A Highly Reliable Strong Physical Unclonable Function Design Based on FPGA

Yi Zhang, Min Zhu, Bohan Yang, Leibo Liu
Institution of Microelectronics, Tsinghua University, Beijing, China, 100084
yi-zhang17@mails.tsinghua.edu.cn

Abstract. Physical unclonable function (PUF) generates unique secret keys from unavoidable IC manufacturing variations, which can eliminate high computation expense and additional key storage. The inherent flexibility and lower time-to-market have made field programmable gate array (FPGA) the platform of choice for faster implementation. However, logic elements on FPGA are predefined while some types of PUFs need strictly mirrored symmetric routing. This paper presents a robust and FPGA friendly PUF based on oscillator collapse (OC-PUF) with million number of challenge-response pairs (CRPs). Mirrored routes are obtained with a novel delay cell design and a signal path configuration technique. In the meanwhile, the response bias issues on FPGA is effectively resolved. Measured on Altera DE2-115, the average inter-chip hamming distance (inter-chip HD) of the proposed OC-PUF is 46.7%. After applying dynamic threshold with a value 255, intra-chip hamming distance (intra-chip HD) dropped to 5.46E-06 at nominal conditions.

1. Introduction
Internet of Things (IoT) devices, integrating functions of computing, interconnection, and communication, play a very important part in our modern life. The security of IoT is becoming a major challenge. Conventional security authentications rely on secret keys, digital signatures, cryptographic algorithms, which cost huge computing resources. Secret keys need to be stored in an on-chip non-volatile memory. However, on-chip non-volatile memory (NVM) is vulnerable to invasive and non-invasive side-channel attacks [1]. This type of security problem is particularly challenging in some lightweight but critical devices.

Physical Unclonable Functions (PUFs) provide a low-cost and high-security method for chip authentication and cryptographic key generation and trusted computing [2]. They eliminate the expense of calculation steps and the risk of storing data by utilizing process variation among chips. PUFs can be considered as a black-box. Each input challenge \( C \) returns a response \( R \), which is called challenge-response pairs (CRPs) [3]. CRPs are hidden in the intrinsic physical parameters, just like fingerprints, expressing inherently unique features of circuits. Even with identical layout length, the delay accumulations can be significantly different due to the manufacturing process variations [4].

Compared with traditional ASIC, the inherent flexibility and lower time-to-market have made FPGA the platform of choice for faster implementation [5]. However, implementing a PUF on FPGA involves significant challenges [6]. On the one hand, transistor-level design (e.g. applying bias to a transistor) cannot be implemented on the FPGA due to their logic elements restrictions. On the other hand, many PUF layouts require a careful routing symmetry while doing lower-level control over
layout is not possible [7]. A typical example is the arbiter PUFs [8]. Although multiple PUF structures are proposed, not all of them are suitable for FPGA implementations [5].

In this work, a novel delay cell design and symmetric path configuration technique are proposed. The proposed technique improves the symmetry of the routing on FPGA.

This paper is organized as follows. Section 2 gives a brief review of the FPGA-based PUF, and discuss the challenges in the state-of-the-art FPGA-based PUF designs. Section 3 describes the working mechanism of oscillator collapse based PUF (OC-PUF) and the FPGA implementation of the proposed PUF. Section 4 presents the experimental results and analyses the properties of the structure. Section 5 summarizes the results and concludes the paper.

2. Background

2.1. Strong PUF and Weak PUF

The existing PUFs are usually divided into strong PUFs and weak PUFs according to the size of CRP space which determines the applications. Strong PUFs have a complex input-output relation with large possible challenge-response pairs. More specifically, the number of CRPs grows exponentially as the number of basic components increases. Weak PUFs have limited CRPs linearly related to the number of basic cells. In the worst case, there is just one, fixed challenge-response pair, for example, SRAM-PUF [2].

Because of the small number of CRPs, weak PUFs are usually used as an alternative to non-volatile key storage and impossible to be read out by invasively way [4]. Strong PUFs has a wider range of applications including key establishment [9], identification, authentication [10], zero-knowledge proofs, bit commitment, and secure multi-party computation [11], without using any cryptographic hardware.

2.2. FPGA-Based PUF

Memory-based PUFs are commonly used as weak PUFs for their limited number of CRPs which is proportional to their size. SRAM-PUF was proposed by Maes et al. in [12]. They generate one response by accessing to uninitialized blocks of SRAM. Unfortunately, most FPGA devices will initialize SRAM blocks by default at power-up and lose all the PUF reaction.

As for delay-based PUF, RO PUF [4] allows an easier implementation for FPGAs because of no need for symmetric routing. However, they are normally used as weak PUFs because their CRPs grow logarithmically with the circuit number. Bistable PUFs provide relatively balanced uniqueness and reliability [13], but they take 47us waiting for the output stabilizing in order to guarantee quality of response. The arbiter PUFs, proposed by Lee et al. [14], are the most popular strong PUF for implementation. The input challenges select the top and bottom signal paths by MUXes. Responses are determined by which path is faster. Arbiter PUF requires carefully matched wiring delays while the routing symmetry is nearly impossible on FPGA. The delay accumulation of the two paths is not dominated by the process variation, but the routing length, resulting in serious bias. Machida et al. proposed double arbiter PUF (DAPUF) [15] aiming to overcome the low uniqueness of arbiter PUF, which is based on a novel technique for generating responses from duplicated arbiter PUFs on FPGAs. But the reliability also decreased sharply.

2.3. Reliability Enhancement

Suffering from noise, responses cannot be used directly as key bits. Outputs are likely to be slightly different on each evaluation, even on the same circuit for the same challenge [4]. Reliability enhancement techniques are necessary to transform poorly noisy measurements into robust secure keys.

Majority Voting [16] derives reliable key values by multiple responses voting, through repeating the measuring (time cost) or by multiple individuals (spatial) at once. It can effectively reduce the impact of noisy measurements. Error Correction Codes (ECC) [17] is a typical method for enhancing stability.
It will cost additional hardware overhead as well as non-volatile memory due to its ECC algorithm computation and data storage. Fuzzy extractors [18] are widely used to extract secure keys from the PUF responses, since raw responses are noisy and not fully random. There are two major steps: correcting possible bit errors and extracting uniform random bits from the random PUF responses after correction. The better stability is achieved at the expense of some response bits. Helper Data Algorithm (HDA) [19] can also help obtain high quality keys. In short, all the stabilizing approaches above are able to deal with the fuzziness of responses and transform noisy responses into credible bits. However, additional testing and calibration effort are inevitable [20]. Therefore, the reliability enhancement circuit also needs carefully design to reduce the overhead.

3. PUF Based on Oscillator Collapse

In this section, we introduce the concept of oscillator collapse based PUF inspired by [20]. Compared to the other FPGA-based strong PUF, the proposed OC-PUF is of higher reliability. Also, it is easier to achieve symmetric paths on FPGAs and generate large number of robust CRPs.

3.1. Concept of oscillator collapse

Figure 1 illustrates a ring oscillator based on multi-edge oscillation collapse. OC-PUF is turned on by injecting two edges into an even-stage RO. Due to manufacturing variation, although the two injected edges travel two symmetrical paths, the speeds are slightly different because of accumulating delay cell mismatch. The output eventually stabilizes because of even stages. The response settles on 0 or 1, determined by which path is faster and which edge collapses the oscillation. Delay cell per stage is selected from eight identical inverters so that the chain of the RO ring has been greatly shortened. 3-bit challenge selecting 1 stage enables a much larger number of CRPs than 1 stage selected by 1-bit challenge [20].

3.2. Delay Cell Unit

To ensure the symmetry of the two paths, the delay symmetry inside the logic unit is the priority. The ideal routing is that, the path of the signal passing through each inverter is as equal as possible, from input to output. Since the EDA tool automatically optimizes the circuit logic, so that the path from
input to output cannot be equal for each inverter. We have to do placement and routing manually. To guarantee symmetry from stage level, we place the 8 inverters as shown in Figure 1. For each path, the signal goes through three 2-to-1 multiplexers from input to output equally.

3.3. Path Configuration
Even if the inverters are placed in symmetric positions, the upper and lower path could still be biased because of the uncontrollable wires on logic elements. As shown in Figure 2, line 3 and line 4 are the signal paths without configuration per stage. However, without configuration, the signal route is asymmetrical in the bottom circuit of Figure 2.

In order to eliminate the bias and ensure the symmetry of the upper and lower paths, we propose a novel symmetric path configuration method, shown in the upper circuit in Figure 2. Line 1 and line 2 are the signal paths with the configuration. Although we could not precisely control the wiring in FPGA logic elements, we can ensure the delay accumulations are symmetric by using the same configuration. The techniques make it possible to implement a mirrored symmetric routing on FPGA.

3.4. Time Setting
Reliability is measured by how many responses can be regenerated to the same challenge from a single PUF. The more responses repeat, the better reproducibility of the PUF outputs. Ideally, the BER should be 0%. Reliability is related to the cycle time of the start signal in Figure 3 and the number of oscillations. Since the OC-PUF oscillates several times before generating one bit response, OC-PUF does not always stabilize in a predefined period [20]. If the period of the start signal is set too short, many configurations are terminated before stabilize and result in low stability. OC-PUF is more suitable for circuits with requirements for lower speed but higher reliability.

![Figure 2. Path Configuration.](image-url)
3.5. Dynamic Threshold

While two paths have nearly matched delay, it results in slow collapse results. Responses that collapse slower are more likely to occur bit-flip because they are determined by noise rather than by process variation. Therefore, we implement a simple dynamic threshold technique [20] to restrict the number of cycles to collapse. Cycles are recorded by a 9-bit counter. CRPs with collapse cycles larger than the threshold are discarded. The increase in stability takes only a 9-bit counter overhead [20]. The working mechanism of dynamic threshold is shown in Figure 4.

3.6. Enrollment and Verification Protocol

In the actual use of encryption, it requires that every bit of key stays constant. Noise may result in unpredictable change on each evaluation. Although we have implemented an effective dynamic threshold technique to ensure the reliability, we still need an authentication protocol to guarantee the outputs being consistently produced even in the changing environment. Figure 5 shows the overall authentication protocol [20] which can be divided into two stages: enrolment and verification. During enrolment stage, a chip is interrogated by random challenges, only CRPs with a collapse cycle smaller than the threshold are recorded. We call them golden CRPs. Only golden CRPs can enter the verification of the 2 stage.

During verification, golden CRPs are sent to PUFs under test. For the safe, collapse cycles are evaluated again. If the collapse cycle is larger than the threshold, the CRP is skipped and the next stored CRP is used. If the collapse cycle is smaller than threshold, the response is checked against the stored golden CRP. Responses which do not match golden response are discarded. The process is repeated k times. The number of k can be adjusted by security level. For example, if k=2, then repeat
twice verification to get consecutive correct response to pass the verification. Similarly, repeat 5 times if $k=5$. The higher the security level, the more repeated.

![Stage 1 Enrollment](image1.png) ![Stage 2 Verification](image2.png)

**Figure 5.** Authentication protocol with dynamic threshold [20].

4. Experimental Results

This section presents the experimental results of the proposed OC-PUF. We evaluate an 8-stage OC-PUF with $2^{24}$ CPR space on Altera DE2-115 FPGA boards on 48 different places at ambient temperature. The period of the start signal is around 20us. This section includes three important metrics: inter-chip HD (spatial uniqueness), intra-chip HD (temporal stability) and randomness.

4.1. Inter-chip HD

Inter-chip HD evaluate the responses of different PUFs to the same challenges. Inter-chip HD represents uniqueness. It is usually estimated as follows [13]:

$$\text{Inter-chip HD} = \frac{2}{m(m-1)} \sum_{i=1}^{m} \sum_{j=i+1}^{m-1} \frac{\text{HD}(R_i, R_j)}{n}$$

(1)

Where $m$ refers to the number of testing PUFs. $R_i$ and $R_j$ are two n-bit responses to the same challenge generated by PUF$_i$ and PUF$_j$. Hamming distance (HD) evaluates how many different bits between $R_i$ and $R_j$, two equal lengths binary vectors.

![Figure 6.](image3.png) ![Figure 7.](image4.png)

**Figure 6.** After implementing the same input configuration, the inter-chip HD comes around 46.7%.

**Figure 7.** Before implementing the same input configuration, the inter-chip HD is about 25.58%.
We measured 48 different PUFs on FPGA with 2000 challenges. Responses come from different PUFs but the same challenges. We compare the distance between every two responses vector and calculate the value. As shown in Figure 6, the average inter-chip HD is 46.7%. It means that two identical PUF on two different places produce a different output bit with a probability of 46.7%, close to the ideal 50% value. The closer to 50%, the harder to predict the response.

As shown in Figure 7, Before implementing the same input configuration, the inter-chip HD is about 25.58%, which results in huge bias. After implementing the same input configuration, the inter-chip HD comes around 46.7%, which is much better than the original one, close to the ideal 50% value.

4.2. Intra-chip HD

Intra-chip hamming distance (intra-chip HD) indicates the reproducibility of the PUF [4]. In other words, intra-chip HD evaluates the responses of a PUF to one challenge repeat thousand times. Ideally, the intra-chip HD should be 0%, which means the response to the same challenge is all the same without bit flipping. The definition[13] is as follows:

\[
\text{Intra-chip HD} = \frac{1}{m \sum_{t=1}^{m}} \frac{\text{HD}(R_{t}, R_{t}')} {n}
\]  

(2)

Where \(m\) and \(n\) are the repeat times and the n-bit response length respectively. \(R_{t}\) refers to response from the \(t\) repeat.

We evaluate 5000 challenges with each challenge were evaluated by 1000 times to the same PUF. The average intra-chip HD is 3.53% at nominal conditions without applying stability enhancement technique. After setting dynamic threshold by 255, the value dropped to 5.47E-06, almost equal to ideal value 0. It means that repeating one challenge 1000 times on the same PUF, only 5.47E-06 proportion responses occur bit flip (0.0055bit out of 1000, almost no error).

As shown in Figure 8, 5k random challenges had been sent to PUF at normal condition with each repeat 1k times. The gray part is the frequency distribution of the cycle value from all the CRPs. Red bars are the cycle value distribution of CRPs which occurred more than one bit flip during the process of the 1k repetition for each challenge.

![Figure 8. Distribution of oscillator collapse cycles.](image)

![Figure 9. Dynamic threshold.](image)

The OC-PUF experiences multiple oscillations before generating 1 stable response. After analysing all the cycle value, unstable CRPs is concentrated in the interval with cycle value larger than 255. As shown in Figure 9, the CRPs larger than the threshold are discarded. As the threshold decreases, inter-chip HD decreases gradually.

Before applying dynamic threshold, the intra-chip HD is around 3.56%. After applying dynamic threshold value 255, the intra-chip HD is drop to 5.47E-06. The reliability is improved at the expense of discarding 45% of CRPs which is acceptable for a strong PUF as the number of CRPs grows.
exponentially $2^{3N}$ with the number of stages. CRP space shrinks can be easily compensated by increasing series can be compensated by adding more stages.

4.3. Randomness

PUFs are vulnerable to cryptanalysis if the responses are not random enough. Randomness indicates the balance of 0 and 1 in the responses of the PUF.

We tested 1 million random responses generated by OC-PUF using the National Institute of Standard and Technology (NIST) statistical test package [21]. One million response bits divided into 100 blocks of 10000 bits each. Table 1 shows the results of NIST tests. The uniform distribution across columns "C1" to "C10" represents a uniform distribution of the frequency of various "P-values". The "P VAL" column shows the "P-value" obtained via a $\chi^2$ test. P-value > 0.01 conclude that the sequence is random. The "PROP" column lists the proportion of "0-1" sequences that passed the corresponding test. The last column indicates the names the tests.

Before applying dynamic threshold, 1 million random responses generated by OC-PUF have passed the (12/15) NIST test shown in Table 1. It indicates that the responses are random enough for a PUF. After applying dynamic threshold value 255, the ratio of 0 and 1 for the remaining 550k responses is 48.2%:51.8%. Response 1 is slightly more than response 0. However, the bias is not serious and seems acceptable.

Table 1. NIST Test Results on Random Sequences Generated by the OC-PUF.

| C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | P VAL | PROP | TEST     |
|----|----|----|----|----|----|----|----|----|-----|-------|------|---------|
| 18 | 11 | 7  | 10 | 9  | 6  | 9  | 6  | 10 | 14  | 0.191687 | 99/100 | Frequency |
| 6  | 7  | 12 | 10 | 7  | 5  | 13 | 21 | 12 | 0.01455 | 100/100 | Block Frequency |
| 18 | 10 | 8  | 13 | 7  | 11 | 6  | 10 | 11 | 0.213309 | 99/100 | Cumulative Sums |
| 18 | 11 | 12 | 5  | 11 | 5  | 3  | 11 | 10 | 10  | 0.01455 | 99/100 | Cumulative Sums |
| 11 | 8  | 17 | 14 | 5  | 7  | 11 | 9  | 9  | 9   | 0.289667 | 99/100 | Runs     |
| 9  | 6  | 19 | 8  | 11 | 8  | 15 | 6  | 11 | 7   | 0.071177 | 100/100 | Longest Run |
| 11 | 13 | 9  | 10 | 18 | 4  | 9  | 7  | 9  | 10  | 0.202268 | 97/100 | Rank     |
| 12 | 11 | 7  | 10 | 12 | 13 | 4  | 12 | 10 | 9   | 0.657933 | 98/100 | FFT      |
| 6  | 5  | 14 | 7  | 17 | 14 | 6  | 10 | 12 | 9   | 0.085587 | 99/100 | Overlapping |
| 7  | 12 | 10 | 20 | 8  | 10 | 5  | 8  | 12 | 8   | 0.080519 | 100/100 | Serial (forward) |
| 8  | 9  | 14 | 15 | 11 | 9  | 13 | 8  | 6  | 7   | 0.474986 | 100/100 | Serial (backward) |
| 9  | 7  | 9  | 6  | 10 | 8  | 14 | 21 | 10 | 0.030806 | 99/100 | Linear Complexity |

Table 2. Statistic of state-of-the-art FPGA-based PUFs.

| PUF     | Name       | Type  | CRPs (N = stages) | Inter-chip HD (Idea 50%) | Intra-chip HD (Idea 0%) | Enhancement |
|---------|------------|-------|-------------------|--------------------------|-------------------------|-------------|
| DAC'07  | RO PUF     | Weak  | $\log_2 N$        | 46.15%                   | 0.48%                   | Mask & ECC  |
| CHES'07 | SRAM       | Weak  | N                 | 49.97%                   | 3.57%                   |             |
| HOST'08 | Butterfly PUF | Weak   | N | 45% | 6% |             |
| CHES'07 | Glitch PUF | Weak  | N                 | 45% | 6% |             |
| VLSI'19 | Anderson PUF | Weak    | N | 45% | 6% |             |
| HOST'11 | Bistable PUF | Strong | $2^N$ | 44% | 2.19% |             |
| VLSI'05 | Abiter PUF  | Strong | $2^N$ | 36.75% | 1.52% |             |
| FdCSIS'14 | 3-1 DAPUF | Strong | $2^N$ | 50% | 12% |             |
| FdCSIS'14 | 2-1 DAPUF | Strong | $2^N$ | 42% | 9.7% |             |
| ISCAS'19 | MID PUF    | Strong | N | 47.2% | 0.73% |             |
| This Work | OC-PUF   | Strong | $2^{3N}$ | 46.7% | 5.47E-06 | Dynamic threshold |

5. Conclusion
In this paper, we have proposed an OC-PUF implemented on Altera DE2-115 FPGAs and a novel delay cell design and symmetric path configuration methodology. Compared with the state-of-the-art FPGA-Based strong PUFs in Table 2, OC-PUF shows a relatively good inter-chip HD of 46.7%. After applying dynamic threshold with a value 255, intra-chip HD dropped to 5.46E-06 at nominal conditions, nearly error free. The proposed OC-PUF could be a desired option for PUF applications on FPGAs. All delay based PUFs are very vulnerable to machine learning attacks since its response determined by the accumulated delay of different path with high linear relationships. We will do further research on how to resist machine learning attacks for the proposed OC-PUF in future work.

6. References

[1] Rostami M, Majzoobi M, Koushanfar F, Wallach D S and Devadas S 2014 Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching IEEE Transactions on Emerging Topics in Computing 2 37–49
[2] Herder C, Yu M-D, Koushanfar F and Devadas S 2014 Physical unclonable functions and applications: A tutorial Proceedings of the IEEE 102 1126–1141
[3] Delvaux J, Gu D, Schellekens D and Verbauwhede I 2015 Helper data algorithms for PUF-based key generation: Overview and analysis IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34 889–902
[4] Suh G E and Devadas S 2007 Physical unclonable functions for device authentication and secret key generation 2007 44th ACM/IEEE Design Automation Conference (IEEE) pp 9–14
[5] Merli D, Stumpf F and Eckert C 2010 Improving the quality of ring oscillator PUFs on FPGAs Proceedings of the 5th workshop on embedded systems security (ACM) p 9
[6] Usmani M A, Keshavarz S, Matthews E, Shannon L, Tessier R and Holcomb D E 2019 Efficient PUF-Based Key Generation in FPGAs Using Per-Device Configuration IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27 364–375
[7] Roel M 2012 Physically unclonable functions: Constructions, properties and applications Katholieke Universiteit Leuven, Belgium
[8] Morozov S, Maiti A and Schaumont P 2010 An analysis of delay based PUF implementations on FPGA International Symposium on Applied Reconfigurable Computing (Springer) pp 382–387
[9] Pappu R, Recht B, Taylor J and Gershenfeld N 2002 Physical one-way functions Science 297 2026–2030
[10] Gassend B, Clarke D, Van Dijk M and Devadas S 2002 Silicon physical random functions Proceedings of the 9th ACM conference on Computer and communications security (ACM) pp 148–160
[11] Rührmair U 2010 Oblivious transfer based on physical unclonable functions International Conference on Trust and Trustworthy Computing (Springer) pp 430–440
[12] Guajardo J, Kumar S S, Schrijen G-J and Tuyls P 2007 FPGA intrinsic PUFs and their use for IP protection International workshop on cryptographic hardware and embedded systems (Springer) pp 63–80
[13] Chen Q, Csaba G, Lugli P, Schlichtmann U and Rührmair U 2011 The bistable ring PUF: A new architecture for strong physical unclonable functions 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (IEEE) pp 134–141
[14] Lim D, Lee J W, Gassend B, Suh G E, Van Dijk M and Devadas S 2005 Extracting secret keys from integrated circuits IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13 1200–1205
[15] Machida T, Yamamoto D, Iwamoto M and Sakiyama K 2014 A new mode of operation for arbiter PUF to improve uniqueness on FPGA 2014 Federated Conference on Computer Science and Information Systems (IEEE) pp 871–878
[16] Maes R, Tuyls P and Verbauwhede I 2008 Intrinsic PUFs from flip-flops on reconfigurable devices 3rd Benelux workshop on information and system security (WISSec 2008) vol 17 p 2008

[17] Yu M-D and Devadas S 2010 Secure and robust error correction for physical unclonable functions IEEE Design & Test of Computers 27 48–65

[18] Dodis Y, Reyzin L and Smith A 2004 Fuzzy extractors: How to generate strong keys from biometrics and other noisy data International conference on the theory and applications of cryptographic techniques (Springer) pp 523–540

[19] Linnartz J-P and Tuyls P 2003 New shielding functions to enhance privacy and prevent misuse of biometric templates International Conference on Audio-and Video-Based Biometric Person Authentication (Springer) pp 393–402

[20] Yang K, Dong Q, Blaauw D and Sylvester D 2015 14.2 A physically unclonable function with BER< 10^-8 for robust chip authentication using oscillator collapse in 40nm CMOS 2015 IEEE International Solid-State Circuits Conference- (ISSCC) Digest of Technical Papers (IEEE) pp 1–3

[21] Bassham III L E, Rukhin A L, Soto J, Nechvatal J R, Smid M E, Barker E B, Leigh S D, Levenson M, Vangel M, Banks D L and others 2010 Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications

[22] Kumar S S, Guajardo J, Maes R, Schrijen G-J and Tuyls P 2008 The butterfly PUF protecting IP on every FPGA 2008 IEEE International Workshop on Hardware-Oriented Security and Trust (IEEE) pp 67–70

[23] Suzuki D and Shimizu K 2010 The glitch PUF: A new delay-PUF architecture exploiting glitch shapes International Workshop on Cryptographic Hardware and Embedded Systems (Springer) pp 366–382

[24] Zhang Z, Gu C, Cui Y, Zhang C, O’Neill M and Liu W 2019 Multi-Incentive Delay-Based (MID) PUF 2019 IEEE International Symposium on Circuits and Systems (ISCAS) (IEEE) pp 1–5