A Tale of Twin Primitives: Single-chip Solution for PUFs and TRNGs*

Kuheli Pratihar[1], Urbi Chatterjee[2], Manaar Alam[1], Debdeep Mukhopadhyay[1] and Rajat Subhra Chakraborty[1]

[1] Indian Institute of Technology, Kharagpur, India
[2] Indian Institute of Technology, Kanpur, India

Abstract.
Physically Unclonable Functions (PUFs) and True Random Number Generators (TRNGs) are two highly useful hardware primitives to build up the root-of-trust for an embedded device. PUFs are designed to offer *repetitive and instance-specific* randomness, whereas TRNGs are expected to be *invariably* random. In this paper, we present a dual-mode PUF-TRNG design that utilises two different hardware-intrinsic properties, i.e. oscillation frequency of the Transition Effect Ring Oscillator (TERO) cell and the propagation delay of a buffer within the cell to serve the purpose of both PUF and TRNG depending on the exact requirement of the application. The PUF design is also proposed to have a built-in resistance to machine learning (ML) and deep learning (DL) attacks, whereas the TRNG exhibits sufficient randomness.

Keywords: True Random Number Generators · Physically Unclonable Functions · Transient Effect Ring Oscillator · Feedback · Recurrent Neural Network · Internet of Things (IoT) · PUF modelling

1 Introduction

Identification and Random Number Generation play a crucial role in modern cryptographic systems. Physically Unclonable Functions (PUFs) exploit the random physical variations inherent to any manufacturing process to generate device-specific and unclonable fingerprints. On the other hand, True Random Number Generators (TRNGs) produce bitstreams that are independent, unpredictable and uniformly distributed from random physical phenomena such as atmospheric noise, clock jitter, phase noise and others.

The circuitry for both PUFs and TRNGs require a very small systematic mismatch such that there exists no bias in their respective responses. While the output for both PUFs and TRNGs are unpredictable, the fundamental difference lies in the fact that for the same challenge, the PUF response is repeatable for every run, whereas for TRNGs, the output varies randomly on every run.

Now, in state-of-the-art literature, mostly four design principles have been followed so far.

- **Stand-alone PUF circuits:** The Arbiter PUF(APUF) [GCvD02b, SD07] is the first proposed electrical, integrated Strong PUF [HF14]. Ring Oscillator(RO) PUF [MS11] and Transition Effect Ring Oscillator (TERO) PUF [BNCF14] are PUFs which utilize oscillation frequency or metastability for response generation. Additionally, several weak-PUFs based on memory storage technologies such as the

*This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

SRAM [HBF09], Flash [PAG+11], Memristors [KKS13] and DRAM [RCC+13] have also been proposed.

- **Stand-alone TRNG circuits:** Noise sources like thermal noise [JK99] and metastability [BAV+92, RLG14a] are commonly used entropy sources for TRNG designs [RLG14b, Gün10]. Also, randomness of mixed signal blocks like PLL [FD03, APFB18, PMBF17], Ring Oscillators [SMS07, WT08, YFH+14] and Flip Flops with high-precision edge sampling [YRG+18] have been presented as TRNG designs.

- **Integrated PUF-TRNG circuit using same entropy source:** Simultaneous PUF and random bits can be generated from RO based PUF-TRNG [MNRS09] and Universal Transition Effect Ring Oscillator (UTERO) based on the TERO loop [VDF13] using same source of entropy. In [LMS20], the authors introduce a secure and low-power integrated PUF-TRNG design by using analog grade embedded flash memories.

- **Integrated PUF-TRNG circuit using different entropy sources:** An unified weak PUF and TRNG design that utilizes two different entropy sources: current-steering digital-to-analog converter (DAC) and ring voltage-controlled oscillator (VCO) has been proposed in [DVK+20].

In this work, we present an unified PUF-TRNG design that brings together a delay-based Strong PUF and an oscillatory metastability based TRNG by sharing and reusing a significant amount of the utilized hardware resources suitable for IoT domain making it low-cost and low-power.

However, the evolution of Strong PUF construction has been significantly influenced by the rapid increase of machine learning based model building attacks. In the current state-of-art, Strong PUFs predominantly rely on delay based APUFs as their core building block [Del19, SD07, DPGV15, VKM+12, GCvD02a, MKP08, SMCN18, YHD+16]. Hence, they can be modelled by using a linear function which forms the basis for a number of attacks that use collected challenge-response pairs (CRPs) to build a mathematical clone of the PUF construction. Logistic Regression (LR) [RSS+10, RSS+13] and Reliability based modeling attacks [Bec15] are some of the examples. Hence, we *present an architectural tweak using recurrence* to make it resistant against modelling attacks.

The novel contributions of this paper in the scope of an unified PUF-TRNG architecture are as follows:

- The first major contribution of our work is to design a dual-mode TERO cell that exhibits the functionalities of both the PUF and the TRNG. We then introduce an auxiliary circuit to extend the TERO cell into a complete hardware structure that can on-the-fly switch to operate as an oscillatory metastability based TRNG and a delay based Strong PUF design. To the best of our knowledge, it is the first compact and unified architecture that effectively blends a delay based Strong PUF and the well-studied TERO TRNG by leveraging two different hardware intrinsic properties in the same circuitry.

- Our initial analysis showed that a linear delay model of the proposed delay based Strong PUF instance can be formed. Hence, to impede the modelling attacks, we present a challenge obfuscation technique inspired by Recurrent Neural Networks (RNNs) [SAS+19] that will make it significantly resistant against ML based modelling attacks

The remainder of the paper is organized as follows. Section 2 discusses the proposed architecture in details. Additionally, we also propose a modified architecture based on recurrence in Section 3 and finally conclude the paper in Section 4.

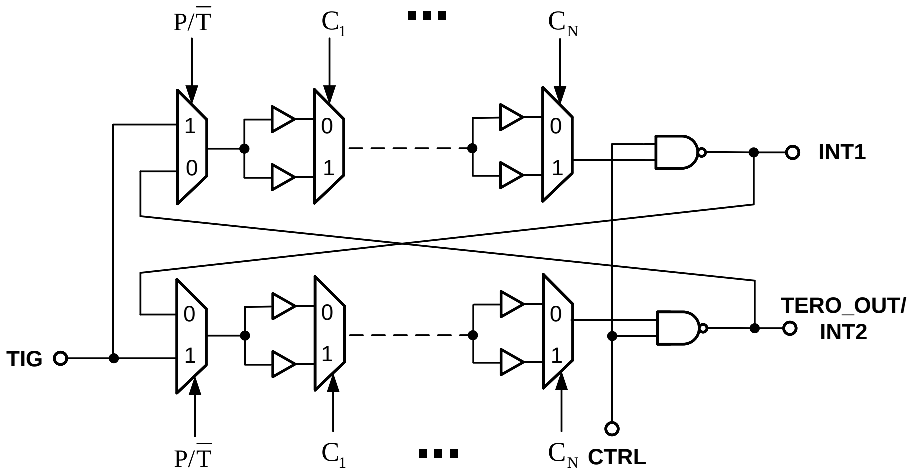

Figure 1: Proposed Architecture of a dual-mode TERO Cell.

2 Proposed Unified PUF-TRNG Architecture

In this section, we propose the architecture for an unified PUF-TRNG and the design rationale over conventional design criteria.

2.1 Design 1: A Dual-Mode TERO Cell

We first use the TERO cell structure as proposed in [PMB+16] to develop a unified PUF-TRNG structure. The major crux of our proposed scheme is to develop a primitive by using *a single TERO Cell with both PUF and TRNG functionalities.* And we have achieved this by replacing the buffers of the TERO cell with delay elements [CDGB12]. As shown in Fig. 1, the TERO cell consists of two symmetrically laid delay chains consisting of total $2N$ identical challenge-driven delay stages along with one 2×1 multiplexer at the end of every stage. Now, as a particular delay stage is realized using two buffer elements, we can apply challenges to the 2×1 multiplexer at every stage to select one of them for propagation of the trigger signal. This enables us to select 2^N combinations of buffers from the single TERO cell using an N-bit binary challenge. This makes our proposed PUF design fundamentally different from the conventional TERO-PUF designs. Now, the next question that arises is as follows:

How can we exploit the delay difference between the two chains as its source of device-specific randomness as opposed to the most significant bits (MSBs) of the number of temporary oscillations?

The answer is to break the feedback loop in the TERO-cell structure, thereby giving rise to two delay chains as shown in Fig 1. We achieve this by introducing two additional 2×1 multiplexers before the upper and lower delay chains to control the operability of the cell. Depending on the control signal $P/\overline{T}$, the dual-mode TERO cell either exploits the temporary oscillation of the feedback loop to generate random bits or leverages the delay differences of the two symmetrically laid delay chains to generate the PUF response. To the best of our knowledge, *this is the first design to utilize two very different entropy sources within the same structure,* thereby bringing together the goodness of a delay based Strong PUF as well as the well-studied TRNG design into a single dual-mode hardware architecture.

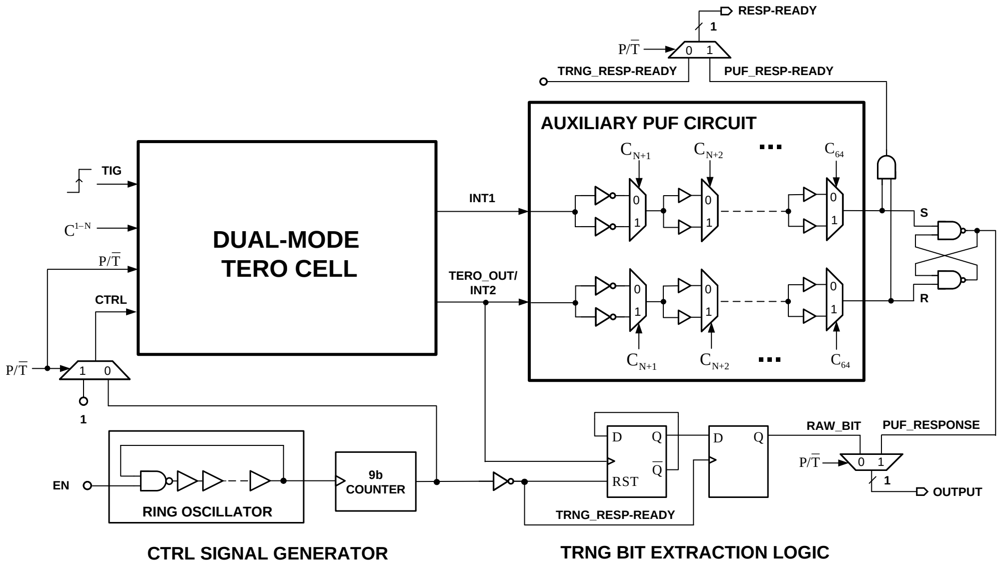

Figure 2: Unified PUF-TRNG Architecture with dual mode operability.

2.2 A Dual-Mode PUF-TRNG Structure

The unified bit PUF-TRNG circuit consists mainly of four components: a) the dual-mode TERO cell, b) an auxiliary PUF circuit, c) TRNG bit extraction logic, and d) a control signal generator. The overview of the same is shown in Fig. 2. The core of the TERO-TRNG is the dual-mode TERO cell built using a multiplexer, a chain of N delay elements, and a NAND gate in both the branches. The TERO cell is followed by the TRNG bit extraction block that comprises of a 1-bit counter. It is implemented using a T-flip flop followed by an output data register. The control signal generator periodically restarts the dual-mode TERO cell during its TRNG operation by using a conventional ring oscillator followed by a 9-bit counter. However, during its operation as a PUF, the control signal is set to high and as a result, the NAND gates in each branch of the TERO loop behave as an inverter. The first N bits of the 64-bit binary challenge are provided identically to the N delay elements of the top and the bottom branch in the dual-mode TERO cell. The remaining $64 - N$ challenge bits are provided to the delay elements in the Auxiliary PUF circuit. *This design principle eventually eradicates the necessity of including all 64 delay elements in the TERO cell itself, thus, in turn maintaining the good TRNG quality.*

Now, we move forward to discuss in detail the working principle of the proposed circuit in TRNG mode and in PUF mode.

2.3 Operation Principle

Initially, we use the $\mathrm{P}/\bar{\mathrm{T}}$ signal to select the mode of operation for the circuit. This signal is given as an input to the select line of the multiplexers as a result *disabling/enabling the feedback loop*, as the case may be. We discuss the operation in detail below:

2.3.1 TRNG Mode

In this case, $\mathrm{P}/\bar{\mathrm{T}}$ is set to 0, thereby enabling the feedback loop and ensuring the bi-stable operation of the TERO cell. All the challenge bits are set to 0 in this mode, thereby selecting a fixed set of delay elements. In the reset phase, the control signal $CTRL = 0$,

Algorithm 1: Working of Design 3 with θ cycles.

```
Input: 64 Bit Challenge $\mathbf{c}$, $\theta$
Output: 1 Bit Response
$k = 0$, $t = 0$, $\mathbf{c}_p = \mathbf{c}_e = \mathbf{c}$
while $t < \theta$ do
    while $k < 4$ do
        /* Cyclic Left Shift of Challenge                    */
        $\mathbf{c}_p = \mathbf{c}_p \ll 16$
        $r_k^{(t)} = \mathrm{PUF}(\mathbf{c}_p)$
        $k = k + 1$
    end
    /* $\mathbf{c}_r$ will have the same bit length as c     */
    $\mathbf{c}_r = [r_0^{(t)}, r_1^{(t)}, r_2^{(t)}, r_3^{(t)}, r_0^{(t)}, r_1^{(t)}, r_2^{(t)}, r_3^{(t)}, \ldots]$
    $\mathbf{c}_e = \mathrm{XOR}(\mathbf{c}_e, \mathbf{c}_r)$
    $t = t + 1$
    $k = 0$
    $\mathbf{c}_p = \mathbf{c}_e$
end
Final Response = $r_0^{(\theta)} = \mathrm{PUF}(\mathbf{c}_e)$
```

therefore the output of the TERO loop is 1. When $CTRL = 1$, the TERO cell starts oscillating and continues to do so until it reaches a stable state. The counter implemented using a T-flip flop counts the number of temporary oscillations and the output data register gives the least significant bit (LSB) of the total number of temporary oscillations as the internal random bit.

2.3.2 PUF Mode

In this mode P/$\bar{\mathrm{T}}$ is set to 1, thereby breaking the feedback loop. The PUF is enabled via the TIG signal which propagates through the two delay chains. We reuse the components of the dual-mode TERO loop and implement a delay based strong PUF.

3 Recurrence PUF Architecture

The resistance of strong PUFs against ML attacks is in general achieved by either introducing non-linearity into the system [VPPK16] or by obfuscating the challenge using randomness [ZIC17]. In this subsection, we propose a recurrence based challenge obfuscation technique to show greater resistance of the PUF design against ML attacks. This approach exploits the device-specific randomness to encode the challenge, thereby hiding the true relationship between the original challenge and the PUF response.

The proposed PUF has a linear delay model similar to that of a PAPUF. ML tools like PAC Learning use Linear Threshold Functions(LTFs) to predict CRPs making the PUF vulnerable to attacks. Recurrence PUF addresses this issue by obfuscating the challenges hence increasing the number of CRPs required to learn the PUF behaviour which cannot be modeled by a LTF anymore.

3.1 Enhanced Design with Recurrence for ML Resistance

We use the idea of encoding the challenge $\mathbf{c}$ with response-bits obtained from left shifted versions of the challenge $\mathbf{c}$. This process is repeated for θ iterations to recurrently encode

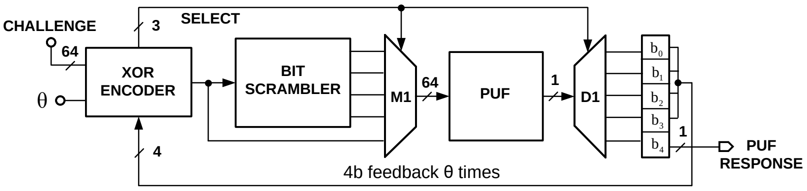

Figure 3: Block diagram showing the high level behaviour of the proposed PUF with recurrence

our original challenge $\mathbf{c}$. Fig. 3 shows the high level block diagram of the recurrence scheme. We have a multiplexer to select between challenge $\mathbf{c}$ or challenge $\mathbf{c} \ll 16p$ where, $\ll 16p$ is the cyclic left shift operation by $16p$ bits with $p \in \{0,1,2,3\}$. The response bits for t^{th} iteration are $r_0^{(t)}, r_1^{(t)}, r_2^{(t)}, r_3^{(t)}$ corresponding to the PUF response for $\mathbf{c}, \mathbf{c} \ll 16, \mathbf{c} \ll 32, \mathbf{c} \ll 48$ for $0 \leq t < \theta$. The challenge $\mathbf{c}$ is now XOR-ed with an extension of response bits repeated 16 times to get an encoded challenge. The final PUF response is $r_0^{(\theta)}$ which is obtained from the θ times encoded challenge.

4 Conclusion

In this paper we initiated the study of a dual mode delay based Strong PUF-TRNG architecture co-residing in a single circuit using two different hardware-intrinsic properties that have not been studied before to the best of our knowledge. We also demonstrate that by breaking the feedback loop of a TERO cell in a controllable manner leads to designing of two different primitives utilising the randomness generated from both propagation delay variation and oscillatory metastability.

References

[APFB18] E. N. Allini, O. Petura, V. Fischer, and F. Bernard. Optimization of the pll configuration in a pll-based trng design. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1265–1270, 2018.

[BAV+92] M. J. Bellido, A. J. Acosta, M. Valencia, A. Barriga, and J. L. Huertas. A simple binary random number generator: new approaches for cmos vlsi. In *[1992] Proceedings of the 35th Midwest Symposium on Circuits and Systems*, pages 127–129 vol.1, 1992.

[Bec15] Georg T. Becker. The gap between promise and reality: On the insecurity of XOR arbiter PUFs. In Tim Güneysu and Helena Handschuh, editors, *CHES 2015*, volume 9293 of *LNCS*, pages 535–555. Springer, Heidelberg, September 2015.

[BNCF14] L. Bossuet, X. T. Ngo, Z. Cherif, and V. Fischer. A puf based on a transient effect ring oscillator and insensitive to locking phenomenon. *IEEE Transactions on Emerging Topics in Computing*, 2(1):30–36, 2014.

[CDGB12] Z. Cherif, J. Danger, S. Guilley, and L. Bossuet. An easy-to-design puf based on a single oscillator: The loop puf. In *2012 15th Euromicro Conference on Digital System Design*, pages 156–162, 2012.

[Del19] J. Delvaux. Machine-learning attacks on polypufs, ob-pufs, rpufs, lhs-pufs, and puf–fsms. *IEEE Transactions on Information Forensics and Security*, 14(8):2043–2058, 2019.

[DPGV15] Jeroen Delvaux, Roel Peeters, Dawu Gu, and Ingrid Verbauwhede. A survey on lightweight entity authentication with strong pufs. *ACM Comput. Surv.*, 48(2), October 2015.

[DVK$^+$20] M. Danesh, A. B. Venkatasubramaniyan, G. Kapoor, N. Ramesh, S. Sadasivuni, S. T. Chandrasekaran, and A. Sanyal. Unified analog puf and trng based on current-steering dac and vco. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(11):2280–2289, 2020.

[FD03] Viktor Fischer and Milos Drutarovský. True random number generator embedded in reconfigurable hardware. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES 2002*, volume 2523 of *LNCS*, pages 415–430. Springer, Heidelberg, August 2003.

[GCvD02a] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Controlled physical random functions. In *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, pages 149–160, 2002.

[GCvD02b] Blaise Gassend, Dwaine E. Clarke, Marten van Dijk, and Srinivas Devadas. Silicon physical random functions. In Vijayalakshmi Atluri, editor, *ACM CCS 2002*, pages 148–160. ACM Press, November 2002.

[Gün10] Tim Güneysu. True random number generation in block memories of reconfigurable devices. In *2010 International Conference on Field-Programmable Technology*, pages 200–207. IEEE, 2010.

[HBF09] D. E. Holcomb, W. P. Burleson, and K. Fu. Power-up sram state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, 2009.

[HF14] Daniel E. Holcomb and Kevin Fu. Bitline PUF: Building native challenge-response PUF capability into any SRAM. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 510–526. Springer, Heidelberg, September 2014.

[JK99] Benjamin Jun and Paul Kocher. The Intel random number generator. *Cryptography Research Inc. white paper*, 27:1–8, 1999.

[KKS13] P. Koeberl, Ü. Kocabaş, and A. Sadeghi. Memristor pufs: A new generation of memory-based physically unclonable functions. In *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 428–431, 2013.

[LMS20] S. Larimian, M. R. Mahmoodi, and D. B. Strukov. Lightweight integrated design of puf and trng security primitives based on eflash memory in 55-nm cmos. *IEEE Transactions on Electron Devices*, 67(4):1586–1592, 2020.

[MKP08] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Lightweight secure pufs. In *2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 670–673, 2008.

[MNRS09] Abhranil Maiti, Raghunandan Nagesh, Anand Reddy, and Patrick Schaumont. Physical unclonable function and true random number generator: A compact and scalable implementation. In *Proceedings of the 19th ACM Great Lakes Symposium on VLSI*, GLSVLSI '09, page 425–428, New York, NY, USA, 2009. Association for Computing Machinery.

[MS11] Abhranil Maiti and Patrick Schaumont. Improved ring oscillator PUF: An FPGA-friendly secure primitive. *Journal of Cryptology*, 24(2):375–397, April 2011.

[PAG$^+$11] Pravin Prabhu, Ameen Akel, Laura M. Grupp, Wing-Kei S. Yu, G. Edward Suh, Edwin Kan, and Steven Swanson. Extracting device fingerprints from flash memory by exploiting physical variations. In Jonathan M. McCune, Boris Balacheff, Adrian Perrig, Ahmad-Reza Sadeghi, Angela Sasse, and Yolanta Beres, editors, *Trust and Trustworthy Computing*, pages 188–201, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[PMB$^+$16] O. Petura, U. Mureddu, N. Bochard, V. Fischer, and L. Bossuet. A survey of ais-20/31 compliant trng cores suitable for fpga devices. In *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–10, 2016.

[PMBF17] O. Petura, U. Mureddu, N. Bochard, and V. Fischer. Optimization of the pll based trng design using the genetic algorithm. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, 2017.

[RCC$^+$13] S. Rosenblatt, S. Chellappa, A. Cestero, N. Robson, T. Kirihata, and S. S. Iyer. A self-authenticating chip architecture using an intrinsic fingerprint of embedded dram. *IEEE Journal of Solid-State Circuits*, 48(11):2934–2943, 2013.

[RLG14a] S. Robson, B. Leung, and G. Gong. Truly random number generator based on a ring oscillator utilizing last passage time. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(12):937–941, 2014.

[RLG14b] Stewart Robson, Bosco Leung, and Guang Gong. Truly random number generator based on a ring oscillator utilizing last passage time. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(12):937–941, 2014.

[RSS$^+$10] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, page 237–249, New York, NY, USA, 2010. Association for Computing Machinery.

[RSS$^+$13] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas. Puf modeling attacks on simulated and silicon data. *IEEE Transactions on Information Forensics and Security*, 8(11):1876–1891, 2013.

[SAS$^+$19] Nimesh Shah, Manaar Alam, Durga Prasad Sahoo, Debdeep Mukhopadhyay, and Arindam Basu. A 0.16pj/bit recurrent neural network based puf for enhanced machine learning attack resistance. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, ASPDAC '19, page 627–632, New York, NY, USA, 2019. Association for Computing Machinery.

[SD07] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *2007 44th ACM/IEEE Design Automation Conference*, pages 9–14, 2007.

[SMCN18] D. P. Sahoo, D. Mukhopadhyay, R. S. Chakraborty, and P. H. Nguyen. A multiplexer-based arbiter puf composition with enhanced reliability and security. *IEEE Transactions on Computers*, 67(3):403–417, 2018.

[SMS07] B. Sunar, W. J. Martin, and D. R. Stinson. A provably secure true random number generator with built-in tolerance to active attacks. *IEEE Transactions on Computers*, 56(1):109–119, 2007.

[VDF13] M. Varchola, M. Drutarovsky, and V. Fischer. New universal element with integrated puf and trng capability. In *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, pages 1–6, 2013.

[VKM+12] Anthony Van Herrewege, Stefan Katzenbeisser, Roel Maes, Roel Peeters, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs. In Angelos D. Keromytis, editor, *FC 2012*, volume 7397 of *LNCS*, pages 374–389. Springer, Heidelberg, February / March 2012.

[VPPK16] A. Vijayakumar, V. C. Patil, C. B. Prado, and S. Kundu. Machine learning resistant strong puf: Possible or a pipe dream? In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 19–24, 2016.

[WT08] K. Wold and C. H. Tan. Analysis and enhancement of random number generator in fpga based on oscillator rings. In *2008 International Conference on Reconfigurable Computing and FPGAs*, pages 385–390, 2008.

[YFH+14] K. Yang, D. Fick, M. B. Henry, Y. Lee, D. Blaauw, and D. Sylvester. 16.3 a 23mb/s 23pj/b fully synthesized true-random-number generator in 28nm and 65nm cmos. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 280–281, 2014.

[YHD+16] M. Yu, M. Hiller, J. Delvaux, R. Sowell, S. Devadas, and I. Verbauwhede. A lockdown technique to prevent machine learning on pufs for lightweight authentication. *IEEE Transactions on Multi-Scale Computing Systems*, 2(3):146–159, 2016.

[YRG+18] Bohan Yang, Vladimir Rožić, Miloš Grujić, Nele Mentens, and Ingrid Verbauwhede. ES-TRNG: A high-throughput, low-area true random number generator based on edge sampling. *IACR TCHES*, 2018(3):267–292, 2018. https://tches.iacr.org/index.php/TCHES/article/view/7276.

[ZIC17] S. S. Zalivaka, A. A. Ivaniuk, and Chip-Hong Chang. Fpga implementation of modeling attack resistant arbiter puf with enhanced reliability. In *2017 18th International Symposium on Quality Electronic Design (ISQED)*, pages 313–318, 2017.