References

[1] 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.

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

[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, 2005, pp. 1200–1205.

[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] 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.

[9] A. Maiti and P. Schaumont, “Improved ring oscillator PUF: An FPGA-friendly secure primitive,” Journal of Cryptology, vol. 24, no. 2, 2011, pp. 375–397.

[10] G. Komurcu and G. Dundar, “Determining the quality metrics for PUFs and performance evaluation of two RO-PUFs,” in IEEE 10th International New Circuits and Systems Conference, (NEWCAS), 2012, pp. 73–76.

[11] A. Maiti, L. McDougall, and P. Schaumont, “The impact of aging on an FPGA-based physical unclonable function,” in International Conference on Field Programmable Logic and Applications (FPL), 2011, pp. 151–156.

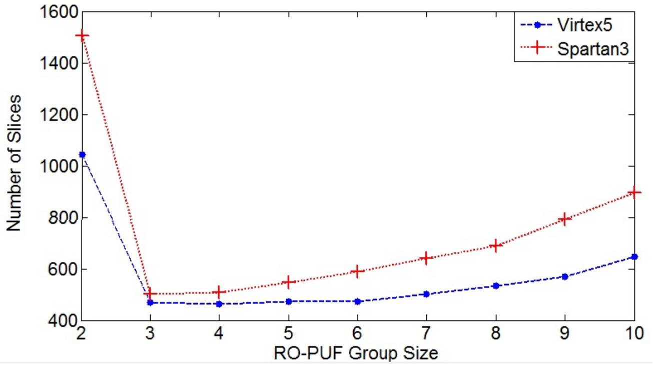


Figure 7. Area Utilization of RO-PUFs.

[12] 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.

[13] 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.

[14] 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.

[15] G. Komurcu, A. E. Pusane, and G. Dundar, “Enhanced challenge-response set and secure usage scenarios for ordering based RO-PUFs,” Devices, and Systems, (IET-CDS) , vol. 9, no. 2, 2014, pp. 87–95.