Design Considerations for Memristive Crossbar Physical Unclonable Functions

Mesbah Uddin, Md. Badruddoja Majumder, Garrett S. Rose, Karsten Beckman, Harika Manem, Zahiruddin Alamgir, and Nathaniel C. Cady

ACM Journal for Emerging Technologies in Computing System (JETC), September 2017.

A

Design Considerations for Memristive Crossbar Physical Unclonable Functions

Mesbah Uddin, University of Tennessee, Knoxville
Md. Badruddoja Majumder, University of Tennessee, Knoxville
Karsten Beckmann, SUNY Polytechnic Institute, Albany
Harika Manem, SUNY Polytechnic Institute, Albany
Zahiruddin Alamgir, SUNY Polytechnic Institute, Albany
Nathaniel C. Cady, SUNY Polytechnic Institute, Albany
Garrett S. Rose, University of Tennessee, Knoxville

Hardware security has emerged as a field concerned with issues such as integrated circuit (IC) counterfeiting, cloning, piracy and reverse engineering. Physical unclonable functions (PUF) are hardware security primitives useful for mitigating such issues by providing hardware specific fingerprints based on intrinsic process variations within individual IC implementations. As technology scaling progresses further into the nanometer region, emerging nanoelectronic technologies such as memristors or RRAMs have become interesting options for emerging computing systems. In this paper, using a comprehensive temperature dependent model of an HfO_x memristor, based on experimental measurements, we explore the best region of operation for a memristive crossbar PUF (XbarPUF). The design considered also employs XORing and a column shuffling technique to improve reliability and resilience to machine learning attacks. We present a detailed analysis for the noise margin and discuss the scalability of the XbarPUF structure. Finally, we present results for estimates of area, power and delay alongside security performance metrics to analyze the strengths and weaknesses of the XbarPUF. Our XbarPUF exhibits nearly ideal (near 50%) uniqueness, bit-aliasing and uniformity, good reliability of 90% and up (with 100% being ideal), a very small footprint and low average power consumption $\approx 104\mu$W.

CCS Concepts: •**Hardware security** → *Physical unclonable function, PUF, memristor PUF;* •**Emerging technologies** → *Memristor, RRAM, ReRAM, Transition metal oxide;*

Additional Key Words and Phrases: Hardware security, crossbar PUF, HfO2 memristor

ACM Reference Format:
M. Uddin *et. al., 2017. Design Considerations for Memristive Crossbar Physical Unclonable Functions.* ACM Trans. Embedd. Comput. Syst. *V, N, Article A (January YYYY), 24 pages.*
DOI: *http://dx.doi.org/10.1145/0000000.0000000*

1. INTRODUCTION

A physical unclonable function (PUF) is a security primitive useful for dealing with a variety of security concerns, such as integrated circuit (IC) counterfeiting, piracy, cloning, and secret key generation [Suh and Devadas 2007; Paral and Devadas 2011]. PUFs exploit uncontrollable physical differences among chips such as delays in circuit paths, leakage currents, start-up characteristics, threshold voltages and internal re-

This material is partially based upon work supported by the Air Force Office of Scientific Research under award number FA9550-16-1-0301.
Author's addresses: Mesbah Uddin, Md. Badruddoja Majumder and Garrett S. Rose, Electrical Engineering and Computer Science Department, University of Tennessee, Knoxville, TN, USA; Karsten Beckmann, Harika Manem, Zahiruddin Alamgir and Nathaniel C. Cady, Colleges of Nanoscale Science & Engineering, SUNY Polytechnic Institute, Albany, NY, USA.

DOI: http://dx.doi.org/10.1145/0000000.0000000

sistance. These differences depend on intrinsic process variations and can be thought of as "fingerprints" for a particular chip. Usually there is a set of inputs or challenges, C applied to a particular IC PUF that yields some output or response, R. That response is a function of the IC hardware itself, being controlled by the challenge input. Thus, a PUF can be defined as: $f_{chip}(C) = R$.

PUFs can be categorized into two types based on its implementation: (i) delay based PUF and (ii) memory based PUF. Memory based PUFs like SRAM PUF [Guajardo et al. 2007], memristor PUF (memristor as a memory element) [Koeberl et al. 2013] or STT-MRAM PUF [Zhang et al. 2014] utilize variations in the start-value of uninitialized memory elements. Variations in initial states could arise from the mismatch in the cross-coupled inverters as in a SRAM cell due to process variation. Therefore, these memory based PUFs could potentially be prone to noise and slight environmental changes and not very reliable in the face of changing operating condition. The responses could also be biased heavily towards a '1' or a '0' rendering them unsuitable for cryptographic applications.

A delay based PUF, as the name suggests, depends on delay variation arising from process variations across unique implementations of ICs. Most delay PUFs utilize the delay difference between two elements or paths, not absolute delays and thus, are robust against noise and environmental changes. One of the most well-known examples of a delay based PUF is the arbiter-based PUF (APUF) that depends on variations in the propagation path delays of chains of switch elements [Suh et al. 2005]. The APUF consists of a series of identically laid out two input/two output switch boxes to create two propagation delay paths for a signal. Due to the inherent process variation among the physical implementations of these switch-boxes, a signal propagates faster through one path relative to the other with an arbiter used to generate a corresponding response. Ring-oscillator PUF or RO-PUF [Suh and Devadas 2007] is another delay based CMOS PUF which can also be implemented easily in an FPGA like an APUF. Process variation dictates that different implementations of ring-oscillators will exhibit different frequencies. This is primarily due to slightly different delays for the inverters used to construct each ring-oscillator. These frequencies are then compared using counters and comparators such that their differences are converted into a unique signature.

As technologies continue to scale, the need has arisen for increasingly lightweight PUFs that leverage nano-scale components for lower area utilization and improved energy efficiency. Memristive PUFs have recently been proposed as nanoelectronic hardware security primitives. We use the word 'memristor' to mean metal-oxide memristor or RRAM (resistive RAM or ReRAM) or transition metal-oxide. A simple PUF based on the write-time of a memristor, dubbed write-time memristive PUF (WTMPUF), was introduced in [Rose et al. 2013a]. The WTMPUF was later experimentally realized and presented by Mazady *et al.* [Mazady et al. 2015]. Kavehei *et al.* presented a PUF built from a combination of RRAM devices (memristors) and RO-PUFs named mrPUF [Kavehei et al. 2013]. Other works that focus on memristor based PUF and memristor characteristics are presented in [Chen 2015; Abunahla et al. 2014; Chen et al. 2015; Liu et al. 2015; Beckmann et al. 2016].

In a previous work, we have presented a PUF designed using a dense crossbar array of memristors, the memristive crossbar PUF (XbarPUF)[Rose and Meade 2015; Uddin et al. 2016]. The XbarPUF leverages the relative write-times of pairs of memristor-based circuits and thus does not require control of the specific minimum write-time of the devices. The crossbar architecture is very regular in shape and therefore, easy to fabricate. They also take less area and allow a dense array implementation. Therefore, the crossbar architecture has become an attractive choice for many memory and PUF implementations with nano-devices [Lugli et al. 2013]. In this work, we have also

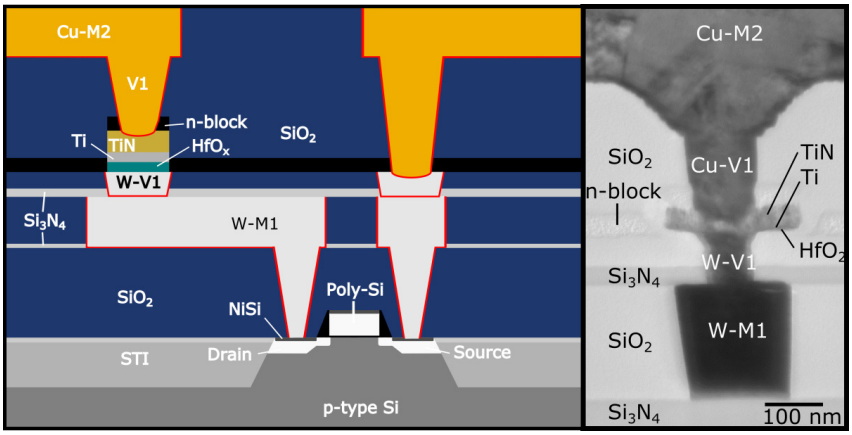

Fig. 1. Illustration (left) and TEM image (right) of a hafnium-oxide memristor embedded between the M1 and M2 layers. A Tungsten V1 via is used for the inert bottom electrode, HfO_x is the oxide layer, the Ti layer serves as an oxygen-getter for the HfO_x, TiN is used for the inert top electrode and copper via (V1) and metal-2 (M2) layer provide a top contact for the device. In addition, the Illustration depicts a seemless integrated 1 memristor 1 transistor (1M1T) structure where the transistor acts as the current limiting device [Beckmann et al. 2016].

implemented a PUF using a crossbar array of memristors by extending our previous design [Rose and Meade 2015; Uddin et al. 2016].

The contributions of this work include: (1) an updated memristor model including temperature dependence on memristive states and threshold voltages; (2) detailed analysis for circuit-level choices under varying parameters; (3) analyzing impact of aging and temperature on reliability of the XbarPUF; (4) analyzing the impact of scaling based on different crossbar size; (5) overhead calculations (power, area and delay) based on the whole circuit simulation; and (6) insights for device designers regarding device parameters targets.

We provide background discussion on memristors, specifically the HfO_x memristor considered in this work, memristive PUFs and the PUF performance metrics in section 2 and 3. The memristor model developed and used for this work is briefly discussed in subsection 2.3. In section 4 we describe the memristive crossbar-based PUF design, including circuit-level details and rationale behind choosing different circuit components. Performance in terms of PUF security metrics using Monte Carlo simulations are presented in section 5. We also provide details on the overhead, namely area, power and delay. Some analyses and discussion about the performance and limitations of the HfO_x based memristor PUF are provided in section 6. Finally, future prospects and concluding remarks are provided in sections 7 and 8, respectively.

2. MEMRISTOR BACKGROUND

2.1. Device Fabrication

The memristive device technology considered in this work was designed and fabricated at the SUNY Polytechnic Institute, Center for Semiconductor Research (CSR) [Beckmann et al. 2016]. Devices were manufactured on 300mm wafers where they were integrated with the IBM 65nm 10LPe CMOS process technology. It is worth noting that CSR facilities allow for an area-efficient and seamless flow of all the layers including front-end CMOS and back-end memristive and metalization layers. The seamless integration of CMOS with memristive technology is a unique feature as compared to related efforts where memristive devices are integrated post-fabrication [Mazady

et al. 2015]. Memristor devices are embedded between metal 1 (M1) and metal 2 (M2) interconnects using a custom and cost-effective build technique. In this process, standard copper is replaced with tungsten metalization layer and also an additional via (W-V1) is used beneath the memristor. HfO_2 layer was deposited layer between two metal layers using a precise atomic layer deposition (ALD) technique. Composition of M2 and V1 layers was altered for using the front-end-of-the-line (FEOL) tools in this HfO_2 deposition process. This metal-oxide film works as the active switching layer for the considered memristor device. The switching layer is capped with a Ti oxygen getter layer and a TiN top electrode to enable the switching behavior of the devices. A cross-section of the fabricated HfO_2 memristor is shown in Fig. 1 [Beckmann et al. 2016].

2.2. Memristor Basics

The memristor or the RRAM, first theorized by Leon Chua [Chua 1971] in 1971, is considered as the fourth fundamental circuit element after resistors, inductors and capacitors. Memristance or "memory resistance" is dependent on the past history of current that has flown through the device. Alternatively, memristance is determined by the magnitude and duration of a voltage applied across the memristor. A metal-oxide TiO_{2-x} device was demonstrated by HP Labs in 2008 [Strukov et al. 2008] and shown to be a memristor. A memristor can have different memristance states and these states can represent various logic states and thus act as a memory element. Memristors are, therefore, non-volatile in nature. A high positive or negative pulse can be applied to switch the memristance states to perform a memory write. On the other hand, the application of a small voltage does not alter the memristance and thus a read operation can be performed without any modification of the memory. The different memristive states and switching voltages show high variability in practice from one device to another and even from one cycle to another. This high variance in nanoscale devices such as memristors are the main source of entropy exploited for PUF operation.

In most metal-oxide memristors there are two stable memristive states; one showing high resistance or OFF state and the other showing low resistance or ON state. Temperature affects the normal operating condition of a memristor. We have data for a limited range of temperature for the fabricated HfO_x memristor in our lab. We have also referred to the literature for temperature dependence of metal oxide memristors, specifically for HfO_x based memristors. The findings are consistent with our observations and expectations. The probability of defect formation at the dielectric/electrode interface in the conduction filament of a HfO_x or metal-oxide memristor increases with the increase of temperature. This enhances trap-assisted tunneling (TAT) leakage current for the off state. Therefore, the OFF state of a memristor can be compared with semiconductor behavior and resistance decreases with increasing temperature as shown in [Fang et al. 2010; Walczyk et al. 2011]. The ON state is more analogous to metallic characteristics such that the resistance increases with increasing temperature [Fang et al. 2010; Walczyk et al. 2011]. As a result, the OFF/ON ratio decreases with increasing temperature.

2.3. Memristor Device Behavior and Modeling

The memristor model used in this work has been adapted from a model first presented by McDonald *et al.* in [McDonald et al. 2012]. The original model was developed to represent unipolar, non-polar or bipolar behavior. However, this work is specifically concerned with bipolar behavior. Model specifications are based on experimental results, similar to those as shown in Fig. 2. It is worth pointing out that the HRS and LRS values for both figures are 11kΩ and 4kΩ, respectively. For the rest of this work,

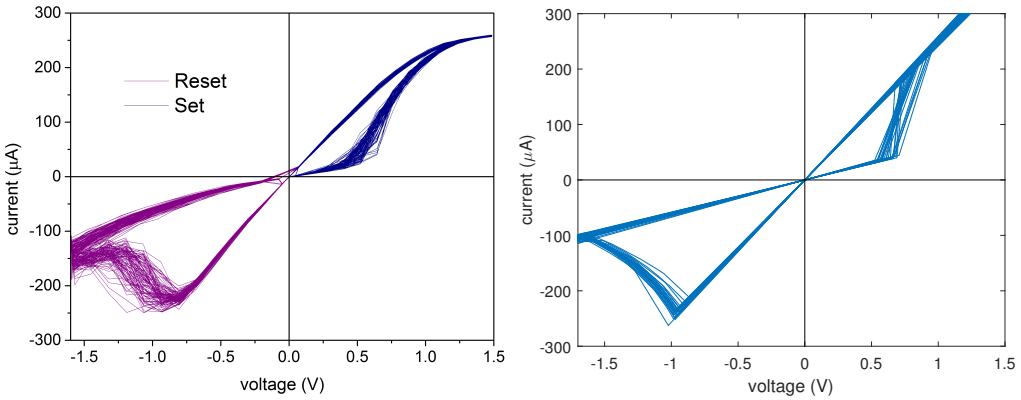

Fig. 2. IV measurements of a memristor (left) [Beckmann et al. 2016] and simulated IV characteristics using our memristor model (right).

we have used 300KΩ and 30kΩ as base values for HRS and LRS, respectively, which are reasonably good assumptions for a HfO_x memristor.

Like other metal-oxide memristors, our HfO_x memristor has two defining memristance levels: the high resistance state (OFF state) or HRS and the low resistance state (ON state) or LRS. The memristance at any time is determined based on the previous memristance state and also the polarity and duration of the applied voltage.

When the magnitude of applied voltage is greater than the set or positive threshold voltage, the memristor transitions from HRS to LRS and the memristance is updated at small steps:

$$M(t_{i+1}) = M(t_i) - \frac{\Delta r \Delta t \left|V(t_{i+1})\right|}{t_{swp} V_{tp}}, \tag{1}$$

while for a reset operation (LRS to HRS) operation the memristance is updated by:

$$M(t_{i+1}) = M(t_i) + \frac{\Delta r \Delta t \left|V(t_{i+1})\right|}{t_{swn} V_{tn}}, \tag{2}$$

where M is the memristance at that instant, Δr is the absolute difference of resistance between the two memristance states, HRS and LRS, Δt is the simulation step-size, $V(t)$ is the applied voltage across the memristor, t_{swp} and t_{swn} are the minimum required time to fully set or reset, respectively, under a bias greater than either of the voltage thresholds, and V_{tp} (V_{tn}) is the minimum threshold voltages needed to switch the memristance.

As memristors are emerging nanoelectronic devices, they typically exhibit relatively larger process variations contributing to die-to-die variations. These larger variations in process translate to a large variance in device parameters, for example as examined by Pouyan *et al.* [Pouyan et al. 2014]. The OFF resistance, or HRS, and ON resistance, or LRS, are the parameters most sensitive to process variability, with the variation of HRS typically being greater than that of LRS. We chose 20% and 10% for correspond-

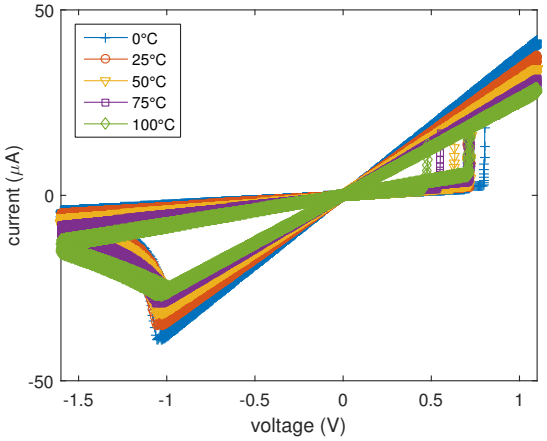

Fig. 3. Simulated IV characteristics of our memristor model with varying temperature.

Table I. Temperature coefficients for memristor parameters

α_{LRS}	α_{HRS}	α_{Vtp}	α_{Vtn}
0.004	-0.008	-0.001	0.0008

ing die-to-die variations σ_{HRS} and σ_{LRS}, respectively, in the model used to simulate the XbarPUF. It is worth mentioning that these assumptions are based only on the average fabricated memristor and the added variance follows the inter-die variation. Two different sets of intra-die variations (across time) are used for simulation. One set includes 10% cycle-to-cycle variance for HRS and 5% variance for the remaining memristor parameters: LRS, t_{swp}, t_{swn}, V_{tp} and V_{tn}. The other set considered a flat 2% variance for all the major six parameters of the memristor, representing the expected variance once the memristor technology matures. For every simulation run, parameter values are reevaluated using their corresponding mean and variance assuming a normal distribution. Figure 2 (right) illustrates simulated I-V characteristics using our model. The multiple paths of this I-V curve indicate the presence of cycle-to-cycle variations for all key parameters. It is worth pointing out the slight bend at both ends of the IV curve of the left of Fig. 2 is due to the current limiting (current compliance) circuit elements and, therefore, is not present in the simulated result (right of Fig. 2).

We also incorporate temperature dependence in our memristor model. We have learned that during LRS, HfO_x memristors show metal-like behavior and, therefore, resistance increases linearly with increasing temperature during this state [Beckmann et al. 2016]. However, during HRS, semiconductor-like behavior is observed and thus the value of HRS decreases fast with increasing temperature [T. Cabout 2013]. Threshold voltages (V_{tp} and V_{tn}) vary very slowly or do not change noticeably [T. Cabout 2013; Walczyk et al. 2011] in the operating regions for temperature considered. Therefore, temperature coefficients for both of these parameters are very small. Equation 3 defines these relationships:

$$X_\theta = X_{ref}[1 + \alpha_x(T_\theta - T_{ref})], \tag{3}$$

where T_{ref} or reference temperature is the room temperature (25°C) and α_x is the temperature coefficient for any of the four parameters X_θ. The values of temperature coefficients are listed in Table I.

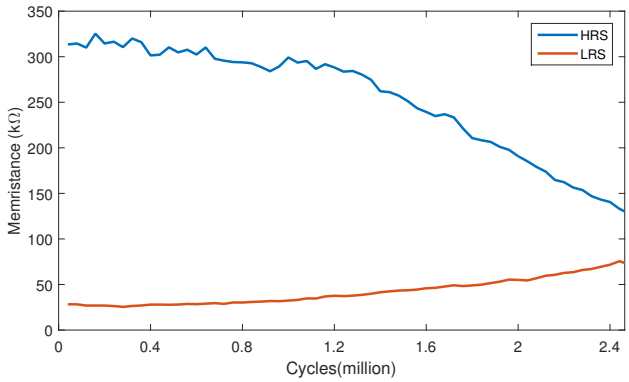

Fig. 4. Simulated effect of (worst case) aging on both HRS and LRS of a memristor over time.

We do not find any suitable information regarding the dependence of switching times of a memristor with temperature and, therefore, haven't included this dependence in the model yet. Fig. 3 shows the IV curve obtained from our memristor model for different temperatures, showing a trend similar to what is found in the literature [Walczyk et al. 2011]. We have used the temperature range 0^oC to 100^oC in this work. The effect of increasing temperature on the memristive states is that the HRS/LRS ratio decreases with increasing temperature. Therefore, noise margin between read and write operations also decrease and would be the eventual limiting factor for scaling the memristive crossbar.

Another important memristor characteristic related to PUF performance is aging or endurance of the device. Researchers have noticed a few different aging patterns as listed in the literature [Benoist et al. 2014; Pouyan et al. 2014; Chen et al. 2011; Beckmann et al. 2015]. The mean and distribution of HRS would decrease with time and that of LRS would increase with time in the worst case. Thus the HRS/LRS ratio would decrease i.e. the margin between logic '1' and logic '0' would get narrower. We assume an almost linear aging rate for both LRS and HRS, similar to Equation 3, while the rate of change of LRS being slower than the rate of change of HRS, as observed in the worst type of aging of these metal-oxide memristors. This is illustrated in Fig. 4 for a clock frequency of 10kHz. Since clock time period is higher than set/reset switching time, a memristor would be set and reset at every clock cycle and thus, accelerated aging is observed in this figure. This figure is interpolated from the partial data gathered in our lab and also based on other works presented in the literature [Benoist et al. 2014; Chen et al. 2011; Beckmann et al. 2015]. However, this is just a demonstration and aging can vary quite drastically across different device recipes.

3. MEMRISTOR PUF BACKGROUND

3.1. Previous Memristor based PUF Works

All PUFs depend on some unpredictable but stable characteristics of a set of identically implemented devices. As mentioned earlier, a memristor 'remembers' its past resistive state and the next state is determined by the magnitude and duration of an applied voltage. If a positive set voltage is applied across a memristor for a duration greater than the positive switching time, then the memristor is said to be set or its resistance decreases to the low resistance state (LRS). On the other hand, if a negative reset voltage is applied with a duration greater in magnitude than the negative switching time, then the memristor is said to be reset to the high resistance state (HRS). The mem-

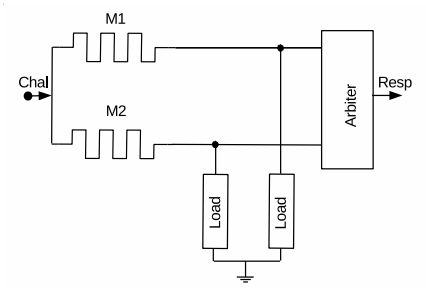

Fig. 5. Memristive PUF depending on relative write-time.

ristance does not change if the applied voltage magnitude is less than the set voltage and at the same time, greater than the reset voltage. Because of process variation, individual memristors have slightly different switching times. If we determine the mean switching time of a set operation for a memristor and apply a set voltage for that specified period of time, a specific memristor may or may not switch its state depending on its actual switching time being greater than the mean or not. This property can be leveraged for a PUF where the set voltage pulse being determined by the switching times of the memristors is the main source of entropy. This concept was first presented in [Rose et al. 2013b] as the absolute write-time based memristor PUF.

There are practical limitations to the absolute write-time memristor PUF. First, one needs to go through a rigorous and exhaustive method to determine the nominal or mean switching time of the set operation for a set of memristors to be used in the PUF. If this switching time distribution is not uniform, then the probability of getting either '1' or '0' could be greater than getting the other and the PUF output would no longer be truly random. Also, any environmental changes such as temperature, supply rail noise and aging could change the switching time of a memristor and thus, degrade performance. Therefore, an alternative approach has been proposed in [Rose et al. 2013a]. Instead of measuring output voltage directly from a single memristor, the same set voltage is applied across two memristors for any specified amount of time as shown in the Fig. 5. Because of internal process variations, one memristor would be set relatively faster than the other and thus, would give rise to a higher output voltage. An arbiter captures which memristor first gives rise to '1' or high voltage and the PUF response is determined in this way.

The crossbar architecture based memristor PUF was proposed in [Rose and Meade 2015] as XbarPUF. The XbarPUF has similar abstract architecture like an arbiter PUF (APUF), where a memristor acts as a switch instead of 2-input, 2-output switches of an APUF. The crossbar architecture is very dense and since memristors are very small compared to those switches, XbarPUF requires much lower area. All the memristors of a column of an XbarPUF contribute to the voltage drop of the column load resistance at the end. The load voltages from two adjacent column are compared in real time to detect voltage difference arising from different memristances of two columns. Moreover, memristors in each row of each column are controlled by external challenge input. Therefore, each output or response is a function of each challenge input and hence the whole architecture is easily scalable. At the start of each data sampling period, all memristors are reset and challenge is applied on a read-monitored write [Manem and Rose 2011] basis. As soon as stable voltage differences are found for all the response bits, challenge is stopped and response is sampled using a low read voltage. This crossbar architecture is our reference circuit block for this work.

In practice, many memristors as well as HfO_x memristor considered here require a forming step, where a higher set voltage is needed to initiate the device operation as a

memristor. It is only required once in real device and we do not have that forming step in our model. However, extra circuitries are required to take care of on-chip forming.

3.2. PUF Performance Metrics

In order to gauge the degree of security provided by a PUF and to be able to fairly compare it with other PUF implementations, standardized performance metrics have been devised. Maiti *et al.* discusses several metrics to quantify a PUF's performance [Maiti et al. 2013]. The major four functional metrics are: uniqueness, reliability, bit-aliasing and uniformity. These measures are used to quantify a PUF's performance across multiple device dimensions: inter-chip space, intra-chip space, and time.

1) Uniqueness: It is a measure of a PUF's ability to produce a unique ID in terms of its challenge-response pairs which are a specific function of its implementation on a given chip. To be able to efficiently distinguish every IC chip (from other equivalent ones) with a PUF circuit, the uniqueness must be tending to 50. That means, for a given challenge set, almost half the responses produced by two PUFs should be different from each other. Uniqueness or also known as inter-chip hamming distance is defined as:

$$Uniqueness(\%) = 100 * \frac{2}{N_{chips}.(N_{chips}-1)} \sum_{i=1}^{N_{chips}-1} \sum_{j=i+1}^{N_{chips}} r_i \oplus r_j, \tag{4}$$

for each response bit and each challenge set, where N_{chips} is the number of PUFs or chips to be measured, r_i and r_j are responses from the i-th and j-th chip respectively.

2) Uniformity: Another PUF performance metric which measures a PUF's ability to produce distinct responses across a set of challenges is the uniformity. With the flipping of even a single bit of the challenge, nearly half of the response bits are expected to flip. Effectively, uniformity is a measure of the ratio of 0's and 1's across the whole of the response set of the PUF. A PUF with poor uniformity would allow the attacker to reduce the possible response space and get a better prediction. This metric is defined as:

$$Uniformity(\%) = 100 * \frac{1}{N_{challenges}} \sum_{c=1}^{N_{challenges}} r_c, \tag{5}$$

for each response bit and each chip, where $N_{challenges}$ is the number of challenges applied to a single PUF and r_c is the response for c-th challenge.

3) Reliability: Also represented as intra-chip hamming distance, this metric quantifies a PUF's consistency over time. If a PUF is unable to produce the same response every time for a given challenge, then the PUF is considered unreliable and may require error-correction. Reliability, as a metric is defined as:

$$Reliability(\%) = 100 - 100 * \frac{2}{N_{cycles}.(N_{cycles}-1)} \sum_{t=1}^{N_{cycles}-1} \sum_{it=t+1}^{N_{cycles}} r_t \oplus r_{it}, \tag{6}$$

per response bit and chip, where N_{cycles} is the number of times of applying a challenge and measuring a response.

4) Bit-Aliasing: Different PUFs might produce similar responses for certain challenges. This would decrease the unpredictability of the PUF. Bit aliasing measures the average hamming distance for the k-th response bit of different PUFs. This metric is defined as:

$$BitAliasing_k(\%) = 100 * \frac{1}{N_{chips}} \sum_{m=1}^{N_{chips}} r_{k,m}, \tag{7}$$

for each bit k of a response.

The above four metrics are very important for a PUF as they are measures of security performance of a system. However, they fail to take into account the real-world application scenario. In order to be implemented as a nano-security primitive in a system, a PUF also needs to be energy-efficient and lightweight as well as providing security. For small deployable devices, area requirement may be a critical factor. Power or Energy consumption is also an essential factor for any battery-run or resource-constraint devices. Therefore, area and power analysis are very performance important metrics in the evaluation of these lightweight security systems as well as PUFs. The delay of a PUF is not as important as other metrics because a PUF would only be used occasionally for security verification or identification and, therefore, it doesn't need to be as fast as the main functional parts of a chip.

4. DESIGN CHOICE

4.1. XbarPUF circuit

A circuit-level block diagram of the crossbar PUF used in this work is shown in Fig. 6. It is a $n \times m$ XbarPUF in terms of number of challenge and response bits, respectively. There are two rows for each challenge bit, thus having a total of $2n$ rows, and there are four columns for each XORed response bit, having a total of $4m$ columns. One of the two rows for one challenge bit is driven the actual challenge being applied and the adjacent row is driven by the inverted challenge signal. If a challenge bit is high, then it sets all the memristors of that row to LRS, but the memristors of the row driven by the inverted challenge bit remains at HRS. The situation reverses when the challenge bit is low.

At the end of each crossbar column, a load resistance is connected in series to measure the voltage differences arising from differences in memristive states of memristors of that column. Because of the high process variation present, the switching rate and both the LRS and HRS of one memristor vary with respect to other memristors. Therefore, the time required to apply a challenge and obtaining a high voltage at the load resistance varies across different columns. An arbiter circuit is used to measure the voltage mismatch arising from this timing mismatch. The voltages across the two load resistances of two adjacent crossbar columns are then connected to the two inputs of an arbiter to determine which crossbar column achieve an equivalent memristance lower than the load, thus having a higher voltage drop at the load. In this way, the effective resistance values of two columns are reduced incrementally with positive set pulses such that the resistance values race one another until one "wins", as determined by the arbiter. A "DNE" signal (in Fig. 6) is used to check if valid output is found for the response bits or not and once all the response bits are valid, it indicates to start new data sampling session and apply a new challenge.

A fast arbiter is needed for reliable sampling and measurement of voltages at its two input nodes. Researchers have used flip-flops or latches for this purpose in other PUFs like arbiter PUFs. However, a flip-flop does not provide symmetric paths from input to output for the two inputs. Thus, one input path is faster than the other, and introduces bias. We have opted for a S-R latch in this work which has symmetric paths from both inputs (S and R) to output. As the number of rows scale up, the crossbar column voltage differences could get narrower in real time, thereby requiring more precise measurements. This situation, however, could be improved further using memristors

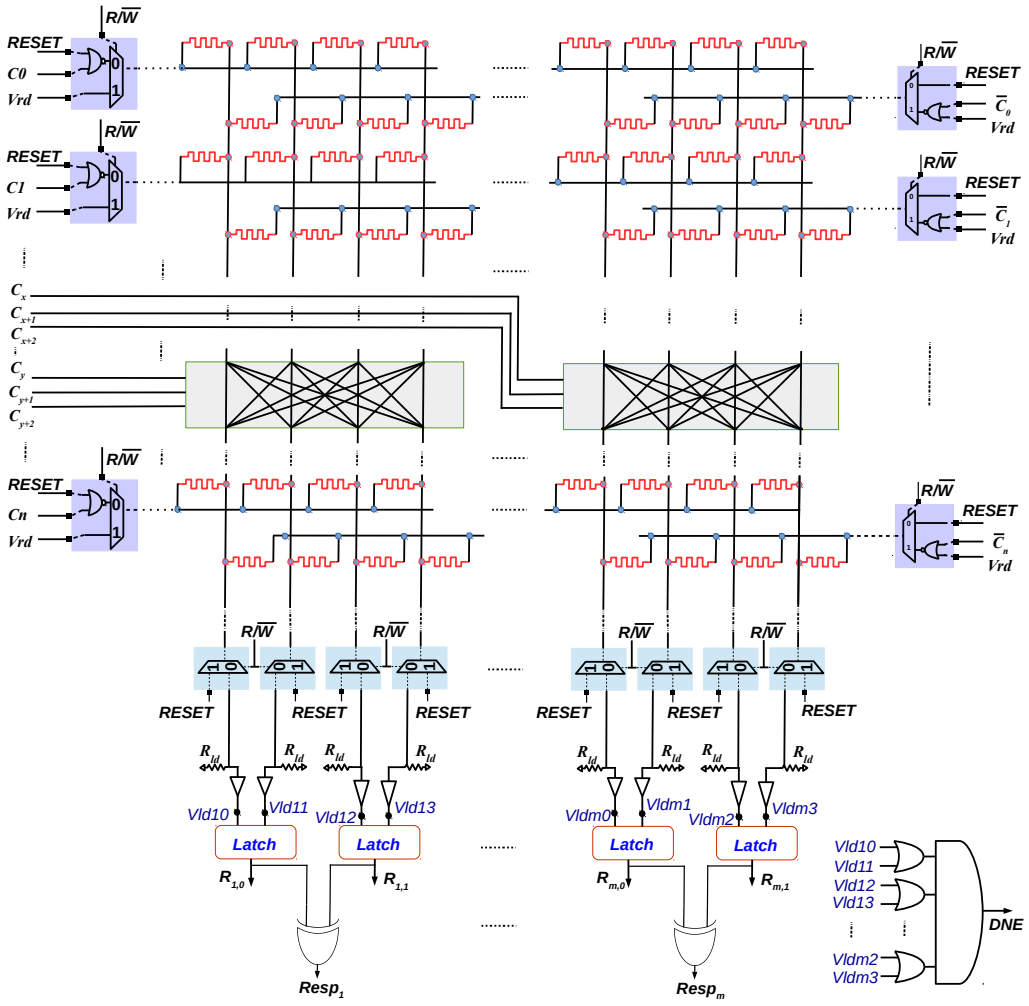

Fig. 6. Schematic of a $n \times m$ XORed crossbar-memristive PUF including column shuffling (crossbar architecture and peripheral read-write circuit) as mentioned. This design requires a memristive crossbar of size $2n \times 4m$ including $8nm$ total memristors [Rose and Meade 2015; Uddin et al. 2017]

with higher OFF/ON ratio. More sensitive circuitry could be used to measure voltage differences more accurately. But this could potentially give rise to a response before it is actually stable and thus disrupt the response generation system in our circuit. In the XbarPUF, a small fixed voltage pulse of 1ns or 100ps is used and whenever one of the inputs gets a voltage above VDD/2 the arbiter has enough time (2-4ps) to reflect that into the output. Since any degradation in performance is not found with this technique for XORed XbarPUF upto 32×2, we propose using the SR latch.

The outputs from a pair of arbiter are XORed to produce a final response bit. A larger fan-in at the XOR gate could reduce the stability of the responses due to racing among the inputs. However, for a two-input XOR, there shouldn't be any negative impact on reliability as is validated by our simulation results. The reason behind including XORing in our XbarPUF is to actually increase reliability and robustness as has been

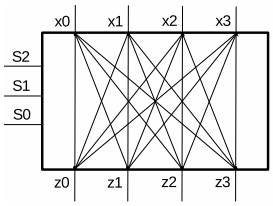

Fig. 7. Block diagram column shuffling circuit.

suggested by Herder $et.al$ in [Herder et al. 2014] for arbiter PUF. In this design, a single response bit is generated from four columns of the crossbar.

However, XORing is applied at the end of crossbar architecture and thus a good non-linear classifier can still separately model PUF responses along with XORing operations and predict responses with sufficient accuracy. Therefore, we have added another circuit block inside the crossbar architecture which would add further non-linearity. We have named this 'column shuffling block' or 'column swapping block'. As the name suggests, it can route current from one crossbar column to a different column and thus, arriving at different arbiters, producing different responses. In our demonstration, we have implemented a 4×4 column shuffling block that can route at any of the four paths of one end to any of the four paths of the other end as shown in Fig. 7.

Three shuffling control bits are needed for this block. These three bits can be a part of the whole challenge set. For each set of four columns inside a crossbar, there could be different or same sets of these selector bits as in Fig. 6. Thus they add additional complexity in the XbarPUF and would make it stronger against modeling attacks. It is a very important point that since this column shuffling logic block is in the path from challenge to response, it shouldn't create any additional bias (delay difference) to PUF circuit. All path delays from 'x' nodes to 'z' nodes in Fig. 7 have to be the same and should be interchangeable as current flows in both directions in the crossbar. Column shuffling control logic is constructed after maintaining all these requirements.

The main source of entropy of our crossbar PUF comes from process dependent parameters such as switching times and differences of HRS and LRS among the memristors in subsequent columns. The challenge bits are applied directly across each row as set voltage pulses for all memristors in this crossbar arrangement. The response from each arbiter or latch is dependent on all possible delays created by 2^N different combinations of memristive states from the columns. Using shuffling challenge bits, columns can be routed to different arbiter input also. Furthermore, outputs from two arbiters are combined with an XOR gate [Uddin et al. 2016] to generate each final response bit. All these features make the response of this PUF a more complex function of internal process variations of all memristors of a column and all the challenges. Due to having these complexity in its structure, we expect our XORed XbarPUF along with column shuffling to be very robust against machine learning based modeling attacks.

4.2. Choosing Load Resistance and Clock Frequency

As mentioned in previous sub-section, we are using a reset first approach to evaluate the response from our memristive crossbar based PUF circuit.

In this approach, we are employing a "read first and then write" method when applying a challenge. A very short pulse depending on the challenge bit is used to nudge a memristor from HRS towards LRS. Then a read operation is performed. If neither column in a pair has reached the desired effective resistance, another write operation is performed. Thus, we perform a read-monitored write [Manem and Rose 2011] approach to incrementally decrease the column resistance until the arbiter determines a

winner, challenge application is stopped and a response bit is produced. The choice of clock frequency is very important for a successful read and to differentiate the voltage between two column load resistance (R_{ld}). A high clock frequency will help in determining tiny differences in voltage and to have more confidence in the response. If the frequency is not high enough, then the memristors could switch fully from HRS to LRS in just one or a few clock pulses and both columns would reach the desired resistance effectively at the same time and would be indistinguishable. Therefore, we prefer a relatively high frequency clock (10MHz) so that it takes many clock pulses to switch the memristor states, thus ensuring notable differences in resistance values and correspondingly, output load voltages.

We have also performed noise margin analysis for our crossbar circuit and have determined optimum load resistance. During a reset operation, all memristors are reset to HRS. After applying a challenge, one half of the rows of memristors switch to LRS while the rest remain in HRS. The percent difference in load voltage due to the resistance difference in crossbar column pairs is what we call noise margin (NM) for our circuit. The load resistance (R_{ld}) selected is critical for the ability of the read circuitry to reliably differentiate between logic '0' and logic '1'. Therefore, load resistance, R_{ld} is desired to be of some value which would have a maximum voltage drop during a reset operation and minimum voltage drop after applying a challenge. When all the memristors of a column are reset to HRS, the equivalent resistance of a column is R_1 = HRS/8 for a 4×2 crossbar which have effectively 8 rows and all the rows are in parallel during a reset. After applying a challenge, half of the memristors move toward LRS (set) and the other half remain at HRS. Therefore, during the read operation, there are 8 memristors in parallel, 4 of which are at LRS and 4 at HRS. The equivalent column resistance is R_2 = (HRS/4) $\parallel$ (LRS/4) at this time. For a N-bit challenge (or 2N rows) XbarPUF, the generalized form of R_1 and R_2 would be:

$$R_1 = \frac{HRS}{2N}, R_2 = \frac{HRS}{N} \parallel \frac{LRS}{N}. \tag{8}$$

Obviously $R_1 > R_2$ and we want R_{ld} to be of some value which would have a larger voltage drop when R_1 is in series and a smaller voltage drop when R_2 is in series with the load. This situation is illustrated in Fig. 8 and the corresponding voltages across R_{ld} are defined in equations 9 and 10:

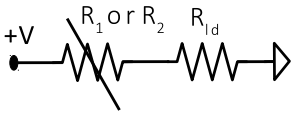

Fig. 8. Two resistance in series, one is the equivalent memristance of a crossbar column and another is the load resistance.

$$V_{eq,R_1} = \frac{R_{ld}}{R_1 + R_{ld}} * V \tag{9}$$

$$V_{eq,R_2} = \frac{R_{ld}}{R_2 + R_{ld}} * V \tag{10}$$

In order to maximize difference between these two voltage levels, we need to maximize the difference (ΔV_R) between the resistances in these two situations with respect

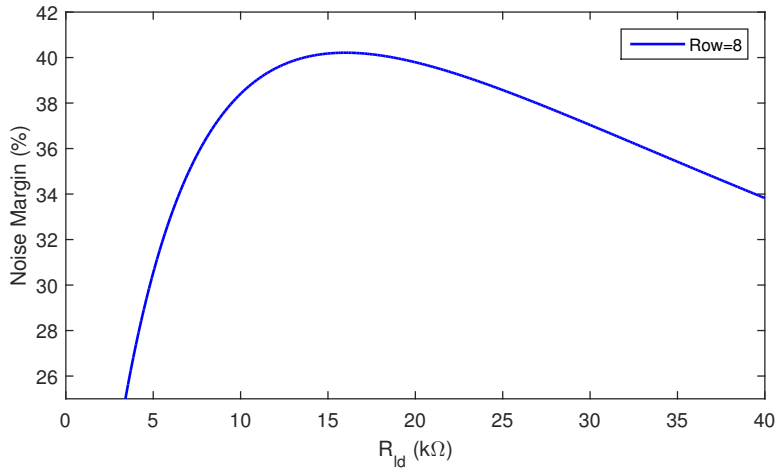

Fig. 9. Relationship between noise margin and the load resistance.

to R_{ld}. This is represented in equation 11 and by differentiating this equation with respect to R_{ld} and then setting to zero, we get the condition for maximum noise margin as defined in equation 12.

$$\Delta V_R = V_{eq,R_2} - V_{eq,R_1} \tag{11}$$

$$R_{ld,best} = \sqrt{R_1.R_2} = \sqrt{\frac{HRS}{2N} * (\frac{HRS}{N} \parallel \frac{LRS}{N})} \tag{12}$$

We also performed a sweep of R_{ld} and determined the differences between $V_{eq,R1}$ and $V_{eq,R2}$. For this test case, we have used LRS = 30KΩ, HRS = 300KΩ and N = 4. Results for this analysis are shown in Fig. 9. For the same 4×2 crossbar circuit with LRS = 30KΩ and HRS = 300KΩ, one can achieve about 40% noise margin which translates to a 240mV noise margin for a 600mV read voltage.

The reset first approach greatly benefits from any increase in the HRS/LRS ratio. We show the plot of maximum achievable noise margin vs. HRS/LRS ratio in Fig. 10. It is evident that the difference between two resistances at these two logic levels increases with increasing HRS (or HRS/LRS ratio while LRS remains the same), such that the noise margin improves.

Increasing the number of rows doesn't change the maximum achievable noise margin. However, including more rows does mean more memristors in parallel which demands lower load resistance and finer resolution. From Fig. 11, it is clear that the curve around the best load resistance point gets narrower as the number of rows increases. Therefore, for an XbarPUF with many rows, a small change in load resistance could adversely affect the noise margin. This poses an upper limit on the maximum number of crossbar rows depending on the resolution of the load resistance and memristance variations.

5. SIMULATION SETUP

In this work, we have experimented with four different crossbar sizes, 4×2, 8×2, 16×2 and 32×2, all XORed XbarPUF. Due to huge simulation time and memory requirements, we have not considered bigger crossbars than these. Instead, we have used

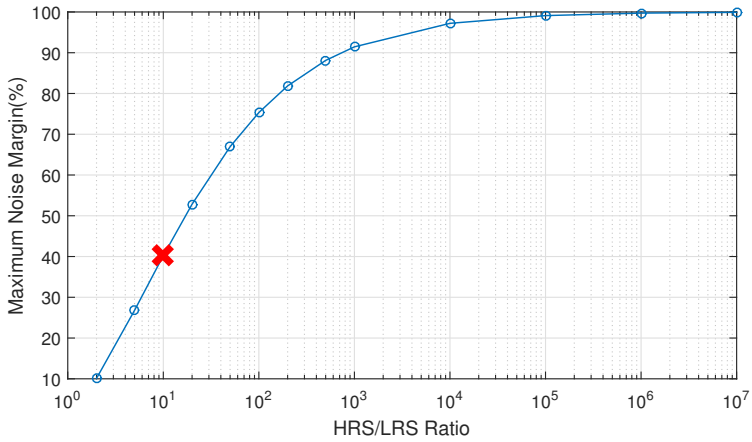

Fig. 10. Relationship between the maximum achievable noise margin and HRS/LRS ratio.

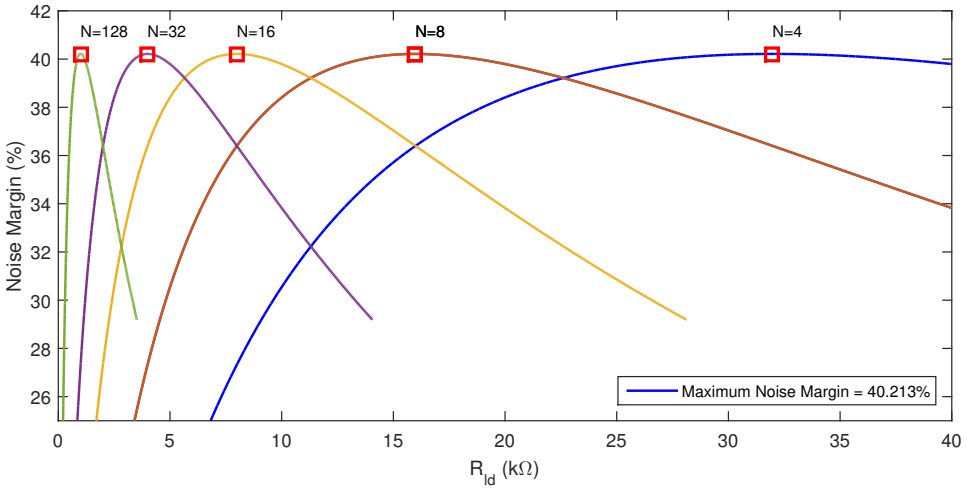

Fig. 11. Relationship between maximum achievable noise margin and the number of rows (N).

exactly the same set of device and circuit level parameters to get a fair comparison across these different sized XbarPUFs. We have tuned the device parameters to explore different regions of operation, Table II lists the primary set of memristor device paramters used in this work. These are reasonable values for a HfO_x memristor.

Table II. Device level parameters (of a memristor)

Parameters	HRS	LRS	V_{tp}	V_{tn}	t_{swp}	t_{swn}
Nominal	300KΩ	30KΩ	0.7V	-1.0V	1μs	1μs

Clock frequency, load resistance, read voltage and write voltage are the circuit level parameters considered for the XORed XbarPUF circuit. Analyses for determining clock frequency and load resistance are presented in the previous section. The read and

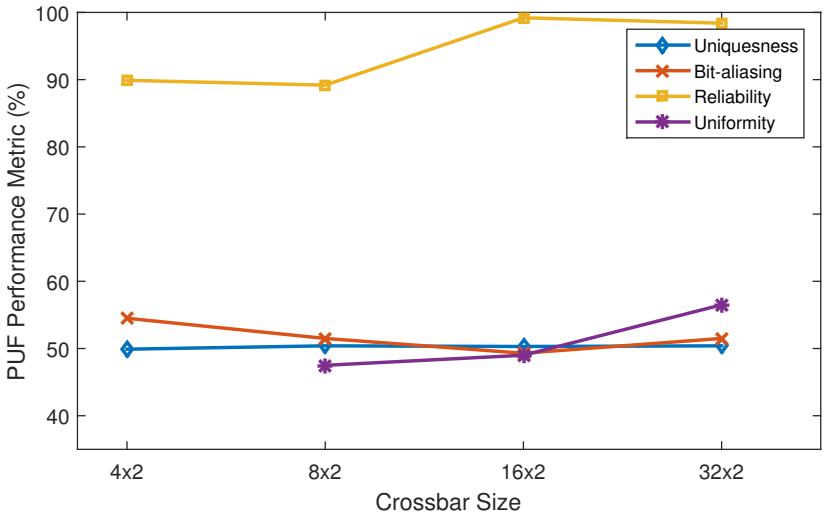

Fig. 12. Security performance of XORed XbarPUF for different crossbar size.

write voltages are also important design parameters. These voltages should be kept as close to minimum as possible to reduce power consumption of the overall circuit. However, write voltage must also be greater than magnitudes of threshold voltages to ensure proper memristor switching. On the other hand, the read voltage must be less than the threshold voltage magnitudes but must be large enough to provide good detection between high and low load outputs from the perspective of buffers and other read circuitry. The voltage drop across the multiplexers and other logic elements also provide a lower limit to these voltages. By iterating with different values and thus determining the performance of the circuit, we have chosen the read and write voltages to be 0.6V and 1.3V, respectively.

To evaluate the uniqueness and bit-aliasing metrics of the XbarPUF circuit, 100 runs of Monte Carlo simulation were performed using Cadence Spectre for a few different challenges. To evaluate uniformity, 100 unique challenges were applied over several different chips. To measure reliability, multiple challenges were applied across ten different chips for 100 cycles per chip. Each challenge set is unique and chosen randomly. To evaluate power consumption, we have used several different LRS and HRS values and with different ratios. Power is measured separately in read and write/challenge stages and then averaged over time to produce the average power consumption of the circuit.

For all these simulations, the base LRS is kept at 30K. As mentioned in subsection 2.3, we have used two sets of statistical cycle-to-cycle variations for these simulations. The temperature was varied in small steps from 10^0C to 100^oC in increasing or decreasing fashion. The simulation results are presented and analyzed in the following section.

6. RESULTS, ANALYSES AND DISCUSSION

At first, we present the efficiency of our XORed XbarPUF in terms of PUF performance metrics for four different crossbar sizes. More specifically, Fig. 12 shows the result for uniqueness, bit-aliasing, uniformity and reliability. From these values, it is evident that our memristor XbarPUF design works well as a strong PUF. Uniqueness almost remains at its ideal value of 50% for all crossbar sizes. Uniformity and bit-aliasing

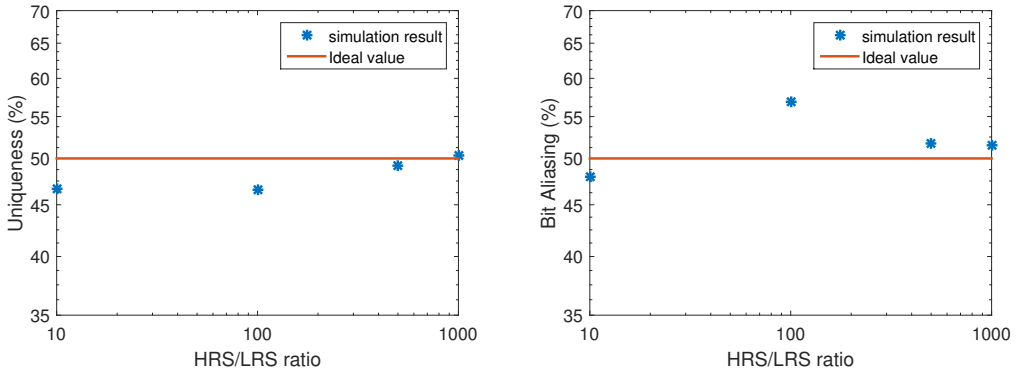

Fig. 13. Dependence of uniqueness (left) and bit-aliasing (right) with HRS/LRS ratio and the absolute values of HRS and LRS. The base LRS was 30KΩ for both of these plot.

Table III. Corner analysis of a 32×2 XORed XbarPUF for 2% variation of load resistance (R_{ld})

Metric	Min. R_{ld}	Nominal R_{ld}	Max. R_{ld}
Uniqueness (%)	50.17	50.40	50.12
Bit-aliasing (%)	51.00	51.50	50.25
Uniformity (%)	53.00	56.50	47.50

are also close to their ideal value of 50%. Reliability is over 90% and close to the ideal of 100% with increasing size of crossbar. This may seem counter-intuitive at first because with an increasing number of rows, the equivalent memristance drops further and thus should decrease noise margin. However, for a 32×2 XORed XbarPUF, the load resistance is found to be 2kΩ using equation 12. This is not too low so that wire resistance and other peripheral resistance would influence and worsen noise margin. However, it can be inferred that with an increasing number of rows the effect of cycle-to-cycle variation for each individual memristor to the overall equivalent memristance of the crossbar is reduced. Therefore, during different cycles, the probability of getting different responses for the same challenge is reduced and thus reliability is improved. However, at some point this would not be the case for very large size crossbar sizes since the equivalent memristance would be comparable with any changes caused by environmental variation or noise. Having high HRS and LRS values would help reduce this effect and improve scalability, which is shown later in this section.

We have also performed a small corner analysis by including CMOS variation. Since the variation of peripheral circuitry would only affect slightly the read/write voltage and thus does not change the overall performance of the circuit we have used higher than minimum read/write voltages to obfuscate any such changes with 2% variation for load resistance. The security metrics either do not change or show only minuscule change and, therefore, the results is not explicitly provided here. Uniformity is also relatively constant across different load resistances as it is a property of challenge. The other two metrics, uniqueness and bit-aliasing also remain effectively constant for up to 32×2 XbarPUF, which is shown in Table III.

Uniqueness and bit-aliasing with respect to different HRS/LRS (OFF/ON) ratios are shown in Fig. 13. The LRS was kept at 30KΩ while HRS was varied to obtain all the data in these two plots. From Fig. 13, we can see an improvement in both uniqueness and bit-aliasing (moves toward 50%) with increasing HRS/LRS ratio, which is expected since an increase in HRS/LRS ratio provides a better noise margin. It is also evident that even at a smaller HRS/LRS ratio (leftmost point on the plots in Fig. 13) and with

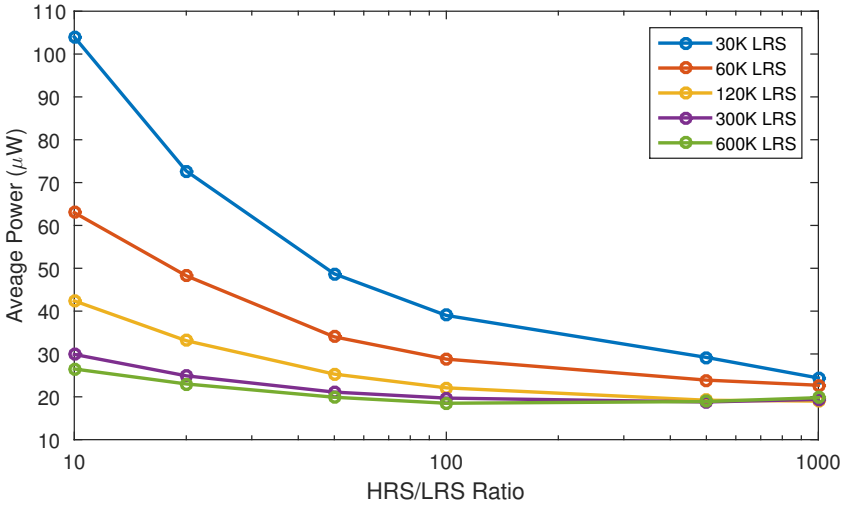

Fig. 14. Dependence of average power with HRS/LRS ratio and the absolute values of HRS and LRS.

Table IV. Performance of the XORed XbarPUF with variable temperature (10^oC- 100^oC) and with different cycle-to-cycle variation for the memristor parameters

Circuit	Reliability (%) (10% variation)	Reliability (%) (2% variation)
XORed XbarPUF (fixed/room temperature)	80	90
XORed XbarPUF (variable temperature)	80	90

10% cycle-to-cycle variation, good values for both uniqueness (≈47%) and bit-aliasing (≈48%) can be observed. This is expected from any PUF consisting of nano-materials such as memristors. Since memristors display much higher process variation compared to CMOS devices, it is statistically intuitive that memristor based PUFs would also exhibit near ideal uniqueness, uniformity and bit-aliasing in hardware implementation.

Change in power consumption for the circuit with respect to LRS and HRS values are shown in Fig. 14. We have simulated and evaluated power for six different HRS/LRS ratios, each with five different sets of HRS and LRS values. The leftmost and topmost point in Fig. 14 represents the power consumption with the current HRS and LRS values considered achievable for HfO_x memristors. Therefore, it is evident that there is room for further reductions in power consumption by improving device characteristics, something we expect to occur as the technology matures. Both an increase in HRS/LRS ratio and any increase in LRS (and HRS to keep the ratio same) result in a reduction in power. However, larger reductions are found at HRS/LRS ratios less than 100, beyond which the power curve tends to flatten out. With increasing LRS, the advantage in power reduction with increasing HRS/LRS ratios also reduces.

Because of the high cycle-to-cycle variation present in memristors and other nano-materials, they tend to show slightly different behavior across different cycles. Thus, the stability or reliability of the circuit is hampered with increasing cycle-to-cycle variation. Temperature also changes the memristor parameters and thus can potentially affect reliability. The reliability results for our XORed XbarPUF with and without temperature variation along with two sets of variation are presented in Table IV. As

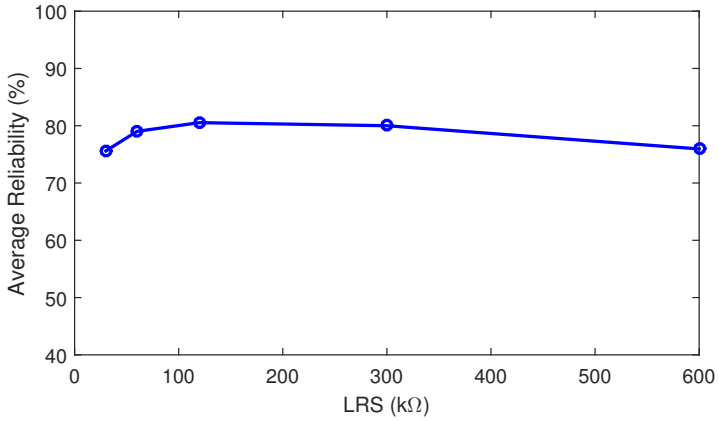

Fig. 15. Reliability of the XbarPUF with respect to increasing LRS.

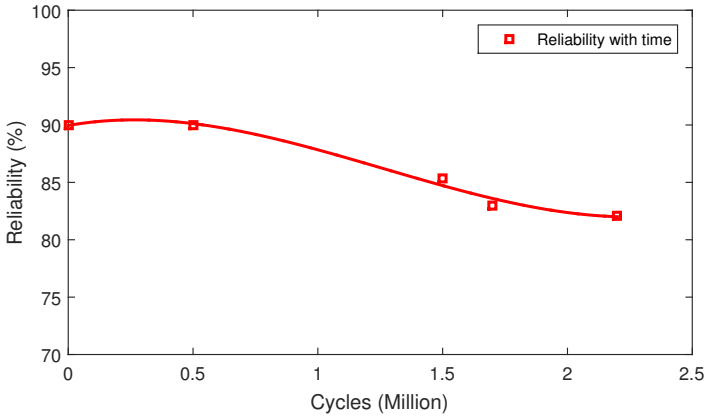

Fig. 16. Reliability of the XbarPUF with age.

expected, reliability is improved, from 80% to 90% for a 4×2 XbarPUF, if the variation decreases from 10% to 2%. We have not found any change in reliability with changing temperature. However, it is expected to have some adverse effect of changing temperature in any circuit. But since in XbarPUF, instead of measuring absolute voltage/current, the relative differences between adjacent columns voltages of the XbarPUF are measured and, therefore, any environmental change should affect both the physically nearby columns almost equally, thereby preserving the original relative differences.

The main reason for including column shuffling in the XbarPUF architecture is to improve resilience against machine learning attacks. We have evaluated the prediction accuracy of each variations of XbarPUF against well-known machine learning algorithms. Another work expands on this [Uddin et al. 2017]. Based on that work, we have found the modeling accuracy for a column swapping XbarPUF to be $\approx 65\%$, less than with XORed XbarPUF where accuracy is roughly 77% or for the intrinsic XbarPUF that can be modeled with an accuracy of 99% [Uddin et al. 2017]. This re-

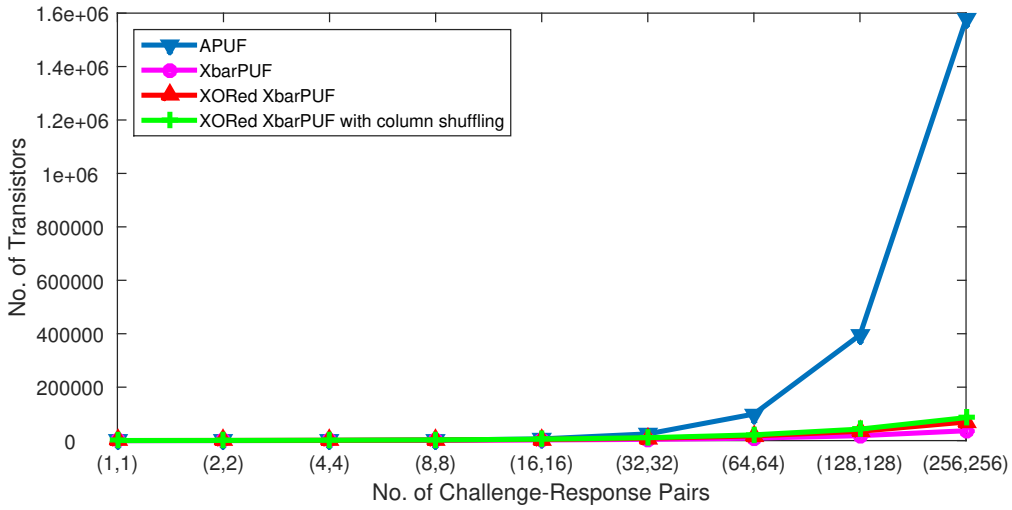

Fig. 17. Area comparison of XbarPUF with Arbiter PUF.

sult suggests that both XORing and column swapping provide increased robustness against modeling attacks over the intrinsic XbarPUF.

Reliability is also presented for different LRS (and HRS) values in Fig. 15. There is little dependence on reliability with LRS values. However, if the LRS is too low, then the equivalent memristance of the crossbar would be very low, thus decreasing noise margin and reliability. Therefore, on the left side of Fig. 15, we see that reliability tends to decrease with decreasing LRS.

We have also measured reliability over 100 cycles for the XbarPUF consisting of fresh memristors and memristors aged over many set-reset cycles. With aging, the HRS/LRS ratio drops as shown in Fig. 4. Therefore, the reliability of the XbarPUF also decreases as illustrated in Fig. 16, as is expected. To tackle this situation, memristors with much higher HRS/LRS ratio should be used so that even after a few million cycles, a good separation between HRS and LRS values exists to differentiate. Besides, there are also memristors where HRS and LRS do not change much with time and thus their relative difference is still intact after many cycles.

In addition, there are no floating rows as the read voltage is applied across all rows simultaneously and each column is measured using a load resistance, unlike in a crossbar memory where a single memory element needs to be read. Therefore, there is no negative impact of sneak-path current [Cassuto et al. 2013] in our crossbar circuit implementation as there are no unselected cells. For these reasons, we expect the reliability of the fabricated XORed XbarPUF also to be very close to its desired value if the variation is improved.

Memristors are very small compared to CMOS transistors and can be placed at each cross-points of a crossbar array. Therefore, the crossbar representation requires minimal area from the perspective of CMOS. Since the XORed circuit generates one response bit from four columns instead of two as was the case in XbarPUF without XORing, its area is double in size compared to an XbarPUF with no XORing. Only a few transistors are added to the whole circuit if column shuffling is used. Fig. 17 shows the area comparison between XbarPUF and CMOS-based arbiter PUF (APUF) in terms of transistor count. The area of the memristive XbarPUF increases linearly with an

increase in rows and/or columns while in the case of the APUF, the increase in area is exponential.

The delay of this circuit is the minimum clock cycle time that can be used to sample the response bits faithfully. This minimum clock cycle time is constrained by the longest time needed for a memristor to switch to either the high or low resistance state. In our model the switching time is 1μs. There is also a time needed to apply a challenge to the crossbar PUF and then get a valid response. From the simulation, we find this delay to be $25.2n$s which includes the sampling time using another clock. Thus, the delay of our XbarPUF circuit would be a function of the switching times of memristors and the time required to apply a challenge and get a valid response. It is worth noting that other memristor material stacks (such as [Yang et al. 2010]) have exhibited faster switching times so the delay of the XbarPUF is expected to improve as technologies mature.

7. FUTURE WORK

Crossbar size considered in our work are not huge in practice but even simulating these circuits to generate one data point using Monte Carlo analysis take several hours. Therefore, it is essential to build a behavioral model of this circuit to perform large-scale simulations in a short time. Right now, we are working on building a complete behavioral model of the whole system. Thus, we will be able to simulate and analyze big PUF circuits with much larger challenge and response set. Since XbarPUF architecture has similarity to APUF (memristor with switch-box), the XbarPUF is also found to be vulnerable to machine learning attacks as suspected in [Gao et al. 2016]. Any information leakage via side channel has not been considered too. It will be easier to generate bigger sets of data using a behavioral model of XbarPUF to perform a complete machine learning based modeling attacks and evaluate robustness of the XbarPUF against these attacks and compare with other PUFs [Ruhrmair et al. 2013]. Besides, the scalability of this XbarPUF is limited by the ability of sensing circuitries and memristor OFF/ON ratio. The accuracy of arbiter and other peripheral circuitries are also crucial for the XbarPUF to operate reliably. We, therefore, are currently working on exploring other options to improve these circuit level aspects. Also, we have no understanding of significant statistical correlation among neighboring memristors. We need to work with our device engineers more with this matter. We can also build a noise model for memristors and the crossbar circuit and work on including temperature dependence in memristor model for a much larger range. Finally, testing every circuit components of the XbarPUF to implement successfully in a real hardware is the next step for us. These are something we hope to fulfill in near future.

8. CONCLUSION

Memristors have very large device-to-device variations which result in a high uniqueness, uniformity and bit-aliasing of the XbarPUF. Memristors also have larger cycle-to-cycle variations, which along with variable temperature and noise could decrease the reliability. Nonetheless, these security performance of our XbarPUF is better than or comparable to other existing PUFs [Machida et al. 2014; Garg and Kim 2014; Sahoo et al. 2013]. Although our work is based on simulation and, therefore, our next big step would be to evaluate all these metrics from a fabricated XbarPUF. One clear advantage of memristive XbarPUFs over other PUFs is that this XbarPUFs require a very small amount of area which makes them very suitable for security measures employed in hand-held devices and other small embedded system. The delay is lower but the power consumption of the XbarPUF using the current memristor data is not as low as we hoped for. As mentioned in section 6, there is a good possibility of improvement if the device parameters evolve.

Thus, the power consumption is comparatively higher for the HfO_x memristor based XbarPUF. The reliability is not 100%. Therefore, this PUF may not be ready to be used in cryptographic key generation applications yet but could be very suitable for authentication purposes. The scalability of the crossbar design is also limited by the number of crossbar rows and in turn, number of challenges. If the memristor device technology matures and memristors have smaller cycle-to-cycle variation and can provide the HRS, LRS and other parameters that we expect, it will result in a much stable XbarPUF with improved security performance as well as less overhead in terms of power consumption. Employing some error-correction schemes would improve reliability further, although could result in a larger area and delay overhead.

Acknowledgment

The authors would like to thank Lok Kwong-Yan of the Air Force Research Laboratory and Sagarvarma Sayyaparaju of our research group for interesting discussions on this topic.

REFERENCES

H. Abunahla, B. Mohammad, and D Homouz. 2014. Effect of device, size, activation energy, temperature, and frequency on memristor switching time. In *26th Int. Conf. on Microelectronics (ICM)*. 60–63.

K. Beckmann, J. S. Holt, N. C. Cady, and J. Van Nostrand. 2015. Comparison of random telegraph noise, endurance and reliability in amorphous and crystalline hafnia-based ReRAM. In *2015 IEEE International Integrated Reliability Workshop (IIRW)*. 107–110. DOI:http://dx.doi.org/10.1109/IIRW.2015.7437079

Karsten Beckmann, Harika Manem, and Nathaniel Cady. 2016. Performance enhancement of a Time-Delay PUF Design by Utilizing Integrated Nanoscale ReRAM Devices. *IEEE Transactions on Emerging Topics in Computing* (2016), 1. DOI:http://dx.doi.org/10.1109/TETC.2016.2575448

A. Benoist, S. Blonkowski, S. Jeannot, S. Denorme, J. Damiens, J. Berger, P. Candelier, E. Vianello, H. Grampeix, J. F. Nodin, E. Jalaguier, L. Perniola, and B. Allard. 2014. 28nm advanced CMOS resistive RAM solution as embedded non-volatile memory. In *2014 IEEE International Reliability Physics Symposium*. 2E.6.1–2E.6.5. DOI:http://dx.doi.org/10.1109/IRPS.2014.6860604

Y. Cassuto, S. Kvatinsky, and E. Yaakobi. 2013. Sneak-path Constraints in Memristor Crossbar Arrays. In *IEEE Int. Symp. on Inform. Theory Proc. (ISIT)*. 156–160. DOI:http://dx.doi.org/10.1109/ISIT.2013.6620207

An Chen. 2015. Utilizing the Variability of Resistive Random Access Memory to Implement Reconfigurable Physical Unclonable Functions. *IEEE Electron Device Lett.* 36 (February 2015), 138–140. Issue 2. DOI:http://dx.doi.org/10.1109/LED.2014.2385870

B. Chen, Y. Lu, B. Gao, Y. H. Fu, F. F. Zhang, P. Huang, Y. S. Chen, L. F. Liu, X. Y. Liu, J. F. Kang, Y. Y. Wang, Z. Fang, H. Y. Yu, X. Li, X. P. Wang, N. Singh, G. Q. Lo, and D. L. Kwong. 2011. Physical mechanisms of endurance degradation in TMO-RRAM. In *2011 International Electron Devices Meeting*. 12.3.1–12.3.4. DOI:http://dx.doi.org/10.1109/IEDM.2011.6131539

P. Y. Chen, , Tempe, R. Fang, R. Liu, C. Chakrabarti, Y. Cao, and S. Yu. 2015. Exploiting resistive cross-point array for compact design of physical unclonable function. In *IEEE Int. Symp. on Hardware Oriented Security and Trust (HOST)*. 26–31. DOI:http://dx.doi.org/10.1109/HST.2015.7140231

L. O. Chua. 1971. Memristor-The missing circuit element. *IEEE Trans. on Circuit Theory* 18, 5 (September 1971), 507–519.

Z. Fang, H. Y. Yu, W. J. Liu, Z. R. Wang, X. A. Tran, B. Gao, and J. F. Kang. 2010. Temperature Instability of Resistive Switching on HfO_2 -Based RRAM Devices. *IEEE Electron Device Lett.* 31 (May 2010), 476–478. Issue 5. DOI:http://dx.doi.org/10.1109/LED.2010.2041893

Yansong Gao, Damith C Ranasinghe, Said F Al-Sarawi, Omid Kavehei, and Derek Abbott. 2016. Emerging physical unclonable functions with nanotechnology. *IEEE Access* 4 (2016), 61–80.

A. Garg and T.T. Kim. 2014. Design of SRAM PUF with Improved Uniformity and Reliability Utilizing Device Aging Effect. In *IEEE Int. Symp. on Circuits and Syst. (ISCAS)*. 1941–1944.

J. Guajardo, G.-J. Kumar, S. Schrijen, and P. Tuyls. 2007. Physical unclonable functions and public-key crypto for FPGA IP protection. In *Proc. of the IEEE Int. Conf. on Field Programmable Logic and Applicat.* 189–195.

C. Herder, Meng-Day Yu, F. Koushanfar, and S. Devadas. 2014. Physical Unclonable Functions and Applications: A Tutorial. 102 (May 2014), 1126–1141. Issue 8.

Omid Kavehei, Chun Hosung, Damith Ranasinghe, and Stan Skafidas. 2013. mrPUF: A Memristive Device based Physical Unclonable Function. *ArXiv e-prints* (Feb. 2013).

Patrick Koeberl, Unal Kocabas, and Ahmad-Reza Sadeghi. 2013. Memristor PUFs: A new generation of memory-based Physically Unclonable Functions. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 428–431.

R. Liu, H. Wu, Y. Pang, H. Qian, and S. Yu. 2015. Experimental Characterization of Physical Unclonable Function Based on 1 kb Resistive Random Access Memory Arrays. 36 (October 2015), 1380–1383. Issue 12. DOI: http://dx.doi.org/10.1109/LED.2015.2496257

Paolo Lugli, Ahmed Mahmoud, Gyrgy Csaba, Michael Algasinger, Martin Stutzmann, and Ulrich Ruhrmair. 2013. Physical unclonable functions based on crossbar arrays for cryptographic applications. *Int. J. Circ. Theor. Appl.* 41 (June 2013), 619–633. Issue 6. DOI: http://dx.doi.org/10.1002/cta.1825

T. Machida, Japan Chofu, D. Yamamoto, M. Iwamoto, and K. Sakiyama. 2014. A new mode of operation for arbiter PUF to improve uniqueness on FPGA. In *Federated Conf. on Comp. Science and Inform. Syst. (FedCSIS)*. 871–878.

Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. 2013. A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions. In *Embedded Systems Design with FPGAs*, Peter Athanas, Dionisios Pnevmatikatos, and Nicolas Sklavos (Eds.). Springer New York, 245–267. DOI: http://dx.doi.org/10.1007/978-1-4614-1362-2_11

Harika Manem and Garrett S. Rose. 2011. A Read-Monitored Write Circuit for 1T1M Memristor Memories. In *Procs. of IEEE Int. Symp. on Circuits and Syst.* 2938–2941.

A. Mazady, H. Manem, M.T. Rahman, D. Forte, and M. Anwar. 2015. Memristor PUF - A Security Primitive: Theory and Experiment. *IEEE J. on Emerging and Selected Topics in Circuits and Syst.* 5, 8 (June 2015), 222–229.

N. R. McDonald, S. M. Bishop, B. D. Briggs, J. E. Van Nostrand, and N. C. Cady. 2012. Influence of the plasma oxidation power on the switching properties of Al/Cu_xO/Cu memristive devices. *Solid-State Electronics* 78 (December 2012), 46–50.

Z Paral and Srinivas Devadas. 2011. Reliable and efficient PUF-based key generation using pattern matching. In *IEEE Int. Symp. on Hardware-Oriented Security and Trust (HOST)*. IEEE, 128–133.

P. Pouyan, E. Amat, and A. Rubio. 2014. Reliability challenges in design of memristive memories. In *5th European Workshop on CMOS Variability (VARI)*. 1–6. DOI: http://dx.doi.org/10.1109/VARI.2014.6957074

G.S Rose and C.A. Meade. 2015. Performance analysis of a memristive crossbar PUF design. In *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6.

G. S. Rose, N. McDonald, L.-K. Yan, and B Wysocki. 2013a. A Write-Time Based Memristive PUF for Hardware Security Applications. In *Proc. of the IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*. 830–833.

G. S. Rose, N. McDonald, L.-K. Yan, B. Wysocki, and K. Xu. 2013b. Foundations of Memristor Based PUF Architectures. In *Proc. of the IEEE/ACM Int. Symp. on Nanoscale Architectures (NANOARCH)*. 52–57.

U Ruhrmair, J. Solter, F. Sehnke, and Xiaolin Xu. 2013. PUF Modeling Attacks on Simulated and Silicon Data. *IEEE Trans. on Inform. Forensics and Security* 8 (August 2013), 1876–1891. Issue 11.

D.P. Sahoo, D. Mukhopadhyay, and R.S. Chakraborty. 2013. Design of low area-overhead ring oscillator PUF with large challenge space. In *Int. Conf. on Reconfigurable Computing and FPGAs (ReConFig)*. 1–6.

Dmitri B. Strukov, Gregory S. Snider, Duncan R. Stewart, and R. Stanley Williams. 2008. The missing memristor found. *Nature* 453 (May 2008), 80–83.

G.E. Suh and S. Devadas. 2007. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *44th ACM/EDAC/IEEE Design Automation Conference (DAC)*. 9–14.

G. Edward Suh, Charles W. O'Donnell, Ishan Sachdev, and Srinivas Devadas. 2005. Design and Implementation of the AEGIS Single-Chip Secure Processor Using Physical Random Functions. In *Proc. of the 32nd Annual Int. Symp. on Comput. Architecture*. 25–36.

V. Jousseaume et. al. T. Cabout, L. Perniola. 2013. Temperature impact (up to 200C) on performance and reliability of HfO_2-based RRAMs. In *5th IEEE International Memory Workshop*. 116–119. DOI: http://dx.doi.org/10.1109/IMW.2013.6582112

M. Uddin, M. B. Majumder, and G. S. Rose. 2017. Robustness Analysis of a Memristive Crossbar PUF Against Modeling Attacks. *IEEE Transactions on Nanotechnology* PP, 99 (2017), 1–1. DOI: http://dx.doi.org/10.1109/TNANO.2017.2677882

Mesbah Uddin, Md. Badruddoja Majumder, Garrett S. Rose, Karsten Beckmann, Harika Manem, Zahiruddin Alamgir, and Nathaniel C. Cady. 2016. Techniques for Improved Reliability in Memristive Crossbar PUF Circuits. In *IEEE Comp. Society Annual Symp. on VLSI*.

Christian Walczyk, Damian Walczyk, Thomas Schroeder, Thomas Bertaud, Magorzata Sowinska, Mindaugas Lukosius, Mirko Fraschke, Dirk Wolansky, Bernd Tillack, Enrique Miranda, and Christian Wenger. 2011. Impact of Temperature on the Resistive Switching Behavior of Embedded HfO_2 - Based RRAM Devices. *IEEE Trans. on Electron Devices* 58 (September 2011), 3124–3131. Issue 9. DOI:http://dx.doi.org/10.1109/TED.2011.2160265

J Joshua Yang, MX Zhang, John Paul Strachan, Feng Miao, Matthew D Pickett, Ronald D Kelley, G Medeiros-Ribeiro, and R Stanley Williams. 2010. High switching endurance in TaOx memristive devices. *Applied Physics Letters* 97, 23 (2010), 232102.

Le Zhang, Xuanyao Fong, Chip-Hong Chang, Zhi Hui Kong, and Kaushik Roy. 2014. Highly reliable memory-based physical unclonable function using spin-transfer torque MRAM. In *IEEE Int. Symp. on Circuits and Syst. (ISCAS)*. IEEE, 2169–2172.