INVITED PAPER

Nano Meets Security: Exploring Nanoelectronic Devices for Security Applications

Among the components that can fortify computer security there is a relatively new entrant known as the memristor (contraction of memory resistor).

By Jeyavijayan Rajendran, *Student Member IEEE*, Ramesh Karri, James B. Wendt, *Student Member IEEE*, Miodrag Potkonjak, Nathan McDonald, Garrett S. Rose, and Bryant Wysocki

ABSTRACT | Information security has emerged as an important system and application metric. Classical security solutions use algorithmic mechanisms that address a small subset of emerging security requirements, often at high-energy and performance overhead. Further, emerging side-channel and physical attacks can compromise classical security solutions. Hardware security solutions overcome many of these limitations with less energy and performance overhead. Nanoelectronics-based hardware security preserves these advantages while enabling conceptually new security primitives and applications. This tutorial paper shows how one can develop hardware security primitives by exploiting the unique characteristics such as complex device and system models, bidirectional operation, and nonvolatility of emerging nanoelectronic devices. This paper then explains the security capabilities of several emerging nanoelectronic devices: memristors, resistive random-access memory, contact-resistive random-access memory, phase change memories, spin torque-transfer random-access memory, orthogonal spin transfer random access memory, graphene, carbon nanotubes, silicon nanowire field-effect transistors, and nanoelectronic mechanical switches. Further, the paper describes hardware security primitives for authentication, key generation, data encryption, device identification, digital forensics, tamper detection, and thwarting reverse engineering. Finally, the paper summarizes the outstanding challenges in using emerging nanoelectronic devices for security.

KEYWORDS | Emerging technologies; hardware security; memristors; PCMs; physical unclonable functions

I. INTRODUCTION

Since the mid–1970s, information security has evolved from primarily focusing on the confidentiality and integrity of stored and in-transit data to incorporating trust, privacy, and remote ground truthing. Over this 40-year span, the usage scenario of security technologies has evolved from securing physical premises with mainframe computers to securing lightweight, low-cost, high-performance, and low-power mobile phones, tablets, and sensors.

On one hand, classical security (i.e., mathematical or algorithmic) has created elegant security primitives and protocols. Unfortunately, these solutions are not only slow and consume significant amounts of energy for most modern security primitives but are also vulnerable to physical attacks (e.g., radiation or exposure to high temperatures). On that other hand, nanoelectronic devices[1] enable conceptually new and strong security primitives. Nanoelectronic security primitives are potentially more robust than conventional complementary metal oxide semiconductor (CMOS) device-based security primitives. Nanoelectronic devices have the potential to yield computing systems with miniscule form factors, ultra low-power consumption, and fast computation times relative to CMOS devices.

Manuscript received May 11, 2013; revised June 27, 2014; accepted November 26, 2014. Date of publication May 13, 2015; date of current version May 22, 2015.
Received and cleared for public release by AFRL on May 9, 2013, case number 88ABW-2013-2239. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or its contractors.

J. Rajendran and **R. Karri** are with New York University, New York, NY 10003 USA (e-mail: jv.ece@nyu.edu; rkarri@poly.edu).
J. B. Wendt and **M. Potkonjak** are with the University of California-Los Angeles, Los Angeles, CA 90024 USA (e-mail: jwendt@cs.ucla.edu; miodrag@cs.ucla.edu).
N. McDonald and **B. Wysocki** are with Air Force Research Labs, Rome, NY USA (e-mail: Nathan.McDonald@rl.af.mil; Bryant.Wysocki@rl.af.mil).
G. S. Rose is with the University of Tennessee, Knoxville, TN 37996 USA (e-mail: garose@utk.edu).

Digital Object Identifier: 10.1109/JPROC.2014.2387353

[1]In this paper, we refer a non-CMOS, nanoscale, emerging technology device as a nanoelectronic device.

A variety of materials and devices including memristors, spin-torque devices, phase change materials, graphene, plasmonics, and quantum dots are being investigated for use in nanoelectronics. These nanoelectronic devices have nonlinear input-output relationships and exhibit inherent process variations similar to current CMOS technologies [1]–[4], while demonstrating technology specific characteristics.

The objectives of this paper are: (i) to emphasize the security capabilities of nanoelectronic devices; and (ii) to highlight the outstanding challenges in exploiting nanoelectronic devices for security For this purpose, we explain the characteristics of different nanoelectronic devices and how they can be used to build the following security primitives: physical unclonable functions, public physical unclonable functions, random numbers, unique signatures, tamper detection circuits, and cryptographic architectures.

Section II explains different nanoelectronic devices and their properties, which can be leveraged for security. In Sections III and IV, we describe how nanoelectronic devices can be used to build physical unclonable functions and public physical unclonable functions, respectively, which can be used in several cryptographic protocols. Section V details how nanoelectronic devices can be used to generate random numbers and unique signatures. Section VI, explains one can use using nanoelectronic devices to build a tamper evident memory. In Section VII, we demonstrate how one can perform forensic analysis on a nanoelectronic device-based digital logic circuits. Section VIII describes crypto-architectures using nanoelectronic devices. Section IX concludes the paper. Overall, we expect to convey our vision of security as an important application for nanoelectronic devices.

II. NANOELECTRONIC DEVICES AND THEIR CHARACTERISTICS

In recent years, device physicists have realized a wide variety of nanoelectronic devices. We now highlight some of the devices which are identified as potential candidates for logic and memory applications by the International Technology Roadmap for Semiconductors [10].

A **memristor** consists of two metal-oxide layers sandwiched between two electrodes as shown in Fig. 1(a) [2]. One of the metal oxide layers has oxygen vacancies, and the other lacks. On applying voltage/current, the resistance of the device changes due to the shift in oxygen vacancies (see Section II-A for more details.)

A **resistive random access memory (RRAM)** consists of a metal-oxide layer sandwiched between two electrodes as shown in Fig. 1(b) [5]. On applying a sufficiently high voltage, a conduction path is formed within the metal-oxide layer, facilitating the flow of current. Consequently, the resistance of the device is decreased.

A **contact-resistive random access memory (CRRAM)** is a variation of RRAM realized by stacked TiN/TiON layers as depicted in Fig. 1(c) [11]. This structure uses the drain of a CMOS transistor as the bottom electrode, enabling a compact cell array. Consequently, one can build a high-density memory. Similar to RRAM, the switching resistor is set and reset with appropriately applied voltages.

A **phase change memory (PCM)** consists of a phase-change material and a heater sandwiched between two electrodes as shown in Fig. 1(d) [7]. On applying a large current pulse for a short duration, a region of the phase-change material changes to amorphous, thus exhibiting a high resistance. On applying a current pulse for a relatively longer duration, the amorphous region is turned into crystalline, thus decreasing the resistance.

A **spin transfer torque random access memory (STT-RAM)** consists of two magnetic layers: a free layer and a fixed layer. An insulating barrier separates these two layers. This structure is sandwiched between two electrodes as shown in Fig. 1(e) [8]. On passing a current through the device, one can change the free layer to be parallel or antiparallel with the fixed layer, thereby decreasing or increasing the resistance.

An **orthogonal spin transfer random access memory (OST-RAM)** consists of three magnetic layers: a free layer, a fixed layer, and a polarizer layer. Each of these layers is separated from its adjacent layers by a barrier. This structure is sandwiched between two electrodes as shown in Fig. 1(e) [8]. The switching mechanism, though similar to that of STT-RAM, is faster than STT-RAM due to the polarizer layer.

We use memristor as the candidate device to explain device characteristics that enable security primitives.

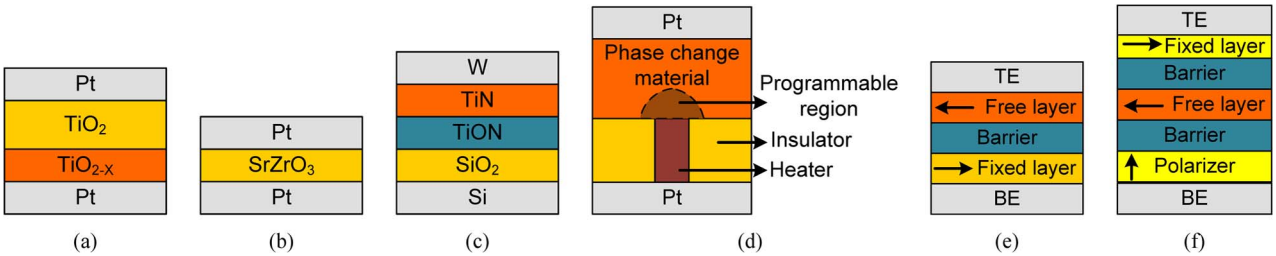


Fig. 1. *Device structure of candidate nanoelectronic devices: (a) Memristor [2]. (b) Resistive RAM [5]. (c) Contact resistive RAM (CRRAM) [6]. (d) Phase change memory (PCM) [7]. (e) Spin-transfer torque RAM (STT-RAM) [8]. (f) Orthogonal spin-transfer torque RAM (OST-RAM) [9].*

A. Memristors

1) Theory: Memory resistance or memristance $M(q)$ relates charge q and flux ϕ, such that the memristance of the device changes with the applied electric field and time [1]:

$$M(q) = \frac{d\phi(q)}{dq} \tag{1}$$

where, $M(q)$ is the memristance of a memristor, measured in ohms. Memristance at any time instance depends on the integrals of the voltage (current) across (through) the device from $-\infty$ to that time. Thus, the memristor behaves like an ordinary resistor at any given instance of time, while its memristance depends on the history of the device [1], [2].

2) Device Structure: Memristors have been fabricated from a variety of materials. For example, a TiO_{2-x} layer with oxygen vacancies on a TiO_2 layer without oxygen vacancies sandwiched between metallic (platinum) electrodes as shown in Fig. 1(a) [2]. More generally, a memristor consists of an insulator sandwiched between two metal layers (metal-insulator-metal or MIM), where the insulating layer may be a variety of materials including chalcogenides [12], [13], metal oxides [14], [15], perovskites [16], [17], or organic films [18], [19].

3) Operation: Memristors have at least two resistance states, a high resistance state (HRS) and a low resistance state (LRS). To switch a memristor from the HRS to the LRS (a SET operation), a voltage bias of the appropriate polarity and magnitude, V_{SET}, must be applied to the device. A device in the LRS may then be returned to the HRS (a RESET operation) by applying a lower voltage, V_{RESET}. Additional resistance states are attainable by limiting the applied voltage or current.

MIM memristors demonstrate several switching styles depending on its material stack. When V_{SET} and V_{RESET} are of opposite polarity, the device is said to be bipolar. When V_{SET} and V_{RESET} are of the same polarity, the device is said to be unipolar.

4) Simulation Models: Simulation models for metal oxide and other types of memristors have been developed based on their device physics [20]. The relation between the flux $\phi(\mathrm{t})$ and the memristance of the device, $M(\phi(t))$, can be written as

$$M(\phi(t)) = \frac{\mathrm{HRS}}{D}\sqrt{(D^2 - 2\eta\frac{\mathrm{LRS}}{\mathrm{HRS}}\phi(t)\mu} \tag{2}$$

where L, W, and D are the length, width, and thickness of the device, respectively. η is $+1$ (-1) for positive (negative) voltages, and μ is mobility of the dopants. The rate at which the domain wall moves is given as [21]

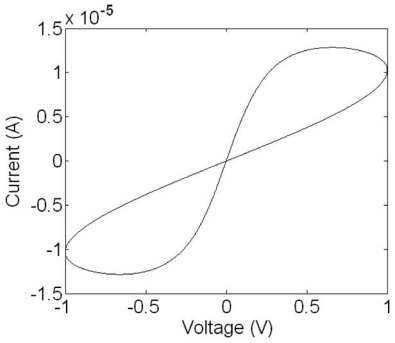


Fig. 2. ***Theoretical current-voltage characteristics of a bipolar memristor [4].***

$$\frac{dW}{dt} = \mu I \frac{R_{\mathrm{ON}}}{D}(1 - x^{2p}) \tag{3}$$

where I is the current flowing through the memristor, and p is a fitting coefficient.

5) Characteristics: Metal-oxide memristors have unique characteristics that may be leveraged for security. Not all types of memristors possess all these characteristics. The specific characteristics exhibited by a memristor depend on the material used.

i) **Nonvolatility**. Memristors retain their memristance value even when the power is turned OFF.
ii) **Bidirectionality**. Some bipolar memristors exhibit similar current-voltage characteristics irrespective of the polarity of the applied voltage or current. This is evident from the symmetric, theoretical I-V curve in Fig. 2.
iii) **Nonlinearity**. The I-V characteristics of memristors are highly nonlinear due to their time-dependent and voltage-dependent behavior, as shown in (2). Also, the HRS to LRS ratio is typically on the order of $10^3 - 10^6$.
iv) **Formation process**. For many types of memristors, a separate forming step (V_f) is required to initialize the memristor to the LRS. Prior to this point, the memristor behaves as a linear resistor; only after forming the devices exhibits the switching characteristics [23].
v) **Memristance drift**. On applying an input voltage (positive or negative) across certain metal-oxide memristors, its memristance changes because of the movement of dopants, a process called memristance drift [2]. The amount of drift depends on the polarity, amplitude, and duration of the applied voltage.

Table 1 Characteristics Exhibited by Select Nanoelectronic Devices. The "–" Mark Indicates That the Corresponding Information is Not Available in Literature

Devices	Non-volatility	Bi-directionality	Nonlinear I-V relationship	Device formation	Run-time drift	Process variations	Radiation hardness	Temperature stability
CMOS	No	No	Yes	No	Yes	Yes	No	No
Memristor	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
RRAM	Yes	Yes	Yes	Yes	–	Yes	Yes	Yes
PCM	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
CRRAM	Yes	Yes	Yes	–	–	Yes	–	–
STT-RAM	Yes	Yes	Yes	No	–	Yes	No	Yes
OST-RAM	Yes	Yes	Yes	No	–	Yes	–	Yes

vi) **Process variations**. According to (2), the memristance of a memristor is affected by process-variation induced changes in its dimensions and dopant concentration. Furthermore, the effects of variation in the thickness of the memristor upon its memristance values are highly nonlinear (more significantly for the LRS than the HRS [24], [25]).

vii) **Radiation-hardness**. Some memristor devices are inherently radiation-hard due to their material properties [2].

viii) **Temperature stability**. The LRS and HRS values are highly stable in the case of a TiO_2 memristor, since the temperature coefficient of resistance for TiO_2 is very small (less than -3.82×10^{-3}/K). However, the switching speed of the memristor varies with temperature because of the change in the mobility of dopant atoms.

Table 1 lists the characteristics exhibited by different nanoelectronic devices. All these characteristics with the exception of nonvolatility and radiation-hardness pose problems when these devices are used to build memory and logic circuits. However, we show that these typically problematic characteristics can be useful in the context of security. Only the memristor exhibits all these characteristics. Hence, we will use memristor as an example device and explain how security primitives can be built by leveraging its characteristics.

III. NANOELECTRONIC PHYSICAL UNCLONABLE FUNCTIONs (NanoPUFs)

In this section, we introduce physical unclonable functions (PUFs). We then describe the architecture of a memristor-based NanoPUF, analyze its security properties, and compare it with CMOS-based PUFs.

Random unclonable physical disorders in the integrated circuit (IC) fabrication process may be leveraged to produce unique responses (outputs) upon the application of challenges (inputs) [26]. A special circuit called the physical unclonable function (PUF) is used for this purpose. PUFs map a challenge to a response.

PUFs have been used for secure software execution on a processor [27], for device authentication, for trusted configuration of Field Programmable Gate Arrays (FPGAs) [28], and for encrypted storage [27].

A. Architecture

Researchers have explored the inherent process variations in memristors to create PUFs [22], [29], called NanoPUF, which are described here. The NanoPUF shown in Fig. 3 consists of three major parts:

i) A crossbar with memristors.

ii) The challenge circuit enables one to apply a challenge (input) to the crossbar. It selects a memristor through the row and column decoders. The row decoder applies a voltage of magnitude V_{dd} to that particular row. The column decoder selects that particular column and connects it to the load resistance, R_{load}. All other rows and columns remain floating. While the amplitude of this pulse is V_{dd}, one can apply it for different durations.

iii) The response circuits collects the response from the crossbar for a challenge. It consists of R_{load} and a current comparator. The comparator compares the current flowing out of the column (I_{out}) to that of the reference current (I_{ref}). If I_{out} is

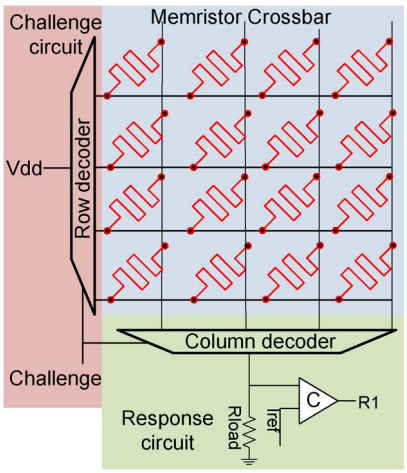


Fig. 3. *A memristor-based nanophysical unclonable function (NanoPUF). A 4 × 4 memristor crossbar that can be used as a NanoPUF (see Section III for more details.) [22].*

Table 2 Quality of Unique Responses Produced by Memristor-Based NanoPUF [22]

Metric	Uniqueness	Uniformity	Bit-aliasing
Value	49.85%	49.99%	49.99%

greater than I_{ref}, then the response bit is logic 1, otherwise it is 0.

B. Operation

A write pulse of a fixed duration is applied to the selected memristor. Due to process variations, some memristors turn ON and the others remain OFF. If a memristor turns ON, the value of the corresponding response bit is assigned to logic 1. Otherwise, it is assigned to logic 0. While [22] varies only the duration of the write pulse, [29] varies both duration and amplitude of the write pulse.

C. Security Analysis

Simulation setup. The variable mobility model from [24] is used. For the considered device, LRS is 121 KΩ, HRS is 121 MΩ, D is nominally 50 nm, and μ 0 is 5^{-18} $m^2/(V \cdot s)$. The nominal write time for the device is fixed at 7.1 μS and the amplitude at 1.2 V. To analyze the capability of NanoPUFs in producing unique IDs, one can use three metrics—uniqueness, uniformity, and bit-aliasing [30].

Uniqueness is defined as the Hamming distance between the responses from two different crossbars upon application of the same challenge. Its ideal value should be 50%.

Uniformity is defined as the proportion of 1's and 0's in a response. It ensures the randomness of the response. Its ideal value should be 50%.

Bit-aliasing is defined as the affinity of a response bit towards either 0 or 1. Ideally, the value for bit-aliasing should be 50%. Because of bit-aliasing, different PUFs may produce similar response bits. Consequently, the responses of these PUFs will be more predictable. Table 2 lists the quality of unique responses produced by a NanoPUF [22]. All the values are close to the ideal value, thus making memristor-based NanoPUFs promising.

D. Other NanoPUFs

Researchers have exploited the process variations in other nanoelectronic devices such as STT-RAM and PCM to design PUF circuits. Domain-wall memory PUF uses the variation in the write time of STT-RAMs [31]. PUF circuits are built using magnetic tunnel junction devices [32], [33] and graphenes [34]. Similarly, one can use the variation in the write time of PCM devices to produce unique responses [35]. One can also use the variation in the resistivity of diodes in a crossbar as a PUF [36]. The randomness in the number of carbon nanotubes, their alignment, and the ratio of semiconductor to metallic tubes has been used to build PUF circuits [37].

E. CMOS PUFs

Several PUFs have been proposed using CMOS devices. In a ring oscillator PUF (RO PUF) [38], the frequencies of identical ring oscillator instantiations are compared with each other. The challenge to the RO PUF selects the specific ring oscillators to be compared. The response of the RO PUF is the output of this comparison. In an arbiter-based PUF [39], the delays of identical circuit paths are compared against each other to generate a response. The paths are configured by the challenge to the arbiter-based PUF. A butterfly PUF exploits the delay variations in interconnects in cross-coupled loops found in latches and flip-flops [40]. Power networks in ICs are also used as PUFs [41]. Memory cells have also been used as PUFs. The random values in SRAM cells during start-up were used to construct a PUF in [28]. Mecca PUF exploits the read failures in SRAM cells to generate responses [42]. Surveys on different PUF circuits are provided in [43]–[45].

F. Advantages Over CMOS PUFs

When compared to SRAM-based PUFs, NanoPUFs can generate more response bits for the same amount of area as memristors are denser than SRAM cells. Furthermore, they consume less power when compared to their SRAM counterparts.

G. Outstanding Challenges

Stability. A PUF circuit should produce a stable response for a challenge at different temperature and voltage conditions. However, the stability of NanoPUF circuits have not been reported yet.

Entropy. The number of responses produced by a PUF circuit should be exponential in the number of elements in the PUF circuit. However, in case of NanoPUFs, the number of responses produced is linear in the number of nanoelectronic devices. Hence, one has to develop techniques to increase the entropy of NanoPUFs.

IV. NANOELECTRONICS-BASED PUBLIC PHYSICAL UNCLONABLE FUNCTIONs (NanoPPUFs)

In this section, we introduce public physical unclonable functions (PPUFs). We then describe the architecture of a memristor-based NanoPPUF [46], analyze their security properties, and propose security protocols. In this section, Alice and Bob want to exchange information securely. Mallory is an attacker trying to obtain this information.

PPUF is a variant of PUF. Its simulation models are made public [46]–[48], unlike a PUF whose simulation models are hidden from the attacker. Although an attacker can simulate the PPUF on a given challenge to obtain a response, the simulation time is too large (e.g., several years) compared to the time it takes to apply a challenge and obtain its response on the PUF primitive (e.g,. a few nanoseconds). A NanoPPUF can implement two-party

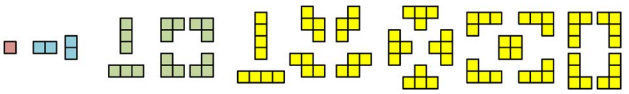


Fig. 4. *All polyominoes of size 1 (red), 2 (blue), 3 (green), and 4 (yellow).*

security protocols such as authentication, key exchange, bit commitment, and time stamping. One cannot use PUF to implement many of these protocols as it requires both the parties to know the challenge-response pairs *a priori*.

One can build a NanoPPUF using memristors by exploiting characteristics such as process variations, bidirectionality, and the simulation complexity of memristor and memristor-crossbar models [46]. Furthermore, a NanoPPUF also leverages polyominoes shapes that can be realized in a memristor crossbar. A polyomino is a geometrical structure formed by connecting a number of individual blocks. An M-omino is formed by connecting M blocks. One can consider each memristor in the memristor crossbar as a block. The number of possible M-ominoes is exponential in the value of M. The total number of possible polyomino shapes in a crossbar with M memristors is $(c\lambda^M/M) \times N$, where λ and c are 4.0626 and 0.3169, respectively [49]. Fig. 4 shows polyominoes of different sizes.

A. Architecture

The NanoPPUF, as shown in Fig. 5, consists of five major parts: (i) a crossbar, (ii) a challenge and characterization circuit, (iii) a refresh and characterization circuit, (iv) a response circuit, and (v) a controller circuit.

Each crosspoint in the **crossbar**, shown in Fig. 5, consists of a memristors. In addition, it consists of tap points (shown as blue colored dots) to measure the boundary conditions. These tap points are connected to voltage sensors to measure the voltages. On selecting a set of tap points, one can realize the polyomino in a crossbar. For example, one can realize an L-shaped 4-omino by tapping the points around the memristors, M1, M5, M9, and M10. The selected tap points define the boundaries of a polyomino.

The **challenge and characterization circuit** enables one to apply the challenge during the protocol and to characterize the memristors to build simulation models. Challenges are applied to the left side of the crossbar through the 2 : 1 multiplexer. Each challenge bit corresponds to one row. When a challenge bit is 1, the multiplexer applies a voltage of magnitude V_{dd} to that particular row. If the challenge bit is 0, that row is floated so that no current from the crossbar leaks through the rows. The floating rows do not eliminate sneak paths within a circuit. Instead, they force all the sneak path currents to drain through the response circuit connected to the columns of the crossbar.

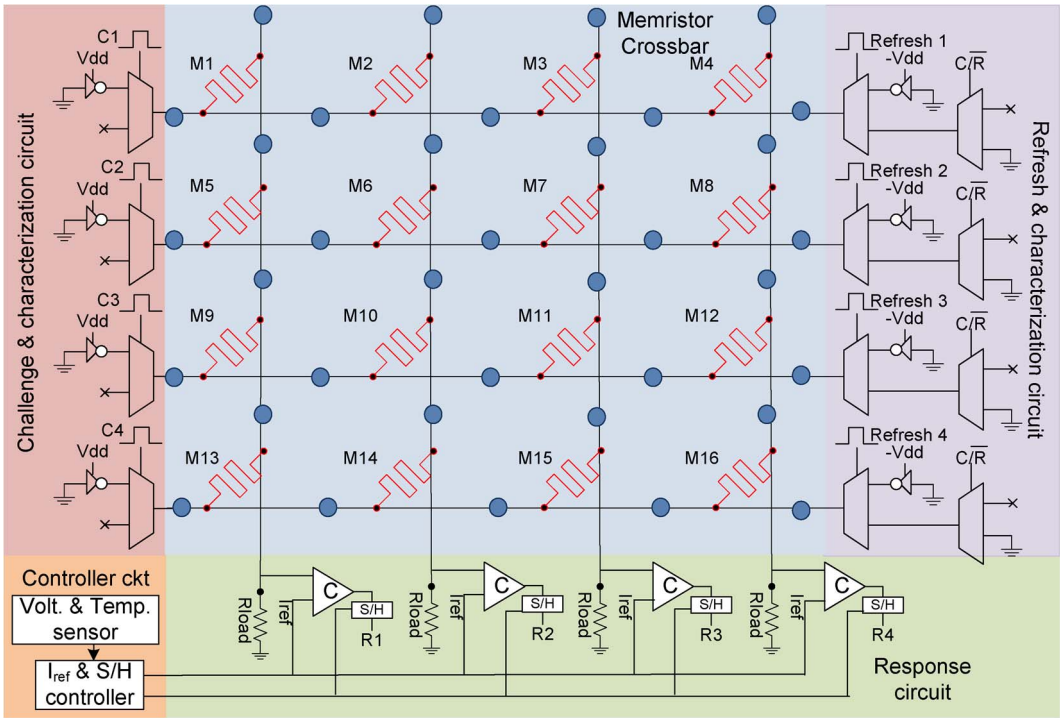


Fig. 5. *A 4 × 4 memristor-based NanoPPUF architecture. It consists of a memristor-based crossbar, a challenge and characterization circuit, a refresh and characterization circuit, a response circuit, and a controller circuit. The blue dots represent the tap points to measure the voltages.*

When the challenge bit is 1, a positive pulse is applied to that corresponding row. While the amplitude of this pulse is fixed at V_{dd}, one can change its width. The pulse is long enough to cause all the memristors in the row to switch to the LRS. In the refresh phase, the left hand side of each row is floated. During characterization, a positive pulse is applied only to the row that contains the selected device resides. All other rows remain floating.

The **refresh and characterization circuit** has two $2:1$ multiplexers per row. In the refresh phase, the multiplexers apply V_{dd} to the row. This pulse is long enough such that all the memristors are switched to the HRS. In the characterization phase, the multiplexers force the right hand side of the crossbar to float so that the characterization pulse passes only through the target device. In the challenge-response phase, the multiplexers force the right hand of side of the crossbar to float so that sneak path currents drain only through the response circuit.

Each column of the crossbar generates a response bit. In the **response circuit**, each column of the crossbar consists of a load resistance (R_{load}), a current comparator, and a sample and hold (S/H) circuit. The comparator compares the current flowing out the column (I_{out}) with that of the reference current (I_{ref}). If I_{out} is greater than I_{ref}, the response bit is logic 1; Otherwise, it is logic 0. The S/H circuit latches the value of the output bit at the sample time, which is controlled by the controller circuit.

The **controller circuit** senses the current supply voltage and temperature and sets the I_{ref} and sample time values. The controller circuit has a look-up-table that stores the predefined I_{ref} and sample time values for different voltage and temperature values. Controller circuit with similar capabilities have been demonstrated in [41].

B. Operation

The NanoPPUF works in four phases as described below.

Characterization phase. This is a one-time operation. In this phase, the individual devices are characterized to build an accurate simulation model for the NanoPPUF. Characterization of a device involves determining its length (L), width (W), and thickness (D). Specifically for devices in [2], the characterization process involves the following steps:

i) **Determining the HRS value**. Switch off the device by applying a negative pulse of long duration. Next, apply a positive pulse of magnitude V_{dd} with smaller duration. Measure the current (I_{load}) flowing through the load resistor (R_{load}). The value of HRS is given as:

$$\text{HRS} = \frac{V_{dd}}{I_{\text{load}}} - R_{\text{load}} \tag{4}$$

ii) **Determining the LRS value**. Switch on the device by applying a positive pulse with sufficient duration to turn it on. Next, apply a negative pulse of magnitude V_{dd} and smaller duration. Finally, measure the current flowing through the load. The value of LRS is given as

$$\text{LRS} = \frac{V_{dd}}{|I_{\text{load}}|} - R_{\text{load}} \tag{5}$$

iii) **Determining the time to switch on** (t_{ON}). Switch off the device by applying a negative pulse of long duration. Next, apply a positive pulse of shorter duration until the device switches on. The time to switch on the memristor, t_{ON}, is the duration of this positive pulse. After measuring HRS, LRS, and t_{ON} values, one can solve the following equations to obtain the values of L, W, and D.

$$M_{\text{off}} = \frac{\rho_{\text{off}} \times D}{L \times W} \tag{6}$$

$$M_{\text{on}} = \frac{\rho_{\text{on}} \times D}{L \times W} \tag{7}$$

$$\frac{dW}{dt} = \frac{M_{\text{off}} - M_{\text{on}}}{t_{\text{ON}}} \tag{8}$$

The above steps must be repeated for every memristor in the crossbar. At the end of this phase, the simulation model for the NanoPPUF is developed. Each invocation of a protocol (e.g., authentication) using NanoPPUF involves the following phases.

Refresh phase. In this phase, all the memristors are refreshed to a known state before the start of the protocol. All the memristors are switched to their HRS by applying a negative voltage pulse of sufficient duration across them. Thus, the changes in the memristance values of the memristors caused by previously applied challenges are erased. This phase uses the challenge and characterization circuit as well as the refresh and characterization circuit.

Challenge-response phase. In this phase, the challenge is applied by the challenge and characterization circuit, and the response is measured by the response circuit. The controller circuit compensates for voltage and temperature fluctuations.

Measurement phase. In this phase, the voltages at the boundaries of the polyomino nodes selected by the verifier will be measured using voltage sensors.

C. Security Analysis

Simulation setup. Simulation models of the memristor device fabricated in [2] were developed based on the TEAM Model proposed in [21]. The parameters of this device are given in Table 3. To analyze the effect of variation, the thicknesses of the devices are varied by $\pm 2\%$. To evaluate the security and stability metrics, we performed

Table 3 Device Parameters for NanoPPUF

Parameter	Value
LRS	121KΩ
HRS	121MΩ
L	50nm
D	50nm
W	50nm
μ	1e-9m^2/(V·s)
p	0.5
α	-40×1e-6

Monte Carlo simulations for 100 crossbars using the HSPICE simulation tool [50]. For each crossbar we applied 100 challenges. We used an Intel(R) Xeon E5-2450L 32 cores CPU with 128 GB memory operating at 1.80 GHz to simulate the NanoPPUF circuits.

Size of the crossbar (N) versus simulation time. The size of the crossbar should be large enough such that it is computationally infeasible for an attacker to simulate the NanoPPUF, but small enough to be fit into a chip. Therefore, one needs to determine the size of the crossbar that satisfies these two constraints. Fig. 6 shows the simulation time for different crossbar sizes and for different resistive devices. It can be seen that the memristor device has a higher simulation time than the other devices, because of it is a highly nonlinear I-V relationship.

One can estimate the simulation time of a NanoPPUF by determining the polynomial equation that fits the data points shown in Fig. 6. Such polynomial equations can be estimated by using the curve fitting tool box in Matlab. On curve fitting the data values in Fig. 6, the simulation time of an NanoPPUF with N rows and N columns $(N \times N)$ is given as

$$\text{Simulation time} = 0.0175N^3 + 0.412N^2 + 4.99N + 2.39s. \quad (9)$$

From this equation, one can determine that in order to have a simulation time of 1000 days, the PPUF should be of size 1782 × 1782.

Furthermore, one can also derive the lower bound on simulation time in the following way. Many SPICE simulation tools perform transient simulations of a circuit by multiplying its voltage and conductance matrices. The size of these matrices is equivalent to the number of nodes in the circuit. In the case of the NanoPPUF with $N \times N$ memristors, these matrices are at least of the size $N \times N$. As the complexity of matrix multiplication of size $N \times N$ is 2.373 N^2 [51], the lower bound of simulation time for the NanoPPUF is also 2.373 N^2.

Polyomino size (M) versus simulation time. Increasing the polyomino size exponentially increases the number of possible polyominoes but also increases the simulation time for the verifier. Fig. 7 shows the number of possible polyominoes and the simulation times for different sizes of polyominoes. The left-hand side (LHS) Y axis is in log scale, while the right-hand side (RHS) Y axis is in linear scale. One can infer from the figure that even for a small size of the polyomino (e.g., 20), the number of possible polyominoes is more than a billion. Such a large number of polyominoes thwarts an attacker from masquerading. The attacker does not know which of these billion polyominoes that the verifier will select for her simulation. With a polyomino size of 20, the simulation time for the verifier is around 25 s which is feasible to perform real-time authentication.

Unique responses. Fig. 8 shows the distribution of the three metrics—uniqueness, uniformity, and bit-aliasing—for different crossbar sizes. It also shows a 95% confidence interval via the error bars. One can see that results are close to the ideal value of 50%. Furthermore, the spread of the confidence intervals is 3% at most.

Stability against voltage fluctuations. The NanoPPUF should produce stable irrespective of environmental variations. To estimate the stability of the NanoPPUF against

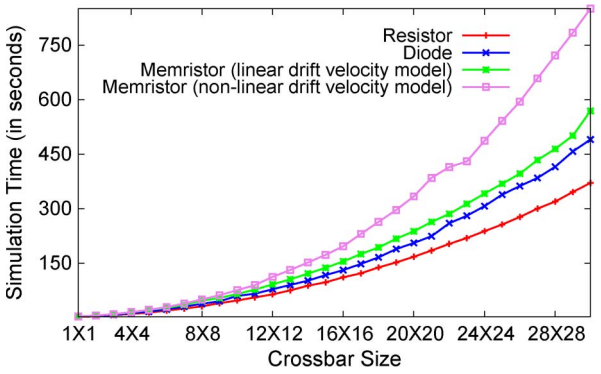


Fig. 6. *Simulation time of different crossbar sizes for different resistive devices.*

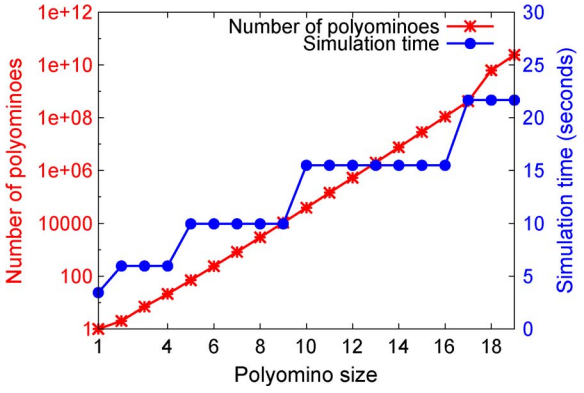


Fig. 7. *Number of possible polyominoes and the simulation times for different sizes of polyomines.*

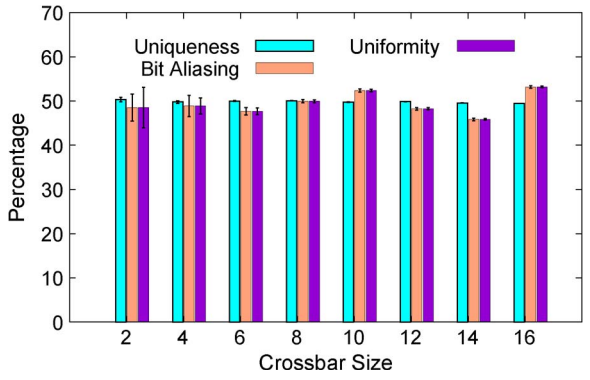


Fig. 8. *Uniqueness, bit-aliasing, and uniformity of NanoPPUF for different sizes of crossbar.*

voltage fluctuations, we first apply a challenge and calculate the response of the NanoPPUF at nominal operating conditions (1 V for 32 nm CMOS technology). We then obtain the response for the same challenge at different voltage conditions. We estimate the reliability of the NanoPPUF using two metrics: correctness and reliability.

i) **Correctness**. For voltage fluctuations, correctness is the maximum closeness between the responses at nominal voltage and at different operating voltages in terms of Hamming distance. Ideally, this value should be 100%.

ii) **Reliability**. This metric is similar to that of correctness, differing only by a factor. Again for voltage fluctuations, reliability is the average closeness between the responses at nominal voltage and at different operating voltages in terms of Hamming distance. Ideally, this value should be 100%.

Fig. 9 shows the correctness and reliability results for voltage fluctuations for an 8×8 crossbar. Even though the correctness and reliability values decrease at higher voltages, the minimum value is still greater than 90%. Furthermore, the spread of the 95% confidence interval is $\pm 1\%$ at most. Thus, the NanoPPUF produces reliable responses even in the presence of voltage fluctuations.

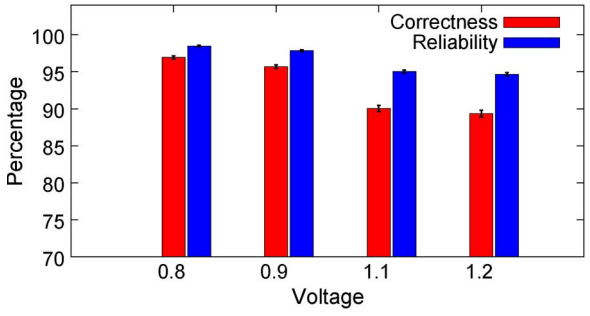


Fig. 9. *Correctness and reliability results for an 8×8 NanoPPUF in the presence of voltage fluctuations.*

D. Protocols

We now outline two 2-party security protocols enabled by a NanoPPUF. Bob's NanoPPUF is denoted by PPUF_B. The challenge C is the input and the response R is the corresponding output of the NanoPPUF. B represents the set of boundary conditions (voltage values) of a selected polyomino in a NanoPPUF crossbar. The challenge set X is the list of pins, where the challenge vector C is applied. The number of bits in C and X are equal.

User authentication. We propose a user authentication protocol using NanoPPUF. It uses a NanoPPUF which is large enough that it is computationally infeasible to simulate it even for a single challenge.

Assume that Alice wishes to authenticate that she is indeed conversing with Bob and not a malicious adversary pretending to be Bob. Alice issues a challenge C to Bob. Bob applies the challenge to his physical NanoPPUF, PPUF_B, and returns the response R to Alice. Given this challenge-response pair, Alice can validate the authenticity of Bob. Fig. 10 shows a time-bounded authentication protocol using a NanoPPUF. Due to the exponentially large number of polyominoes in the NanoPPUF and the bidirectionality of the NanoPPUF, Alice can simulate selected polyominoes and validate the inputs and outputs along the boundaries of the selected polyomino.

An adversary, Mallory, masquerading as Bob has to respond to Alice with the same output response R. However, this is not possible as Mallory cannot simulate PPUF_B to obtain response R, and R is random for different challenges. Alice can safely pick a random polyomino from among the exponential number of polyominoes in this large NanoPPUF grid to validate the challenge and response on PPUF_B. Mallory has near-zero probability of randomly guessing the chosen polyomino. Consequently, Mallory cannot choose the polyomino *a priori* and simulate it. For additional security, Alice could use two or more polyominoes or even request responses from Bob to two or more challenges.

Remote secret key exchange allows Alice and Bob to securely communicate by encrypting their messages with a secret key. A challenge to the NanoPPUF can be used as the secret key. For this protocol, the size of the NanoPPUF is substantially reduced so that one can simulate it in real time.

When Alice decides to send a message to Bob, she simulates a secret key C_B on PPUF_B and calculates R_B. Bob receives a copy of R_B, the encrypted message, $M = C_B \oplus m$, and the input challenge set X on which the challenge was applied. Since Bob owns the NanoPPUF that originally output R_B, he determines the secret key to decrypt the message by iterating through all the challenges in X on the given input set until the challenge C that produced R is found. While can perform this in real-time as he has access to PPUF_B, an attacker can only simulate it. Since the simulation time of PPUF_B is long (say, several years), and Mallory has to iterate through all the challenges in X, she cannot determine the key. For these reasons an eavesdropper is unlikely to find the secret key through simulation.

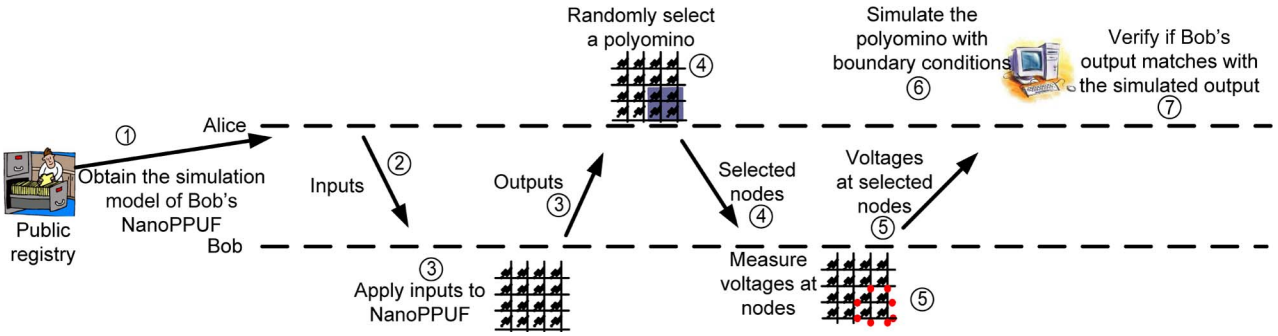


Fig. 10. ***Protocol for time-bounded authentication using NanoPPUF [46].***

E. Other NanoPPUFs

Till now, NanoPPUFs using only memristors have been proposed [46]. One needs to investigate the applicability of using nanoelectronic devices other than memristors as PPUFs.

F. CMOS PPUFs

SIMulation Possible but Laborious (SIMPL) systems were proposed for time-bounded authentication [48]. SIMPL systems were constructed using cellular nonlinear networks and CMOS-based static random-access memory (SRAM) cells. Another CMOS-based PPUF uses XOR networks [47]. This PPUF also assumes that the simulation of the entire PPUF circuit is computationally impossible.

G. Advantages Over CMOS-Based PPUFs

In a SIMPL PPUF, the time difference between the execution of an input on a SIMPL primitive and simulation of a SIMPL model has not been demonstrated. This may preclude their use in two-party protocols like bit commitment, oblivious transfer, zero-knowledge proofs, and coin flipping that requires an exponential difference [47].

The simulation model of XOR-based PPUF is simpler than that for nanoelectronic devices for a variety of reasons including their unidirectional nature. Hence, to achieve a desired security level (simulation time), the XOR-based PPUF has to be substantially larger; at least 10 000 XOR gates are required [47]. Such large XOR-based PPUFs have a large latency (e.g., several seconds).

NanoPPUF solves these problems as they have a high simulation time compared to SIMPL and CMOS-based PPUFs due to the inherent bidirectionality of memristors. Furthermore, the size and speed of the memristors ensure that NanoPPUF is more compact and can operate at higher speeds than SIMPL and CMOS-based PPUFs.

H. Outstanding Challenges

It is essential to consider several challenges in designing a NanoPPUF. Failure to do so could jeopardize the integrity of the system by wrongly authenticating a fraudulent user or disavowing a legitimate user.

Modeling errors. NanoPPUFs require accurate modeling of all device and circuit parameters, including the resistances and parasitic capacitances in the crossbar. However, achieving a high degree of modeling accuracy is a significant challenge. Thus, there will likely be tradeoffs between the size of the crossbar and the achievable degree of model fidelity.

Impact of peripherals (sense amplifiers and row/column drivers). The sense amplifiers, which are used to measure the output voltage in the crossbar, have an inherent noise margin. This noise margin can lead to ambiguous results, thereby resulting in uncharacteristic outputs.

Impact of temperature and voltage fluctuations on stability. The behavior of nanoelectronic devices typically vary with temperature. Thus, the outputs of crossbars built with these nanoelectronic devices will also vary with temperature and result in uncharacteristic outputs. This may lead to false authentication and/or rejection. Device physicists have demonstrated significant progress in fabricating nanoelectronic devices that are stable over a range of temperatures. For instance, researchers have demonstrated a CRRAM device that exhibits stable operation even when the device is operated at 150 °C for over 10^6 clock cycles [52].

Reduced order simulation of crossbars and its impact on security. The security of the NanoPPUF strongly depends on the complexity of the device model. If one can build a reduced model of the device (e.g., a piece-wise linear model) and still can accurately predict both the device and crossbar behaviors, the security of the system will be reduced as the computation effort for an attacker is reduced.

V. NANOELECTRONICS-BASED TRUE RANDOM NUMBER GENERATORS (NanoTRNGS)

A. NanoTRNGS

Random number generators are important security primitives as they are used in generating session keys that are essential to establish secure communication channels.

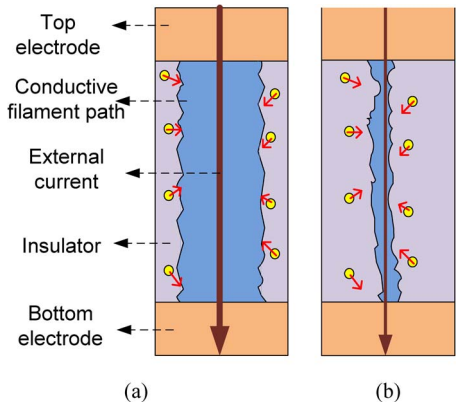


Fig. 11. ***NanoTRNG using CRRAM devices [3]. (a) Conductive filament formation on applying a high voltage. Trapped electrons have negligible impact on the current flowing through the filament. The bottom electrode is the drain region of a CMOS transistor. (b) Conducive filament on applying a low voltage. Trapped electrons significantly impact on the current flowing through the filament, leading to randomness in it.***

NanoTRNGs leverage inherent randomness in nanoelectronic devices, specifically CRRAM, to generate random numbers.

1) Architecture: CRRAMs can be used to generate random numbers as shown in Fig. 11 [3]. A CRRAM is formed from a layer of silicon dioxide between two electrodes with the bottom electrode being the drain of a CMOS transistor. This structure makes them compatible with CMOS device processes.

2) Operation: In these devices, electrons trapped in the insulation layer will randomly impact the current flowing through the filament channel. Upon applying a high voltage (3 V for the device in [3]), the current flowing through the filament will be too large to be impacted by the trapped electrons. However, on applying a low voltage (1.2 V), the width of the filament shrinks. The trapped electrons strongly influence the current flowing through the device. Since the number of trapped electrons is random, the output current, which depends on the number of trapped electrons, will also be random.

3) Security Properties: Randomness of a TRNG is quantified using the suite of randomness tests designed by the National Institute of Standards and Technology (NIST). Researchers have reported that the CRRAM-based TRNG has successfully passed all these tests [3].

4) Other NanoTRNGs: Since the premise of this TRNG is based on the randomness in the current passing through filaments, one can exploit a similar phenomenon in memristors, PCMs, and RRAM devices to build TRNGs. Similar to generating random numbers using CRRAM devices, randomness in the amount of dopants in diodes is used to generate random numbers [53].

5) CMOS-Based TRNGs: In CMOS technology, the inherent process variations in the FPGAs are leveraged to generate random numbers [54], where circuits such as ring oscillators are used for randomness extraction. However, the extracted random values are unstable since the frequency of a ring oscillator highly depends on the operating temperature. Randomness in the power-up state of SRAMs have been used to generate random numbers [55].

6) Advantages Over CMOS-Based TRNGs: NanoTRNGs are attractive over their CMOS counterparts due to their low-power consumption, high-density, and increased amount of randomness in their physical properties.

7) Outstanding Challenges: Though all nanoelectronic devices possess randomness, one needs to design circuits that extract randomness from these devices and convert into digital values. Such a circuit requires a highly-sensitive sense amplifier to act as a threshold circuit between logic 0 and logic 1. Furthermore, the stability of NanoTRNGs is yet to be studied. Since the randomness in the device current varies with respect to temperature and voltage fluctuations, the quality of random numbers generated using NanoTRNGs may also vary.

B. Nanoelectronic Device-Based Unique Signatures

Nanoelectronic devices can be used to generate a unique signature for hardware by exploiting two characteristics [56]: 1) inherently nonuniform, irreproducible process variations during fabrication and 2) forming step required to make them functional. Researchers use a pair of nonpolar nanoelectronic devices in series as a random bit generator, where the bit generation is a function of the location of a low resistance filament [56]. Multiple instances of such random bit generators can produce a random word. Since this signature is nonvolatile, it may be used for hardware identification purposes. This embedded hardware ID can be used to thwart electronic counterfeiting or detect refurbished components.

1) Architecture: Consider a pair of memristors in series as shown in Fig. 12. The bottom metal electrode (BE) and the insulator layer are common for the two devices. Each memristor has its own top metal electrode (TE).

2) Operation: During the forming step, one TE is biased while the other TE is grounded. Two low-resistant filaments are formed; one beneath each TE through the insulator material layer. During the RESET operation, the resistance values of the two series memristors are returned to the HRS. During this operation, only one of the low resistance filaments becomes highly resistive; the other filament remains of low resistivity. The location of this

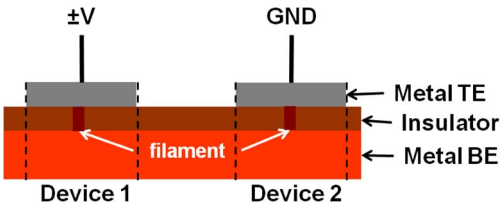


Fig. 12. ***Electrical configuration for generating unique signatures using memristors [56].***

latter filament serves as the random bit value. This location depends on the process-induced variations in insulator layer thickness and dopant concentration in the memristors. The location of the low-resistant filament is also impervious to additional SET and RESET operations. Thus, a unique signature is generated for the hardware. This signature will not be determined prior to the "formation" step, thus precluding an attacker in the manufacturing unit from passively reading this unique device ID and spoof it.

3) Security Analysis: The uniqueness of the signatures produced by a nanoelectronic-based unique signatures is quantified by the uniqueness metric. The ideal value is close to 50%. Memristor-based unique signatures, demonstrated in [22], has a uniqueness value of 49.9%.

4) Other Nanoelectronic-Based Unique Signatures: Similar to comparing the resistivity of two memristors to generate a random bit, researchers compared the resisitivity of two STTRAM cells to generate a random bit [57]. PCKGen uses the variation in the write time of PCM devices to produce unique device signatures [58]. Furthermore, these keys can be dynamically generated by applying different sets of write pulses with varying amplitude and duration.

5) CMOS-Based Unique Signatures: A unique device signature in CMOS can be derived from an unwritten SRAM circuit [40]. An SRAM cell consists of two transistors connected in a butterfly-like fashion. Due to threshold voltage mismatch caused by process variations, one transistor will be stronger than the other. This mismatch is then used to generate the random signature.

6) Advantages Over CMOS Unique Signatures: In case of CMOS-based unique signatures, an attacker in the manufacturing facility can easily read-out the unique signature and use it to spoof the hardware. Unlike with the nanoelectronic-based unique signatures, this tampering is not irrefutable.

7) Outstanding Challenges: Similar to NanoTRNGs, memristor-based unique signatures require a sense amplifier with high resolution and high stability to convert the randomness to signatures. Furthermore, they suffer from stability issues as the characteristics of the nanoelectronic devices vary with temperature and voltage fluctuations.

VI. NANOELECTRONICS-BASED TAMPER DETECTION CIRCUITs (NanoTDCs)

Tamper detection entails identifying unauthorized usage of or access to the target hardware. Tamper evident-memories not only ensure the confidentiality of the stored data but also expose any attempt to read or write into them [60]. We will show how the device formation, nonvolatility, and run-time drift properties in nanoelectronic devices can be leveraged for tamper detection. Specifically, we demonstrate manufacture-time and run-time tamper detection in memristor-based memories.

A. Architecture

Consider the crossbar with memristors shown in Fig. 3. There are two kinds of paths in the crossbar: direct path and sneak path. In a direct path, current flowing from an input (row) to an output (column) is the function of the resistance of the device at the crosspoint of that input and output. In a sneak path, the current flowing from an input to an output is a function of the resistance of devices at other crosspoints in the crossbar. Such memristor-based crossbars have been used to build nonvolatile memories [61], [62]. In these memories, the HRS and LRS are used to represent logic 0 and 1, respectively.

Write operation. To write to a particular device, V_{RESET} (for logic 0) or V_{SET} (for logic 1) is applied to the corresponding row, and 0 V is applied to the corresponding column.

Read operation. To read a particular device, a read voltage, usually a positive pulse of small amplitude, is applied to the corresponding row. The current flowing out of the corresponding column is compared with a reference current. If the output current is greater than the reference current, then a logic 1 is read, otherwise a logic 0 is read. In devices that exhibit memristance drift, applying a read voltage across the memristor can cause its memristance to drift. Hence, in order to undo this change caused by memristance drift during the read operation, a two-stage read operation is used [63]. For a bipolar memristor, the ideal read pattern uses a positive pulse followed by a negative pulse of the same magnitude and duration, creating a zero net-change in memristance.

B. Operation

1) Manufacture-Time Tamper Detection: The device formation step in memristors enables one to differentiate a virgin (i.e., nformed) device from one that has been used (i.e., ormed). Any user (authorized or unauthorized) would need to form these devices before gleaning any useful information. One can check whether a device has been used or not. Such a technique is useful to verify the

trustworthiness of the new integrated circuits received from an untrustworthy fabrication facility. In this technique, one first writes a known value to the memristor(s), reads that value back, writes the complement of the known value to the memristor(s), reads the next value back, and compares the results. If the formation step had not occurred, it would not be possible to write to the memristor(s), and the result of the comparison would show the values read were the same. However, if the memristors have been formed, then the comparison will show that the values read are different. This second case is an evidence of possible tampering of the circuit.

2) Run-Time Tamper Detection: Unauthorized memory reads in memristor-based memories can be detected as follows [59]. The key idea to detect an unauthorized read operation is to monitor the associated drift in memristance.

In order to cover his trail, the attacker (after performing the unauthorized read) may restore the memristance of the device to its original value by "unreading" (e.g., applying a read pulse of opposite polarity but with the same magnitude and width) the device. The memristance then drifts in the opposite direction by the same amount and returns to the original value.

To prevent the attacker from restoring the memristance value, the memory read operation is modified as shown in Fig. 13 [59]. The modified memory read operation uses two consecutive read pulses. While the magnitude and duration of the first pulse are public (i.e., known to the attacker as well), the magnitude and duration of the second pulse are private and known only to an authorized user. Thus, even though an attacker can restore the memristance value to its initial value using a pulse of opposite polarity, he cannot revert back the change in memristance caused by the second pulse with its private parameters. This way, the memristance value following an unauthorized read operation will be different from the initial memristance value before the read operation and cannot be undone. The authorized user can detect this change in memristance, and consequently, the tampering.

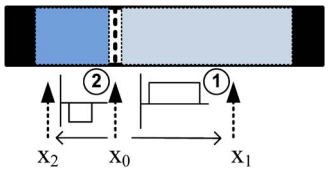


Fig. 13. ***Run-time tamper detection using nanoelectronic devices [59]. The dark and lightly shaded regions represent the high-resistive and low-resistive regions, respectively. The dotted line represents the location of the domain wall which determines the current resistance value. Every read operation uses two pulses. The magnitude and duration of the first pulse are public, whereas the magnitude and duration of the second pulse are known only to the defender.***

C. Security Analysis

In case of manufacture-time tamper detection, it is impossible for an attacker to use a memristor and evade the detection mechanism of a defender (designer). This is because an attacker has to perform "device formation" on each device before using them, changing the resistance from a few gigaohms to few megaohms; a defender can always detect this change in the resistance of the device. In case of run-time tamper detection, it has been demonstrated that a 50-mV change in the amplitude of the second pulse causes the device resistance to change by 5% [59].

D. Other NanoTDCs

Similarly, one can extend the run-time tamper detection technique to other nanoelectronic devices, such as PCM and STT-RAM, which exhibit run-time drift in their resistance. Recently, researchers have proposed a tamper detection circuit using magnetic RAM (MRAM) devices, whose device operation is similar to that of STT-RAM devices except that the magnetic field changes the free layer from parallel to anti-parallel [65]. Any attempt to read the devices using external magnetic field changes its alignment. A defender can sense this change to detect the tampering.

Recently, researchers used ambipolarity property in Silicon Nanowire Field Effect Transistors (SiNW-FETs) to prevent manufacture-time tampering [64]. An ambipolar SiNW-FET can behave either as an NMOS or as an PMOS device, depending up on the value applied to its "polarity gate" terminal. For example, Fig. 14(a) shows a circuit with SiNW-FETs. Depending upon the values applied to the polarity gate terminals, the circuit can exhibit either the configuration shown in Fig. 14(b) or the one in Fig. 14(c). The values to the polarity gate terminals are not applied during manufacturing. Thus, an attacker in the foundry will not be able to tamper with the design, as he does not know whether a particular SiNW-FET behaves as an NMOS device or as an PMOS device. This prevents piracy and overbuilding of circuits.

One can use nanoelectronic mechanical switches (NEMS) to thwart reverse engineering of a chip [66]. NEMS devices convert an external mechanical vibration into electrical energy through the piezoelectric effect. Fig. 15 conceptually illustrates the schematic of a chip using a NEMS-based energy harvester to thwart delayering attacks. The NEMS/MEMS device is connected to the erasure device through an antifuse, which is enabled after fabrication. Any attempt to delayer will trigger the NEMS device due to the induced vibration. Consequently, the erasure device is triggered, thereby destroying the design and preventing an attacker from gaining any information.

E. Tamper detection in CMOS

A CMOS device can be tampered with during manufacturing. Researchers have proposed circuit-level techniques to detect insertion of additional circuits or

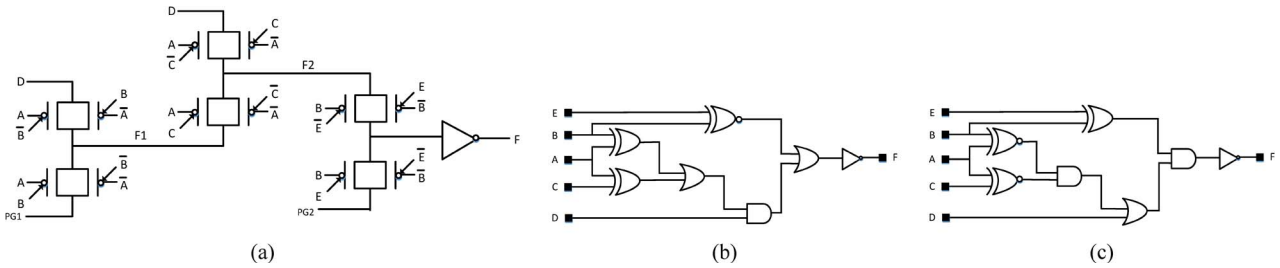


Fig. 14. *Preventing manufacture-time tampering using SiNW-FETs [64]. (a) An example circuit with SiNW-FETs. PG1 and PG2 are the polarity gate terminals. (b) Equivalent gate-level diagram when PG1 is grounded, and PG2 is connected to Vdd. (c) Equivalent gate-level diagram when PG1 is connected to Vdd, and PG2 is grounded.*

modification to existing circuits [67]. To detect run-time write operations, there are no device- or circuit- level techniques for CMOS devices to the best of our knowledge.

F. Advantages Over Tamper Detection in CMOS

As mentioned above, CMOS devices require system-level techniques to enable tamper detection, resulting in high power and performance overhead. However, Nano-TDCs use inherent device properties to enable detection, thus requiring less power and performance overhead.

G. Outstanding Challenges

In the context of tamper detection in nanoelectronic devices-based circuits, the amplitude and duration of the second pulse have to be adjusted so that the read and write margins of the devices are honored. Such fine adjustments require a complex write circuit, which may consume more power than its counterpart in conventional memory. However, researchers have proposed similar write circuits, though not in the context of tamper detection, to tolerate process variations in memistors [61], [62]. Thus, these circuits can be repurposed for security. Furthermore, to perform tamper detection in nanoelectronic device-based memories, changes in resistance values have to be accurately sensed. This requires designing highly sensitive sense amplifiers. Such sense amplifiers are typically large and consume more power. Hence, one needs to develop a power-efficient, high-resolution read and write circuits. Similar to SiNW-FETs, graphenes also exhibit the ambipolarity property [68] and thus can be used to prevent piracy and overbuilding of circuits.

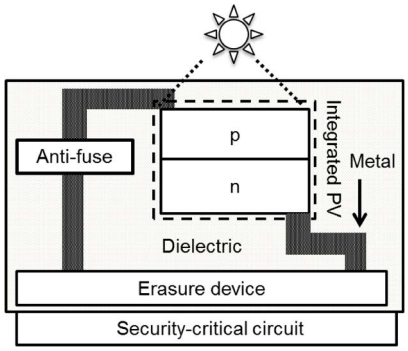


Fig. 15. *Side-view of a chip using NEMS/MEMS-based devices to detect mechanical attacks [66]. The inset shows the structure of a NEMS/MEMS cantilever.*

VII. NANOELECTRONICS-BASED FORENSICS (NanoForensics)

We now explain how one can use the run-time drift property of memristors for forensics using memristor-based digital logic gates. Specifically, one can determine the set of input patterns applied to the circuit in the past using forensics.

A. Architecture: Memristor-Based Threshold Gates

Consider the memristor-based threshold logic (MTL) gates proposed in [69]. In an MTL gate, the memristors are used as weights on the inputs of the gate. Fig. 16(a) shows a 3-input threshold gate which uses the memristors M_A, M_B, and M_C to weigh the current flowing from the inputs A, B, and C, respectively. The current mirrors isolate the currents flowing through the different inputs. The current comparator is then used to compare the sum of the weighted currents against the reference current I_{ref}. If the sum of the weighted currents is greater than I_{ref}, then the output is logic 1; Otherwise, the output is logic 0. A positive voltage denotes logic 1, and 0 V denotes logic 0.

To program the memristors of the MTL circuit, one can use a programming circuit, which is similar to the write circuit of the memristor-based memory. One can use the same write circuit to address all the memristors in the design, or a write circuit for every set of memristors [70], [71].

B. Operation

As discussed previously, when logic 1 (positive voltage) is applied to an input, the memristance value of the corresponding memristor changes. On the other hand, there

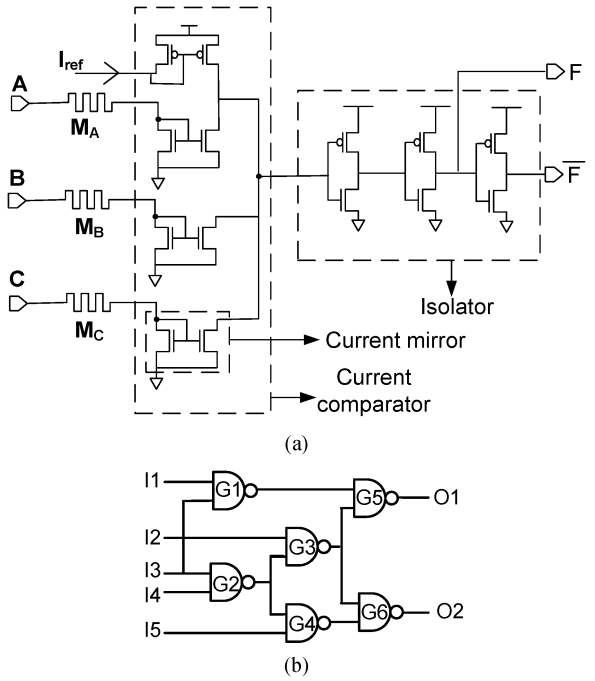


Fig. 16. *Forensic analysis in nanoelectronic architectures. (a) A 3-input memristor-based threshold logic gate (MTL) [69]. (b) C17, an ISCAS'85 benchmark circuit.*

will not be any change in memristance when logic 0 is applied. The amount of drift depends on the number of logic 1's applied to that input. For example, consider an MTL gate implementing an AND function. The memristance values of the memristors are 2 MΩ. Let the amplitude and duration of input pulses be 1.1 V and 2 ns, respectively. Consider one million, two million and three million 1's are applied to inputs A, B, and C, respectively. The final memristance values of memristors M_A, M_B, and M_C will be 2.12 MΩ, 2.25 MΩ, and 2.38 MΩ, respectively. The changes in memristance values are caused by memristance drift.

Conversely, if one determines that the final memristance values of memristors M_A, M_B, and M_C are 2.12 MΩ, 2.25 MΩ, and 2.38 MΩ, respectively, one can estimate that about one million, two million, and three million 1 s have been applied to the inputs A, B, and C, respectively. This is referred to as *forensic analysis*. Consider extending this forensic analysis from individual gates to circuits. The number of 1's received by an input of a gate depends on its location within the circuit. Consequently, the change in the memristance of the memristors at the inputs of different gates will be different.

Consider the C17 circuit, one of the ISCAS'85 benchmark circuits, shown in Fig. 16(b). Let us name the memristors based on the signals/gates that feed them. On applying the input pattern 11111, the memristance values of the memristors I1–I5 and G3–G5 will change. Similarly, on applying the input pattern 00000, the memristance values of the memristors G1–G4 will change. Note that the memristance values of the memristors G3 and G4 change for both patterns.

By measuring the change in the memristance of a memristor, one can determine the number of 1's received at that input. Similarly, the number of 1's received by all the memristors in the circuit can be determined. After measuring the changes in memristance values, a forensic analyst can make the following observations. If none of the memristors had drifted, then the hardware was never used. If a set of memristors had drifted, then he can identify a set of input patterns that may have been applied to the hardware which caused that drift. For instance, if only the memristors *I1–I5* and *G3–G5* had drifted, then he will identify the input pattern applied to the hardware as 11111. If the memristor *G3* had drifted more than the other gates, then he infers that input patterns applied are d0ddd, d100d, d101d, and/or d110d, where d represents a "don't care" bit value.

C. Security Analysis

While the above example demonstrates the feasibility of forensics in MTL gates, its security analysis is still lacking. The ability of an attacker to apply a set of input patterns which cannot be recovered by the above analysis are yet to be developed. Furthermore, metrics for hardware forensics is yet to be developed.

D. Other NanoForensics

One can extend a similar line of research to other nanoelectronic devices such as PCM which exhibit runtime drift.

E. Forensics in CMOS

Forensic analysis of CMOS-based designs has not been explored to the best of authors' knowledge. However, similar to memristance drift for memristors, one can leverage the Negative Bias Temperature Instability (NBTI) effect in CMOS for forensics [72]. NBTI occurs in a CMOS transistor when electron traps are formed at the silicon-silicon dioxide interface. The NBTI effect in PMOS is more dominant than it is in NMOS. Applying logic 1 to the PMOS transistor subjects it to NBTI stress which then degrades the threshold voltage of the transistor and thereby increases its delay. A forensic analysis can detect this change if one can determine the number of logic 1 s received by that transistor.

F. Advantages Over Forensics in CMOS Designs

Unfortunately, the rate of change in the delay of a CMOS transistor due to NBTI is slow (on the order of a few years) when compared to the instantaneous change in memristance values due to memristance drift. Thus, performing forensics in memristor-based circuits is relatively easier when compared to CMOS.

G. Outstanding Challenges

To perform forensic analysis in MTL gates, changes in memristance values have to be accurately sensed. This requires designing highly sensitive sense amplifiers. Such sense amplifiers are typically large and consume more power. However, researchers have designed power-efficient sense amplifier circuits [70], [71].

Additionally, increasing either the duration or amplitude of the input pulses will significantly change the memristance value of the device, thereby making forensic analysis easier. However, changes in memristance values over time also move the weights of MTL gates out of range, making the hardware nonfunctional. An authorized user has to once again reset the memristors to their initial memristance values. While decreasing the duration or amplitude of the input pulses will increase the usage time of the hardware, it makes the forensic analysis harder as the change in memristance values will be small. Thus, the duration and amplitude of input pulses have to be optimized for hardware usage time and ease of forensic analysis.

VIII. NANOELECTRONICS-BASED CRYPTO ARCHITECTURES

While nonvolatile main memories (NVMM) are attractive as they retain data even during power-off, an attacker can read-out its contents. One can prevent such attacks by encrypting the data using algorithms (e.g., AES), storing it in the memory and decrypting it while reading it out. However, this results in huge power, performance, and area overhead. Furthermore, these cryptographic algorithms are susceptible to side-channel attacks, where an attacker can retrieve the secret key by monitoring the side-channel information such as power and timing. To thwart such attacks and enable low-cost encryption, researchers have proposed encryption using nanoelectronic devices [73], [74]. Sneak path encryption (SPE) uses sneak paths in a crossbar to encrypt the data, thereby creating a secure NVMM (SNVMM) [73].

A. Architecture

Consider a typical two-level memory architecture shown in Fig. 17(a). The processor and cache operate on the unencrypted data. The sneak-path encryption control unit (SPECU) is placed between the NVMM and the level-2 cache. The SPECU controls the SPE to encrypt the data stored in NVMM.

The SNVMM consists of a crossbar and a transistor control circuit shown in Fig. 17(b). The crossbar is similar to that of the one shown in Fig. 3, except that each cross-point in the crossbar, in addition to a memristor, has a transistor. This transistor enables the selection of a particular memristor for read and write operations. The selected memristor is called the *point of encryption* (PoE). The transistor control circuit switches on a selected set of transistors to create sneak paths used by the SPECU for encryption and decryption. During conventional read and write operations, the transistor control circuit disables sneak paths, thereby facilitating valid read and write operations.

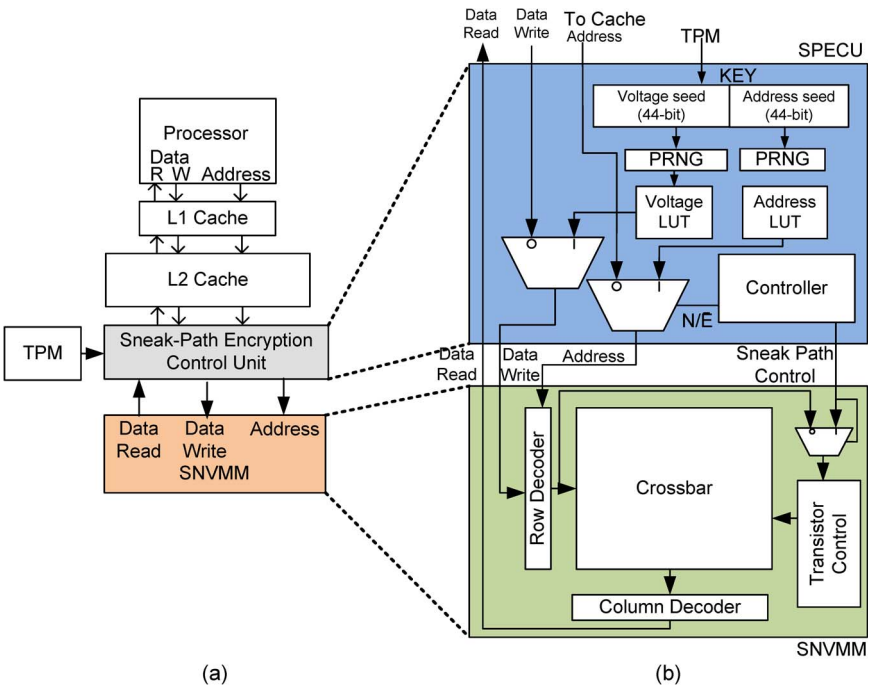


Fig. 17. *(a) Architecture of a SNVMM. (b) SPECU architecture and NVMM modifications for sneak path encryption [73].*

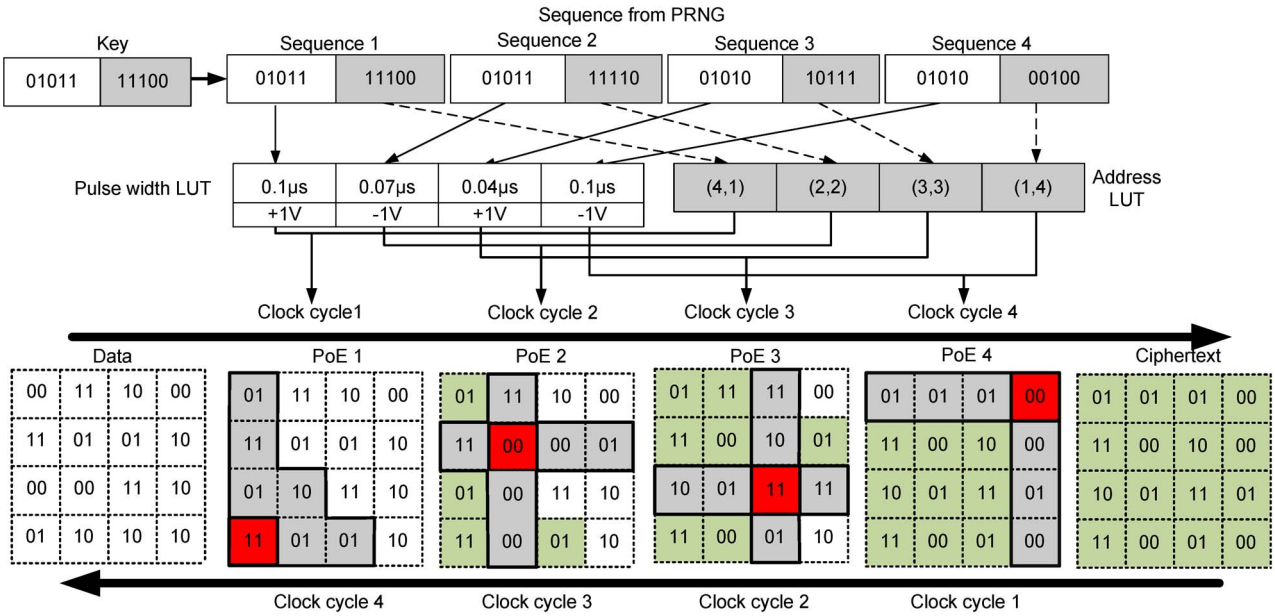


Fig. 18. ***SNVMM operation: Encryption/Decryption in a 4 × 4 crossbar. The component in red shows the PoE, grey the affected memristors, and green the memristors whose contents are encrypted [73].***

In addition to enabling sneak paths for encryption and decryption, SNVMM modifies the amplitude and duration of write pulses, which are determined by the PRNGs and the look-up-tables. One PRNG uses the first-half of the key to generate random numbers that control the amplitude and duration of the write pulse; the other PRNG uses the second-half of the key to generate random numbers that select the PoEs. The two LUTs map the random numbers to the amplitude and duration of the write pulse and the location of PoEs.

B. Operation

The write and read operations of SNVMM are similar to the memory described in Section VI. We now describe encryption and decryption.

Encryption. The user applies the key which is used as the seed for the PRNGs. The PRNGs and LUT generate the duration and amplitude of write pulses and the location of PoEs. Applying a voltage pulse to a PoE, in the presence of sneak paths, results in a voltage difference across the selected and its adjacent memristors. Their memristance is either increased or decreased, thereby changing the data stored in them.

The same process is repeated for several PoEs, changing the content of memristors. A new PoE is selected for each clock cycle. The process is repeated for several clock cycles until the contents of all the memristors in the SNVMM are changed. Fig. 18 shows how the contents of different memristors are modified at every clock cycle. The set of memristors affected are unique to each PoE and are based on the physical parameters of the crossbar and the data stored in each cell.

Decryption. As shown in Fig. 18, the PoEs are addressed in an order inverse to that of encryption. Though the duration of the write pulses remains the same as the one during encryption, their amplitude is inverted. This reverses the change in the memristances caused during encryption and retains the original data. Any change in the sequence of the addressed PoEs alters the set of memristors affected, thereby resulting in a data different from the original data.

C. Security Analysis

The SPE scheme has the following security features [73]. (i) high-sensitivity to changes in the key, data, and physical parameters of the crossbar, (ii) low correlation between the data and the encrypted data, and (iii) randomness in encrypted data when the either the data has low-density of 1 s and 0 s or the key has low density of 1 s and 0 s. These features ensure that the encryption scheme is secure.

D. Other Crypto-Architectures Using Nanoelectronic Devices

Since nanoelectronic devices, especially RRAMs, consume less power than CMOS devices, RRAM-based crossbars are used to implement cryptographic algorithms. Such crossbars are resilient to power analysis-based side-channel attacks [74].

Recently, researchers used Graphene-based symmetric tunneling FETs (SymFETs) to thwart fault attacks [64]. SymFETs cut off the source-drain current if the source-drain voltage is outside a narrow voltage band. Such

Table 4 Possible Security Primitives That Can be Implemented by Different, Select Nanoelectronic Devices. The "*" Mark Indicates the Lack of Circuit or Device-Level Techniques

Devices	NanoPUF	NanoPPUF	NanoTRNG	Nanoelectronics-based unique signatures	NanoTDCs		NanoForensics	Nanoelectronics-based crypto architectures
					Manufacture-time	Run-time		
CMOS	Yes	Yes	No	Yes	No*	Yes	No*	No*
Memristor	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
RRAM	Yes	Yes	Yes	Yes	Yes	–	–	Yes
PCM	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
STT-RAM	Yes	Yes	Yes	Yes	No	–	–	Yes
CRRAM	Yes	Yes	Yes	Yes	–	–	–	Yes
OST-RAM	Yes	Yes	Yes	Yes	No	–	–	Yes

SymFETs can be used to build fault-tolerant power supply circuits to thwart fault attacks. Any fault, which is injected in the power supplies, changes the source-drain voltage. Consequently, the device will not output current to drive the cryptographic circuits and hence no computation will be performed, preventing an attacker from learning any useful information.

E. CMOS-Based Encryption Algorithms

Traditionally, the data stored in a CMOS memory is protected through encryption algorithms. However, there is no technique that leverages the inherent properties of CMOS devices to encrypt the memory contents.

F. Advantages Over CMOS Crypto-Architectures

As mentioned before, CMOS-based crypto-architectures have high power and performance overhead and are susceptible to side-channel attacks. However, crypto-architectures using nanoelectronic devices have less power and performance overhead and are resilient to power analysis-based side-channel attacks [73], [74].

G. Outstanding Challenges

One needs to evaluate if the cryptographic strength of SPEs is on par with that of mathematically proven encryption algorithms such as AES. Furthermore, one needs to evaluate the resiliency of crypto-architectures using nanoelectronic devices to other side-channel attacks such as timing and fault analysis.

IX. CONCLUSIONS

We surveyed the important characteristics of nanoelectronic devices and demonstrated how they can be used to build new security primitives. Researchers have focused mostly on nanoPUFs and nanoPPUFs. As a result of this, several metrics and design criteria have been developed for nanoPUFs and nanoPPUFs. To a lesser extent, design and evaluation of nanoelectronic device-based random number generators have been reported. Applications such as tamper detection and forensics using nanoelectronic devices are evolving. Furthermore, different nanoelectronic devices have different sets of properties, thus enabling different security primitives. Table 4 lists the possible security primitives that can be implemented by different nanoelectronic devices. These security primitives have found applications in industry. For instance, Crocus Technology has started providing magnetic RAM-based memory that enables one to perform authentication without reading out from the memory [75]. One can perform a red-team/blue-team approach. To understand the security capabilities of different nanoelectronic devices, we organized a red-team/blue-team competition as part of the 2014 New York University's Embedded Security Challenge [76]. Participating teams proposed different nanoelectronic device-based security primitives, including [31], [33], [64].

An important direction for device physicists is to engineer nanoelectronic devices not only for memory and logic circuits but also for security primitives. Security researchers should develop new security primitives, protocols, and associated mathematical proofs by abstracting the detailed characteristics of nanoelectronic devices. Circuit and computer-aided design engineers should bridge the gap between device engineers and security researchers, harnessing these devices characteristics to satisfy security requirements. Research challenges for engineers at different abstraction levels are listed in Table 5. These challenges are a consolidation of challenges of individual security primitives. Only when all these challenges are solved, one can harness the security benefits of nanoelectronics. ■

Table 5 Challenges for Device, Circuit, Computer-Aided Design, and Security Engineers

Abstraction	Challenges
Device-level	Temperature stability Dependency of device current on filament thickness
Circuit-level	Accurate sense amplifier and comparator circuits Stable voltage sources
Computer-aided design	Accurate and complex device models Algorithms for fast and efficient simulation of crossbars
Security	New security primitives, protocols, and associated proofs

REFERENCES

[1] L. Chua, "Memristor-The missing circuit element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[2] R. Williams, "How we found the missing memristor," *IEEE Spectrum*, vol. 45, no. 12, pp. 28–35, 2008.

[3] C. Huang, W. Shen, Y. Tseng, C. King, and Y. C. Lin, "A Contact-resistive random-access-memory-based true random number generator," *IEEE Electron Dev. Lett.*, vol. 33, no. 8, pp. 1108–1110, 2012.

[4] N. McDonald, S. Bishop, B. Briggs, J. Van Nostrand, and N. Cady, "Analysis of nonpolar resistive switching exhibited by Al/CuxO/Cu memristive devices created via room temperature plasma oxidation," in *Proc. Int. Semicond. Device Res. Symp.*, 2011, pp. 1–2.

[5] H. Akinaga and H. Shima, "Resistive random access memory (reram) based on metal oxides," *Proc. IEEE*, vol. 98, pp. 2237–2251, Dec. 2010.

[6] Y.-H. Tseng *et al.*, "A new high-density and ultrasmall-cell-size contact RRAM (CR-RAM) with fully CMOS-logic-compatible technology and circuits," *IEEE Trans. Electron Dev.*, vol. 58, no. 1, pp. 53–58, 2011.

[7] H. S. P. Wong *et al.*, "Phase change memory," *Proc. IEEE*, vol. 98, no. 12, pp. 2201–2227, 2010.

[8] M. Stiles and J. Miltat, "Spin-transfer torque and dynamics," *Spin Dyn. Confined Magn. Struct. III*, vol. 101, pp. 225–308, 2006.

[9] H. Liu *et al.*, "Ultrafast switching in magnetic tunnel junction based orthogonal spin transfer devices," *Appl. Phys. Lett.*, vol. 97, no. 24, p. 242510, 2010.

[10] Emerging Devices Summary, 2013. [Online]. Available: http://www.itrs.net/Links/2013ITRS/Summary2013.htm

[11] Y.-H. Tseng, C.-E. Huang, C. H. Kuo, Y. D. Chih, and C.-J. Lin, "High density and ultra small cell size of Contact ReRAM (CR-RAM) in 90 nm CMOS logic technology and circuits," in *Proc. IEEE Int. Electron Dev. Meet.*, 2009, pp. 1–4.

[12] A. Oblea, A. Timilsina, D. Moore, and K. Campbell, "Silver chalcogenide based memristor devices," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2010, pp. 1–3.

[13] R. Waser and M. Aono, "Nanoionics-based resistive switching memories," *Nature Mater.*, vol. 6, pp. 833–840, 2007.

[14] L. Goux *et al.*, "Coexistence of the bipolar and unipolar resistive-switching modes in NiO cells made by thermal oxidation of Ni layers," *J. Appl. Phys.*, vol. 107, no. 2, 2010.

[15] B. Briggs *et al.*, "Influence of copper on the switching properties of hafnium oxide-based resistive memory," *New Funct. Mater. Emerg. Dev. Architect. Nonvolatile Memories, MRS Proc.*, vol. 1337, 2011.

[16] A. Sawa, T. Fujii, M. Kawasaki, and Y. Tokura, "Interface resistance switching at a few nanometer thick perovskite manganite active layers," *Appl. Phys. Lett.*, vol. 88, no. 23, pp. 232112-1–232112-3, 2006.

[17] K. Szot, W. Speier, G. Bihlmayer, and R. Waser, "Switching the electrical resistance of individual dislocations in single crystalline SrTiO3," *Nature Mater.*, vol. 5, pp. 312–320, 2006.

[18] J. C. Scott and L. D. Bozano, "Nonvolatile memory elements based on organic materials," *Adv. Mater.*, vol. 19, pp. 1452–1463, 2007.

[19] N. B. Zhitenev, A. Sidorenko, D. M. Tennant, and R. A. Cirelli, "Chemical modification of the electronic conducting states in polymer nanodevices," *Nature Nanotechnol.*, vol. 2, p. 237242, 2007.

[20] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: Properties of basic electrical circuits," *Eur. J. Phys.*, vol. 30, p. 661, 2009.

[21] S. Kvatinsky, E. Friedman, A. Kolodny, and U. Weiser, "TEAM: ThrEshold Adaptive Memristor Model," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 60, no. 1, pp. 211–221, 2013.

[22] G. Rose, N. McDonald, L.-K. Yan, and B. Wysocki, "A write-time based memristive PUF for hardware security applications," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2013, pp. 830–833.

[23] Q. Xia *et al.*, "Memristor-CMOS hybrid integrated circuits for reconfigurable logic," *Nano Lett.*, vol. 9, no. 10, pp. 3640–3645, 2009.

[24] J. Rajendran, H. Manem, R. Karri, and G. Rose, "An approach to tolerate process related variations in memristor-based applications," in *Proc. IEEE Int. Conf. VLSI Design*, 2011, pp. 18–23.

[25] D. Niu, Y. Chen, C. Xu, and Y. Xie, "Impact of process variations on emerging memristor," in *Proc. IEEE/ACM Design Autom. Conf.*, 2010, pp. 877–882.

[26] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2002, pp. 148–160.

[27] G. Suh, C. O'Donnell, I. Sachdev, and S. Devadas, "Design and implementation of the AEGIS single-chip secure processor using physical random functions," in *Proc. IEEE Int. Symp. Comput. Architect.*, 2005, pp. 25–36.

[28] J. Guajardo, S. Kumar, G.-J. Schrijen, and P. Tuyls, "Physical unclonable functions and public-key crypto for FPGA IP protection," in *Proc. Int. Conf. Field Program. Logic Appl.*, 2007, pp. 189–195.

[29] P. Koeberl, U. Kocabas, and A.-R. Sadeghi, "Memristor PUFs: A new generation of memory-based physically unclonable functions," in *Proc. IEEE/ACM Des., Autom. Test in Eur. Conf. Exhib.*, 2013, pp. 428–431.

[30] A. Maiti, I. Kim, and P. Schaumont, "A robust physical unclonable function with enhanced challenge-response set," *IEEE Trans. Inf. Forens. Secur.*, vol. 7, no. 1, pp. 333–345, 2012.

[31] A. Iyengar, K. R. Ramclam, and S. Ghosh, "DWM-PUF: A low-overhead, memory-based security primitive," in *Proc. IEEE Int. Conf. Hardware Orient. Secur. Trust*, 2014, pp. 154–159.

[32] T. Marukame, T. Tanamoto, and Y. Mitani, "Extracting physically unclonable function from spin transfer switching characteristics in magnetic tunnel junctions," *IEEE Trans. Magn.*, vol. 50, no. 11, pp. 1–4, 2014.

[33] J. Das, K. Scott, D. Burgett, S. Rajaram, and S. Bhanja, "A novel geometry based MRAM PUF," in *Proc. IEEE Int. Conf. Nanotechnol.*, 2014, pp. 859–863.

[34] C. Dimitrakopoulos, D. Pfeiffer, and J. Smith, "Authentication using graphene based devices as physical unclonable functions," U.S. Patent App. 13/712 455, 2014. [Online]. Available: http://www.google.com/patents/US20140159040

[35] L. Zhang, Z. H. Kong, C.-H. Chang, A. Cabrini, and G. Torelli, "Exploiting process variations and programming sensitivity of phase change memory for reconfigurable physical unclonable functions," *IEEE Trans. Inf. Forens. Secur.*, vol. 9, no. 6, pp. 921–932, 2014.

[36] U. Ruhrmair *et al.*, "Security applications of diodes with unique current-voltage characteristics," *Finan. Cryptogr. Data Secur.*, vol. 6052, pp. 328–335, 2010.

[37] S. T. C. Konigsmark, L. K. Hwang, D. Chen, and M. D. F. Wong, "CNPUF: A carbon nanotube-based physically unclonable function for secure low-energy hardware design," in *Proc. IEEE/ACM Asia and South Pacific Des. Autom. Conf.*, 2015.

[38] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. IEEE/ACM Des. Autom. Conf.*, 2007, pp. 9–14.

[39] D. Lim *et al.*, "Extracting secret keys from integrated circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 10, pp. 1200–1205, 2005.

[40] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended Abstract: The Butterfly PUF Protecting IP on Every FPGA," in *Proc. IEEE Int. Workshop on Hardware-Orient. Secur. Trust*, 2008, pp. 67–70.

[41] R. Helinski, D. Acharyya, and J. Plusquellic, "A physical unclonable function defined using power distribution system equivalent resistance variations," in *Proc. IEEE/ACM Design Autom. Conf.*, 2009, pp. 676–681.

[42] A. Krishna, S. Narasimhan, X. Wang, and S. Bhunia, "MECCA: A robust low-overhead PUF using embedded memory array," *Proc. Cryptogr. Hardware Embedded Syst.*, vol. 6917, pp. 407–420, 2011.

[43] U. Ruhrmair, S. Devadas, and F. Koushanfar, "Security based on physical unclonability and disorder," *Introduct. Hardware Secur. Trust*, pp. 65–102, 2012.

[44] F. Armknecht, R. Maes, A. Sadeghi, O.-X. Standaert, and C. Wachsmann, "A formalization of the security features of physical functions," in *Proc. IEEE Symp. Secur. Privacy*, May 2011, pp. 397–412.

[45] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," *Towards Hardware-Intrinsic Secur.*, pp. 3–38, 2010.

[46] J. Rajendran, G. Rose, R. Karri, and M. Potkonjak, "Nano-PPUF: A memristor-based security primitive," in *Proc. IEEE Comput. Soc. Ann. Symp. VLSI*, 2012, pp. 84–87.

[47] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," *Inf. Hiding*, pp. 206–220, 2009.

[48] U. Ruhrmair *et al.*, "Towards electrical, integrated implementations of SIMPL systems," *Inf. Secur. Theory Practices. Secur. Priv. Pervas. Syst. Smart Dev.*, vol. 6033, pp. 277–292, 2010.

[49] I. Jensen and A. J. Guttmann, "Statistics of lattice animals (polyominoes) and polygons," *J. Phys. A: Math. Gener.*, pp. L257–L263, 2000.

[50] SynopsyshSpice. [Online]. Available: http://www.synopsys.com/tools/Verification/AMSVerification/CircuitSimulation/HSPICE/Pages/default.aspx

[51] A. Stothers, *On the Complexity of Matrix Multiplication*. [Online]. Available: http.maths.ed.ac.uk/pg/thesis/stothers.pdf

[52] Y.-H. Tseng, W. C. Shen, and C.-J. Lin, "Modeling of electron conduction in contact resistive random access memory devices as random telegraph noise," *J. Appl. Phys.*, vol. 111, pp. 073701-1–073701-5, 2012.

[53] U. Ruhrmair *et al.*, "Applications of high-capacity crossbar memories in cryptography," *IEEE Trans. Nanotechnol.*, vol. 10, no. 3, pp. 489–498, 2011.

[54] P. Kohlbrenner and K. Gaj, "An embedded true random number generator for FPGAs," in *Proc. ACM Int. Symp. Field Program. Gate Arrays*, 2004, pp. 71–78.

[55] D. Holcomb, W. Burleson, and K. Fu, "Power-Up SRAM State as an identifying fingerprint and source of true random numbers," *IEEE Trans. Computers*, vol. 58, no. 9, pp. 1198–1210, 2009.

[56] N. McDonald, "Al/CuxO/Cu memristive devices: Fabrication, characterization, and modeling," M.S. Thesis, College of Nanoscale Sci. Eng., Univ. Albany, Albany, NY, USA, 2012, vol. 1517153.

[57] L. Zhang, X. Fong, C.-H. Chang, Z. H. Kong, and K. Roy, "Highly reliable memory-based physical unclonable function using spin-transfer torque MRAM," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2014, pp. 2169–2172.

[58] L. Zhang, Z. H. Kong, and C.-H. Chang, "PCKGen: A Phase Change Memory based cryptographic key generator," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2013, pp. 1444–1447.

[59] X. Wang and Y. Chen, "Spintronic memristor devices and application," in *Proc. IEEE/ACM Des., Autom. Test in Eur. Conf. Exhib.*, 2010, pp. 667–672.

[60] Maxim Integrated, DS28CN01: 1 KbitI^2C/SMBus EEPROM With SHA-1 Engine. [Online]. Available: http://www.maximintegrated.com/datasheet/index.mvp/id/5369

[61] K.-H. Jo, C.-M. Jung, K.-S. Min, and S.-M. S. Kang, "Self-adaptive write circuit for low-power and variation-tolerant memristors," *IEEE Trans. Nanotechnol.*, vol. 9, no. 6, pp. 675–678, 2010.

[62] H. Manem, J. Rajendran, and G. S. Rose, "Design considerations for multilevel CMOS/Nano memristive memory," *J. Emerg. Technol. Comput. Syst.*, vol. 8, no. 1, pp. 6:1–6:22, 2012.

[63] Y. Ho, G. Huang, and P. Li, "Nonvolatile memristor memory: Device characteristics and design implications," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2009, pp. 485–490.

[64] R. Katti, J. Tucker, and A. Kohli, "Magnetoresistive random access memory (MRAM) package including a multilayer magnetic security structure," U.S. Patent 8 811 072, 2014.

[65] Y. Bi *et al.*, "Leveraging emerging technology for hardware security—Case Study on silicon nanowire FETs and graphene SymFETs," in *Proc. IEEE Asian Test Symp.*, 2014, pp. 342–347.

[66] D. Shahrjerdi, J. Rajendran, S. Garg, F. Koushanfar, and R. Karri, "Shielding and securing integrated circuits with sensors," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2014, pp. 170–174.

[67] M. Tehranipoor *et al.*, "Trustworthy hardware: Trojan detection and design-for-trust challenges," *Comput.*, vol. 44, no. 7, pp. 66–74, 2011.

[68] J. F. Tian, L. A. Jauregui, G. Lopez, H. Cao, and Y. P. Chen, "Ambipolar graphene field effect transistors by local metal side gates," *Appl. Phys. Lett.*, vol. 96, no. 26, p. 263110, 2010.

[69] J. Rajendran, H. Manem, R. Karri, and G. Rose, "An energy-efficient memristive threshold logic circuit," *IEEE Trans. Comput.*, vol. 61, pp. 474–487, 2012.

[70] F. Alibart, E. Zamanidoost, and D. B. Strukov, "Pattern classification by memristive crossbar circuits using *ex situ* and *in situ* training," *Nature Commun.*, vol. 4, 2013.

[71] H. Manem, J. Rajendran, and G. Rose, "Stochastic gradient descent inspired training technique for a CMOS/Nano memristive trainable threshold gate array," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 59, no. 5, pp. 1051–1060, 2012.

[72] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Dedham, MA, USA: Addison-Wesley, 2010.

[73] S. Kannan, N. Karimi, and O. Sinanoglu, "Secure memristor-based main memory," in *Proc. IEEE/ACM Design Autom. Conf.*, 2014, pp. 178:1–178:6.

[74] G. Khedkar and D. Kudithipudi, "RRAM motifs for mitigating differential power analysis attacks (DPA)," in *Proc. IEEE Comput. Soc. Ann. Symp. VLSI*, 2012, pp. 88–93.

[75] B. Cambou, *Match-In-Place: A Novel Way to Perform Secure and Fast Users Authentication*, 2014. [Online]. Available: http://www.crocus-technology.com/pdf/Crocus_MIP_White_Paper_v6.pdf

[76] The Embedded Security Challenge, 2014. [Online]. Available: http://isis.poly.edu/esc/

ABOUT THE AUTHORS

Jeyavijayan Rajendran (Student Member, IEEE) is a PhD Candidate in the Electrical and Computer Engineering Department, New York University, NY, USA.

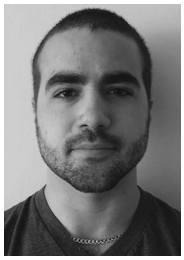
His research interests include hardware security and emerging technologies.

Mr. Rajendran has won three Student Paper Awards (ACM CCS 2013, IEEE DFTS 2013, IEEE VLSI Design 2012); four ACM Student Research Competition Awards (DAC 2012, ICCAD 2013, DAC 2014, and the Grand Finals 2013); Service Recognition Award from Intel; Third place at Kaspersky American Cup, 2011; and Myron M. Rosenthal Award for Best Academic Performance in M.S. from NYU, 2011. He organizes the annual Embedded Security Challenge, a red-team/blue-team hardware security competition. He is a student member of the Association for Computing Machinery (ACM).

Ramesh Karri received the Ph.D. degree in computer science from the University of California at San Diego, La Jolla, CA, USA.

He is a Professor of Electrical and Computer Engineering at the Polytechnic Institute of New York University, New York, NY, USA. His research interests include trustworthy ICs and processors; High assurance nanoscale IC architectures and systems; VLSI Design and Test; Interaction between security and reliability. He has over 150 journal and conference publications in these areas. He was the recipient of the Humboldt Fellowship and the National Science Foundation CAREER Award, and Best Student Paper Awards at ACM Computer and Communications Security 2013, IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems 2013, and IEEE VLSI Design conference 2011. He is the area director for cyber security of the NY State Center for Advanced Telecommunications Technologies at NYU-Poly; Hardware security lead of the Center for research in interdisciplinary studies in security and privacy-CRISSP (http://crissp.poly.edu/), co-founder of the Trust-Hub (http://trust-hub.org/) and organizer of the annual red team blue team event at NYU, the Embedded Systems Security Challenge (http://esc.isis.poly.edu). He also co-founded CRISSP and CRISSP-AD.

Dr. Karri co-founded and served as the chair of the IEEE Computer Society Technical Committee on Nanoscale architectures. He is a cofounder and steering committee member of the IEEE/ACM Symposium on Nanoscale Architectures (NANOARCH), Program Chair (2012) and General Chair (2013) of IEEE Symposium on Hardware Oriented Security and Trust (HOST), Program Co-Chair (2012) and General Co-Chair (2013) of IEEE Symposium on Defect and Fault Tolerant VLSI and Nanotechnology Systems and the General Chair of the 2013 NANOARCH. He serves on several program committees including, VTS, DAC, HOST and ICCD. He is the Associate Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF VLSI SYSTEMS, and *ACM Journal of Emerging Technologies in Computing*. He is an IEEE Computer Society Distinguished visitor and has organized invited tutorials on Trustworthy Hardware (including at 2012 VLSI Test Symposium, 2012 International Conference on Computer Design, 2013 IEEE North Atlantic Test Workshop, 2013 Design Automation and Test in Europe and 2013 International Test Conference and 2014 IEEE/ACM Design Automation Conference).



James B. Wendt (Student Member, IEEE) received the B.A. degree in physics from Pomona College in 2009, the M.S. degree in computer science from the University of California Los Angeles in 2011, where he is currently a Ph.D. candidate in Computer Science, under the advisement of Professor Miodrag Potkonjak.

His research interests include hardware security, emerging technologies, and ultralow power design.

Miodrag Potkonjak received the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, CA, USA, in 1991.

He is a Professor with the Computer Science Department at the University of California, Los Angeles. He created the first watermarking, fingerprinting, and metering techniques for integrated circuits as well as the first remote trusted sensing and trusted synthesis approaches, compilation using untrusted tools, and public physical unclonable functions.

Nathan McDonald received his B.S. in physics from the University at Albany, SUNY, in 2008 and his M.S. in nanoscale engineering from the College of Nanoscale Science Engineering, University at Albany, SUNY, in 2012. He is a researcher at the Air Force Research Laboratory, Information Directorate in Rome, NY. His areas of interest include hardware-based reservoir computing architectures for size, weight, and power constrained neuromorphic computing systems and copper oxide based memristive device fabrication and characterization for use in von Neumann and neuromorphic computing architectures.

Garrett S. Rose received the B.S. degree in computer engineering from Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA, USA, in 2001 and the M.S. and Ph.D. degrees in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in 2003 and 2006, respectively. His Ph.D. dissertation was on the topic of circuit design methodologies for molecular electronic circuits and computing architectures.

Presently, he is an Associate Professor with the Department of Electrical Engineering and Computer Science at the University of Tennessee, Knoxville. Prior to joining the University of Tennessee, he was with the Air Force Research Laboratory, Information Directorate, Rome, NY where his work was focused on research in the areas of trusted hardware, information security and nanoelectronic computing. From August 2006 to May 2011, he was an Assistant Professor in the Department of Electrical and Computer Engineering at the Polytechnic Institute of New York University, Brooklyn, NY. From May 2004 to August 2005 he was with the MITRE Corporation, McLean, VA, involved in the design and simulation of nanoscale circuits and systems. His research interests include low-power circuits, emerging computer architectures, hardware security, and developing VLSI design methodologies for novel nanoelectronic technologies.

Dr. Rose is a member of the Association of Computing Machinery, IEEE Circuits and Systems Society and IEEE Computer Society. He serves and has served on Technical Program Committees for several IEEE conferences (including ISCAS, GLSVLSI, NANOARCH) and workshops in the area of VLSI design. In 2010, he was a guest editor for a special issue of the *ACM Journal of Emerging Technologies in Computing Systems* that presents key papers from the IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH, 09). Since April 2014, he has been an associate editor for the IEEE Transactions on Nanotechnology.

Bryant Wysocki received the Ph.D. degree from Cornell University, Ithaca, NY, USA.

He is the Chief Engineer, Information Directorate, Air Force Research Laboratory, Rome, N.Y. The Information Directorate is a center for the advancement and application of information systems science and technology. His recent work examines the nonlinear dynamics and delayed feedback (short term memory) of hardware-based reservoir computing methods as applied to process perception, prediction, and control. He is actively involved in community STEM outreach.