A Designer’s Rationale for Nanoelectronic Hardware Security Primitives

Garrett S. Rose, Mesbah Uddin, and Md. Badruddoja Majumder

Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Pittsburgh, PA, July 2016.

Citation Information (BibTex):

```
@INPROCEEDINGS{ISVLSI:Rose2016,
  author="G. S. Rose and M. Uddin and M. B. Majumder",
  title="A Designer’s Rationale for Nanoelectronic Hardware
      Security Primitives",
  booktitle="{IEEE} Computer Society Annual Symposium on
      {VLSI} ({ISVLSI})",
  month="July",
  year="2016",
  pages="194-199",
  address="Pittsburgh, PA"
}
```

A Designer's Rationale for Nanoelectronic Hardware Security Primitives

Garrett S. Rose, Mesbah Uddin, and Md. Badruddoja Majumder
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
Knoxville, Tennessee 37996 USA
{garose, muddin6, mmajumde}@utk.edu

***Abstract*—A variety of hardware security primitives have been developed in recent years, aimed at mitigating issues such as integrated circuit (IC) piracy, counterfeiting, and side-channel analysis. For example, a popular security primitive for mitigating such hardware security vulnerabilities is the physical unclonable function (PUF) which provides hardware specific unique identification based on intrinsic process variations in individual integrated circuit implementations. At the same time, as technology scaling progresses further into the nanometer region, emerging nanoelectronic technologies are becoming viable options for many next-generation computing technologies. At the intersection between nanoelectronics and security, several examples of nano-enabled security primitives have been proposed in the last few years. In this paper, we consider a few examples of nanoelectronic security in the context of how such nanoscale technologies impact power, area and delay as compared to conventional CMOS-based approaches. Our analyses show that leveraging novel nanoelectronic technologies not only provide area benefits but also energy-efficient solutions that enable security with a small footprint.**

***Keywords*-Nanotechnology; memristor; computer security; physical unclonable function; encryption; chaos computing; obfuscation;**

I. Introduction

Use of nanoelectronic circuits for building hardware security primitives has become a topic of interest to the researchers . Memristor, a basic nanoelectronic circuit element has been proposed for using in a number of hardware security applications. Physical unclonable function (PUF) is one of the security primitives where memristors have emerged as a promising candidate. A 1-bit memristive PUF was proposed by Rose *et al.* where the write time of a memristor is used as the source of entropy [1]. Mazady *et al.* presented an experimental demonstration of 1-bit memristive PUF [2]. Rose *et al.* extended the concept of 1-bit PUF to a new one (XbarPUF) based on crossbar array of memristors [3].Kavehei *et al.* presented a concept of "mrPUF" using the combination of memristors and RO-PUF [4]. There are some more works focusing on different aspect of memristive PUF and memristor characteristics [5]–[8].

In addition to memristive PUFs, a variety of other nanoelectronic security primitives have also been proposed over the years. For example, Rajendran *et al.* explored the use of memristive crossbars to construct a form of nanoelectronic public PUF (NanoPPUF) [9]. A key enabling feature for some of the memristive PUFs mentioned, including the NanoPPUF, are sneak path currents in dense memristive crossbar arrays. Sneak paths are also exploited by Kannan *et al.* in [10] where a lightweight encryption engine is presented based on memristive crossbars. In this paper, we explore these various nanoelectronic security primitives, highlighting the nanodevice features exploited for security applications. Additionally, we describe the concept of a new nanoelectronic security primitive that leverages sneak path currents for integrity checking in memristive memories.

Nanoelectronic devices exhibit novel behaviors that can be exploited in interesting ways. In addition to specialized circuitry, nanoelectronics offers the potential to implement novel non Von Neumann computer architectures. A clear example would be memristor-based neuromorphic computing. In this paper, we also consider chaos computing, a form of computing that relies on the complexity of chaotic oscillators. Such complex systems have been explored from the perspective of security in the sense that this complexity is used to obfuscate the functionality of the computer system itself [11], [12]. Such novel architectures, even chaos computing, are now becoming realizable due to the high density and energy-efficiency of nanoelectronic technologies.

A brief overview of an example nanoelectronic device, the memristor, is provided in section II. In section III we describe a crossbar-based PUF, specifically in terms of how to leverage sneak path currents and variability in memristor parameters. Section IV provides a greater discussion of the nanoelectronic crossbar and how this structure is leveraged in the implementation of a variety of nanoelectronic security primitives. Novel nanoelectronic computer architectures and their use for improved computer security are discussed in section V. Final thoughts and conclusions regarding some of the features of nanoelectonic memristors and their use for security are provided in section VI.

II. The Memristor: A Nanoelectronic Device

CMOS is the basic building block of most integrated circuits (ICs) and has been shrinking regularly following Moore's Law from a few micrometers in the 1960s to tens of nanometers today. But this trend is under threat now. In order to continue the miniaturization of circuit elements,

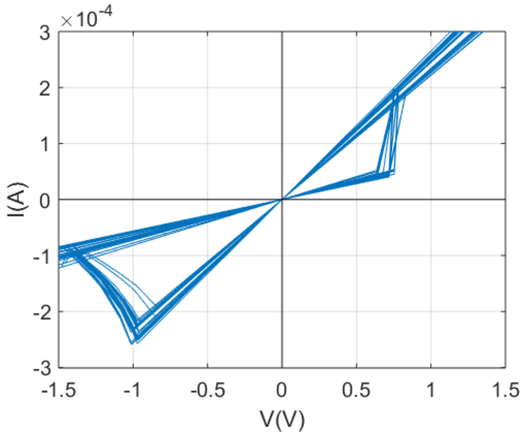

Fig. 1. Simulated I-V characteristics of a HfO_2 memristor.

researchers are investigating several alternatives to build ultra-dense circuitry. One of these nanoelectronic alternatives is the memristor. The term ‘memristor’ (memory resistor) was first coined by Chua in 1971 [13] and was later demonstrated by HP Labs in 2008 [14]. The memristor is considered as the fourth fundamental circuit element after resistor, capacitor and inductor, that relates the charge (q) with flux linkage (ϕ). A memristor is a two-terminal device and its memristance (or “memristor resistance”) is variable and depends on the “past history” of applied voltage. In most applications, memristors are modeled as two-level (or multi-level) resistors where the resistance is changed when the magnitude and direction of the applied voltage changes. Furthermore, there are also two threshold voltages which determines the state of its resistance. For a bipolar memristor, these two thresholds are positive and negative threshold voltages. When the voltage across a memristor is increasing and becomes greater than the positive threshold, the memristor is set to a low resistance state (LRS). When the voltage is decreasing and becomes smaller than the negative threshold, the memristor is reset to a high resistance state or HRS. An example I-V characteristic of a memristor is shown in Fig. 1.

One important thing to note is that memristor behavior is highly dependent on the materials used for fabrication and, therefore, one memristor can vary quite a lot from another in terms of current-voltage relationship. The first memristor fabricated at HP lab [14] has a very thin metal-insulator-metal (MIM) structure with TiO_2 being the substrate. Metal-oxide (TiO_2, HfO_2, etc.) substrate memristors display similar behavior as in Fig. 1 but other types like spintronic, magnetic and polymeric memristors have different internal physics and, therefore, their behaviors are also quite different.

Their versatile nature and CMOS fabrication compatibility of memristors motivates several potential applications in many fields. Memristors are non-volatile, which means they can retain their state even if the power supply is turned off. Because of this non-volatile behavior and the fact that the crossbar implementation of memristors can achieve ultra-high density along with memory access times comparable to DRAM, memristors are considered to be the future for very dense memory architectures. Memristor devices can also be used in analog applications like programmable logic design, chaos computing and oscillators [15]–[17] and also in digital logic applications. Memristors are also proposed to be the building blocks of the new paradigm of computing e.g. neuromorphic computing system [18], [19].

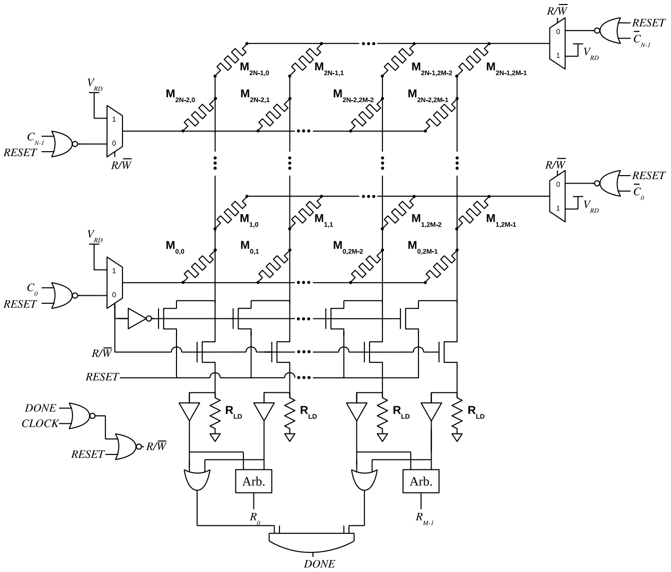

Fig. 2. Schematic of the proposed crossbar-memristive PUF design for N configuration and M response bits. The design as proposed requires a memristive crossbar of size $2N \times 2M$ including $4NM$ total memristors [3].

III. Low-Overhead, Nanoelectronic Physical Unclonable Functions

Physical unclonable functions have emerged as solutions to a variety of security concerns, including integrated circuit (IC) piracy, counterfeiting, and secret key storage [20]. A PUF is a unique hardware identifier where intrinsic process variations are used to create a “fingerprint” for a particular device. For example, it can be shown that the frequency of a CMOS ring oscillator is sensitive to variations in transistor device parameters such that the same ring oscillator PUF (RO PUF) design implemented on two different ICs will generate unique signatures for each IC due to differences in their frequencies of oscillation [21].

Using memristors as opposed to CMOS circuits is largely motivated by area and power constraints. Memristor-based designs are expected to take up less physical area and use fewer transistors than their CMOS counterparts. In order to maximize this property, in prior work we have considered a 2D crossbar array design that makes very efficient use of the available space [3]. Improved area efficiency results from the fact that the crossbar array consists of two-terminal memristive devices at the crosspoints of perpendicular nanowires.

The crossbar memristive PUF or XbarPUF (shown in Fig. 2) is based on the write-time memristive PUF (WTMPUF) described in [1]. As with the WTMPUF, the XbarPUF’s

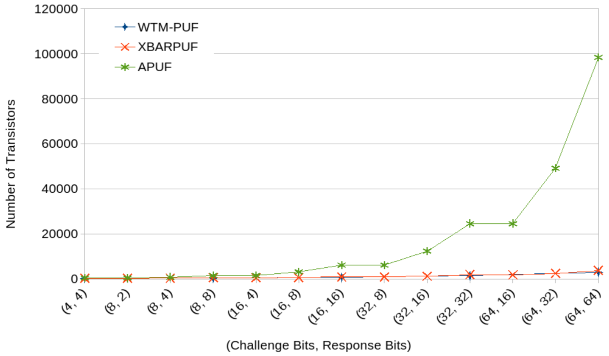

Fig. 3. Plot of area (transistor count) vs. number of challenge and response bits for three different PUF designs. CMOS APUF area increases quadratically in area while the memristive PUF designs only require a linear increase in the number of transistors [3].

primary entropy source is the minimum time it takes for memristors to SET during a write operation. Additionally, the XbarPUF is somewhat a reimagining of the arbiter PUF [20] where in place of racing delay paths, memristors are in a race to see which is SET first. Given the crossbar configuration, there are actually two corresponding rows of memristors for each challenge bit: one written based on the challenge bit and another by the inverted challenge bit. This arrangement ensures only one row of devices is actively written to while the other remains inactive (remains at HRS) during the write operation. To generate each response bit, the effective resistance values of two columns are compared. Once the resistance of one column reaches a certain threshold an arbiter selects the "winner," thus determining the corresponding response [3].

The security performance (also reported in [3]) of the XbarPUF is similar to that of the WTMPUF, while removing the dependency on an absolute write-time. Additionally, the XbarPUF has comparable performance to that of the arbiter PUF, while significantly decreasing the transistor count. For the transistor count of the APUF, we make the assumption that all response bits are determined simultaneously. A comparison of the three PUF designs (CMOS arbiter PUF, WTMPUF, XbarPUF) in terms of transistor count can be seen in Fig. 3. Due to the constraint of single-cycle response generation, the transistor count for the arbiter PUF is quadratic with the number of challenge-response bits while that of the two memristive PUF designs is linear. This is an expected result as the entropy source for the memristive PUFs is the memristor itself which isn't being counted toward the area even though the XbarPUF does include a quadratic number of memristors as a function of number of challenge-response bits.

IV. Crossbars and Security

The memristive crossbar used in the construction of the XbarPUF is a common structure found in a wide range of nanoelectronic architectures. For example, nanoelectronic memories [22], dense programmable logic arrays [23]–[25], and neuromorphic computer architectures [26]. The crossbar structure is attractive for its simplicity and the ability to integrate complex logic and storage capacity in a small area. However, memristive crossbars also suffers from the issue of sneak path currents which limit their scalability. The so-called sneak path current is simply the current contribution from unselected circuit paths in a resistive (or "memristive") memory where one particular device or memory cell has been selected for a read or write operation. As the size of the crossbar grows, the contribution from the unselected circuit paths (the sneak paths) eventually outweighs that of the selected device. Sneak paths can be mitigated through a variety of techniques, including the inclusion of transistors at each memory cell, leading to a 1-transistor, 1-memristor (1T1M) structure [23].

For security purposes, we consider the sneak path currents as means to providing useful information. This is actually evident in the XbarPUF where multiple memristive memory cells are selected simultaneously, leading to aggregate behavior along the length of the columns. A variety of other nanoelectronic security primitives have also emerged that leverage sneak path currents in crossbar arrays.

A. The NanoPPUF

The Nanoelectronic Public PUF (NanoPPUF) was proposed by Rajendran *et al.* [27] as an authentication method that depends on the long duration of time required to simulate a large memristive crossbar system. This is an extension of prior research exploring the concept of a Public PUF (PPUF) in a more general sense [28]. For any PPUF, an accurate model for the system is made available to the public. However, since simulation takes a very long time relative to the time taken to measure the physical PPUF, the physical system is easy to distinguish and thus authenticate.

The NanoPPUF is based on a large memristive crossbar that can be configured by (1) setting the states of the various memristive devices and (2) selecting a subset of the system for measurement. As with any other PPUF, setting up and running the simulation takes a much longer time than simply measuring the physical NanoPPUF. Thus, the NanoPPUF provides similar security features found in any PPUF. However, since the NanoPPUF is constructed from nanoelectronic devices (memristors), it consumes lower power and is much smaller than CMOS counterparts.

B. Sneak Path Encryption

As a more direct use of sneak path currents, Kannan *et al.* [10], [29] presented a technique that directly utilizes the sneak paths in memristive memories as a method of encryption. Again, the major attraction to such systems is the ability to implement truly lightweight and power-efficient encryption by leveraging the high density nature of the crossbar.

C. Integrity Checking in Memristive Memory

Another concept we've begun exploring is the use of sneak path currents in dense memristive memory arrays to efficiently generate reliable tags for integrity checking. For this technique we depend on the sneak path current being unique for any

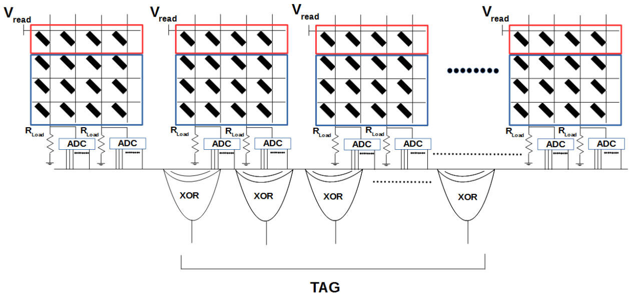

Fig. 4. System realization for authenticating large number of crossbar blocks with limited number of tag bits.

memory state stored into the memory. Put another way, we find that even the update of a small number of memristive memory cells leads to a measurable change in the resulting sneak path currents. Since the sneak path current is indirectly related to the data stored in memory, we treat it as being analogous to a hash typically used for integrity checking. However, since the tags are generated from the sneak paths already present in the crossbar, the sneak path based integrity checking approach requires significantly less overhead as compared to any CMOS-based technique.

Fig. 4 shows a schematic for a potential sneak path integrity checking technique. Here, tags are generated from the sneak paths of individual banks of memristive memory, each implemented as crossbars. The tags from the various banks are then combined via an XOR-based circuit to generate a tag for the overall memory. It should also be noted that for this technique 1T1M memories are used to selectively turn on the sneak path currents. For a normal read or write operation, the transistors in the 1T1M memory cells would be used to eliminate the sneak paths. However, multiple 1T1M cells across multiple rows would be activated during tag generation in order to allow the sneak paths to flow through the entire memory structure. In this way, sneak paths are being controlled to provide specific functionality, in this case integrity checking.

V. Nanoelectronics Begets Novel Architecture

Nanotechnology certainly provides interesting possibilities and even advantages when implementing lightweight security primitives. As mentioned, crossbar structures with memristive switches at every crosspoint can be particularly useful in the implementation of low-power and low overhead security. However, such emerging nanoelectronic structures are also being considered to implement novel computing architectures. For example, many researchers, including our group at the University of Tennessee, are actively exploring the use of memristors in the construction of neuromorphic computing systems. As such novel forms of computing emerge, it is important to consider the security implications including any potential advantages. Here, we provide some examples of such emerging nanoelectronic computer architectures and discuss from the perspective of security.

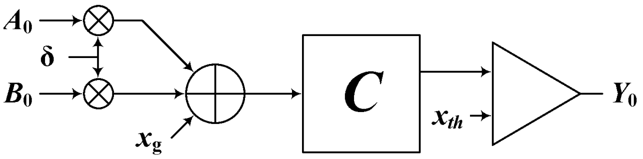

Fig. 5. Schematic of a basic chaos logic gate [11].

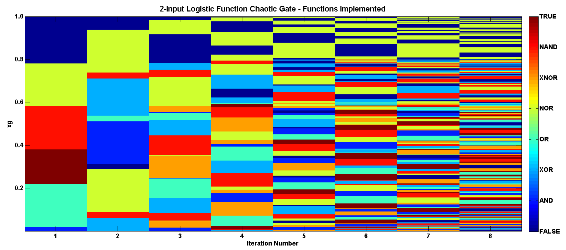

Fig. 6. State space of a 2-dimensional chaos logic gate [11].

A. Chaos Computing

Nanoelectronic technology not only provides the ability to implement small scale, energy-efficient add-ons for conventional systems but can be an enabler for novel computer architectures. One example of a novel computer architecture is based on chaos logic gates (see Fig. 5) that operate by leveraging the complexity of chaotic oscillators [11]. The basic chaos logic gate is constructed from a chaotic oscillator (e.g. Chua's circuit), initialization circuitry, and a thresholding circuit that essentially converts the output of the oscillator to a digital value. While most electronic chaotic oscillators are implemented using somewhat large and power hungry analog circuits, researchers have recently presented more efficient memristor-based chaotic circuits [15].

For security purposes chaos computing is interesting due to the inherent complexity of the system. Specifically, a given chaos logic gate, or for that matter a chaos arithmetic logic unit [11], can be configured to implement any function in a variety of different ways. Moreover, since the system is based on chaos, there is no perceptible relationship between different implementations of the same function. This is illustrated in Fig. 6 where the possible functions are plotted for the different configurations for the system. For example, the red slices in the figure represent implementations of NAND functionality. While there are several ways a NAND gate can be realized there is no periodicity or clear relationship between these various implementations.

B. Concept: Physically Unclonable Computing Systems

Building off chaos computing, we can consider more generalized physically unclonable computing systems (PUCS) that offer an inherent ability to obfuscate sensitive information. Certainly, the complexity of a chaos computing system would lend itself well to one implementation of a PUCS. Taking this a step further, the unclonable nature of a PUCS could be determined by exploiting the underlying process variations of emerging nanoelectronic technologies, specifically metal-oxide memristors. These variations lead to a variety of unique ways for implementing functional operations in the computing system such that the task of attacking the system and obtaining sensitive information becomes exceedingly difficult. Further, side-channel signatures and implementation details for any given op code set are unique across multiple PUCS implementations or chips.

Key differentiating features for the PUCS concept include: (1) uniqueness across IC implementations, (2) uniqueness across op code sets and (3) reconfigurability of op code sets. These attributes are enabling factors for obfuscation across multiple abstraction layers of computer system design. At the logic level, reconfigurability ensures that the physical design is not easily determined from functionality. Further, uniqueness requirements apply to side-channel information leakage such that reverse engineering via side-channel analysis (e.g. power analysis) is likely to be unfruitful. Finally, the very existence of multiple op code sets and the uniqueness requirement thereof are expected to lead to hardware handles for code-level obfuscation.

We mention the PUCS concept here as a sort of bridge across the concepts considered. Certainly, given the uniqueness requirements of a PUCS system it is easy to draw similarities with physical unclonable functions, such as the XbarPUF.

VI. Conclusion

Nanotechnology offers the potential for implementing security primitives that provide reasonable levels of security for minimal cost in terms of area and power. For example, physical unclonable functions constructed from memristors have been proposed and even demonstrated which offer strong security in terms of uniqueness and reliability but are also much smaller and more energy-efficient as compared to their CMOS counterparts. Other techniques such as NanoPPUF, sneak-path encryption and potentially memristor-based integrity checking offer similar overhead advantages while still providing strong security for their respective applications.

Finally, it is worth noting that novel nanoelectronic devices, such as memristors, carbon nanotubes and graphene, exhibit exotic behavior which can be exploited for novel computer architectures. Memristors, for example, are often considered for use as synaptic elements in nanoelectronic neuromorphic systems. Another emerging architecture worth considering for security purposes is chaos computing. Such novel approaches to computing, which are often well suited for nanoelectronic implementation, may provide significant benefits in terms of mitigating security threats to future computer systems. However, regardless of what benefits may be gained, the emergence of nanoelectronics and novel computer architectures necessitates security-based design analysis, especially as such technologies approach maturity.

Acknowledgement

The authors would like to thank Jeyavijayan Rajendran of the University of Texas at Dallas for useful and interesting conversations on the topics discussed.

References

[1] G. S. Rose, N. McDonald, L. Yan, and B. Wysocki, "A write-time based memristive PUF for hardware security applications," in *Proc. of the IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, November 2013, pp. 830–833.

[2] A. Mazady, M. T. Rahman, D. Forte, and M. Anwar, "Memristor PUF—a security primitive: Theory and experiment," *IEEE J. on Emerging and Selected Topics in Circuits and Syst.*, vol. 5, no. 8, pp. 222–229, June 2015.

[3] G. S. Rose and C. A. Meade, "Performance analysis of a memristive crossbar PUF design," in *Proc. of the Annual Design Automation Conf. (DAC)*, June 2015, pp. 75:1–75:6.

[4] O. Kavehei, C. Hosung, D. Ranasinghe, and S. Skafidas, "mrPUF: A Memristive Device based Physical Unclonable Function," *ArXiv e-prints*, Feb. 2013.

[5] A. Chen, "Utilizing the variability of resistive random access memory to implement reconfigurable physical unclonable functions," *IEEE Electron Device Lett.*, vol. 36, pp. 138–140, February 2015.

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

[7] P. Y. Chen, , Tempe, R. Fang, R. Liu, C. Chakrabarti, Y. Cao, and S. Yu, "Exploiting resistive cross-point array for compact design of physical unclonable function," in *IEEE Int. Symp. on Hardware Oriented Security and Trust (HOST)*, May 2015, pp. 26–31.

[8] R. Liu, H. Wu, Y. Pang, H. Qian, and S. Yu, "Experimental characterization of physical unclonable function based on 1 kb resistive random access memory arrays," *IEEE Electron Device Lett.*, vol. 36, pp. 1380–1383, October 2015.

[9] J. Rajendran, R. Karri, J. B. Wendt, M. Potkonjak, N. McDonald, G. S. Rose, and B. Wysocki, "Nanoelectronic solutions for hardware security," Cryptology ePrint Archive, Report 2012/575, 2012, http://eprint.iacr.org/.

[10] S. Kannan, N. Karimi, and O. Sinanoglu, "Secure memristor-based main memory," in *Proc. of the Annual Design Automation Conf. (DAC)*, 2014, pp. 178:1–178:6.

[11] G. S. Rose, "A chaos-based arithmetic logic unit and implications for obfuscation," in *IEEE Computer Society Annual Symposium on VLSI*, July 2014, pp. 54–58.

[12] J. Bohl, L. K. Yan, and G. S. Rose, "A two-dimensional chaotic logic gate for improved computer security," in *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, August 2015, pp. 1–4.

[13] L. O. Chua, "Memristor-the missing circuit element," *IEEE Trans. on Circuit Theory*, vol. 18, no. 5, pp. 507–519, September 1971.

[14] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008.

[15] B. Muthuswamy and P. P. Kokate, "Memristor-based chaotic circuits," *IETE Technical Review*, vol. 26, no. 6, pp. 417–429, 2009.

[16] J. Rajendran, H. Manem, R. Karri, and G. S. Rose, "Memristor based programmable threshold logic array," in *IEEE/ACM Int. Symp. on Nanoscale Architectures*, June 2010, pp. 5–10.

[17] T. Raja and S. Mourad, "Digital logic implementation in memristor-based crossbars," in *International Conference on Communications, Circuits and Systems (ICCCAS)*, July 2009, pp. 939–943.

[18] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," *Nanotechnology*, vol. 23, no. 7, p. 075201, 2012.

[19] H. Wang, H. Li, and R. E. Pino, "Memristor-based synapse design and training scheme for neuromorphic computing architecture," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–5.

[20] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. of the 9th ACM Conf. on Comput. and Commun. Security*, 2002, pp. 148–160.

[21] L. Goux, J. G. Lisoni, M. Jurczak, D. J. Wouters, L. Courtade, and C. Muller, "Coexistence of the bipolar and unipolar resistive-switching modes in NiO cells made by thermal oxidation of Ni layers," *J. of Applied Physics*, vol. 107, no. 2, January 2010.

[22] G. S. Rose, Y. Yao, J. M. Tour, A. C. Cabe, N. Gergel-Hackett, N. Majumdar, J. C. Bean, L. R. Harriott, and M. R. Stan, “Designing cmos/molecular memories while considering device parameter variations,” *ACM J. of Emerging Technologies in Computing Syst.*, vol. 3, no. 1, April 2007.

[23] H. Manem, J. Rajendran, and G. S. Rose, “Design considerations for multi-level cmos/nano memristive memory,” *ACM J. of Emerging Technologies in Computing Syst.*, vol. 8, no. 1, February 2012.

[24] H. Manem and G. S. Rose, “A read-monitored write circuit for 1t1m memristor memories,” in *Proc. of IEEE Int. Symp. on Circuits and Syst.*, May 2011, pp. 2938–2941.

[25] G. S. Rose, H. Manem, J. Rajendran, R. Karri, and R. Pino, “Leveraging memristive systems in the construction of digital logic circuits,” *Proc. of the IEEE*, vol. 100, no. 6, pp. 2033–2049, June 2012.

[26] G. S. Snider, “Spike-timing-dependent learning in memristive nanodevices,” in *IEEE Int. Symp. on Nanoscale Architectures, 2008 (NANOARCH)*, June 2008, pp. 85–92.

[27] J. Rajendran, G. S. Rose, R. Karri, and M. Potkonjak, “Nano-PPUF: A memristor-based security primitive,” in *IEEE Computer Society Annual Symposium on VLSI*, August 2012, pp. 84–87.

[28] M. Potkonjak and V. Goudar, “Public physical unclonable functions,” *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1142–1156, August 2014.

[29] S. Kannan, J. Rajendran, R. Karri, and O. Sinanoglu, “Sneak-path testing of crossbar-based nonvolatile random access memories,” *IEEE Trans. Nanotechnol.*, vol. 12, no. 3.