ABSTRACT
With rapid advancements in electronic gadgets, the security and privacy aspects of these devices are significant. For the design of secure systems, physical unclonable function (PUF) and true random number generator (TRNG) are critical hardware security primitives for security applications. This paper proposes novel implementations of PUF and TRNGs on the RRAM crossbar structure. Firstly, two techniques to implement the TRNG in the RRAM crossbar are presented based on write-back and 50% switching probability pulse. The randomness of the proposed TRNGs is evaluated using the NIST test suite. Next, an architecture to implement the PUF in the RRAM crossbar is presented. The initial entropy source for the PUF is used from TRNGs, and challenge-response pairs (CRPs) are collected. The proposed PUF exploits the device variations and sneak-path current to produce unique CRPs. We demonstrate, through extensive experiments, reliability of 100%, uniqueness of 47.78%, uniformity of 49.79%, and bit-aliasing of 48.57% without any post-processing techniques. Finally, the design is compared with the literature to evaluate its implementation efficiency, which is clearly found to be superior to the state-of-the-art.

CCS CONCEPTS
• Security and privacy → Hardware security implementation; Embedded systems security.

KEYWORDS
PUF, TRNG, RRAM, Memristors, Hardware Security

1 INTRODUCTION
Physical unclonable functions (PUFs) are being considered the most reliable hardware security primitives, especially in the IoT era for device authentication and key generation [22]. PUF usually exploits the variations in the manufacturing process and physical randomness inherent to the device, making them highly reliable and unclonable [27]. PUF utilizes the idea that a device-specific unique key can be derived each time rather than storing a set of keys in the memory. PUF works on the principle of computing a unique output response for the applied input challenge, thus creating a challenge-response pair (CRP), which is crucial for designing security protocols, ranging from device attestation to data encryption.

Various CMOS-based PUF designs such as Arbiter PUF [13, 21] and SRAM PUF [12, 18] are employed, which depend on the randomness introduced during their fabrication process for their operation. However, they suffer from various demerits, as they are area inefficient [9], sensitive to temperature variations [19], and require post-processing [24]. Also, carbon nanotube-based PUF have been proposed, which utilize the random nature of the bits generated from the unpredictable connections in the CNT devices [7]. This approach showed some reliability. However, it is not usually employed due to the complexity involved.

In recent years, resistive random access memory (RRAM) has been investigated widely for use as the main memory and in-memory computing. A major drawback to RRAMs being widely adopted as next-generation memories is their stochastic nature and the intrinsic variation in switching parameters [28]. This uncertain behavior of RRAM is critical for implementing hardware security...
primitives, including PUF design [23]. The schematic structure of RRAM consists of a metal-insulator-metal stack as depicted in Fig. 1b. The RRAM can be considered a variable resistance, where memory states are stored as a high-resistance state (HRS) and a low-resistance state (LRS). The RRAM state is switched by applying a suitable voltage pulse. The devices significantly impact the resistance state of the device due to the device-to-device (D2D) and cycle-to-cycle (C2C) variations [14].

This stochasticity of RRAM is desirable for implementing PUF and other hardware security primitives. It utilizes variations during the manufacturing and fabrication process for authentication and secret key storage purposes [10]. Lately, multiple research efforts have been focused for utilizing emerging memories such as RRAM [17] and STT-MRAM (Spin Transfer Torque Magnetic RAM) [26] for various security applications. RRAM has emerged as the most feasible option for designs of PUFs since it allows extraction of intrinsic hardware secrets from process variations, similar to that of the traditional silicon-based PUFs. For RRAM, the key is dependent only on the CF variations in the device during LRS and HRS, which are unique for each switching cycle. The secret is independent of the configuration that is selected by a challenge [16]. These primitives are usually employed for generating unique keys that connect inseparably to the hardware. A typical PUF schematic based on RRAM crossbars is depicted along with the crossbar structure for collecting the CRP. For RRAM, the key is dependent only on the CF variations in the device during LRS and HRS which are unique for each switching cycle.

In addition to PUFs, TRNGs are another important element for hardware encryption modules. TRNGs typically exploit the randomness in physical processes to generate a stream of random numbers [15]. Although various CMOS-based designs have been implemented, they have limitations as they only provide limited security-specific properties. The emerging technologies, especially RRAM, have shown great potential due to high density, low power operation, and new randomness sources [25].

The remainder of the paper is organized as follows: Section II details the methods employed to realize the TRNG and PUF designs based on RRAM. Section III presents the results of the RRAM-based security primitives and discusses some critical performance metrics such as uniformity, bit aliasing, and reliability. Section IV concludes the paper.

2 METHODOLOGY

Firstly, we demonstrate and validate a novel TRNG based on RRAM that relies on the variations in the material stack of a device in the resistive switching layer. We utilize the inherent stochasticity of the RRAM as the randomness source to achieve the TRNG design. In this work, we propose two variants of TRNG based on RRAM as design elements, as depicted in Fig. 2. The first design utilizes the write-back technique for generating random numbers in the crossbar, whereas the second design produces random numbers by applying a 50% switching probability pulse [2]. The proposed RRAM-based TRNG designs have lower circuit complexity advantages as randomness is generated and harvested using simple elements. The proposed TRNG designs can be incorporated into memory subsystems, enhancing security and area efficiency. The random bit streams generated using both designs are depicted in Fig. 2a and Fig. 2b, respectively.

Fig. 2a shows the flow to implement the TRNG using the write-back technique. Initially, all the devices are formed into the LRS, followed by resetting all devices back to HRS. Next, a pair of devices is selected, and a READ voltage (0.2V) is applied across the rows of selected devices. As the variations on the crossbar are normally distributed, one device will provide more current than another. The currents following through these devices are compared, and the fast device in the selected pair is explicitly switched to LRS. This will ensure an equal probability of 0’s and 1’s in each row of the
When a 50% switching probability pulse is applied, random devices TRNGs are used to construct the hardware security primitives. The bitmap generated using this technique on a single RRAM crossbar structure. The proposed architecture shows an implementation of TRNG and PUF. There are majorly two types of variations, known as D2D and C2C which can affect the switching behavior of RRAM devices. The following are the parameters affected during the manufacturing process of RRAM used in this study: 

- Radius of the disc ($r_{disc}$)
- Oxygen vacancy concentration in the disc, maximum ($N_{disc, max}$) and minimum ($N_{disc, min}$)
- Length of the disc ($l_{disc}$)

Table 1 gives the ranges of variable parameters that are used for this study. These variations are experimentally verified and that provides validation to our design choices [11].

C2C variations. C2C variations in the devices are analyzed by changing the variable parameters before every SET and RESET operation. The change of parameters in the next cycle is based on the previous state parameters and random walk of parameters in +ve and -ve directions with an equal probability. The variability parameters are exported in a CSV file using MATLAB scripts. The final parameter file contains variation values and the simulation time stamp at which these parameters will be adopted. This file is then loaded into the Cadence Spectre simulation as a parameter list.

Fig. 4a shows the distribution of HRS and LRS for 1000 cycles. Each cycle represents a voltage pulse with a 20ns pulse duration, 2.5ns rise time, and 2.5ns fall time. The new set of values are applied after the 100ns period. Within these 100ms, the device is RESET.
with a 25ns pulse of -2V and SET with a 2V pulse with the same duration. The state of the device is read by applying a READ pulse of 0.2V after every RESET/SET operation. In the next phase (after 100ns), new values are fetched from the parameter file and applied in the simulation. It has been observed that there is a ±5% change in HRS and a ≈1% change in LRS during C2C variations.

**D2D variations.** To simulate D2D variations, the random set of values for each device’s variate parameters are drawn from the experimentally verified Gaussian distribution [11]. These variations were then independently applied to the available devices in the crossbar. Fig. 4b shows the distribution of HRS and LRS states in a 10x10 crossbar (100 devices) without a transistor. To measure the HRS distributions, all the devices are first switched to LRS by applying a 2V SET pulse with 150ns pulse width along with 50ns rise and fall time. The devices are switched to the HRS state in the next cycle by applying a RESET voltage of -2V with a similar pulse duration used during SET operation. The resistance state of the device is read at 0.2V. The same procedure is used to find LRS distribution. It has been observed that HRS has more impact of process variations compared to LRS. Due to variations, the HRS varies from $31\,\text{K}\Omega$ to $155\,\text{K}\Omega$ with an average of $65.56\,\text{K}\Omega$ and LRS from $1.55\,\text{K}\Omega$ - $1.67\,\text{K}\Omega$ with an average of $1.64\,\text{K}\Omega$. However, HRS has a wide distribution; all HRSs are distinguishable from LRS. A carefully chosen pulse to RESET the devices will lead to switching the devices randomly from LRS to HRS or vice-versa, which can eventually be exploited to design the TRNGs.

**Pulse width study.** This study gives the pulse width required to switch a device from LRS to HRS and vice-versa. We analyze the switching characteristic of devices on different pulse duration with a fixed voltage. Table 2 shows the switching probability of crossbar with different pulse duration. To conduct this experiment, all the devices are considered initially into HRS, and a fixed voltage of -0.8V with 1ns rise & fall time for pulse duration less than 100ns and 10ns for other pulse duration is applied. A total of 100 devices in a 10x10 crossbar are connected to an individual voltage source. A READ voltage of 0.2V is applied to read the state of the devices. The crossbar contains D2D and C2C variations as discussed in Section 3.2.

Fig. 5 shows the switching behavior of devices with different pulse widths. The devices’ states are read at 0.2V and sorted according to their resistance value. Initially, all the devices were in HRS. As the pulse width increases, more devices start switching into LRS from the HRS state. This analysis gives a distribution of HRS and LRS, which can further be used to choose the 50% probability pulse. Finding a pulse width that can randomly switch 50% of devices in the crossbar is exploited to design the TRNG. A 3us pulse is used for these particular devices to switch the 50% of devices in the crossbar, as given in Table 2.

### 3.3 PUF properties

For evaluating a PUF design, a number of properties need to be analyzed, such as uniformity, uniqueness, bit-aliasing, and reliability. In the following subsections, we provide the analysis of the proposed design on these properties in detail.

| Pulse duration | SET probability | LRS (KΩ) | HRS (KΩ) |
|----------------|-----------------|----------|----------|
| 10 ns          | 0%              | 1.67 - 1.7 | 23 - 140.7 |
| 100 ns         | 11%             | 1.62 - 1.67 | 38 - 140.8 |
| 1 us           | 36%             | 1.6 - 1.675 | 32.2 - 140.8 |
| 3 us           | 50%             | 1.58 - 1.675 | 47 - 140.8 |
| 10 us          | 64%             | 1.585 - 1.67 | 47 - 140 |
| 1 ms           | 94%             | 1.58 - 1.67 | 94 - 128 |
| 10 ms          | 100%            | 1.58 - 1.68 | - |

### Table 2: Pulse width study to find out 50% pulse to generate random numbers in RRAM crossbar. SET voltage = -0.8V, READ voltage = 0.2V, Rise and Fall time = 1ns.
The first step is calculating the intra Hamming distance (intra-HD), which shows the difference in the output response bits corresponding to different input challenges. Fig. 6a shows the histogram for intra-HD in responses of 16-bit RRAM-based PUF, which is calculated by Equation 1. Ideally, PUF should result in a 50% intra-HD in the response bits. In this case, with a particular entropy and variations, the proposed PUF shows 39.68% intra-HD.

\[
\text{Intra-HD} = \frac{1}{n} \sum_{i=1}^{n} HD(R_i, R_{i+1}) \times 100 \quad (1)
\]

Where ‘n’ is the total number of challenges given to the PUF, \( R_i \) and \( R_{i+1} \) are the responses to the challenges \( C_i \) and \( C_{i+1} \) respectively. Note that \( R_i \) is a bit vector structured from each response bit position at the columns from \( r_1 \) to \( r_m \) for \( i_{th} \) challenge.

**Uniformity.** Uniformity property shows how the response bit from each device is distributed in 0’s and 1’s. It is expected from the PUF for an equal number of switching between 0 and 1 for each response bit. The ideal value for uniformity is 50%, calculated by taking the average of all the response bits’ distribution. Fig. 6b shows the distribution of 0’s (green) and 1’s (red) in each response bit position. Even though some response bits are more inclined towards 0’s, and some are towards 1’s, the proposed PUF shows 49.79% uniformity. The distributions of 0’s and 1’s are a function of the entropy source and will change its inclination on another entropy set for other crossbars.

**Uniqueness.** Uniqueness characterizes the ability to distinguish between different devices based on their response to the same challenge. As the applied challenges are identical, the difference between the responses are entirely based on the process variations. It is expected that the PUF should respond with 50% HD in the responses to the same applied challenges for different devices. The uniqueness experiment is conducted using different crossbar variations randomly drawn from the Gaussian distribution. A total of 50K CRPs are collected from each device. Further, Fig. 6c shows the histogram of uniqueness, which is calculated as per Equation 2. The proposed PUF gives 47.78% uniqueness, which is very close to the ideal value.

\[
\text{Uniqueness} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100 \quad (2)
\]

Where \( i \) and \( j \) are two different devices. \( R_i \) and \( R_j \) are the responses from \( device_i \) and \( device_j \) for the challenge \( C \) respectively. \( k \) is the number of devices and \( n \) is total number of responses.

**Bit-aliasing.** Bit-aliasing investigates the switching of each response bit across the number of available crossbars and can be calculated as per Equation 3. It is essential to ensure that no bit positions of responses are strongly biased towards 0 or 1. To calculate bit-aliasing for the proposed design, CRPs are collected from two different crossbars, and then each bit position of response is analyzed. Ideally, bit-aliasing should be 50% for a well-balanced design, and the proposed design shows a bit-aliasing of 48.57%.

\[
\text{Bit-aliasing} = \frac{1}{n} \sum_{i=1}^{n} (R_{i,p} == 1) \times 100 \quad (3)
\]

where \( R_{i,p} \) is the response bit at the \( p \)th cell location in the \( i \)th chip and ‘n’ is the total number of responses including responses from all devices.

**Reliability.** Reliability captures how efficient a PUF is in repeating the same response for the same challenge. The higher the reliability, the less cost for the error-correcting techniques. Reliability is calculated by intra-HD between responses during the same applied challenges. Ideally, intra-HD for the same response will be zero, and reliability is given as (100% − (intra-HD)). To check the reliability of the proposed design, we applied the same challenge of 10K times and collected the responses. The proposed design shows a reliability of 100% without any error-correcting techniques.

### 3.4 Machine Learning-based Modelling Attacks

The machine learning (ML) models can be trained to predict the responses of a given PUF. It has been shown that nearly all variants of CMOS-based PUFs can be efficiently modelled using ML [5] and correspondingly attack-resistant PUF structures are also being proposed [8]. These attacks characterize the silicon behavior from CRPs. However, in RRAM-based PUF, the underlined devices have different characteristics and physics compared to the CMOS-based design, which requires a deeper analysis. We reserve the study of detailed analysis of modeling attacks for the future.

### 3.5 Randomness tests and Comparison

The process variations in RRAM devices result in different switching behavior of these devices and are exploited to design the TRNG. We proposed two methods (50% probability pulse and write-back) to generate TRNGs in the 16x16 RRAM crossbar, which are further used to construct the PUF. The national institute of standards and technology (NIST) randomness tests are conducted to analyze the randomness. The method with a 50% probability pulse passes 11 tests, and the write-back technique passes 08 out of 15 tests. NIST tests are conducted on just 16x16 crossbar data, and it passes all critical tests. The NIST tests can further be improved by increasing the size of the crossbar.

Table 3 compares the proposed PUF with other RRAM-based PUFs. Table 3 shows that the proposed design on a 16x16 crossbar outperforms other RRAM-based designs, even without the post-processing. MR-PUF [1] analyzes the single-cell behavior (not in crossbar) and presents the result after post-processing. Our technique is implemented on the passive crossbar, which exploits the process variations along with the sneak-path current. Construction using a passive crossbar leads to less overall area.

| Parameters | Proposed PUF | Nano PUF [6] | RRAM PUF [3] | Memristor PUF [4] | MR-PUF [1] |
|------------|--------------|--------------|--------------|------------------|------------|
| Crossbar   | 16x16        | 1 cell       | 64x8         | 2x10x10 (3D)     | 1 cell     |
| Uniqueness | 47.78        | 47           | 49.85        | 50               | 49.3       |
| Uniformity | 49.79        | 47           | 47.28        | 49.5             | 49.9       |
| Reliability| 100          | 90           | 98.67        | 97               | 100        |
| Bit-aliasing | 48.57        | -            | 49.86        | ≈ 50             | 49.6       |
4 CONCLUSIONS

This work presents novel TRNG and PUF designs based on RRAM technology in the passive crossbar. The inherent variability of the switching mechanism of RRAMs is exploited to generate random bits. The TRNG output was evaluated using the NIST test suite, confirming the generated bits’ random nature. Next, the exploitation of device variations and sneak-path currents in the passive RRAM crossbar is shown to implement the PUF. The results are evaluated with extensive SPICE simulations, which verify the robust nature of the proposed PUF as 100% reliability is achieved without the need for post-processing. The proposed PUF design also demonstrates good performance for various metrics such as uniformity, uniqueness, and bit-alasing. The results are encouraging and demonstrate the potential for using RRAM-based designs in hardware security applications. We will explore the prototyping of the proposed designs and a detailed study of resilience against attacks in the future.

ACKNOWLEDGMENTS

This work was partially funded by Deutsche Forschungsgemeinschaft (DFG – German Research Foundation) under the priority programme SPP 2253 and was partially funded by the grant NRF-CRP21-2018-0003.

REFERENCES

[1] Ibrahim at al. 2022. Memristor-based PUF for lightweight cryptographic randomness. Scientific Reports 12 (05 2022). https://doi.org/10.1038/s41598-022-11240-6
[2] John at al. 2021. Halide perovskite memristors as flexible and reconfigurable physical uncloneable functions. Nature Communications 12 (06 2021). https://doi.org/10.1038/s41467-021-24057-0
[3] Kim at al. 2018. A Physical Unclonable Function With Redox-Based Nanointrinsic Resistive Memory. IEEE Transactions on Information Forensics and Security 13, 2 (2018), 437–448. https://doi.org/10.1109/TIFS.2017.2756562
[4] Nili at al. 2018. Hardware-unclonable security primitives enabled by analogue state and nonlinear conductance variations in integrated memristors. Nature Electronics 1 (03 2018). https://doi.org/10.1038/s41928-018-0039-7
[5] Rihmair at al. 2013. PUF Modeling, Attacks on Simulated and Silicon Data. IEEE Transactions on Information Forensics and Security 8, 11 (2013), 1876–1891. https://doi.org/10.1109/TIFS.2013.2279798
[6] Rajendran at al. 2015. Nano Meets Security: Exploring Nonvolatile Devices for Security Applications. Proc. IEEE 103, 5 (2015), 829–849. https://doi.org/10.1109/JPROC.2014.2387353
[7] Srinivasa at al. 2021. Cycle PUF: A Cycle operator based PUF in Carbon Nanotube FET Technology. In IEEE 21st International Conference on Nanotechnology (NANO). 13–16.
[8] Shah at al. 2021. Introducing Recurrence in Strong PUFs for Enhanced Machine Learning Attack Resistance. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 11, 2 (2021), 319–332. https://doi.org/10.1109/JETCAS.2021.3075767
[9] Armknecht et al. 2010. Memory leakage-resilient encryption based on physically uncloneable functions. In Towards Hardware-Intrinsic Security. Springer, 135–164.
[10] Balatti at al. 2016. Physical unbiased generation of random numbers with coupled resistive switching devices. IEEE Transactions on Electron Devices 63, 5 (2016), 2029–2035.
[11] Bengel et al. 2020. Variability-Aware Modeling of Filamentary Oxide-Based Bipolar Resistive Switching Cells Using SPICE Level Compact Models. IEEE TCSIS 67, 12 (2020), 4618–4630. https://doi.org/10.1109/TCSIS.2020.3018502
[12] Guajardo et al. 2007. FPGA intrinsic PUFs and their use for IP protection. In International workshop on cryptographic hardware and embedded systems. Springer, 89–89.
[13] Ganta at al. 2013. Easy-to-build arbiter physical uncloneable function with enhanced challenge/response set. In IEEE International Symposium on Quality Electronic Design (ISQED). 733–738.
[14] Gonzalez et al. 2014. Analysis of the Switching Variability in Ni/TiO2-Based RRAM Devices. IEEE Transactions on Device and Materials Reliability 14, 2 (2014), 769–771.
[15] Govindaraju et al. 2018. CSRO-based reconfigurable true random number generator using RRAM. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 26, 12 (2018), 2661–2670.
[16] Gao et al. 2020. Physical uncloneable functions. Nature Electronics 3, 2 (2020), 81–91.
[17] Gao et al. 2022. A unified PUF and TRNG design based on 40-nm RRAM with high entropy and robustness for IoT security. IEEE Transactions on Electron Devices 69, 2 (2022), 536–542.
[18] Holcomb et al. 2008. Power-up SRAM state as an identifying fingerprint and source of true random numbers. Proc. IEEE Trans. Comput. 58, 9 (2008), 1198–1210.
[19] Herder et al. 2014. Physical uncloneable functions and applications: A tutorial. Proc. IEEE 102, 8 (2014), 1126–1141.
[20] Hardhegen et al. 2018. Improved Switching Stability and the Effect of an Internal Series Resistor in HfO2/TiOx Bilayer ReRAM Cells. IEEE Trans on Electron Devices 65, 8 (2018), 3229–3236. https://doi.org/10.1109/TED.2018.2849872
[21] Lim et al. 2005. Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13, 10 (2005), 1208–1205.
[22] Liu et al. 2016. A highly reliable and tamper-resistant RRAM PUF: Design and experimental validation. In IEEE HOST. 13–18.
[23] Pang et al. 2017. Optimization of RRAM-based physical uncloneable function with a novel differential read-out method. IEEE Electron Device Letters 38, 2 (2017), 168–171.
[24] Rajendran et al. 2012. Nano-PPUF: A memristor-based security primitive. In IEEE Computer Society Annual Symposium on VLSI 84–87.
[25] Pang et al. 2021. RRAM random number generator based on train of pulses. Electronics, 84 (2021), 1381.
[26] Zhang et al. 2014. Highly reliable memory-based physical uncloneable function using spin-transfer torque MRAM. In IEEE International Symposium on Circuits and Systems (ISCAS). 2169–2172.
[27] Zhang et al. 2018. Nanoscale diffusive memristor crossbars as physical uncloneable functions. Nanoscale 10, 6 (2018), 2721–2726.
[28] Zahoore et al. 2020. Resilient random access memory (RAM): an overview of materials, switching mechanism, performance, multilevel cell (MLC) storage, modeling, and applications. Nanoscale research letters 15, 1 (2020), 1–26.