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

Kuheli Pratihar¹, Urbi Chatterjee², Manaar Alam¹, Debdeep Mukhopadhyay¹ and Rajat Subhra Chakraborty¹

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 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 APUs 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.
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 $2 \times 1$ 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 $2 \times 1$ 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 $2 \times 1$ multiplexers before the upper and lower delay chains to control the operability of the cell. Depending on the control signal $P/T$, 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.
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 $P/\bar{T}$ 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, $P/\bar{T}$ 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 $\theta$ cycles.

\textbf{Input:} 64 Bit Challenge $c$, $\theta$

\textbf{Output:} 1 Bit Response

$k = 0$, $t = 0$, $c_p = c_e = c$

while $t < \theta$ do

\hspace{1em} while $k < 4$ do

\hspace{2em} /* Cyclic Left Shift of Challenge */

\hspace{3em} $c_p = c_p \ll 16$

\hspace{3em} $r_k^{(t)} = \text{PUF}(c_p)$

\hspace{3em} $k = k + 1$

\hspace{2em} end

\hspace{1em} /* $c_r$ will have the same bit length as $c$ */

\hspace{2em} $c_r = [r_0^{(t)}, r_1^{(t)}, r_2^{(t)}, r_3^{(t)}, r_0^{(t)}, r_1^{(t)}, r_2^{(t)}, r_3^{(t)}, \ldots]$ $c_e = \text{XOR}(c_e, c_r)$

\hspace{2em} $t = t + 1$

\hspace{2em} $k = 0$

\hspace{2em} $c_p = c_e$

end

Final Response = $r_0^{(\theta)} = \text{PUF}(c_e)$

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 $P/T$ 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 (LTIs) 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 $c$ with response-bits obtained from left shifted versions of the challenge $c$. This process is repeated for $\theta$ iterations to recurrently encode
Figure 3: Block diagram showing the high level behaviour of the proposed PUF with recurrence

our original challenge \( c \). Fig. 3 shows the high level block diagram of the recurrence scheme. We have a multiplexer to select between challenge \( c \) or challenge \( c \ll 16p \) 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 \( c, c \ll 16, c \ll 32, c \ll 48 \) for \( 0 \leq t < \theta \). The challenge \( c \) is now XOR-ed with an extension of response bits repeated 16 times to get an encoded challenge. The final PUF response is \( r_\theta^{(\theta)} \) which is obtained from the \( \theta \) 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


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


