Md Tanvir Arafin & Gang Qu

Memristor-based Security

Contents

List of Figures

List of Tables

Chapter 1

Memristor-based Security

Md Tanvir Arafin

University of Maryland, College Park

Gang Qu

University of Maryland, College Park

CONTENTS

1.1 Introduction

In this chapter, we discuss memristors and memristor-based hardware security primitives. A memristor is a two-terminal non-volatile memory component. The word '*memristor*' is coined from the words '*memory*' and '*resistor*' by circuit theorist Leon Chua [8]. In 1971, Chua first predicted the existence of the fourth basic circuit element that fundamentally relates electric charge(q) and flux-linkage(ϕ) [8]. Electrical properties of this *missing circuit element* was explored in detail by Chua and Kang [13]. However, the physical realization of memristor remained elusive.

In 2008, researchers from Hewlett-Packard(HP) Labs announced the *discovery* of memristors [34]. Their analysis on Titanium Dioxide($Ti0_2$) thin-films in metal/oxide/metal (MIM) cross-point nano-devices revealed memristive properties similar to the ones predicted by Chua [34]. This discovery leads to a renewed interest in memristive circuit and system. Numerous implementations of thin-film-based MIM systems have been designed since this breakthrough.

Memristive properties of thin-film-based cross-point nano-devices were not investigated before 2008; however, these systems were actively researched for decades for designing resistive memory components. These devices are commonly denoted as resistive random access memories (RRAMs or ReRAMs). Chua has argued that such resistive-switching systems can be generalized as memristors [12]; however, there is a debate on this claim [36]. A common trend of using memristors and RRAMs in an analogous fashion exists in circuit and systems design research. Hence, in this chapter, we will use *memristor* as a general term for branding common variations of resistive switching memories *i.e.*, memristors, RRAMs, ReRAMs *etc*.

Progress in memristor-based systems research has poised memristors as a promising solution for low power and high-density non-volatile storage. Unique electronic properties of memristors have attracted several research directions such as memory applications, neuromorphic computation, and hardware security. As memristors gradually become a commodity component/product in computing systems, security issues related to the design and implementation of memristors-based hardware should be studied in detail. Moreover, memristors can be used for designing novel security primitives, which will employ unique intrinsic properties of these devices for security and cryptographic applications. In the next section, we introduce the basic device physics for modeling memristor and memristive system to explore and understand current research efforts in detail.

1.2 Basics of Memristor

According to the classical definition by Leon Chua, instantaneous resistance of a charge-controlled memristor can be written as [9]:

$$M(q) = \frac{d\phi(q)}{dq} \equiv \frac{v(t)}{i(t)} \tag{1.1}$$

where, $q = \int_{-\infty}^{t} i(\tau)d\tau$ is the electric charge, $\phi(q)$ is the magnetic flux, $v(t)$ is the voltage across the device, and $i(t)$ is the electric current flowing in the device at time t [8]. Therefore, from equation 1.1, we can see that the resistance (also known as memristance $M(q)$) of an ideal memristor is dependent on the current that has previously passed through the device.

Chua and Kang generalized the concept of memristors and memristive system in [13]. It was theoretically argued that the dynamic properties of a current controlled memristive system can be expressed with the help of an internal state variable w as:

$$\frac{dw}{dt} = f(w,i) \tag{1.2}$$

$$v(t) = M(w,i) \times i(t) \tag{1.3}$$

where, $M(w,i)$ is the generalized resistance of the memristive system and $f(w,i)$ is a function that captures the boundary behavior and various nonlinear dynamical effects. From these equations, it can be seen that for zero input current there will be a zero output voltage, irrespective of the state variable w. Hence, this dynamic system has a zero-crossing Lissajous figure-like input-output relationship [13]. This input-output characteristic in the $v-i$ plane is also known as pinched hysteresis loop of memristors, and it is considered to be a signature property of this circuit element [12, 11]. As the frequency of the signal along the memristor increases, this zero-crossing pinched hysteresis loop shrinks in size, and it becomes a straight line when the frequency approaches infinity [11].

This fundamental circuit theoretic model was first realized in an MIM-based device structure at HP Labs [34]. The memristor discovered by HP Labs consists of a thin film (5 nm) with of insulating TiO_2 sandwiched between platinum contacts in the simple metal-insulator-metal (MIM) structure [34]. Several transport models have been proposed to explain the electronic properties of this memristor and devices of similar construction. These models primarily attempt to relate the carrier transport mechanism in the thin films with the system of equations 1.1-1.3. Each model provides a definition of the generalized resistance/memristance $M(w,i)$ and the function $f(w,i)$ which encapsulates the memristive nature of these devices. We discuss some of these memristor models below. Before going into details we introduce common circuit symbols used for memristors in Figure 1.1.

1.2.1 Linear and Non-Linear Ion-Drift Models

The simplest model that relates the resistivity of the HP-devices with memristors is the linear ion-drift model proposed by Strukov *et al* [34]. In this model, the governing carrier transport in the thin film of the memristor is described by linear drift of oxygen vacancies. Assume $w(t)$ as the thickness of the doped region in the thin-film which is created by the linear drift of charged oxygen vacancies (dopants) at a given applied bias. Then, considering a linear ion-drift model, one can write the state equation for the state variable $w(t)$ as:

$$\frac{dw}{dt} = f(w,i) = \frac{\mu_v R_{on} i(t)}{D} \tag{1.4}$$

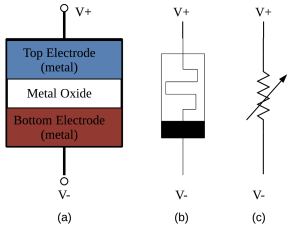

Figure 1.1: (a) Simplified view of MIM structure for memristors; 1.1(b) and 1.1(c) shows the common circuit symbols used for memristors in circuit design.

where D is the film thickness of the memristor, μ_v is the average ion mobility of oxygen vacancies in TiO_2, $(D-w)$ is the size of the undoped region. R_{on} is the resistance of the memristor when it is completely doped (*i.e.,* $w = D$), and R_{off} is the resistance when it is completely undoped (*i.e.,* $w = 0$) as shown in Figure 1.2. The current-voltage relationship for a memristor in this model is defined as [34]:

$$v(t) = \left(\frac{w(t)}{D}R_{on} + (1 - \frac{w(t)}{D})R_{off}\right)i(t) \tag{1.5}$$

Overall, the effective memristance of this structure can be expressed as:

$$M(w) = \frac{w(t)}{D}R_{on} + (1 - \frac{w(t)}{D})R_{off} \tag{1.6}$$

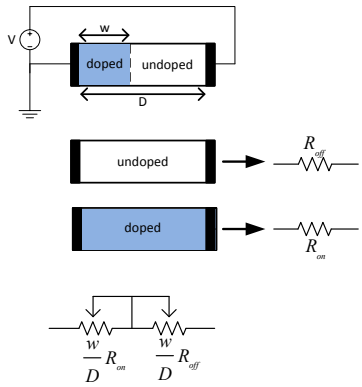

Figure 1.2: Strukov model of a $Pt - TiO_{2-x} - Pt$ device [34].

The linear ion-drift model provides a simple explanation of the transport mechanism in a memristor. It does not consider boundary effects and non-linear dopant kinetics. To implement boundary behaviors, equation 1.4 can be modified as:

$$\frac{dw}{dt} = \frac{\mu_v R_{on} i(t)}{D} g(w,i) \tag{1.7}$$

where $g(w,i)$ is a window function that captures the physics near the device boundary. Several approximations for the window function can be found in the literature [16, 7, 26, 6]. These window functions are listed in Table 1.1.

Table 1.1 Common window functions to capture physics near device boundary in memristors

Author	Window Function
Joglekar *et al.* [16]	$g(w,i) = 1 - \left(\frac{2w}{D} - 1\right)^{2p}$
Biolek *et al.* [7]	$g(w,i) = 1 - \left(\frac{w}{D} - u(-i)\right)^{2p}$
Prodromakis *et al.* [26]	$g(w,i) = j\left(1 - \left[\left(\frac{w}{D} - 0.5\right)^2 + 0.75\right]^p\right)$

Note that, for these window functions, p is a positive integer that controls the rate of change of w near the device boundary, $u(i)$ is the step function and j controls the maximum value of $g(w)$. Although the models described above capture basic device properties, they are insufficient to describe the underlying higher order non-linearity of actual devices.

To simulate these models for VLSI designs, simple SPICE representations are presented in [7, 6, 19] *etc*. The SPICE model proposed by Biolek *et al.* in [7] and its derivatives are widely used in the literature for simulating HP memristors. In this circuit, the window function is incorporated using user-defined function $f()$ and the memory effects are incorporated using a feedback controlled integrator. The circuit diagram for the model is given in Figure 1.3.

All of the discussed linear and non-linear drift models assume that electron transport in memristors is due to the drift of carriers under electric field in the doped and undoped regions. However, quantum mechanical effects need to be considered for accurately describing carrier transport in these nano-devices. Pickett *et al.* first incorporated the quantum mechanical effects in the basic memristor model to give a detailed description of the complex carrier dynamics in memristors [25] as discussed next.

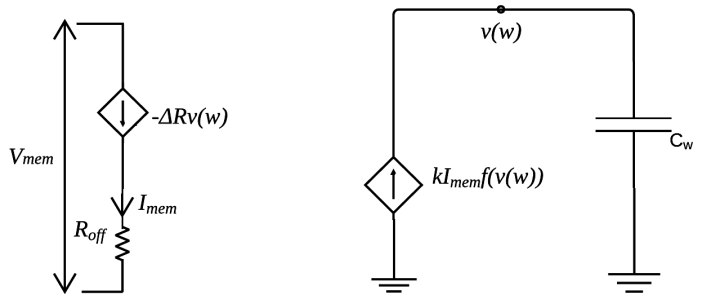

Figure 1.3: SPICE model of a memristor proposed in [7]. Here V_{mem} and I_{mem} are the voltage and current across the memristor, $k = \frac{\mu_v R_{on}}{D^2}$, $\Delta R = R_{on} - R_{off}$, C_w represents the doped layer of width w and $v(w)$ is the voltage across the layer.

1.2.2 Quantum Mechanical Models

Pickett *et al.* explain the observed complex carrier dynamics in TiO_2 based memristors using Simmons tunneling theory [33] and drift mechanisms of the carriers. A tunnel barrier is considered between the conducting channel and the platinum electrode of the device. An ohmic resistor in series with this tunnel barrier is used to explain the device characteristics. The state variable in this model is represented by the width of the tunnel barrier. This model is called the Picketts model or Simmons tunnel barrier model. The state equation for Pickett's model is written as [25]:

$$\frac{dw}{dt} = \begin{cases} f_{off} sinh\left(\frac{|i|}{i_{off}}\right) exp\left[-exp\left(\frac{w-a_{off}}{w_c} - \frac{|i|}{b}\right) - \frac{w}{w_c}\right], & i>0 \\ -f_{on} sinh\left(\frac{|i|}{i_{on}}\right) exp\left[-exp\left(-\frac{w-a_{on}}{w_c} - \frac{|i|}{b}\right) - \frac{w}{w_c}\right], & i<0 \end{cases} \tag{1.8}$$

where, f_{off}, f_{on}, i_{off}, i_{on}, a_{off}, a_{on}, w_c, and b are fitting parameters. The current i through the device is given by [1]:

$$i = \frac{qA}{2\pi h(\Delta w)^2}\left\{\phi_I e^{-B\sqrt{\phi_I}} - (\phi_I + q|v_g|)e^{-B\sqrt{\phi_I + q|v_g|}}\right\} \tag{1.9}$$

Here, q is the elementary electronic charge, A is the average channel area, ϕ_I is the modified barrier height, h is Plancks constant, m is the average effective mass of the carrier and

$$B = 4\pi\Delta w\frac{\sqrt{2m}}{h} \tag{1.10}$$

If we consider R_s as the series resistance of the channel and v_g is the voltage across the tunnel barrier, then the voltage across the device v can be written as:

$$v = v_g + v_R = v_g + iR_s \tag{1.11}$$

To incorporate this model in circuit simulation, Abdalla *et al.* provided a SPICE model based on equation 1.8-1.11 in [1] as given in Figure 1.4. This circuit is derived from equations 1.8-1.11 and uses experimental values to define its parameters. Although this model tries to accurately represent the device physics, simulating this model overestimates current by around 20% and causes the simulated memristor to switch faster than the experimental memristor.

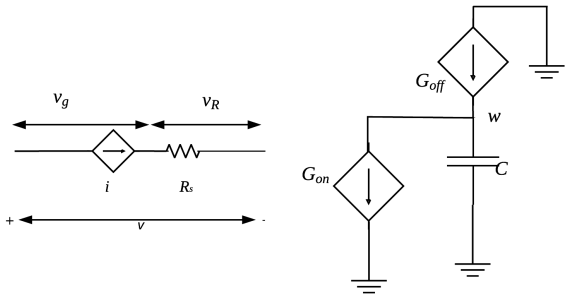

Figure 1.4: (a) SPICE model of a memristor proposed by Abdalla *et al* [1]. Here, the current and voltages are given by equations 1.9-1.11. Figure 1.4(b) represents the state-space model of the device represented by Figure 1.4(a) where C is the width of the tunnel barrier, w is the voltage across the barrier, and $\frac{dw}{dt} = \frac{1}{C}(G_{off} - G_{on})$ where G_{off} and G_{on} are the right hand side of equation 1.8 for $i > 0$ and $i < 0$ respectively.

Picketts model provides a good insight of the carrier dynamics and considers a near accurate physical model. However, this model is computationally expensive. For device simulation, Kvatinsky *et al.* proposed a simplified version known as the *threshold adaptive memristor (TEAM) model* to reduce the computational complexity of Picketts model [18]. There exist similar models such as the Boundary Condition Memristor (BCM) model presented in [5] which are based on the simplified versions of equations 1.8-1.11. Overall, Pickett's model and the subsequently simplified models of memristors provides a good starting point for understanding basic carrier dynamics in a physical memristor.

The equations 1.4-1.11 are specifically derived for the TiO_2-based memristive devices. For other metal-oxide thin film based memristors, the characteristic equation of the device follows a similar pattern. For example, HfO_x-based memristors (also

known as RRAMs) have the state equation as [14]:

$$\frac{dw}{dt} = v_0 e^{-E_{a,m}/kT} sinh\left(\frac{qa\gamma v}{DkT}\right) \tag{1.12}$$

where, w is the state variable for the device as before which represents the spatial distance between the conductive filament in the oxide and the metal boundary, q is the electron charge, D is the device filament thickness, v is the applied voltage, T is the device temperature, $E_{a,m}$, γ, v_0 are all device dependent physical parameters. The current-voltage relationship in the device is given in the following equation [14]:

$$i = i_0 e^{-\frac{w}{w_0}} sinh\frac{v}{v_0} \tag{1.13}$$

where i_0, v_0, w_0 are device dependent physical parameters.

1.3 Memristor-based Security Application

Memristors have some unique properties useful for security applications such as non-volatility, bias-dependent write-time, fabrication variations in a filament, non-linearity in the current-voltage relationship [3].

1. **Non-volatility**: Memristors are non-volatile, *i.e.*, a memristor can retain its resistive state even without power. For single-level memory application, a memristor is used as a binary storage: the resistance remains in a higher resistive state (HRS) or lower resistive state (LRS). During state-transition (*i.e.*, moving from low to high resistive state), a significant amount of current needs to pass through the device. Moreover, after fabrication memristor can have random initial state and this can be used for generating secure random keys.

2. **Bias dependent write-time**: The state transition of a memristor requires the memristor to be kept under a certain bias (V_{bias}) for a given amount of time. This time is usually referred to as the write-time (t_{wr}) of the memristor. Write time is highly dependent on the voltage bias and by adjusting the bias voltage, one can manipulate the write-time required for a given state transition [38].

3. **Fabrication variation in filament thickness**: Physical properties of a memristor vary significantly with the fabrication variation in the filament thickness. For example, equation 1.7-1.9 shows that the change in memristive state is non-linearly dependent on the filament thickness w. Therefore, a small change in film thickness can lead to a significant measurable difference in the memristance of the device. Such random variability resulting from fabrication variations can be exploited to design security features such as device authentication.

4. **Non-linearity:** From the basic equations presented in the previous sections, it can be seen that memristors are highly nonlinear in nature. Device nonlinearity

is a sought after property for secure hardware design, and therefore, harnessing this property will provide a novel implementation of common hardware security protocols such as Physically Uncloneable Functions (PUFs).

There have been many reported efforts on memristors related to security. In this chapter, we present several of these security primitives. First, we introduce our reader to the memristor PUFs that are emerging as an important hardware security component. We also discuss memristor-based true random number generators (TRNGs), encryption schemes that leverage the chaotic behavior of the memristor circuit *etc.*

1.3.1 Memristor-based Physically Uncloneable Function Design

Physically Uncloneable Functions are hardware-dependent security primitives that harness physical variation of a silicon chip to generate unique chip-dependent challenge-response pairs (CRPs). PUF CRPs can be used for secret key generation, seeding of random number generators, and in CRP-based authentication and attestation. Physical dependence of PUF primitives has opened up new avenues for implementing hardware intrinsic security and trust.

Physical variation in the thin film of memristors has a pronounced effect on their device characteristics such as the random variation in write-time for different devices, random distribution of measured resistance after the device forming step during fabrication, and a random resistive path between higher and lower resistive states. These fundamental properties provide an ideal situation to build memristor-based PUFs.

There have been several potential memristor-based PUF-designs in current literature with different use case and attack models. Since PUFs are inherently related to hardware-based authentication, we will assume a simple example scenario where an entity Alice wants to authenticate another entity, Bob. Malice plays the role of an attacker who subverts the authentication mechanism. Below we have summarized the basics of operation of these PUFs.

1.3.1.1 Nano PPUF

Memristor-based crossbars are used for designing one of the first memristor-based PUFs called nano-PPUF [28]. A simulation model of the physical design of a public PUF is publicly available; however, simulation complexity can create a time bound authentication protocol. The attack model assumes a computationally bounded adversary unable to simulate the exact output for a given PUF design. The non-linear equations governing the current-voltage relationship of memristors and the viability of fabricating large memristive crossbars provides the simulation complexity required by this PUF model.

In this PUF design, a public registry contains the simulation model for a given user's (Bob's) memristive PUF. When Alice wants to authenticate Bob, she first sends a random challenge vector $\mathbf{V_C} = \{v_1, v_2, \ldots v_n\}$, where, v_i represents a physical input. For the given PPUF at [28], $\mathbf{V_C}$ is the voltage applied to an $n \times n$ memristor-crossbar. Since Bob has the physical memristor, he can correctly respond to Alice's challenge.

He sends the correct response vector $\mathbf{V_R}$. For a computationally-bounded attacker Malice, completing this step would require simulating the complete crossbar, which would be computationally prohibitive. For completing the authentication, Alice then picks a subsection of Bob's crossbar (a polyomino) and requests the voltages at the boundaries of this polyomino. Bob sends the measurement and simulation results. Alice can accurately simulate the smaller polyomino using $\mathbf{V_C}$ and $\mathbf{V_R}$, and verify Bob's results. Thus, Alice can authenticate Bob.

This initial PUF design suffers from several crucial drawbacks. The crossbar simulation and the results from the physical crossbar would only match if the physical conditions that affect the current in a memristor (such as temperature, history of current flow, aging) remain the same. This is a difficult condition to fulfill for such design. Moreover, Malice can try machine learning and model building attacks on passively obtained challenge responses to breaking the authentication scheme. Additional improvements considering these physical effects on this PUF design are discussed in [37] and [27].

1.3.1.2 CMOS-Memristive PUF

Nano-PPUF uses memristive-crossbars to generate unique challenge-response pairs. Unique device properties of single memristor cells are also used for designing other PUFs and authentication system discussed in [30, 29, 17, 4, 3]. Most of these works depend on the bias-dependent write-time and fabrication variations of memristors. For example, memory based PUF cells proposed in [30] uses the fabrication variation dependent write-time differences as an entropy source. Circuit design for this PUF is shown in Figure 1.5.

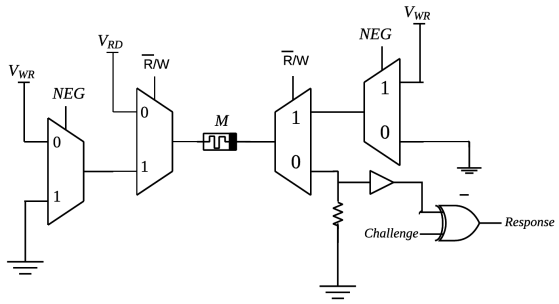

Figure 1.5: 1-bit memristive memory based PUF cell proposed in [30].

The principle of operation for this PUF is simple. First a RESET signal is applied using $NEG = 1$ and $\bar{R}/W = 1$. This puts the memristor M in a high resistive state

(HRS). Then a SET operation is performed with a write pulse $t_{wr,min}$ applied at V_{WR} and $NEG = 0$. The write pulse $t_{wr,min}$ is selected in a way that the likelihood of state transition is 50%. Since the write-time of memristors are random for a given voltage due to the physical randomness in their construction, the circuit (Figure 1.5) can harness 1-bit of entropy from the memristor. With a read operation $\bar{R}/W = 0$ and a challenge bit, one can receive a 1-bit response from this PUF circuit.

Similar weaknesses such as the effect of aging, noise at the supply voltage as discussed for the Nano-PPUF exist for this design also. An improvement of this design can be made by dividing the $t_{wr,min}$ into smaller pulses and use the number of pulses as a challenge vector as described in [3]. Improved PUF design with proper SET time determination for the cell shown in Figure 1.5 is also reported by Rose *et al.* in [29]. Mazady *et. al.* [21] have experimentally verified the design proposed

The second PUF design proposed in [30] is dependent on the stochastic nature of filament formation of memristors during fabrication. Two memristors connected laterally (*i.e.,* they share the same bottom electrode) is used for a PUF cell in this design. The top electrodes of these two memristors are connected to V_{DD} and *GND* respectively as shown in Figure 1.6. The operation for bit-generation is simple. Experimental results have suggested that a lateral SET operation can SET both of the devices; however, a lateral RESET puts one of the devices to HRS and the other remains in LRS. This switching is dependent on the difference in the stochastic variation of these two laterally connected device. By comparing which one of the device has state transition, a single bit of entropy can be generated. The circuit schematics for the bit extraction is given in detail at [30].

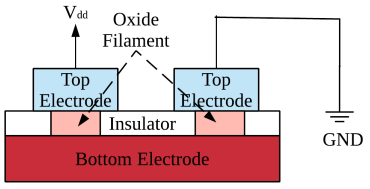

Figure 1.6: Laterally connected memristors for designing PUF cell [30]

1.3.1.3 Memristive Super-High Information Content (SHIC) PUF

The high packing density of memristive crossbar is useful for creating large memories with small area footprints. Passive crossbar arrays for memristors can hold a large amount of information content which can be used for designing SHIC-PUFs discussed in [31]. SHIC-PUFs have two fundamental component: a high-density mem-

ory with random information content and a slow readout technique for this memory. It should be noted that the common memory designs are focused on fast readout circuitry for memristors due to the requirement of faster data processing and storage. However, this leads to fast leaking the random information stored in an SHIC-PUF. Therefore, this PUF requires special readout mechanism to reduce the information leakage. These are strong PUFs with linear CRPs.

The high density of the memristor crossbars makes memristors an ideal candidate for an SHIC-PUF. Some memristor designs require a initial programming step (also known as the *forming* step) after fabrication, where the device is *formed* (*i.e.,* becomes a memristive device instead of a common resistor). Memory contents in a memristive memory are found to be random after a forming step. This random startup resistive states can be the entropy source for such PUFs. Furthermore, probing attack on compact memristive crossbar is difficult, which makes such design secure against probing attacks.

1.3.1.4 Memristive Ring Oscillator PUF

Initial programming variation of memristors is also used in mrPUF which integrates a memristor device into the conventional ring oscillator PUF (RO-PUF) [17]. This design uses the randomness in state distribution in a memristive crossbar as a source of entropy and uses these random resistances on the delay paths of the ring oscillators. In this design, memristors can be viewed as an entropy enhancer to the original RO-PUF designs. These designs enjoy a higher degree of stability since the scheme does not require reprogramming of the memristors.

Memristive PUF is a new concept in PUF design. Other memory-based PUFs (such as SRAM PUFs) is studied thoroughly in the literature with fabricated prototypes. However, fabrication requirements of nano-electronic devices are demanding, and therefore, recent advances in memristive PUF design depends on the existing device model discussed in section 1.2. Therefore, the designs and the reported results for memristive PUFs still need to be verified on fabricated designs against known modeling, learning and side-channel attacks.

1.3.2 Memristor-Based Secret Sharing

One of the most fundamental applications of memristive PUFs is in authentication where Alice tries to authenticate Bob. Password-based authentication can be viewed as a simple case of secret sharing. Common memristor-based PUFs uses simple CRP mechanism for authentication solution. However, non-volatility and multi-level operation of memristor can be useful in generalized secret sharing protocols. Recent works by Arafin and Qu [3, 4] details a device dependent secret sharing mechanisms that uses the bias-dependent write-time properties of the memristors to provide authentication and secret sharing solutions for single and multiple users.

Secret sharing is a well-studied problem in cryptography which can be defined as follows:

For a given secret S, construct and distribute pieces of this secret $(S_1, S_2, \ldots S_n)$ *to n parties in a way such that the knowledge of k or more pieces would be sufficient to reconstruct the secret. However, when knowledge of any* $k-1$ *pieces or less is available, it would be impossible to reconstruct the secret.*

One of the key solutions for this problems is given by Shamir's secret sharing algorithm [32]. This solution requires number theoretic calculations and might be infeasible for resource constrained systems. Naor *et al.* provided another solution to this problem using visual cryptography [22]. This solution requires printing of the secret shares $(S_1, S_2, \ldots S_n)$ on plastic transparencies and when k of these transparencies are placed on top of each other, the secret is revealed. However, for $k-1$ transparencies, no information is leaked to the participants. This is known as a k-out-of-n visual secret sharing.

Visual cryptography remained a solution on transparencies for the last several decades. However, multi-level memory designs and distributed key sharing schemes can harness this idea in hardware. Arafin *et al.* have demonstrated that multi-level memristors can be used for solving secret sharing problem in memory hardware [4]. The scheme proposed in [4] uses basic visual cryptographic construct to create the secret shares $(S_1, S_2, \ldots S_n)$. For programming the memristors, the authors used smaller write voltage which elongated the write-time of the device. Also, instead of a (single) longer write pulse, the authors used multiple short-duration write pulses to write the memristors. It has been shown that the number of pulses required for such write is dependent on the write voltage and the fabrication variation of memristors. Hence, this number can be used as a source of entropy.

For secret sharing, the proposed solution requires not only valid users but also a valid authenticator. Assume, Alice wants to authenticate k-users simultaneously. To do so, she generates a secret to share with them according to the hardware she possesses. For sharing a 1-secret bit with k-users, first Alice uses the following protocol to generate k-shares. This protocol for a k-out-of-k scheme given below [23]:

- Consider a ground set G consisting of k elements $g_1, g_2, \ldots . g_k$; subsets of G with even cardinality are $p_1, p_2, .., p_{2^{k-1}}$ and the subsets with odd cardinality are $q_1, q_2, , q_{2^{k-1}}$.
- Define, $S_0[i, j] = 1$ iff $g_i \in p_j$ and $S_1[i, j] = 1$ iff $g_i \in q_j$. The resulting Boolean metrics S_0 and S_1 will have dimensions $k \times 2^{k-1}$.
- Permute all the column of S_0 and S_1 metrics to derive matrices C_0 and C_1 respectively.
- If the ith-bit of the secret is 0, distribute the rows of C_0 to the participants.
- If the ith-bit of the key is 1 distribute the rows of C_1 to the participants.

Using the techniques discussed in [23], this k-out-of-k scheme can easily be converted into a k-out-of-n scheme. For sharing 1-bit of secret, Alice gives out $2^{(k-1)}$

bits to all the users. In visual cryptography, to reconstruct the secret, the users need to print these bits (a dark pixel for a 1 and a transparent pixel for a zero) on a transparency, and all of these transparencies needs to be stacked up. If the stack looks darker than a given threshold ϕ (*i.e.,* there are ϕ or more 1's in the combined share), the bit is a 1, otherwise, it is a 0.

Memristive hardware can offer a simpler way of bit-reconstruction. First, assume Alice posses a memristor for which she can set the voltage V_{read} such that the it requires exactly ϕ-pulses for a write operation. Alice saves the value of V_{read} for later bit-reconstruction.

To reconstruct the bit, all the users puts their share as a bit-stream to their respective ports (*i.e.*, U_1, U_2...U_k of the circuit in Figure 1.7). Once the bits (related to the i'th bit of the secret) for k shares are superimposed by logical OR, the bit-string consists of x number of 1's. If x is greater than the number of write pulses required for state transition in Alice's memristor(i.e., ϕ), the secret bit is reconstructed as a 1, otherwise, the bit is decoded as a 0. Here, Alice requires the knowledge of the required V_{read} that correctly reconstruct the share. If used in authentication, Alice can authenticate k-users simultaneously using this technique.

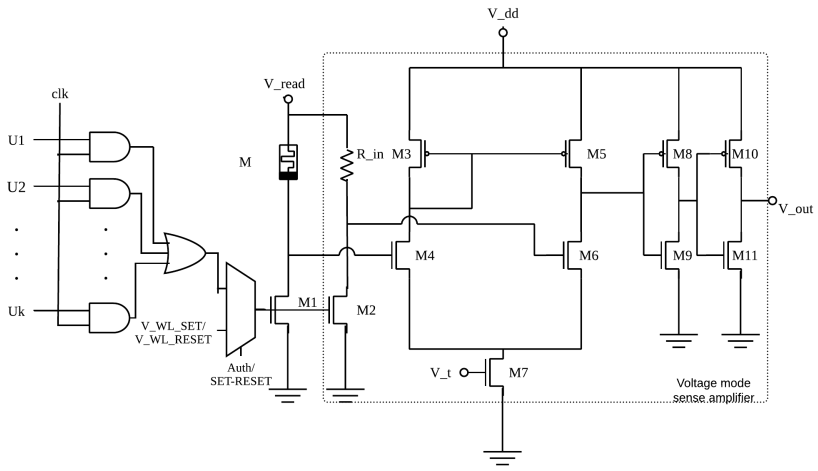

Figure 1.7: Share commitment circuits for k users. At each clock cycle, all of the k-users commit 1-bit of their secret synchronously to the $U_1, U_2, \ldots, U_i$ ports. Logical OR of this signal is applied to the memristor R_x. Since 1-bit of the secret is deconstructed into several bits, it needs multiple cycles for reconstructing the bit.

1.3.3 Random Number Generation and Memristors

Random number generators are an essential primitive of common security protocols. Entropy sources for the true random number are scarce in practice, and proper entropy extraction mechanism is required for generating *true* random numbers. One of the preliminary designs of TRNGs using resistive memories can be found in [15]. This random number generator uses random trapping and detrapping of carriers in the defects of the oxide thin film in a contact-resistive random access memory (CR-RAM). This trapping process results in random fluctuations of the resistance of the device, and this fluctuation is captured using a comparator and flip-flops to generate random bit-streams. The fabricated design satisfies several statistical randomness standards and tests set by National Institute of Standards and Technology (NIST).

1.3.4 Memristor-based Chaotic Circuits in Secure Communication

Memristor-based circuits are an interesting primitive in Chaos theory. The fundamental equations governing the memristive-RL circuit can be translated into state equations governing a chaotic system. This memristor-resistor-capacitor circuit (as shown in Figure 1.8) is known as a Chua circuit. This circuit provides new ways of chaos generation and experimentation. Absence of a "true" memristor impeded research progress in using memristive circuits for chaos generation. However, after the discovery of HP-memristors, there is a renewed interest in memristive chaos circuits and networks.

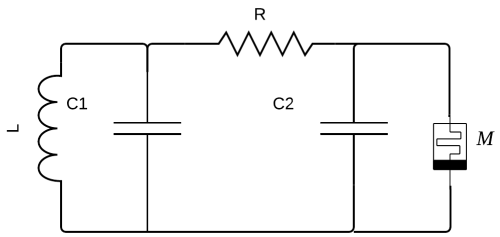

Figure 1.8: Chua circuit for chaos generation [10].

The chaotic memristive system is used for implementing an image encryption technique in [20]. The authors used a modified Chua circuit for generating chaotic sequences and use these sequences as keys for image scrambling (or pixel replacement) based encryption. The image can be unscrambled using the same sequences.

The authors claimed that such techniques would be resistive towards brute-force and tampering attacks.

Researchers have also proposed chaotic sequences for stream cipher-based encryptions. Since memristor provides hardware generated chaos, studies on active and passive memristive circuits for chaos-based encrypted communication can be found in recent literature. One of the common examples can be found in [24] where a secure communication protocol is proposed using memristive Van der Pol oscillator. In this protocol, both of the communicating parties (the transmitter and the receiver) have matched Van der Pol oscillators. The message to be transmitted is used for driving this chaotic oscillator which generates a chaotic signal. This signal is transmitted over an untrusted channel. The security model assumes that since it is difficult for an attacker to have an exactly matched oscillator, it will be impossible for an attacker to decode the message. Only a valid receiver with a matched oscillator will be able to recover the message.

Chaotic circuits are sensitive to noise. Therefore, small device mismatch or shift in the operating conditions can thwart synchronization between valid transmitter and receiver. Therefore, an adaptive synchronization feedback protocol for reconstructing the message from received chaotic signals is proposed in [24]. An advanced secure multi-party (up to four parties) communication using four memristive chaotic circuits is proposed in [35].

Security analysis of chaos-based secure communication protocols needs to be properly addressed by the cryptographic community before validating their application. Hence, new research initiative is required for merging these two branches of technological progress.

1.4 Performance Issues of a Memristive Circuit

In this section, we discuss the common performance issues in a memristive system. Environmental variations such as temperature, and noise in the supply voltage can affect reliable performance of a memristive system. Moreover, crossbar implementation of memristors without control devices suffers from sneak path issues. Bias dependent write-time of memristors also make them susceptible to unbalanced SET-RESET problem which can aggravate system performance over time. We have discuss these issues in details below.

1. **Sneak Path Current** : In a crossbar memory array, a memristor unit can contain a single memristor (1M configuration) as shown in Figure 1.9 or a transistor and the memristor (1T1M configuration). Memristors can have high packing density if fabricated as a 1M crossbar. This simple yet powerful design can lead to unprecedented storage density, however, this design faces issues with controlling the current during operation. As there is no direct access control dedicated for each memory cell, when accessing one memristor, current can flow in the adjacent memristors sharing the same row or column. This current is known as sneak path current and it can create severe performance issues

in a crossbar array. Therefore, for better control and reliability some current control mechanism is usually deployed. A simple solution is to use a transistor to control the flow of current in each device. Thus, 1T1M configuration has immunity to sneak path currents and can deliver more reliable performance. However, transistors are larger than memristors and require elaborate fabrication techniques which make crossbar-based designs less appealing.

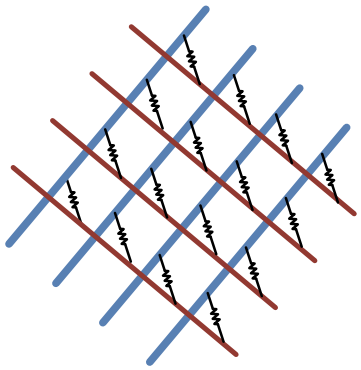

Figure 1.9: Simple view of a 4×4 memristor crossbar. The blue wires denote the top metal electrode and the brown wires depict the bottom metal electrodes. Resistance symbols are used to depict the memristors. Applying voltage (+V) on any of the top blue wire and (-V) on any brown wire with the other wires grounded creates sneak paths (from $\pm V$ to ground) involving the memristors sharing the top electrode with the blue wire and the bottom electrode with the brown wire.

2. **SET-RESET Unbalance**: Some security application may require multiple read-write on the same memristor. However, as if the bias voltage varies in between the SET/RESET cycle then the output of the circuit used for secure hardware design may become completely unreliable. An example of this is given by Arafin and Qu in [3]. Moreover, the resistance of an ideal memristor depends on the history of current that has passed through the device before. As a result, inaccurate operation using wrong bias can create a cascade of failures for a number of subsequent SET/RESET cycles. Therefore, proper balancing must be met when designing sensitive security protocol depending on the analog properties of a memristor.

3. **Variations in Operating Conditions**: Fault injection attacks on common hardware platforms depend on the voltage glitches, synchronization mismatches and temperature manipulations for inducing a fault in cryptographic circuits. Hence, memristor-based security primitives must be designed keeping the attacks in check. Furthermore, if changes in the operating condition

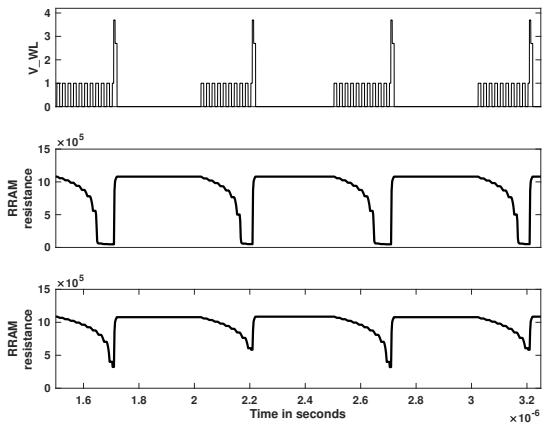

Figure 1.10: Example of SET/RESET unbalance [4] for the circuit given in Figure 1.7. The top plot shows the typical pulsed programming for a memristive device. The second figure (middle one) shows the changes in a memristor resistance for a balanced SET/SRESET condition. The plot at the bottom shows the effect of unbalanced biasing where the SET voltage is lowered by 12mV than the previous balanced condition and the RESET voltage is kept the same as before.

make the output of a security primitive unstable, then proper error correction measures need to be in place to prevent these unwanted errors. Therefore, memristor-based secure designs must meet common operating corner points in terms of operating voltage, temperature, and process variation.

Dependence on the bias-voltage stability for state-transition is discussed before. Stability of memristor's resistive switching is also dependent on temperature variations. For example, the closed form equations 1.12 and 1.13 for HfO_x-based memristors show a non-linear dependence of the state variable (w) on the operating temperature T. Hence, memristor-based security designs should either experiment their prototype in different operating temperature points or consider proper device model that accounts temperature variations during operation.

4. **Aging**: Aging model for memristors is still under investigation. There are two known temporal effects in memristors: (a) short-term variation due to filament development, and (b) long-term read-out effects. Random filament formation during operation and spatial variation of filament development in different write cycles can create cycle-to-cycle variation in low and high resistive state values within a shorter period. This is a short term variation and can occur in random [3]. Moreover, for reading the state information of a memristor, a read/sense current must pass the through the device. Although this read current

does not change the resistive state of the device dramatically, it can degrade the resistance value of a given state over time. The PUFs based on the initial programming variations may experience a wide range of errors as the device ages due to such state degradation. Furthermore, aging can change the SET/RESET dynamics and internal state transitions over time. A detailed recommendation for the designers on developing memristor-based security primitives can be found in [2].

1.5 Conclusions

In this chapter, we discuss the current efforts in memristor-based security primitive design. For our readers, we first present common models describing the underlying physics and principles of operation, then we discuss the recent progress and development in hardware security primitive design using memristors, and finally, we provide the common performance and operational issues that must be properly investigated before integrating memristive designs in existing hardware platform.

1.6 Acknowledgment

This work was supported in part by the Air Force Research Laboratory under agreement number FA8750-13-2-0115 and by an Air Force Office of Scientific Research Office MURI under award number FA9550-14-1-0351.

References

[1] Hisham Abdalla and Matthew D Pickett. SPICE Modeling of Memristors. In *International Symposium on Circuits and Systems (ISCAS)*, pages 1832–1835. IEEE, 2011.

[2] Md Tanvir Arafin, Carson Dunbar, Gang Qu, Nathan McDonald, and Lok Yan. A Survey on Memristor Modeling and Security Applications. In *Sixteenth International Symposium on Quality Electronic Design*, pages 440–447, March 2015.

[3] Md Tanvir Arafin and Gang Qu. RRAM Based Lightweight User Authentication. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '15, pages 139–145. IEEE Press, 2015.

[4] Md Tanvir Arafin and Gang Qu. Secret Sharing and Multi-user Authentication: From Visual Cryptography to RRAM Circuits. In *Proceedings of the 26th Edition on Great Lakes Symposium on VLSI*, pages 169–174. ACM, 2016.

[5] Alon Ascoli, Ronald Tetzlaff, Fernando Corinto, and Marco Gilli. PSpice Switch-Based Versatile Memristor Model. In *International Symposium on Circuits and Systems (ISCAS)*, pages 205–208. IEEE, 2013.

[6] S Benderli and TA Wey. On SPICE Macromodelling of TiO_2 Memristors. *Electronics Letters*, 45(7):377–379, 2009.

[7] D Biolek, V Biolkova, and Z Biolek. SPICE Model of Memristor with Nonlinear Dopant Drift. *Radioengineering*, 2009.

[8] Leon Chua. Memristor-the Missing Circuit Element. *IEEE Transactions on Circuit Theory*, 18(5):507–519, 1971.

[9] Leon Chua. Memristor-the Missing Circuit Element. *IEEE Transactions on Circuit Theory*, 18(5):507–519, 1971.

[10] Leon Chua. *The Genesis of Chua's Circuit*. Electronics Research Laboratory, College of Engineering, University of California, 1992.

[11] Leon Chua. The Fourth Element. *Proceedings of the IEEE*, 100(6):1920–1927, 2012.

[12] Leon Chua. Resistance Switching Memories are Memristors. In *Memristor Networks*, pages 21–51. Springer, 2014.

[13] Leon Chua and Sung Mo Kang. Memristive Devices and Systems. *Proceedings of the IEEE*, 64(2):209–223, 1976.

[14] Ximeng Guan, Shimeng Yu, and H-S Philip Wong. A SPICE Compact Model of Metal Oxide Resistive Switching Memory With Variations. *IEEE Electron Device Letters*, 33(10):1405–1407, 2012.

[15] Miao Hu, Hai Li, Yiran Chen, Qing Wu, and Garrett S Rose. BSB Training Scheme Implementation on Memristor-Based Circuit. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2013*, pages 80–87. IEEE, 2013.

[16] Yogesh N Joglekar and Stephen J Wolf. The Elusive Memristor: Properties of Basic Electrical Circuits. *European Journal of Physics*, 30(4):661, 2009.

[17] Omid Kavehei, Chun Hosung, Damith Ranasinghe, and Stan Skafidas. mrPUF: A Memristive Device Based Physical Unclonable Function. *arXiv preprint arXiv:1302.2191*, 2013.

[18] Shahar Kvatinsky, Eby G Friedman, Avinoam Kolodny, and Uri C Weiser. Team: Threshold Adaptive Memristor Model. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1):211–221, 2013.

[19] Eero Lehtonen and Mika Laiho. CNN Using Memristors for Neighborhood Connections. In *12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA)*, pages 1–4. IEEE, 2010.

[20] Zhao-hui Lin and Hong-xia Wang. Image Encryption Based on Chaos With PWL Memristor in Chua's Circuit. In *International Conference on Communications, Circuits and Systems (ICCCAS)*, pages 964–968. IEEE, 2009.

[21] Anas Mazady, Md Tauhidur Rahman, Domenic Forte, and Mehdi Anwar. Memristor PUFa Security Primitive: Theory and Experiment. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 5(2):222–229, 2015.

[22] Moni Naor and Benny Pinkas. Visual Authentication and Identification. In *Annual International Cryptology Conference*, pages 322–336. Springer, 1997.

[23] Moni Naor and Adi Shamir. Visual Cryptography. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 1–12. Springer, 1994.

[24] EB Megam Ngouonkadi, HB Fotsin, and P Louodop Fotso. Implementing a Memristive Van Der Pol Oscillator Coupled to a Linear Oscillator: Synchronization and Application to Secure Communication. *Physica Scripta*, 89(3):035201, 2014.

[25] Matthew D Pickett, Dmitri B Strukov, Julien L Borghetti, J Joshua Yang, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. Switching Dynamics in Titanium Dioxide Memristive Devices. *Journal of Applied Physics*, 106(7):074508, 2009.

[26] Themistoklis Prodromakis, Boon Pin Peh, Christos Papavassiliou, and Christofer Toumazou. A Versatile Memristor Model with Nonlinear Dopant Kinetics. *IEEE Transactions on Electron Devices*, 58(9):3099–3105, 2011.

[27] Jeyavijayan Rajendran, Ramesh Karri, James Bradley Wendt, Miodrag Potkonjak, Nathan R McDonald, Garrett S Rose, and Bryant T Wysocki. Nanoelectronic Solutions for Hardware Security. *IACR Cryptology ePrint Archive*, 2012:575, 2012.

[28] Jeyavijayan Rajendran, Garrett S Rose, Ramesh Karri, and Miodrag Potkonjak. Nano-PPUF: A Memristor-based Security Primitive. In *Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 84–87. IEEE, 2012.

[29] Garrett S Rose, Nathan McDonald, Lok-Kwong Yan, and Bryant Wysocki. A Write-time Based Memristive PUF for Hardware Security Applications. In *Proceedings of the International Conference on Computer-Aided Design*, pages 830–833. IEEE Press, 2013.

[30] Garrett S Rose, Nathan McDonald, Lok-Kwong Yan, Bryant Wysocki, and Karen Xu. Foundations of Memristor Based PUF Architectures. In *IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pages 52–57. IEEE, 2013.

[31] Ulrich Rührmair, Christian Jaeger, Matthias Bator, Martin Stutzmann, Paolo Lugli, and György Csaba. Applications of High-Capacity Crossbar Memories in Cryptography. *IEEE Transactions on Nanotechnology*, 10(3):489–498, 2011.

[32] Adi Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, 1979.

[33] John G Simmons. Generalized Formula for the Electric Tunnel Effect Between Similar Electrodes Separated by a Thin Insulating Film. *Journal of Applied Physics*, 34(6):1793–1803, 1963.

[34] Dmitri B Strukov, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. The Missing Memristor Found. *Nature*, 453(7191):80–83, 2008.

[35] Junwei Sun, Yi Shen, Quan Yin, and Chengjie Xu. Compound Synchronization of Four Memristor Chaotic Oscillator Systems and secure communication. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(1):013140, 2013.

[36] Sascha Vongehr and Xiangkang Meng. The Missing Memristor has not been Found. *Scientific Reports*, 5, 2015.

[37] James B Wendt and Miodrag Potkonjak. The Bidirectional Polyomino Partitioned PPUF as a Hardware Security Primitive. In *Global Conference on Signal and Information Processing (GlobalSIP)*, pages 257–260. IEEE, 2013.

[38] H-S Philip Wong, Heng-Yuan Lee, Shimeng Yu, Yu-Sheng Chen, Yi Wu, Pang-Shiu Chen, Byoungil Lee, Frederick T Chen, and Ming-Jinn Tsai. Metal–Oxide RRAM. *Proceedings of the IEEE*, 100(6):1951–1970, 2012.