Nanoelectronic Solutions for Hardware Security

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

***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 based security solutions overcome many of the limitations of classical security while consuming less energy and improving performance. Nanoelectronics based hardware security preserves all of these advantages while enabling conceptually new security mechanisms and security applications. This paper highlights nanoelectronics based security capabilities and challenges. The paper describes nanoelectronics based hardware security primitives for device identification, digital forensics, and tamper detection. These primitives can be developed using the unique characteristics of emerging nanoelectronic devices such as complex device and system models, bidirectional operation, and temporal drift of the state variable. We conclude by identifying important desiderata and outstanding challenges in nanoelectronics based security.**

***Index Terms*—Digital integrated circuits, hardware security, nanoelectronics, and memristors.**

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 forty-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.

Classical security (i.e., mathematical or algorithmic) has created elegant security primitives and protocols. Unfortunately, these solutions are not only slow and consume more energy for most modern applications but are also vulnerable to physical and side channel attacks (e.g., radiation or exposure to high temperature). Classical and emerging security requirements and metrics may be addressed in superior ways using nanoelectronics.

Nanoelectronics enable conceptually new and strong security primitives and applications. Nanoelectronic security primitives create intrinsic feedback mechanisms that provide security superior to that offered by Shannon's diffusion and confusion principles while subsuming these two fundamental principles as special cases. Nanoelectronic security primitives are potentially more robust than conventional complementary metal oxide semiconductor (CMOS) device-based security primitives. They can be the basis for provable security in an information theoretic sense since the complexity of compromising a nanoelectronics based security primitive is equivalent to the hard problem of solving a large system of nonlinear equations. Finally, emerging, unconventional nanoelectronics have the potential to yield computing systems with miniscule form factors, ultra low-power consumption, and fast computation times relative to CMOS technologies.

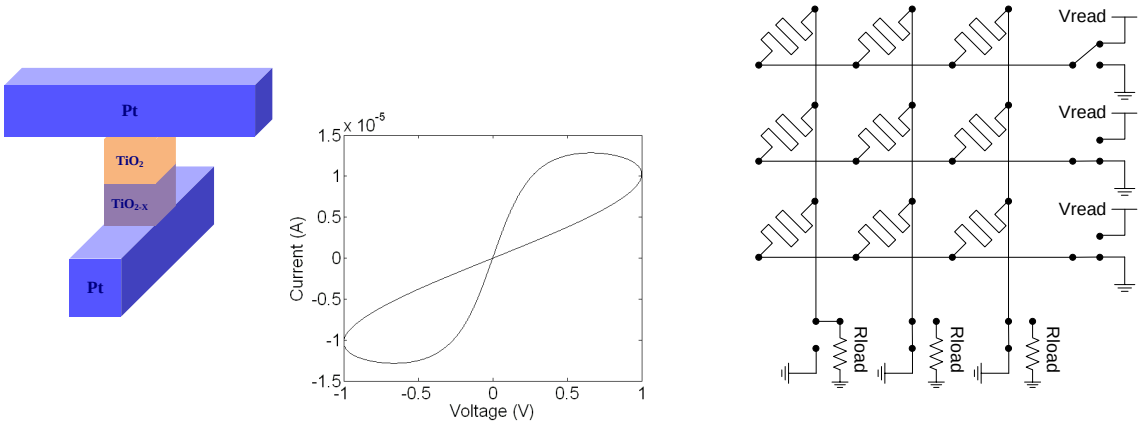
A variety of materials and devices including memristors, graphene, plasmonics, and quantum dots are being investigated for use in nanoelectronics. These nanoelectronic devices have non-linear input-output relationship, exhibit inherent process variations much like current CMOS technologies [1-5] while demonstrating technology specific characteristics.

Our objective is to quantitatively and qualitatively explain the security relevant capabilities of one such nanoelectronics technology namely, memristor. We will explain why the non-linear, bidirectional input-output characteristics of these two terminal devices [1-5], and their inherent non-volatility, combined with temporal drift, and unique device forming step are interesting from a security perspective. We will introduce memristor-based security primitives for device identification, digital forensics, and tamper detection by using these unique characteristics. Finally, we will summarize outstanding challenges. Overall, we expect to convey our vision of security, digital forensics, and tamper detection as important applications for nanoelectronics.

II. Nanoelectronics and Nanoarchitectures

In recent years, device physicists have realized a wide variety of nanoelectronic devices. These include metal-oxide memristors, phase change devices, spin-torque transfer devices, carbon nanotubes, graphene, and quantum-dots. We will show how security primitives can be built mainly using metal-oxide memristors by using some of their unique characteristics.

Received and cleared for public release by AFRL on September 14, 2012, case number 88ABW-2012-5023. 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.



(a) (b) (c)

Fig. [1]. (a) Metal-insulator-metal (MIM) memristor structure [2], (b) Current-voltage characteristics of a bipolar memristor (Source: [15]), (c) A 3X3 memristor-based memory that can function as a memory.

A. *Memristors*

1) *Theory*

Leon Chua showed that memory resistance or memristance *M(q)* relates charge q and flux φ and memristance of the device changes with the applied electric field and time [1]:

$$M(q) = \frac{\mathrm{d}\varphi(q)}{\mathrm{d}q}. \quad (1)$$

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*

Memristive devices have been fabricated from a variety of materials. For example, a TiO_{2-X} layer with oxygen vacancies is placed on a TiO_2 layer without oxygen vacancies, and these two layers are sandwiched between metallic (platinum) electrodes as shown in Fig. [1](a) [2]. In another structure an insulator is sandwiched between two metal layers (Metal-insulator-metal or MIM), where the insulating layer may be a variety of materials including chalcogenides [6,7], metal oxides [8,9], perovskites [10,11], or organic films [12,13].

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 (the 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 via a RESET operation, by applying a lower voltage, V_{RESET}. Additional resistance states are attainable by limiting the applied voltage or current.

MIM memristors afford 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. Nonpolar memristors demonstrate both bipolar and unipolar switching styles.

4) *Simulation models*

Memristor models for metal oxide and other types have been developed based on their device physics [14]. When the ratio of HRS to LRS is very high, the relation between the flux $\varphi(t)$ (time integral of the applied voltage difference across the device, $\int_{-\infty}^{t} V(\tau)d\tau$ at time t) and the memristance of the device, $M(\varphi(t))$, can be written as [14]:

$$M\big(\varphi(t)\big) = \frac{HRS}{D}\sqrt{\left(D^2 - 2\eta\frac{LRS}{HRS}\varphi(t)\mu\right)}, \quad (2)$$

where L, W and D are the length, width, thickness of the devices, respectively. η is ± 1 depending on the polarity of the applied voltage and μ is mobility of the dopants.

5) *Characteristics*

Metal-oxide memristor devices have unique characteristics many of which are not typically found in traditional CMOS devices. We list the characteristics that are leveraged for security[1].

1. Non-volatility: Memristors retain their memristance value even when the power is turned OFF.

2. Bi-directionality: Some bipolar memristors exhibit similar current-voltage characteristics irrespective of the polarity of the applied voltage or current. This is evident from the symmetric I-V curve in Fig. [1](b).

3. Non-linearity: The I-V characteristics of memristors are highly non-linear due to their time-dependent behavior, as shown in Equation 2. Also, the HRS to LRS ratio is typically on the order of 10^3-10^6.

4. Formation process: In most memristor types, when the device is first fabricated it will not switch when the V_{SET} and V_{RESET} voltages are applied. Rather it behaves like a linear resistor [15]. Applying a large formation voltage, Vf, which will force the memristor to its LRS, initializes such a device. After this formation step, the device starts behaving as a memristor.

5. Memristance drift: On applying an input (positive or negative) voltage across certain metal-oxide memristors,

[1] All types of memristors do not possess all of these characteristics. The specific characteristics exhibited by a memristor depend on the material used.

the memristance changes because of the movement of dopants called memristance drift [2]. The amount of drift depends on the polarity, amplitude, and duration of the applied voltage. Consider a 50nm × 50nm × 50nm memristor in [2]. Also assume the HRS and LRS memristance values to be 121MΩ and 121KΩ respectively, and the dopant mobility to be 10^{-14} m/V·s. According to Equation 2, it takes 0.82s for this memristor to drift from 10MΩ to 1MΩ when 1V is applied across the device.

6. Process variations: According to Equation 2 the memristance of a memristor is affected by process-variation induced changes in its dimensions and dopant concentration. Furthermore, the effect of variation in thickness of the memristor on its memristance value is highly non-linear (this effect is more evident in the LRS than HRS [16]).

7. Radiation-hardness: Some memristor devices are inherently rad-hard due to the material properties [2].

All of these characteristics with the exception of non-volatility and radiation-hardness pose problems when designing memory and logic circuits using a metal oxide memristor. However, we show that these problematic characteristics can be useful in the context of security.

III. Nanoelectronics-based Tamper Detection and Forensics

Memristors can be used for tamper detection and digital forensic analysis. Tamper detection entails identifying unauthorized usage of or access to the target hardware. Digital forensics entails recovering data from the hardware. We will show how the device formation, non-volatility, and the memristance drift characteristics can be combined for tamper detection and forensics.

A. Manufacture-time Tamper Detection for Trust Verification

The memristive device formation step enables one to differentiate a virgin (i.e. unformed) device from one that has been used (i.e. formed). Consider a system where a memristor or group of memristors is required for a particular circuit to function. For example, a group of memristors could be part of a circuit used to encrypt data in a secure microprocessor. Any user (authorized or unauthorized) would need to 'form' these devices before gleaning any useful information. A "trust but verify" technique can be used to obtain some information as to whether or not a circuit has been tampered with. Such a technique is useful to verify the trustworthiness of the new integrated circuits received from an untrustworthy fabrication facility. The technique would work by first writing a known value to the memristor(s), reading that value back, writing the complement of the known value to the memristor(s), reading the next value back, and comparing the results. If the formation step had not occurred, then 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 were different. This second case could be an evidence of possible tampering of the circuit.

B. Run-time Tamper Detection in Memristor-based Memories

1) Memristor-based crossbar memories

An N×N memristor crossbar consists of two sets of N wires running orthogonal to each other, where a memristor is grown at the cross points as shown in Fig. 1 (c). 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 cross point of that input and output. In a sneak path, the current flowing from an input to an output is a function of resistance of devices at other cross points in the crossbar.

Such memristor-based crossbars have been used to build non-volatile memories. In these memories, the HRS and LRS are used to represent logic '0' and '1', respectively [17,18]. Let us now look at the memory write and read operations in these crossbar memories.

Write operation: To write into 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 '1' is read, otherwise a '0' is read. In devices where the memristance drifts, 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 [19]. For a bipolar memristor, the ideal read pattern uses a positive pulse immediately followed by a negative pulse of the same magnitude and duration, creating a zero net change in memristance.

2) Tamper detection

Unauthorized memory reads in memristor-based memories can be detected as follows [17]. Consider the memristor based crossbar memory shown in Fig. 1(c). In this memory, the HRS and LRS are used to represent logic '0' and '1', respectively. 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 (by applying a read pulse of opposite polarity with the same magnitude and for the same duration) 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. 2 [17]. The modified memory read operation uses two consecutive read pulses. While the magnitude and duration

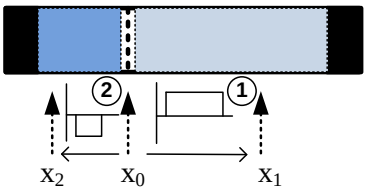


Fig. 2 The dark and lightly shaded regions represent the high-resistive and low-resistive regions, respectively. The dotted line represnts the location of the domain wall which determines the current resistant value. Every read operation uses two pulses. The magnitude and duration of the first pulse are public, whereas the magnitude and duration of the second pulse are known only to the defender.

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 the authorized user. Thus, even though an attacker can restore the memristance value to its initial value using a pulse of opposite polarity, he/she cannot revert back the change in resistance 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 and cannot be undone. The authorized user can detect this change in memristance and consequently the tampering.

C. Forensics in Memristor-based Threshold Logics

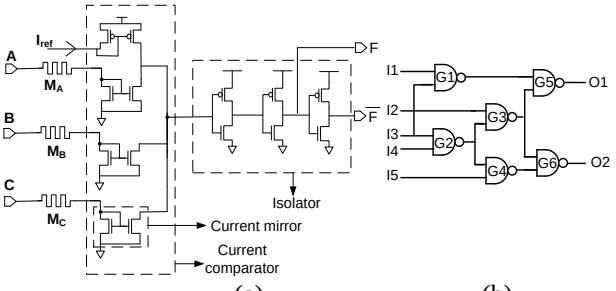
Memristor based digital logic gates have been proposed [20]. We show that such gates can be useful from a forensics perspective.

1) Preliminaries: Memristor-based threshold gates

Consider the memristor-based threshold logic (MTL) gates proposed in [20]. In an MTL gate, the memristors are used as weights on the inputs of the gate. Fig. 3(a) shows a 3-input threshold gate which uses three 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 summed current is greater than I_{ref} then the output is logic '1,' else the output is logic '0'. A positive voltage denotes logic '1' and a 0V denotes logic '0'.

2) Effect of memristance drift on MTL gates

As discussed previously, when logic '1' (positive voltage) is applied to an input, the memristance value of the corresponding memristor drifts. On the other hand, there will not be any drift 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 2MΩ. Let the amplitude and duration of input pulses be 1.1V and 2ns, respectively. Let us apply one million, two million and three million 1's to inputs A, B, and C, respectively. Now the final memristance values of memristors M_A, M_B and M_C will be 2.12MΩ, 2.25MΩ, and 2.38MΩ, respectively. These changes in memristance values are caused by memristance drift.



(a) (b)

Fig. 3. (a) A 3-input memristor-based threshold logic gate (MTL) [20], (b) C17, an ISCAS85 benchmark circuit.

3) Forensic analysis

Conversely, if one determines that the final memristance values of memristors M_A, M_B and M_C are 2.12MΩ, 2.25MΩ, and 2.38MΩ, respectively, then forensic analysis can estimate that about one million, two million, and three million 1's have been applied to the inputs A, B, and C, respectively.

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 memristance change at the inputs of different gates will be different. Consider the C17 circuit, one of the ISCAS'85 benchmark circuits, shown in Fig. 3(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 is '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'[2].

D. Advantages over Forensics in CMOS-based Designs

Forensic analysis of CMOS-based designs has not been explored to the best knowledge of the authors. However, similar to memristance drift in memristor, one can leverage the Negative Bias Temperature Instability (NBTI) effect in CMOS for forensics. NBTI occurs in a CMOS transistor when electron traps are formed at the silicon-silicon dioxide interface. NBTI effect in PMOS is more dominant than in NMOS. Applying logic '1' to the PMOS transistor subjects it to NBTI stress which then degrades the

[2] 'd' represents a don't care value.

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. Unfortunately, the rate of change in transistor delay 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.

E. Outstanding Challenges

To perform tamper detection in memristors and memristors-based memories, and 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. In the case of tamper detection in memristor-based memories, the amplitude and duration of the second pulse have to be adjusted so that the read and write margins of the devices are honored. Similarly, in the case of forensic analysis on MTL gates, 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 non-functional. An authorized user has to once again restore 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.

IV. NANOELECTRONICS SECURITY TOKENS

A. Memristor-based Random Number Generator

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

Certain memristors can be used to generate random numbers as shown in Fig 4 [4]. In these devices, trapped electrons in the insulation layer will randomly impact the current flowing through the filament channel. Upon applying a high voltage (3 V for the memristor device in [4]), 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 filament shrinks down; and the current flowing through the filament is strongly influenced by the trapped electrons. The output current will now be a strong function of the trapped electrons. Since the electrons are trapped in a random number, the output current will also be random. This method shows a great promise as it successfully passes several randomness tests designed by National Institute of Science and Technology (NIST).

B. Memristor-based Unique Signatures (MUS)

Certain memristors can be used to generate a unique signature for hardware [21] by exploiting two different

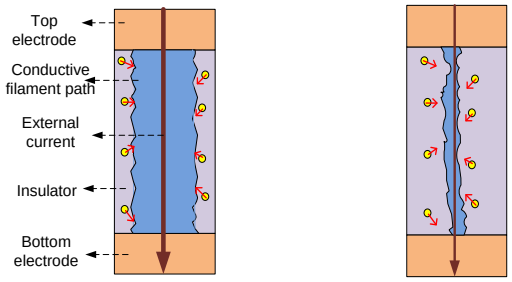


Figure 4 (a) Conductive filament formation on applying a high voltage. Trapped electrons have negligible impact on the current flowing through the filament. (b) Conducive filament on applying a low voltage. Trapped electrons have a larger impact on the current flowing through the filament.

characteristics of memristors: 1) inherent, non-uniform, irreproducible process variations during fabrication and 2) requirement of "forming" to make them functional. [21] uses nonpolar memristors in series of pairs as random bit generators, where the bit generation is a function of the location of a low resistance filament. Multiple instances of such random bit generators can produce a random word, which can be used for a unique signature for that hardware. Unlike the random number generator described in the previous section, this signature is non-volatile and thus may be used for hardware identification purposes. Such hardware IDs can be used to detect electronic counterfeits.

1) Architecture

Consider a pair of memristors in series as shown in Fig. 5. 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) Protocol

During the forming step, one TE is biased while the other TE is grounded. Two low resistance filaments are formed, one beneath each TE, through the insulator material layer. Then during the RESET operation, the resistance values of the two series memristors are returned to the HRS. During this switch, only one of the low resistance filaments becomes highly resistive; the other filament remains of low resistivity. The location of this latter filament serves as the random bit value. This location depends on the process-induced variations on insulator layer thickness and dopant concentration. The low resistance filament location 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, an attacker in the manufacturing unit could not read this unique signature and spoof it.

C. Advantages over CMOS-based Random Number Generators

In CMOS technology, the inherent process-induced variations in the Field Programmable Gate Arrays were leveraged to generate random numbers [22]. Circuits such as ring oscillators were used for randomness extraction.

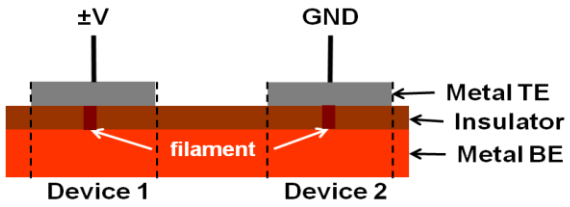


Fig 5. Electrical configuration for random bit generation

However, the extracted random values are unstable as the frequency of the ring oscillators is strongly dependent on temperature.

A unique device signature in CMOS can also be derived from an unwritten Static Random Access Memory (SRAM) circuit. 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. However, an attacker in the manufacturing unit can easily read-out this unique signature and use it to spoof the hardware. Unlike with the MUS, this tampering is not irrefutable.

Similar to generating random signatures using memristor-based crossbars, diode-based crossbars were also used to generate random signatures [23]. The randomness arises from the variation in the amount of dopants in the diodes. Unfortunately, an attacker may still read-out the signature from the diodes, since they do not require a forming process.

Nanoelectronic devices whose operations are inherently invariant with temperature can be used to generate random numbers. Recently, Contact Resistive Random Access Memory (CRRAM), whose characteristics are stable over a range of temperatures (0-150 ˚C), was used to generate random numbers [4,33].

D. Challenges

In the case of MUS, initial analysis indicates that the location of the persistent low resistance filament is random; however, the generated values must still be tested against the NIST randomness test suite.

V. NANOELECTRONICS-BASED PUBLIC PHYSICAL UNCLONABLE FUNCTIONS (NANOPPUFS)

A. Preliminaries

Physical Unclonable Functions: Random unclonable physical disorders in the IC fabrication process may be leveraged to produce unique responses (outputs) upon the application of challenges (inputs) [25]. A special circuit called the Physical Unclonable Function (PUF) is used for this purpose. A PUF is a hardware token that maps a challenge to a response is the secret. PUFs have been used for secure software execution on a processor [26], for device authentication, for trusted configuration of FPGAs [28], and for encrypted storage [26]. However, they cannot be used in advanced two-party cryptographic protocols such as time stamping and bit commitment as PUFs require one of the parties to store the challenge-response pairs.

Public Physical Unclonable Functions (PPUFs): PPUFs are an extension of PUF whose simulation models are made public [29, 31]. Although an attacker can simulate the PUF on a given challenge to obtain a response, the simulation time is too large (several years) compared to the time it takes to apply a challenge and obtain its response on the PUF token (a few nanoseconds).

Memristive nanoelectronic devices are ideal for implementing PPUFs (the NanoPPUF) [24]. A NanoPPUF can implement two-party security protocols such as authentication, key exchange, bit commitment, and time stamping. The NanoPPUF exploits important characteristics of memristors such as process variations, bi-directionality, crossbars, complexity of simulation models of memristors and memristor crossbars.

B. Architecture: Polyominoes and crossbars

NanoPPUFs [24] leverage the complex simulation models of memristor crossbars with special geometric structures called polyominoes that are embedded within large memristor crossbars.

Complexity simulation models of memristor-crossbars: Simulation of a memristor crossbar is computationally expensive when the inputs are applied to all the rows and the outputs are observed at all the columns. This operation is different from the one in memory crossbar where only one row is selected at a time. The simulation complexity of a memristor crossbar can be traced to the non-linearity and bi-directionality of the devices at the crosspoints, and the exponential number of sneak paths in the crossbar [24].

Polyominoes in a memristive crossbar: A polyomino is a geometrical structure formed by connecting a number of individual blocks. An *M*-omino is formed by connecting *M* blocks. 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* resistive devices in a crossbar is $\frac{c\lambda^M}{M}\times N$, where λ and c are 4.0626 and 0.3169, respectively [30].

We will outline some important two-party security protocols enabled by a NanoPPUF. Alice and Bob are represented by A and B, and their NanoPPUFs are denoted as $PPUF_A$ and $PPUF_B$, respectively. 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 the NanoPPUF crossbar. The challenge set X is the list of pins where the challenge vector C is applied; the length of C and X are equal.

C. Protocols

1. User Authentication

NanoPPUF-based authentication protocol can prevent adversarial spoofing and identity theft. Assume that Alice wishes to authenticate that she is indeed conversing with Bob and not a malicious adversary pretending to be Bob.

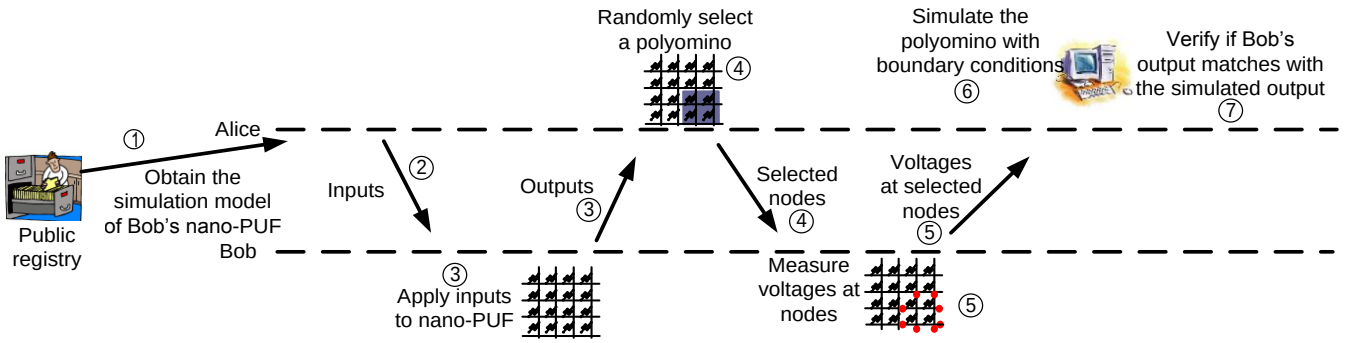


Fig. 6: Protocol for time-bounded authentication using NanoPPUF.

Alice issues a challenge C to Bob. Bob applies the challenge to his physical NanoPPUF, $PPUF_B$ and returns the response $C \rightarrow R$, to Alice. Given this challenge-response pair, Alice can validate the authenticity of Bob. Fig. 6 shows a time-bounded authentication using nano-PPUF. Due to the exponentially large number of polyominoes in the NanoPPUF and the bi-directionality of the NanoPPUF, Alice can simulate selected polyominoes and validate the inputs and outputs along the boundaries of the selected polyomino using node voltage analysis.

An adversary masquerading as Bob has to respond to Alice with a full output response, R, and he cannot guess the polyomino that Alice will choose. 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 $PPUF_B$. The adversary has near-zero probability of randomly guessing the chosen polyomino. For additional security, Alice could use two or more polyominoes or even request responses from Bob to two or more challenges. The computational cost to Bob is negligible while an adversary would have to simulate the two challenges.

2. Remote Secret Key Exchange

This protocol, also known as public key communication, allows Alice and Bob to securely communicate by encrypting their messages with a secret key. A polyomino in the NanoPPUF can be used as the secret key.

When Alice decides to send a message to Bob, she simulates a secret key,C_B, on a polyomino of $PPUF_B$ and calculates R_B. Bob receives a copy of R_B, the encrypted message, $M = C_B \oplus m$, and the input set X, on which the challenge was applied. Bob discovers the secret key to decrypt this message since he owns the NanoPPUF that originally output R_B. He accomplishes this by iterating through all possible combinations of inputs on the given input set until the challenge,C, is found such that $C \rightarrow R_B$.

Protocol 1. Authentication

1: A sends challenge to B
2: B applies challenge to $PPUF_B$ and records response
3: B sends to A
4: A picks a random polyomino of $PPUF_B$
5: A simulates the boundary conditions (and) on to check that node voltage analysis converges correctly
6: B is authenticated

Protocol 2. Remote Secret Key Exchange

1: A simulates secret challenge on $PPUF_B$, recording response (is the secret key)
2: A chooses message *m*, computes
3: A sends , , and challenge set to B
4: B iterates through challenges on until is found
5: B computes

An eavesdropper is unable to find the secret key since C_B can only be calculated from M, R_B and X by iterating over an exponential number of possible input vectors on X.

We can make this guessing C_B even less successful by sending a set of input sets, X', which contains X and an excess of random input sets to increase the search space for the attacker. This has a minimal effect on the NanoPPUF owner since applying a set of inputs to Nano-PPUF is not time consuming when compared to simulating it.

3. Commitment Scheme

The commitment scheme enables one party to commit to a hidden value and reveal that value to a remote party at a later date without having the opportunity of changing that value in between the time it was committed and the time it was revealed [34]**Error! Reference source not found.**. An application of the commitment scheme is a fair coin flipping game played over a remote connection [27]. Bob has the coin and Alice chooses heads or tails. If Bob reveals the value of the coin flip before Alice tells Bob whether she chose heads or tails, Alice can change her answer to win the game. Conversely, if Alice reveals her decision before Bob reveals the coin flip, Bob can lie about the outcome of coin flip. The key to the commitment scheme is that both values, the coin flip and the choice, are revealed at the same time to both parties, so that no party gains an advantage over the other.

NanoPPUFs can be used for coin flipping as follows. Bob applies two challenges to his NanoPPUF and sends the corresponding responses to Alice. Alice does not know which challenges Bob used to compute these responses. She chooses one of these responses and informs Bob of her decision. Then, Bob informs Alice of the corresponding challenge, C_i and C_j. If the challenge corresponding to the response that Alice chose is larger than the other then

Protocol 3. Coin Flipping

1: A sends range of challenges to B
2: B applies two C's at random to PPUF_B, $C_i \rightarrow R_i$ and $C_j \rightarrow R_j$
3: B sends R_i and R_j to A
4: A commits to either R_i or R_j and informs B
5: B informs A of $C \rightarrow R$ mappings
6: A confirms $C_i \rightarrow R_i$ and $C_j \rightarrow R_j$ using polyomino partitioning
7: If $C_i > C_j$, then R_i wins, otherwise, R_j wins

Alice wins the coin toss. Alice can be sure that Bob sent the correct challenges by selecting polyominoes, much like in Protocol 1, to validate $C_i \rightarrow R_i$ and $C_j \rightarrow R_j$.

4. Time Stamping

Time stamping is useful to date a digital document, whether it is for creation, deletion, or alteration, and to lock documents or digital media until a specific time. Consider a movie that is to be released on a specific date and time. If a subscriber wishes to watch this video in high definition on a mobile device at exactly the time it is released, the video must be pre-fetched or at least buffered some time before the official release date. This is a potential security risk to the distributor if the pre-fetched video is accessible before the release date.

The time stamping protocol prevents this by encrypting the video using a key, $R = T \oplus M$, where T is the timestamp that the movie is to be unlocked and M is some secret. One can ensure that the user cannot tamper with the device in order to apply a phony timestamp on the NanoPPUF by integrating the NanoPPUF circuitry into the Global Positioning Systems (GPS). Assuming that the GPS is secure, any tampering of the device in order to spoof the timestamp or access the NanoPPUF input pins directly would damage the circuitry (changing the resistances and resulting in a different NanoPPUF).

D. Advantages over CMOS-based PPUFs

SIMulation Possible but Laborious (SIMPL) systems were proposed for time-bounded authentication [31]. SIMPL systems were constructed using Cellular Non-linear Networks and Static Random Access Memory (SRAM) cells [31]. However, the time difference between the execution of an input on a SIMPL token and simulation of a SIMPL model is polynomial. 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 [32].

Another CMOS-based PPUF uses XOR networks [29]. This PPUF also assumes that the simulation of the entire PPUF circuit is computationally impossible. However, their simulation model is simpler than that for nanoelectronic devices for a variety of reasons including because they are unidirectional.

E. Challenges

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

Modeling errors: NanoPPUFs require accurate modeling of all the devices, 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 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 I-V characteristics of nanoelectronic devices vary with temperature. Thus, the outputs of crossbars built with these nanoelectronic devices will also vary with temperature and result in uncharacteristic outputs. Device physicists have demonstrated significant progress in fabricating nanoelectronic devices that are stable over a range of temperatures. For instance, [33] has built a memristor that exhibits a stable operation over temperatures from 0-150˚C.

Reduced order simulation of crossbars/cubes 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 of model of the device, for instance 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 attacker is now required to spend only a little computational effort.

Choice of viable inputs: Not all input values are permissible for the NanoPPUF; only inputs that cause enough circuit activities to make the simulation complex and the outputs unpredictable can be used. For instance, an input comprising of all 0's should not be used as it does not causing any switching of the devices and it produces an output of all 0's. Techniques to select viable inputs that retain the crypto properties have to be developed.

VI. CONCLUSIONS

We highlighted the important characteristics of memristors and demonstrated how they can be used to build new security primitives. These characteristics are based on experimental and theoretical device research. Characteristics of other nanoelectronic devices such as spintronics, phase change materials, graphene, and quantum dots remain to be explored for their applications in security. Another important direction is for device physicists to engineer nanoelectronic devices not only for memory and logic applications but also for security applications. Security researchers should develop new security primitives, protocols, and associated mathematical proofs by abstracting the detailed characteristics of nanoelectronic devices. Circuit designers are the bridge between device engineers and security researchers and design security circuits that harness these devices characteristics to satisfy mathematical strengths.

REFERENCES

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

[2] *D. B. Strukov*, G. S. Snider, D. R. Stewart and R. S. Williams "How we found the Missing Memristor," Nature, vol. 453, pp. 80–83, 2008.

[3] Y.T. Chiu, "A Memristor True Random-Number Generator", *IEEE Spectrum*, 2012.

[4] C.Y. Huang, W.C. Shen, Y.H Tseng, Y.C. King, and C.J. Lin, "A Contact-Resistive Random-Access-Memory-Based True Random Number Generator," *IEEE Electron Device Letters*, vol.33, no.8, pp.1108-1110, 2012

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

[6] Oblea, A.S.; Timilsina, A.; Moore, D.; Campbell, K.A., "Silver chalcogenide based memristor devices," *International Joint Conference on Neural Networks (IJCNN)*, pp.1-3, 2010

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

[8] L. Goux, J. G. Lisoni, M. Jurczak, D. J. Wouters, L. Courtade, and Ch. Muller, "Coexistence of the bipolar and unipolar resistive-switching modes in NiO cells made by thermal oxidation of Ni layers," J*ournal of Applied Physics*, vol. 107, no. 2, pp. 024512 - 024512-7, 2010.

[9] B.D. Briggs, S.M. Bishop, K.D. Leedy, B. Butcher, R. L. Moore, S. W. Novak and N.C. Cady, "Influence of Copper on the Switching Properties of Hafnium Oxide-Based Resistive Memory," *MRS Proceedings*, vol. 1337, 2011

[10] A. Sawa, T. Fujii, M. Kawasaki, and Y. Tokura, "Interfaces resistance switching at a few nanometer thick perovskite manganite layers," *Applied Physics Letters*, vol. 88, no. 23 pp. 232112 - 232112-3, 2006.

[11] K. Szot, W. Speier, G. Bihlmayer, and R. Waser, "Switching the electrical resistance of individual dislocations in single crystalline $SrTiO_3$," *Nature Materials*, vol. 5, pp. 312–320, 2006

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

[13] N. B. Zhitenev, A. Sidorenko, D. M. Tennant, and R. A. Cirelli, "Chemical modification of the electronic conducting states in polymer nanodevices," *Nature Nanotechonlogy*, vol. 2, pp. 237–242, 2007.

[14] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: properties of basic electrical circuits," *European Journal of Physics*, vol. 30, no. 4, pp. 661–675, 2009

[15] Q. Xia, et. al. "Memristor−CMOS Hybrid Integrated Circuits for Reconfigurable Logic", *Nano Letters*, vol. 9, no. 10, 2009

[16] J. Rajendran, H. Manem, R. Karri and G.S. Rose, "Approach to Tolerate Process Related Variations inMemristor-Based Applications," Intl Conf. on VLSI Design, pp. 18–23, 2011.

[17] Xiaobin Wang; Yiran Chen;, "Spintronic memristor devices and application," *in the Proc. of the IEEE Intl. Conf. on Design, Automation and Test in Europe*, pp. 667-672, 2010.

[18] H. Manem, J. Rajendran, G. S. Rose, "Design Considerations for Multilevel CMOS/Nano Memristive Memory", *ACM Journal of Emerging Technologies in Computing*, vol. 8, no. 1, pp. 1-22, 2012.

[19] Y. Ho, G. M. Huang, and P. Li, "Nonvolatile memristor memory: device characteristics and design implications," *in the Proc. of IEEE/ACM Intl. Conf. on Computer-Aided Design*, pp. 485 - 490, 2009

[20] J. Rajendran, H. Manem, R. Karri, G. S. Rose, "An Energy-Efficient Memristive Threshold Logic Circuit", *IEEE Transactions on Computers*, vol. 61, no. 4, pp. 474-487, 2012.

[21] N. R. McDonald, *Al/Cu_xO/Cu Memristive Devices: Fabrication, Characterization, and Modeling* (Unpublished master's thesis). University at Albany, SUNY, College of Nanoscale Science and Engineering, Albany, NY

[22] P. Kohlbrenner and K. Gaj. "An embedded true random number generator for FPGAs", *in the Proc. of the ACM Intl. Conf. Field programmable gate array*, pp. 71-78, 2004.

[23] U. Rührmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, and G. Csaba, "Applications of High-Capacity Crossbar Memories in Cryptography," *IEEE Transactions on Nanotechnology*, vol.10, no.3, pp.489-498, 2011.

[24] J. Rajendran, G. S. Rose, R. Karri, and M. Potkonjak, "Nano-PPUF: A Memristor-based Security Primitive", *in the Proc. of IEEE Intl. Conf. on VLSI*, pp. 84-87, 2012.

[25] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," *in the Proc. of the ACM Intl. Conf. on Computer and Communications Security*, pp. 148–160, 2002.

[26] G. E. Suh, C. W. O'Donnell, I. Sachdev, and S. Devadas, "Design and implementation of the AEGIS single-chip secure processor using physical random functions," *in the Proc. of IEEE/ACM Intl. Conf. on Computer Architecture*, pp. 25–36, May 2005.

[27] M. Blum, "Coin flipping by telephone a protocol for solving impossible problems," Proceedings of CRYPTO, pp. 11-15, 1981.

[28] J. Guajardo, S. Kumar, G.-J. Schrijen, and P. Tuyls,"Physical unclonable functions and public-key crypto for FPGA IP protection," *in the Proc. of the IEEE Intl. Conf. on Field Programmable Logic and Applications*, pp. 189–195, 2007.

[29] N. Beckmann and M. Potkonjak, "Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions", *in the Proc. of Intl. Workshop on Information Hiding*, pp. 206-220, 2009

[30] I. Jensen and A. J. Guttmann, "Statistics of lattice animals (polyominoes) and polygons," *Journal of Physics A: Mathematical and General*, vol. 33, pp. L257–L263, 2000.

[31] U. Rührmair, Q. Chen, P. Lugli, U. Schlichtmann, M. Stutzmann, and G. Csaba, "Towards Electrical, Integrated Implementations of SIMPL Systems", *Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices, Lecture Notes in Computer Science*, vol. 6033, pp. 177-292, 2009.

[32] Marten van Dijk and Ulrich Rührmair, "Physical Unclonable Functions in Cryptographic Protocols: Security Proofs and Impossibility Results," *IACR Cryptology ePrint Archive 2012:228*, 2012, [Online] Available: eprint.iacr.org/2012/228.pdf

[33] Y. H. Tseng, W. C. Shen, and C. J. Lin, "Modeling of electron conduction in contact RRAM (CR-RAM) devices as random telegraph noise," *Journal of Applied Physics*, vol. 111, no. 7, pp. 073701 - 073701-5, 2012.

[34] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," Journal of Cryptology, vol. 3, no. 2, pp. 99-111, 1991.