2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

Novel Randomized & Biased Placement For FPGA Based Robust Random Number Generator with Enhanced Uniqueness

Arjun Singh Chauhan
Dept. of ECE ,MNIT Jaipur
2015rec9055@mnit.ac.in

Vineet Sahula
Dept. of ECE, MNIT Jaipur
sahula@ieee.org

A. S. Mandal
Cognitive Computing Group, CEERI Pilani
atanu@ceeri.ernet.in

***Abstract*—The physical unclonable functions (PUF) have widely been used to provide software as well as hardware security for the cyber-physical systems. They are used for performing significant cryptography tasks such as generating keys, device authentication and securing against IP piracy. They have also been used to produce the root of trust. However, they lack in reliability metric. We present a novel approach for improving the uniqueness as well as the reliability of field programmable gated arrays (FPGAs) based ring oscillator PUF and derive a random number, consuming a very small area concerning look up tables (LUTs). We use profiling method for observing frequency variations in ring oscillators (RO), spatially placed across the FPGA floor, and are able to spot the suitable locations for RO mapping, which leads to enhanced ROPUF reliability. We have implemented proposed methodology on Xilinx -7 series FPGAs and tested the robustness against environmental variations e.g. temperature and supply voltage variations. The proposed approach achieves 6% higher uniqueness of 49.83% along with the reliability of 99.35%, which as a group of PUF characteristics, is a significant improvement as compared to characteristics provided by existing ROPUF methods. The random number generator so realized passes all applicable nine tests of NIST uniformity statistical test suite.**

***Index Terms*—Hardware Security, PUF, Ring Oscillator, FPGA, Random Number Generator, Biased Placement, NIST Statistical Test.**

I. Introduction & Related Work

Physical unclonable functions are used to generate a random unique identification sequence from hardware, which is used to enable security of the hardware devices. These functions exploit process variability of the silicon-based chips to produce a n bit binary identification sequence from each device i.e. FPGAs and application specific integrated circuits (ASICs). These identification sequences are used to provide the essential level of hardware security against IC overbuilding and IP piracy [1]; moreover it can also be employed in cryptographic keys generation.

The physical unclonable functions are used to protect against device tempering, for cryptographic key generation and it also enables the other security solutions for the FPGA devices. The cryptography methods require a key to perform encryption/decryption and this key should be different among devices. There are many existing software and hardware methods to generate cryptography keys, but keys are vulnerable, traceable and not secure.

The physical one way function and physical unclonable functions were introduced by Pappu et al. [2] and S. Devadas et al. [3]. These functions are capable of generating a unique signature from the hardware. Many PUF designs has been reported in the literature e.g. memory-based [4], arbiter-based PUF (APUF) [5], latch-based [6] and butterfly PUF [7].

Ring oscillator physical unclonable functions (ROPUF) are quite popular due to simple construction and are less affected by routing skew. Consequently, it offers better reliability in comparison with APUF [3]. Each Ring Oscillator (RO) cell produce a different frequency, which is considered as the source of random variation. The ROPUF generated keys are less reliable because ROs are prone to environmental fluctuations (i.e. supply voltage, temperature and device aging). These variations may cause unreliable bits in the response.

There have been many proposals reported in the literature, made to improve the reliability of ROPUF through configurable approaches [8] [9]. These approaches offer better reliability by utilizing extra LUTs. The other compact implementation of ROPUF is proposed, and it requires comparatively less area on FPGA devices, but the effect of environmental variations are not reported [10]. The author J. Gan et al. introduced a LUT based self compare structure, which improves the uniqueness with significantly less area but it requires post-processing for stable response selection [11].

Many researchers have addressed the PUF reliability issues, but many of these approaches require extra hardware, some of them need post-processing, and some of them are unable to provide the required reliability.

The authors present an approach to improve ROPUF reliability by using the biased placement of ring oscillators [12]. The approach uses the biased place & route to increase ring oscillator frequency difference using RO frequency characterization. The reliability improvement through this method is significant. However, the proposed approach is limited concerning uniqueness and uniformity.

We propose a novel technique by addressing uniqueness and uniformity issues and introducing certain augmentations to derive the random number from the proposed design.

The rest of the paper is organized as follows. The scope of the present work and manuscript contribution are discussed in section II and III. In section IV we have discussed prelimi-

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00079

naries and ROPUF modeling. The proposed hardware design, method and experimental results are presented in sections V, VI and VII, respectively. We have concluded in section VIII.

II. Scope of the Present Work

The reliability of our previously proposed method is good but the evaluation method is limited to the consideration of temperature variations only [12]. The supply voltage variation affects ring oscillator frequency in a significant manner, which is not addressed.

The previously reported design has a low area and is able to manage 2^N number of responses, where each ring oscillator is compared to other ring oscillators only for once. The ring oscillator selection is dependent on the applied challenge, which is traceable and not enough to pass NIST random number statistical tests.

This method uses biased ring oscillators, which improves the reliability but on the other side results in degradation of other metrics, i.e. uniformity and uniqueness. The reduction in the uniqueness leads to misidentification of the devices.

These issues are addressed and have been solved in the proposed approach.

III. Contribution of the Manuscript

The contribution of the paper is as follows.

- The ring oscillator allocation scheme is employed along with the routing validation to preserve the uniformity in ROPUF responses, It is observed that all the generated responses pass NIST statistical test.
- Following the routing validation method, the reliability improvement is significant and the robust testing has been performed in the presence of temperature and supply voltage variation.
- The location-based irregularities are inserted during placement of the ring oscillators, which improves the overall uniqueness.
- A sequence generator based on the Galois linear feedback shift register (GLFSR) is used to produce a sequence of random signature suitable for device authentication.

IV. Preliminaries

ROPUF is used to derive the unique random signature from each device using silicon device based manufacturing process variation. ROPUF is designed with total $2M$ ring oscillators, and it is assumed that each ring oscillator produces a different frequency due to manufacturing variations. The propagation delay τ_g of a logic element due to manufacturing variation is manifested in (1).

$$\tau_g = \Phi(V_{th}, W, L, \dots) \tag{1}$$

Here, Φ is the multivariate Gaussian distribution and it is dependent on the process parameters i.e. threshold voltage (V_{th}), the channel length (L), width (W), etc. The τ_g is composed of mainly three components; 1.) nominal delay component τ_g^{nom}, 2.) systemic delay component τ_g^s and, 3.) random delay component τ_g^r. The propagation delay of i^{th} logic element is represented in (2).

$$\tau_{g_i} = \tau_{g_i}^{nom} + \tau_{g_i}^{s} + \tau_{g_i}^{r} \tag{2}$$

The propagation delay (D_{ro}) of n logic element based ring oscillator is manifested in (3).

$$D_{ro} = \left(\sum_{i=1}^{n} \left(\tau_{g_i}^{nom} + \tau_{g_i}^{s} + \tau_{g_i}^{r} \right) \right) + D_w \tag{3}$$

Here D_w is the path delay of the ring oscillator. The nominal value appears due to the effective value of process parameters. The systemic delay component arises due to non-uniformity during the fabrication whereas the random delay component is present due to irregular doping concentration in transistors. Therefore each ring oscillator produces a distinct frequency and further it is used to generate random signatures.

V. Proposed Hardware Design

The proposed hardware design has total $2M$ ring oscillator cells, and two multiplexers are used to select two ring oscillator cells out of $2M$ with the help of select lines (also termed as Challenges) as shown in Figure 1.

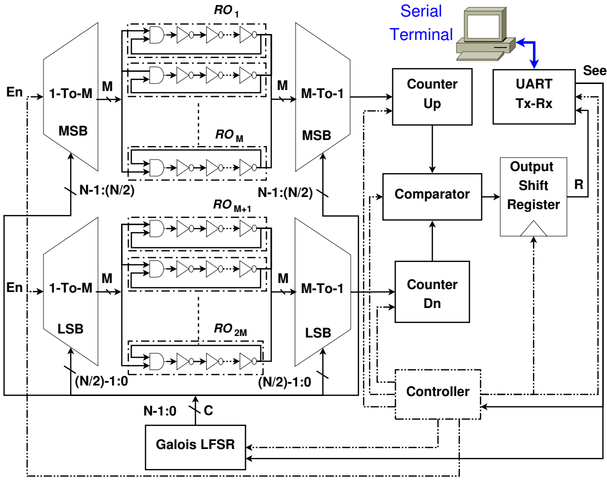


Fig. 1: Proposed hardware design to produce random sequence.

Each selected ring oscillator is connected to a conventional counter which counts the pulses and finally, a comparator is used to produce the binary output, termed as ROPUF response. The enable signal is applied using a de-multiplexer to facilitate a single ring oscillator at an instance, which reduces self-heating. The propagation delay difference for a pair of ring oscillator is written as (4).

$$\Delta D_{ro} = \Delta D_g^s + \Delta D_g^r + \Delta D_w \tag{4}$$

Here ΔD_g^s, ΔD_g^r and ΔD_w are the systemic, random and path delay difference between a pair of ring oscillators. The counters are used to count the value of ring oscillator pulses (α) and the pulse difference ($\Delta\alpha$) is obtained through comparator design. The pulse difference is proportional to the ring oscillator frequency difference (Δf). If the frequency difference is $\Delta f \geq 0$ then the response is 1 else 0.

The proposed design consist of ROPUF cell and a challenge generator, which is used to produce the random sequences of $2^N - 1$ bit from a single seed value, here $N = 2log_2 M$. A maximal cycle GLFSR has been used to provide the series of challenges to ROPUF design, and a controller provides control signals to the GLFSR [13]. The controller timing sequence is manifested in Algorithm 1.

Here en_{lfs}, en_{ro}, en_{osr}, en_{cnt} and en_{cmp} are the enable signal for GLFSR, ring oscillators, counter, output shift register and comparator unit, respectively. Here t_{on} and t_{ofs} are the counter on and counter offset pulse duration. The t_{ofs} is defined as the time required for the settling of the RO pulses, which reduces the transient effect of ring oscillator on output pulse. The function ***resetAllEnable*** clears the value of all the enable signals.

Algorithm 1 Pseudocode for controller design

```
begin
    Loop Forever
        if $S \neq 0$ then
            resetAllEnable();
            for $(i = 1 \text{ to } (2^N - 1))$ do
                $en_{\text{lfs}} = en_{\text{ro}} = 1$;
                wait($t_{ofs}$);
                $en_{\text{cnt}} = 1$;
                wait($t_{on}$);
                $en_{\text{cmp}} = 1$;
                $en_{\text{osr}} = 1$;
                resetAllEnable();
            end
        end
    EndLoop
end
```

The controller is configured in such a way that it requires $0.16ms$ to produce a single bit response (data transmission time is not included). Thus, we need at least $43ms$ to produce 255 bit length sequence for 8 bit GLFSR (The zeroth challenge is not used). GLFSR is used to enhance the obfuscation in the generated sequence. A 16-bit length counter is utilized to reduce the counter resolution error.

VI. Details of The Proposed Method

The proposed method is the extension of the previous work by addressing most of the limitations and providing their solutions [12]. The flow of the proposed method is shown in Algorithm 2.

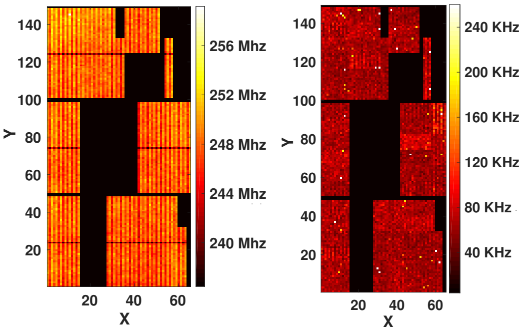
The proposed approach is completed in two different phases. The first phase is used to collect frequency data using on-chip frequency monitors. The captured frequencies are transmitted to *MATLAB* software through universal asynchronous receiver-transmitter *(UART)* communication protocol. Further, in the second phase, the data analysis and PUF constraints generation are reformed. The complete approach is automated using a software and hardware interface (*VIVADO-TCL-MATLAB*) designed using tool command language *(TCL)* scripting. The software communication is performed using *MATLAB-TCL*, and the hardware-software interaction is performed using standard *UART* protocol.

Algorithm 2 Work flow of the proposed method

```
Phase-1 : Frequency Variation Profiling
begin
    1: Generate Place & Route Constraint
    2: Capture Frequency Data using Monitor
end
Phase-2 : ROPUF Creation
begin
    1: Erroneous Frequency Rejection
    2: Clustering Based Grouping
    3: Group Frequency Selection
    4: Ring Oscillator Allocation
    5: Routing Path Validation
end
```

A. Phase-1: Frequency Variation Profiling

The frequency variation monitors are arranged to capture frequencies of ring oscillators, which are distributed across the entire FPGA chip [12]. We have placed a total 1024 ring oscillator instances at a time and the process is repeated six times to cover 6144 slices. Ring oscillators are constrained not to occupy the central region of FPGA fabric because we have appropriated it for other required circuitry. Figures 2a and 2b represent the mean and standard deviation heatmap of RO frequencies. The mean value is varied from 240 MHz to 278 MHz, and the standard deviation value is varied from 30.70 KHz to 381.26 KHz.



(a) Mean Value (b) Standard Deviation

Fig. 2: Heatmap of Frequency mean and standard deviation values for Artix-7 FPGA device

The ring oscillator placement at the different type of slices provides different routing path length, e.g. Slice-M and Slice-L offer different path lengths. Therefore simultaneously placed ring oscillators to all kind of slices introduce placement bias which significantly increases the frequency range [12].

B. Phase-2: ROPUF Creation

The second phase adaptively searches for the best suitable location which maximizes the pairwise frequency difference. The frequency difference for i^{th} pair of ring oscillator in the absence of environmental variation is $\Delta \widehat{f_i}$, thus the frequency difference for k^{th} environmental condition is expressed as (5).

$$\Delta f_i = \Delta \widehat{f}_i + \varepsilon_i^k, \varepsilon_i^k \in \mathcal{R} \tag{5}$$

Here ε_i^k is the frequency difference drift appears due to environmental variation. Therefore, the essential condition required to avoid unstable response generation is manifested in (6).

$$|\Delta \widehat{f}_i| \geq max\left(|\varepsilon_i^k|\right), \forall k \in e \tag{6}$$

Here e is the all possible environmental conditions. The estimation of $max\left(|\varepsilon_i^k|\right)$ is practically not feasible. Consequently, we have designed an adaptive approach to increase $\Delta\widehat{f}_i$. The following steps have been used to increase $\Delta\widehat{f}_i$.

1) Erroneous RO Rejection: The highly deviated ring oscillator frequencies are identified as erroneous RO frequencies and should be discarded. The remaining error-free frequencies (F_S) are evaluated using (7).

$$F_S = \left[x : \frac{std(F_x)}{mean(F_x)} \leq Th\right], \forall x \,\epsilon\, F_{RO} \tag{7}$$

Here, F_{RO} is the RO frequencies obtained from characterization. The parameter Th is defined as the maximum allowable value of the normalized standard deviation.

We have used normalized standard deviation instead of standard deviation to reduce the effect of mean value on the standard deviation. We have analyzed the data of 34 FPGA boards with the conclusion that the maximum of 1% ring oscillator frequencies show bit flipping error. Therefore, we have fixed Th such that 2% of the total frequencies are discarded and not used for further processing.

2) Clustering Based Grouping: The obtained error-free ring oscillator frequencies (F_S) are randomly distributed across the frequency scale. The proposed ROPUF design has total $2M$ ring oscillators, thus we need to choose a total of $2M$ frequencies such that each frequency should have the significant distance with the adjacent neighbor frequency. To perform grouping we have used *k-means* clustering based approach which increases the frequency difference between $2M$ groups [14].

The effectiveness of the proposed grouping based approach is evaluated by comparing it with some other techniques, i.e. mean based frequency selection (Each selected frequency has same frequency difference with neighbors), median based frequency selection, random frequency selection.

We have used 2560 biased ring oscillators and applied *k-means* clustering. The obtained centroid values are used to evaluate the minimum pairwise frequency difference. The *k-means* based approach substantially increases the minimum pairwise frequency difference for $M > 8$, and it is comparable for M $\leq$ 8 as shown in Figure 3.

The qualitative performance of the placement bias is evaluated by applying *k-means* on 2560 biased and unbiased ring oscillators separately. The desired minimum pairwise frequency difference is increased with biased RO frequency as shown in Figure 3. The increment in frequency difference is about 2.1 times as compared to unbiased placement. This increment improves the ROPUF reliability.

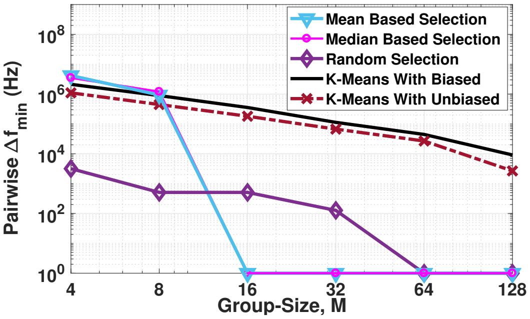


Fig. 3: The comparative analysis of minimum pairwise frequency difference employing different frequency selection approaches for both biased and unbiased placement.

The results obtained after the clustering represents $2M$ groups and each group has a set of frequencies with a centroid frequency $\left(f_{GC}^i\right)$ and the set of all ring oscillator frequency instances in a corresponding i^{th} group $\left(F_G^i\right)$.

3) Group Frequency Selection: Each group consists of multiple frequencies and a centroid. There are possibilities that centroid frequencies are not associated to any ring oscillators, therefore we select a single frequency from each i^{th} group, i.e. f_{GPUF}^i, such that the selected group frequency has minimum distance with group centroid frequency f_{GC}^i. This step increases the frequency difference between two consecutive ring oscillator frequencies. This is performed using (8).

$$f_{GPUF}^i = min\left[\left|f_{GC}^i - f_{G_x}^i\right|\right], \forall x \,\epsilon\, F_G^i \tag{8}$$

Where $f_{G_x}^i$ is the frequency of x^{th} RO in i^{th} group. The each obtained frequency is associated with a ring oscillator.

4) Ring Oscillator Allocation: All the selected $2M$ ring oscillators frequencies (f_{GPUF}) are placed into two groups; upper group (UG) and lower group (LG). The UG and LG locations are connected to upper and lower counters in ROPUF design, respectively. Each group contains M ring oscillator frequencies, such that $\frac{3M}{4}$ frequencies are uniformly allocated and remaining $\frac{M}{4}$ frequencies are randomly allocated out of M frequencies. The procedure to perform uniform allocation is expressed using (9) and (10).

$$F_{UG_{PUF}} = \left[\bigcup_{i=2j-1} F_{srt}^i\right] \bigcup \left[\bigcup_{i=\frac{5M}{4}+2j} F_{srt}^i\right] \tag{9}$$

$$F_{LG_{PUF}} = \left[\bigcup_{i=2j} F_{srt}^i\right] \bigcup \left[\bigcup_{i=\frac{5M}{4}+2j-1} F_{srt}^i\right] \tag{10}$$

Here, F_{srt} is the set of all mean values in sorted order and $j \in \{1, 2 \ldots, \frac{3M}{8}\}$, similarly $F_{UG_{PUF}}$ and $F_{LG_{PUF}}$ are the output frequencies of each group. This allocation scheme improves randomness.

These frequencies follow the sorting order consequently we have performed group based randomization (random placement of ring oscillators on FPGA floor), which is termed as placement based irregularity. These irregularities increase the ring oscillator location randomization, consequently, it improves the uniqueness.

5) ROPUF Path Validation: The proposed approach performs place & route (P&R) during both phases, thus it is necessary to validate the internal routing path delay of each ring oscillators during both phases to avoid routing mismatch which degrades the ROPUF reliability. The delay mismatch validation procedure is shown in Figure 4.

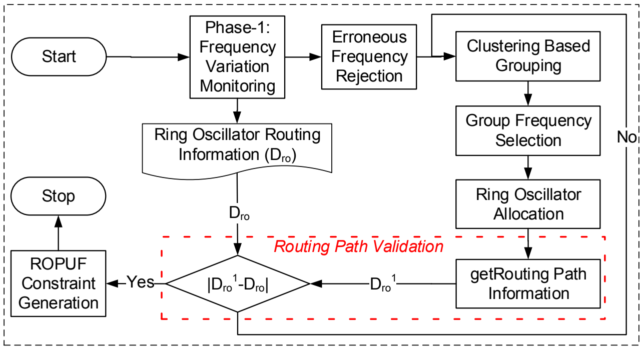

Start
Phase-1: Frequency Variation Monitoring
Erroneous Frequency Rejection
Clustering Based Grouping
Group Frequency Selection
No
Ring Oscillator Routing Information (D_{ro})
Ring Oscillator Allocation
Stop
D_{ro}
Routing Path Validation
ROPUF Constraint Generation
Yes
$|D_{ro}^1-D_{ro}|$
D_{ro}^1
getRouting Path Information

Fig. 4: Work Flow of proposed Routing Validation Scheme

VII. Experimental setup and Results

The proposed method is evaluated using standard metrics, i.e. uniqueness and reliability. The randomness is tested using NIST random statistical suite. We have used total 34 FPGA devices (24 *Nexys-4 DDR* and 10 *Basys-3* FPGA) and each of them able to produce 255 bit response from a single seed value. To apply the NIST statistical test we have created a bitstream of 8670 bits (255 × 34).

A. Experimental Setup

We have designed "Hardware-Software" based Interface to perform the complete experiment. The temperature variation is achieved with a temperature environment enclosure (*Espec SH-241 Temperature & Humidity Chamber*) and the range is kept in between $-5°C$ to $75°C$ with a step size of $10°C$. Similarly, supply voltage variation is achieved by modifying one *Basys-3* FPGA device in the range of $0.9V$ to $1.1V$ with 25mV step size. The real-time temperature and the supply voltage is monitored by *XADC* interfacing and *TCL*. The reference operating conditions are (T_r=25$°C$ and V_r=1V).

B. Reliability Evaluation

The reliability is defined for PUF response variation under different environmental conditions (e). The reliability metric reliability$_i$ of i^{th} device is evaluated using (11).

$$\text{reliability}_i = \left[1 - \left(\frac{1}{e}\sum_{j=1}^{e}\frac{HD(R_i, R_{i,j})}{k}\right)\right] \times 100\% \quad (11)$$

Here, R_i and $R_{i,j}$ both are the k bit response for the i^{th} device. R_i is the reference response and $R_{i,j}$ is the response obtained in the presence of the j^{th} environmental condition.

The reliability obtained by the proposed approach with supply voltage and temperature variation is summarized in Table I. The obtained reliability for M=16 represents that the proposed approach is suitable for device authentication and with some lightweight post-processing this scheme could further be used for cryptographic key generation.

TABLE I: % Reliability value variation in the presence environmental variation

Voltage variation			Temperature variation			%Avg
Reliability Value						
%Min	%Max	%Avg	%Min	%Max	%Avg	
98.03	100	99.01	99.21	100	99.69	99.35

We have used 32 RO based design (M=16), but it can be varied according to the requirement. There is a trade-off between the number of CRP and the maximum unreliable response bits in the presence of environmental fluctuations as shown in Figure 5. It is evident that the unreliable response bits decreases with the increment in the Group Size (M) or number of CRPs.

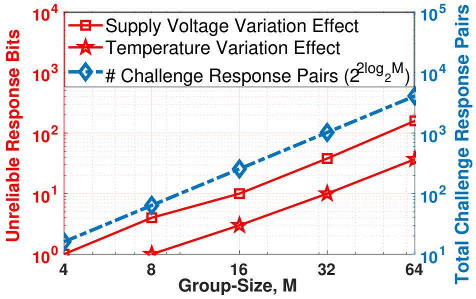


Fig. 5: Effect of Group-Size on unreliable response bits and total possible CRPs.

C. Randomness Test

The randomness of the response is evaluated by applying NIST statistical test suite [18]. The generated bitstream is passed through a total of nine suitable statistical tests such that the P-value of each test should be higher than 0.0001 to pass the test. The test results are illustrated in Table II.

The results describe that at least 32 sequences out of 34 have been passed for each test and it is evident that the proposed ROPUF produces a random sequence.

D. Uniqueness

The uniqueness is the important metric to differentiate among q number of PUF based on their generated signatures. The uniqueness metric (U) is evaluated using (12).

$$U = \frac{2}{q(q-1)}\left(\sum_{i=1}^{q-1}\sum_{j=i+1}^{q}\frac{HD(R_i, R_j)}{k}\right) \times 100\% \qquad (12)$$

Here R_i and R_j are k bit responses for i^{th} and j^{th} PUF, respectively. The uniqueness value for proposed ROPUF is

TABLE II: NIST Randomness Test Results

Test Name	P-Value	$^{(1)}$Prop
Frequency	0.028181	33/34
BlockFrequency	0.122325	34/34
Cumsums Forward	0.911413	32/34
Cumsums Reverse	0.043745	33/34
Runs	0.066882	34/34
LongestRun	0.122325	34/34
Entropy	0.534146	34/34
Serial	0.739918	34/34
Serial	0.082177	34/34

(1) Proportion Ratio.

TABLE III: Comparison of group of PUF characteristics of proposed scheme with those of earlier schemes

Method	Rel	U	RL	Resources	FPGA	Tech	ΔT	$\frac{\Delta V}{V_{ref}}$
ROPUF [15]	**99.52**	46.15	128	1024 ROs	Vertex-4	90	-	-
Maiti-CRO [16]	99.14	47.31	127	128 ROs	Spartan-3E	90	$40°C$	**40%**
Compact RO [10]	99.16	47.13	**256**	**32 Slices**	Spartan-6	45	-	-
PUF-ID [17]	$99.4^{\dagger}$	45.60	128	128 Slices	**Artix-7**	**28**	$75°C$	20%
Previous Work [12]	99.05	35.83	**256**	**32 Slices**	**Artix-7**	**28**	$40°C$	-
Proposed Work	99.35	**49.83**	255	**32 Slices**	**Artix-7**	**28**	$\mathbf{80°C}$	20%

† represents the post processing. **RL** represents the Response Length. Temperature Variation Range $\Delta T = T_{max} - T_{min}$ and Supply Voltage Variation Range $\Delta V = V_{max} - V_{min}$. **Tech** represents the fabrication technology used in nm.

about 49.83% which is near to ideal value of 50%. The uniqueness improvement is obvious because of the irregularities introduced during ring oscillator placement.

The average ROPUF inter-chip and intra-chip hamming distances for the responses are 0.4983 and 0.006, which is obtained from 34 FPGA devices during reference environmental condition is shown in Figure 6.

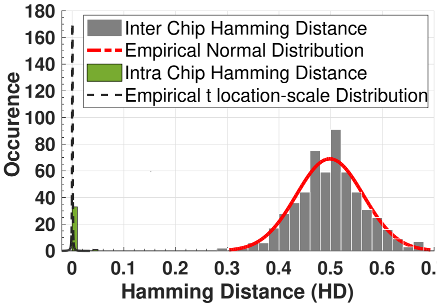


Fig. 6: The hamming distance histogram distribution .

E. Comparison

The proposed method is examined along with some of the existing approaches, and the comparison is reported in Table III. The proposed methodology has been implemented on *28nm* technology FPGA, and evaluated. It is evident that the proposed approach requires much less area and provides excellent uniqueness and reliability values.

VIII. Conclusion

The proposed method provides the significant improvement in reliability and uniqueness. The proposed approach has been primarily implemented and tested on *28nm-technology Xilinx* FPGA. The proposed method is a two-Phase approach, where the First phase requires additional time and it requires a trusted environment to perform frequency characterization on FPGA. Nevertheless, this process is a one time process and can be performed before PUF enrollment with a trusted environment.

ACKNOWLEDGEMENT

We also gratefully acknowledge *SMDP-C2SD project (ODRC No. 1000110086)* project funded by *Ministry of Electronics & IT, Government of India* for technical support.

We gratefully acknowledge CSIR-Central Electronics Engineering Research Institute (CSIR-CEERI), Pilani for providing us the *Temperature Enclosure*.

References

[1] M. Rostami, F. Koushanfar, and R. Karri. A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE*, 2014.

[2] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[3] J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *2004 Symposium on VLSI Circuits*, pages 176–179, June 2004.

[4] F. Tehranipoor, W. Yan, and J. A. Chandy. Robust hardware true random number generators using DRAM remanence effects. In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 79–84, May 2016.

[5] J. W. Lee et al. A technique to build a secret key in integrated circuits for identification and authentication applications. In *Symposium on VLSI Circuits. Digest of Technical Papers*, pages 176–179, June 2004.

[6] Dai et al. Yamamoto. Variety enhancement of PUF responses using the locations of random outputting RS latches. *Journal of Cryptographic Engineering*, 3(4):197–211, Nov 2013.

[7] S. S. Kumar, J. Guajardo, R. Maes, G. Schrijen, and P. Tuyls. Extended abstract: The butterfly PUF protecting IP on every FPGA. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*.

[8] Mingze Gao, Khai Lai, and Gang Qu. A highly flexible ring oscillator PUF. In *DAC '14*, 2014.

[9] S. R. Sahoo, S. Kumar, and K. Mahapatra. A modified configurable ro PUF with improved security metrics. In *ISNIS' 15*, 2015.

[10] N. N. Anandakumar, M. S. Hashmi, and S. K. Sanadhya. Compact implementations of FPGA-based PUFs with enhanced performance. In *2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID)*, 2017.

[11] J. Gan, J. Zhou, and N. Wang. A FPGA-based RO PUF with LUT-based self-compare structure and adaptive counter time period tuning. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS'18).*

[12] A. Singh Chauhan, V. Sahula, and A. S. Mandal. Novel placement bias for realizing highly reliable physical unclonable functions on FPGA. In *2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6, March 2018.

[13] Roy Ward and Tim Molteno. Table of linear feedback shift registers. Technical report, University of Otago, Box 56, Dunedin, New Zealand, http://courses.cse.tamu.edu/walker/csce680/lfsr_table.pdf, oct 2007.

[14] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. SODA '07, pages 1027–1035, 2007.

[15] Blaise Gassend, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. Silicon physical random functions. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002.

[16] Abhranil Maiti and Patrick Schaumont. Improved ring oscillator PUF: An FPGA-friendly secure primitive. *J. Cryptol.*, 24(2), April 2011.

[17] Chongyan Gu, Neil Hanley, and Máire O'neill. Improved reliability of FPGA-based PUF identification generator design. *ACM Trans. Reconfigurable Technol. Syst.*, 10(3), May 2017.

[18] Andrew Rukhin et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications, 2001.