RRAM Based Lightweight User Authentication

Md Tanvir Arafin
ECE Department
University Of Maryland
College Park, USA
marafin@umd.edu

Gang Qu
ECE Department
University Of Maryland
College Park, USA
gangqu@umd.edu

***Abstract*—Resistance switching memories have emerged as a promising solution for low power and high density non-volatile storage. Unique electronic properties of resistive RAMs (and memristors) have attracted not only memory applications, but other applications such as neuromorphic computation and security as well. In this paper, we investigate how to take advantage of the availability of RRAM devices or components in the system to perform lightweight user authentication. Based on several well-known features of RRAM devices, we argue that the basic requirements for user authentication are met in RRAM devices. Then, we design three RRAM utility functions, namely Read State, Read Pulse Write State, and Copy State that are critical to develop RRAM based user authentication protocols. We propose two such protocol primitives to illustrate the concepts, layout the hardware design, and discuss the potential attacks to these protocols and the corresponding countermeasures. We conclude that under realistic attacking assumptions, the proposed protocols are secure. Finally, we use PTM's 65nm MOSFET models and perform HSPICE simulation of our proposed RRAM based hardware authentication units to demonstrate the reliability of our protocols against environmental variations such as temperature, noise, unbalanced set/reset, filament formation variation and device aging.**

Keywords—Resistive-Random Access Memory (RRAM), Memristor, User-authentication, Challenge-response pair.

I. INTRODUCTION

In recent years, metal-oxide-based resistive switching memories have drawn significant research attention due to their potential application in next generation nano-electronic non-volatile memory systems [1]. Simple device geometry, high density and inherent non-volatile features have made resistive random access memories (RRAMs) very appealing as a replacement of flash memory and DRAM. Moreover, studies on on-chip CMOS compatibility of resistive RAMs and memristors have shown promising results indicating the capability of replacing on-chip SRAMs [2]. Logic processing and possible neuromorphic computation in functional resistive switching memories may one day replace CMOS with memristors [3].

In spite of its numerous promises, commercial products based on memristors and RRAM is still quite elusive. Extensive research on fabrication process, yield, novel computer architecture, and circuit design is required for developing competitive products. Furthermore, applications outside memory operation need to be explored to harness the unique characteristics of resistive switching memories. In this work, we have explored one of such avenues- user authentication on resistive switching based memory devices.

Variation in the implementation and the minute details in the description of underlying physics have introduced different terms such as RRAMs, memristors, ReRAM etc. for branding resistive switching memories. Leon Chua, the pioneer of the memristor theory, has theoretically argued about naming all the resistive-switching based non-volatile memories as memristors [4]. However, there exist some debates on this claim [5]. In this paper, we have performed our experiments on HfOx-based resistive memories due to their high packing density, large ON/OFF ratio, multibit storage capacity and good switching endurance [1]. Since it is commonly referred as resistive random-access memory or RRAM, we use this term for the paper. Other memristors such as that developed in HP Lab also possess the features we need for the proposed user authentication protocols. So, our approaches should be applicable on memristors too.

Hardware security is an emerging field that studies the application of security primitives in hardware and their vulnerabilities. As new technologies emerge, the nature of security primitives and vulnerabilities change, hence, studying security features of such technologies have its unique benefits. Moreover, new technologies can provide exclusive hardware features useful for security purpose. Instead of making hardware the weakest link in security (for example, side channel attacks), one can employ hardware's intrinsic properties to enhance security.

In this paper, we focus on using RRAM devices or components available on future systems for user authentication. To the best of our knowledge, this is the first attempt for such applications. The main advantage of this approach is that it does not necessarily need the computational expensive cryptographic techniques. In that sense, our approach can be considered as a lightweight cryptography technique for authentication. RRAM device's intrinsic energy efficiency makes this even more attractive.

The rest of the paper is organized as follows. In Section 2, we survey the related work. Section 3 provides the background of RRAM and several key features we will need later. Section 4 describes the device requirements, the basic RRAM utility functions, two primitive authentication protocols, the hardware implementation, and security analysis. We report our experiment platform and the results on reliability validation in Section 5.

II. RELATED WORK

This work is a continuation of current scholarly efforts in the field of memristor and RRAM based secure circuit design. Efforts on non-volatile memory based secure hardware designs are currently focused on the different categories, such as

developing novel physically unclonable functions (PUFs) and true random number generators, designing memristor-based chaos circuits for secure communication, exploring secure memory architecture designs for non-volatile systems.

Different design schemes for memristive physically unclonable functions can be found in current literature. One of the earliest schemes is nano-PPUF where the authors used the increasing complexity of simulating a crossbar of resistive memories as a tool for time-bound authentication [6,7]. Discrete memristor based PUFs are also being proposed by several authors [7, 8]. For example, the PUF design proposed by Rose et al. [8] uses the write-time variability of memristors. Write-time for a memristor can vary due to physical randomness, and by using a mean write-time one can introduce uncertainty of the state of a given memristor. Therefore, a cell's physical unclonability is derived from its uncertain resistive state for a given write-time. Che *et al*. introduced a memristive PUF design that uses the random resistance distribution in a crossbar array as the source of entropy [9].

Identity and entity authentication depends on the sharing a secret between two parties- the verifier and the claimant. Usually, the verifier authenticates the claimant based on a secret that can be derived from (1) something that is known by both parties (such as passwords), or (2) something possessed by the claimant (such as hardware keys), or (3) something inherent (such as signatures, biometric signals *etc.*) of the claimant [10]. According to Menezes et al. common properties of authentication protocols includes: "*reciprocity of identification, computational efficiency, communication efficiency, nature of security guarantee and the nature of security storage"* [10].

Passwords are the most commonly used user authentication mechanism where the authenticator stores the (user-name, password (or its hashed value)) pair for different claimant and use this pair to identify a claimant. To strengthen this protocol several steps can be taken such as: (i) different password rules can be introduced, (ii) password salting can be performed, or (iii) password mapping can be slowed down which will make it difficult for an attacker to test large number of trial passes [10]. Common attacks on password schemes are replay attack and exhaustive and dictionary password search. Furthermore, leaking of an authenticators database containing (user-name, password) can cause significant threat.

For all of these weaknesses of password schemes, challenge response identification becomes a step toward strong authentication where, the authentic parties share sequence of secret one time passwords or challenge response pairs which is usually derived from some one-way functions or a challenge-response table. Furthermore, zero-knowledge protocol that verifies an entity through its possession of the knowledge of the secret instead the exact secret is also a strong authentication scheme [10].

However, applying cryptographic schemes in an untrusted or weak hardware platform can thwart all of the benefits earned. Also, cryptographic schemes can become energy-hungry and for systems with energy constrains such as low power sensor nodes *etc*. Hence in this work, we present simple authentication techniques using emerging resistive memory system and explore the feasibility of using these devices for hardware security.

III. Preliminaries of RRAM

A. Device Model

Metal oxide resistive memories are fundamentally thin-film based metal-insulator-metal (MIM) devices. We choose to demonstrate our work and perform the experiments on the HfO_x-based resistive memories commonly known as resistive–RAMs (RRAM). These devices can also be considered as memristors. However, they have certain design and characteristic differences from the TiO_2 based memristors developed by HP Labs. HfO_x-based RRAMs have a prominent future as a universal memory due to their high packing density, large ON/OFF ratio, multibit storage capacity and good switching endurance ($>10^6$ cycles) [1]. As we will show later, HP's memristors also possess the features we need for the proposed user authentication protocols. So our approach is not limited to HfO_x-based RRAMs.

Conductive filament formation in the HfO_x thin-film and electron tunneling in metal-insulator boundary determine the basic device properties of these RRAMs. The analytical model of the thin-film formation in HfO_x-based RRAM can be described using the following equation as given in [11]:

$$\frac{dg}{dt} = v_0 e^{-Ea,m/kT} \sinh(\frac{qa\gamma V}{LkT}) \quad (1)$$

where g is the state variable for the device which represents the spatial distance between the conductive filament in the oxide and the metal boundary, q is the electron charge, L is the device filament thickness, V is the applied voltage, T is the device temperature, $E_{a,m}$, γ, v_0 are device dependent physical parameters. The current-voltage relationship in the device is given in the following equation [11]:

$$I(g,V) = I_0 e^{-g/g_0} \sinh(\frac{V}{V_0}) \quad (2)$$

where I_0, V_0, g_0 are device dependent physical parameters. It should be noted that equation (1) and (2) have similar features that one would expect from a memristive system. For this reason, according to Chua, one can consider this device as a memristor [4].

In a crossbar memory array, an RRAM unit can contain a single resistance (1R configuration) or a transistor and a resistance (1T1R configuration). Since 1T1R configuration have immunity to sneak path currents and can deliver more reliable performance [1], we have used this configuration in this work. This basic configuration is shown in Figure 1.

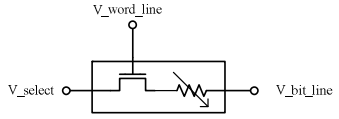

Fig 1. 1T1R configuration of a RRAM cell.

B. Selected RRAM Features

We list intrinsic hardware properties of resistive memories that are useful to design the proposed user authentication protocols and other security primitives.

F1. Non-volatility: Resistive switching memories are non-volatile in nature. Without the application of sufficiently large bias, the resistance of the given RRAM remains either in a low resistive state (LRS) or a high resistive state (HRS), and thus the device stores the information. Moreover, the initial states of a given distribution of RRAM population can be random. This random distribution along with the non-volatile nature of the memory can be used as a seed for generating secure random keys.

F2. Bias dependent write-time: Write-time of a resistive memory device can be defined as the time required for switching a device from one resistive state to another (*i.e.,* from LRS to HRS or from HRS to LRS). It was found that the write-time of an RRAM is highly dependent on the bias voltage applied for performing a write operation [1,11], and by adjusting the bias voltage, one can manipulate the write-time required for a given state transition. The aforementioned memristor-based PUF approaches have utilized this feature [8].

Further control and reliability on the filament growth can be achieved by employing pulsed bias voltages, where the duration of the pulse (t_p) is a fraction of the complete write-time (t_{WT}).

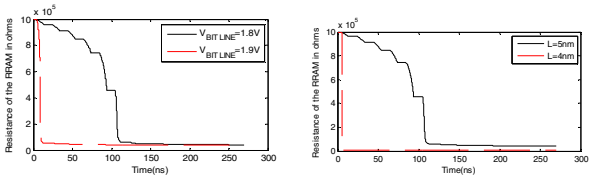

Fig 2. (a) Variation of RRAM write-time as a function of bias voltage. We have used the 1T1R configuration as shown in figure 1 and applied 1V 50MHz square wave at V_word_line, 0V at V_select, and, 1.8 and 1.9V 50MHz square wave at V_{bit_line} node respectively, (b) Variation of RRAM write-time as a function of filament thickness. We have applied a 1V 50MHz square wave atV_word_line, 0V at V_select, and 1.8V 50MHz square wave at V_{bit_line}.

We have performed simulations to validate this feature. For example, Figure 2(a) depicts an RRAM's write-time with two different bias voltages. We can see that when the RRAM switches from HRS to LRS, the write-time will be significantly longer for the low bias voltage case.

F3. Fabrication variation in filament thickness: Physical properties of an RRAM vary significantly with the fabrication variation in the filament thickness. For example, equation (1) shows that the change in resistive state is proportional to the *sinh* function of the inverse of the film thickness *L*. Therefore, a small change in film thickness can lead to significant measurable change in the resistance of the device. Such random variability resulting from fabrication variations can be exploited to design security features such as device authentication. As shown in Figure 2(b), the write-time of HRS to LRS transition is long for RRAM with large filament thickness.

F4. Non-linearity: From the basic equations presented in the previous sections, it can be seen that the resistive-switching devices are highly nonlinear in nature. Device nonlinearity is a highly sought after property for secure hardware design, and therefore, harnessing this property will provide novel implementation of common hardware security protocols such as PUFs. This feature is confirmed in our experiments and can be seen from the figures presented in this section.

IV. User Authetication Protocols Based on RRAM

In this section, we describe how to use RRAM to implement the classic username-password authentication protocol where a system (Alice) verifies the password of a user (Bob). We will first propose and validate the RRAM system requirements, then describe several basic RRAM utility functions and user authentication protocols. Next, we will show the design of the RRAM authentication circuit, and conclude this section with some discussion about the protocols and their implementation. For simplicity, we focus on a single RRAM (or verifying one bit).

A. RRAM Device Requirements

We specify the device requirements (or assumptions) that are needed for the design of the user authentication protocols and also give the justification of these requirements and assumptions.

R1. Unobservability of the intermediate resistance during resistive state transition.

As we have seen in RRAM's non-linearity feature *F4*, when the RRAM device transitions from one resistive state to another, its intermediate resistance does not change in a linear fashion with time. Instead the change is exponential in nature and has random variations due to local Joule heating and stochastic nature of ionic transportation through the thin-film. The can be verified from the I-V relationship of RRAM [1]. Also, manifestation of this can be seen from our experiment in Figure 2. For example, in Figure 2, for the case with V_{bit_line}=1.8V, the device starts at HRS (t=0ns) and completely switches to the LRS after 150ns. The change from HRS to LRS happens abruptly at around 80-100ns and after that the device settles in LRS. To observe the resistance value during the transition and use that value to infer the write-time or the applied bias voltage will be impossible. For example, from Figure 2(a) we can see that to reach the same resistance value, we can use high bias voltage with short write-time, or low bias voltage with long write-time. And from Figure 2(b), we know this also depends on the filament thickness (feature *F3*).

R2. Use of pulse train, sequence of short duration pulses, for state transition.

When given sufficiently high frequency and proper amplitude, these short duration pulses can cause RRAM resistance state transitions just like a longer single voltage pulse. As we can see from equation (1), with V=0, the rate of change of the state is zero, so we would expect a device to retain its resistive state for zero input voltage. Therefore, the device should need the same amount of time with the bias voltage on when it is written using multiple short pulses or a single long pulse. For example, if a device requires a 1.5V-100ns pulse to go from HRS to LRS, it should show similar

transition when ten (or similar number of that order) of 1.5V-10ns pulses are applied. This is validated in our simulations as well.

R3. Robustness of the number of short duration pulses to cause state transitions.

As we have analyzed in above, a number of short duration pulses are required for a state transition at a given bias. We assume that this number remains approximately the same against environment variations such as noise, device aging, and temperature. In addition, such numbers for different RRAMs should be same when these devices are in close proximity, *e.g.* neighboring RRAMs in a crossbar architecture, but may vary randomly when they are far apart. This is based on the locality of the fabrication variation and needs to be verified experimentally.

R4. Unobservability of the number of short duration pulses during the state transition.

This is the discrete version of requirements *R1* under the assumption of *R2*. In another word, one can not imply how many short duration pulses have been applied by observing the intermediate resistance value during the state transition. Furthermore, to learn the exact/approximate number of pulses required for a given state transition at a given bias, one need to observe a complete programming cycle. At a given bias, it would be impossible to extrapolate the number of pulses required by measuring the resistance change of the device for only small period during a transition.

B. Basic RRAM Utility Functions

Before elaborate our proposed RRAM based user authentication protocols, we first describe the basic RRAM utility functions that are critical for these protocols.

RESET: A RESET operation uses a fast negative pulse to put an RRAM to the high resistive state (HRS).

SET: A SET operation puts an RRAM to the low resistive state (LRS). For authentication purpose, we will use the bias dependent write time variability during set (transition from HRS to LRS) operation. We will use multiple short duration ON pulses for setting the device instead of a longer ON pulse.

SET Pulse Count (SPC): The number of ON pulses required to transit an RRAM from HRS to LRS.

Pre-condition: A k-bit pre-condition (k<SPC) operation applies k consecutive ON pulses to the RRAM device.

Read Pulse (Write State): A read pulse operation consists of two steps: (i) Pre-condition the RRAM with *k=SPC-1* ON pulses, (ii) apply the incoming pulse train in the next cycle to detect whether it put the RRAM at LRS (which indicates the applied pulse was ON) or keep the RRAM at HRS (which implies that there is no pulse in the past cycle). Note that this operation has the potential to change the state of the RRAM, thus we also refer to this operation as **write state**.

Read State: No pulse for one cycle. Then detect whether the RRAM device is at HRS or LRS.

Copy State (S, C): This operation "copies" the approximate state information from one RRAM S device to another RRAM C. We use the phrase "approximate" because one cannot make an identical copy of the analog resistance value of an RRAM. The copy created from this operation behaves the same as the original RRAM when the short duration pulses are applied. A copy operation (from RRAM S to RRAM C) has the following steps:

```
S_state = Read State (S);
if (S_state == LRS){SET (C); done;}
k=0;
while (S_state == HRS)
do {apply one ON pulse to RRAM S;
    S_state = Read State (S);
    k++;}
Pre-condition (C, SPC-k);
Pre-condition (S, SPC-k);
```

The goal of the while-do loop is to determine the state, at the cycle-accurate level of short duration pulse, of the original RRAM S. It is worth noting that we need to copy this original state of RRAM S to both RRAM C and itself because its original state has been changed when we repetitively apply ON pulse on RRAM S to detect its state. The Pre-condition(C or S, SPC-k) operation guarantees that S and C will response in the same way as the original RRAM S would do to any incoming pulse train.

C. Authentication Protocols

The problem is how the system (Alice) can take advantage of RRAM devices to authenticate a user (Bob) by the classical username-password protocol. That is, how Alice can verify the password provided by Bob. In our case, Bob's password will be an input pulse train. We describe two authentication protocols here to illustrate the basic ideas. The analysis of these protocols and several variations of them will be discussed later.

Protocol 1.

i. Password registration: for the i^{th} cycle in the pulse train (Bob's password), store it in RRAM R_i by, for example, the following scheme
   a) if the pulse is ON, **RESET** (R_i) to HRS;
   b) if no pulse, **SET**(R_i) to LRS.

ii. Password verification: for the i^{th} cycle in the pulse train
   a) pls = **Read Pulse**;
   b) state = **Read State** (R_i);
   c) if ((pls is ON && state is HRS) || (pls is no pulse && state is LRS)) the i^{th} cycle in the pulse train is correct;
   d) else the i^{th} cycle in the pulse train does not match Bob's;

Protocol 2.

i. Password registration: for the i^{th} cycle in the pulse train (Bob's password), store it in RRAM P_i by as follows
   a) assign R_i a random initial resistance level;
   b) **Copy State** (R_i, P_i);
   c) apply the i^{th} cycle in the pulse train to P_i;.

ii. Password verification: for the i^{th} cycle in the pulse train
   a) **Copy State** (R_i, Q_i);
   b) apply the i^{th} cycle in the pulse train to Q_i;
   c) Q_state = **Read State** (Q_i);
   d) P_state = **Read State** (P_i);
   e) if (P_state and Q_state are the same) the i^{th} cycle in the pulse train is correct;
   f) else the i^{th} cycle in the pulse train does not match Bob's;

It is trivial to see the correctness of both protocols. Next, we show the circuit implementation of these protocols and then discuss their variations with enhanced security features.

D. Authentication Hardware

For developing a RRAM based authentication circuit, we need to extract the current state of a given device. It should be noted that, improper reading of an RRAM cell with high bias voltage across the cell can change its states, and therefore, the read operation should be done carefully. To accomplish this, we have used a simple CMOS compatible differential sense amplifier. The resistance of the device R_x is compared with another resistance R_in. If R_x<R_in the circuit outputs 1, else it outputs a zero. The basic circuit is shown in figure 3.

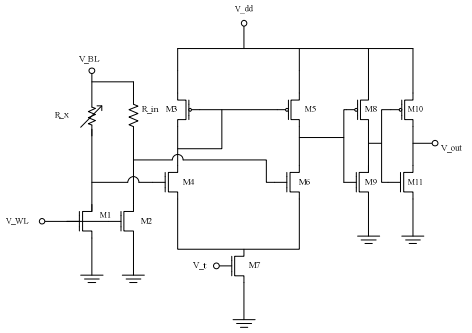

Fig 3. RRAM and CMOS logic-based authentication unit. The difference in voltage due to the difference in R_x and R_in values is amplified in the next stage using a differential amplifier. The output of the amplifier is applied to a buffer for attaining proper logic values.

E. Protocol Analysis

One of the most promising advantages of our proposed user authentication protocols is that they don't need the computational expensive cryptographic techniques (such as performing the modular exponentiation operation for large numbers of hundreds of bits). In that sense, our approach can be considered as a lightweight cryptography technique for authentication. RRAM device's intrinsic energy efficiency (storing data when power is off) makes this even more attractive. We now consider the potential attacks to our proposed user authentication protocols and how to defend again these attacks.

1. **Password stealing.** In Protocol 1, Bob's password (ON pulse or no pulse) is stored in RRAMs in the form of their resistance state (HRS or LRS, respectively). If an attack manages to observe these states, he can retrieve the pulse train that is Bob's password.

There are two simple solutions for this. First, standard cryptographic methods such as encryption and hash can be used so these RRAMs will not store the plain pulse train information. Instead, the encrypted version or hash results will be stored. When the attacker gets access to such information, he will not be able to reconstruct the pulse train and corresponds to Bob's password unless he can break the encryption scheme and the one-way hash function. However, this requires the system to have a unit that does encryption and/or hash and can convert the pulse train into digital format (or binary 0's and 1's).

Another solution is what we have proposed in Protocol 2, where we keep the RRAM's response to the input pulse (the state of the RRAM P_i). Based on device requirement *R1* and *R4*, the attacker may be able to obtain whether P_i is at HRS or LRS, but he won't be able to find the cycle-accurate state information of these RRAMs. Thus, he cannot reconstruct the pulse train which, when applied to RRAM R_i, will change R_i to P_i.

2. **Password guessing**. Attackers may launch the exhaustive password guessing attack or some more advanced version based on RRAM's special features (for example, applying certain pulse train such as all ON and then observe the states of the RRAM devices to reveal password information).

This should not be a big concern because it can be easily prevented by disabling the username after certain number of consecutive failures. Alternatively, Alice can create her own pulse train and combine it with the input pulse train to validate whether the input is Bob's password. As a result, the states of the RRAMs P_i and Q_i will depend on both Bob's password and Alice's pulse train for each particular password authentication. Alice can change her pulse train every time and this will not leave any clue about the Bob's password or the state information of the original RRAM R_i to the attacker.

3. **Password collision (false negative)**. This refers to the case when different passwords are deemed to be authentic for one user.

This is a problem of password selection. One standard solution is to add salt like the UNIX password management system does. When users choose their passwords, the system (Alice) can append/combine certain unique data to the user selected pulse train. This will ensure that uniqueness of each user's password. However, the system needs to store the salt data associated with each user.

4. **False positive alarm**. This is the case when an authentic password is declined. It is more of a reliability of the proposed protocols than security breaches. We will elaborate this in the experiment section.

Here we point out that environmental variations such as temperature, noise, and device aging may all create false positive alarms. We don't expect this to be a serious problem because we do authentication at cycle-accurate level. Furthermore, like most of the biometric authentication schemes (such as fingerprint or face recognition), it is possible to authenticate the password with a non-perfect match. That is, we don't need to match every pulse in the pulse train to authentic the password.

5. **Denial of service**. In this attack, the attacker attempts to alter the state of the RRAM devices that store the password related information such that when the legitimate user enters the authentic password, the system will not be able to verify that.

Due to the large amount of RRAM devices in the authentication architecture, it may not be easy for the attacker to launch this attack to a specific victim unless the attacker correctly identifies which set of RRAM devices correspond to the victim. However, the attacker may succeed to attack random victims.

To countermeasure such attack, we can restrict user's access to the RRAM devices. For example, in Protocol 2, we

can disable the **Write State** utility to RRAM R_i and P_i for protection purpose. This can be done by not allowing input pulse train to be applied to these RRAM devices.

6. **Side channel attack**. This is a group of powerful attacks that targets the vulnerabilities in hardware implementation of the security primitives and protocols. By measuring the side channel information (such as power, timing, and EM emission) during system's execution, secret information of the system can be revealed.

We are not aware of any side channel attacks to RRAM based systems. But should such attacks emerge, most of the existing countermeasures (such as careful engineering, using redundancy, design obfuscation, and so on) should be applicable.

In summary, when we assume that an outside attacker is only allowed to provide pulse train as password for authentication and can measure the state changes at a fixed time during the cycles, the attacker can gain very little information and our user authentication protocols will be strong and secure. However, if the attacker gains physical access to the RRAM devices that we use to store user's password related information, there will be a good chance for malicious attacks such as denial of service.

V. EXPERIMENTS

We have conducted experiments on the performance of the proposed circuit and protocols and explored their reliability over physical and environmental variation. For these experiments we have used the Verilog-A model of RRAMs proposed by Guan *et al.* [**Error! Reference source not found.**]. This is a variability aware RRAM model that takes account of the critical impact of temperature change and temporal variations [**Error! Reference source not found.**]. We have used PTM's 65nm MOSFET models [**Error! Reference source not found.**] for designing the authentication unit. The simulations are performed in HSPICE platform. For calculations presented in this section we have used the parameters listed in Table 1.

TABLE I. Common parameters used for the experiments in this work.

Parameter Name	Value	Parameter Name	Value
Clock frequency (f_{clk})	50MHz	V_WL_RESET	2.8V
V_SET	1.8V	V_RESET	1.9V
V_WL_SET	1.0V	R_in	250k

For the protocols discussed in the previous section, any of the generated response of the RRAM authenticator is completely dependent on the choice of bias voltage and the hardware used. Therefore, the reliability of the protocols will be dependent on the reliable performance of the authentication unit over time. Here, we have analyzed the common causes for which an authentication unit will fail to produce the correct response. It should be noted that these weaknesses are mostly due to the imperfect operating conditions of RRAMs and by addressing these issues one can develop reliable authentication mechanisms in resistive memory platforms. Primary concerns for the reliability of the proposed protocols are listed below:

Unbalanced Set-Reset: The resistive levels of the RRAM devices can change due to unbalanced set-reset pulses over a long period of programming cycles, and hence, one of the major sources of error in the authentication protocols can be unbalanced set-reset cycle. If the RESET pulse pulls up the resistance more than the previous SET pulses, then the overall resistance of the HRS will drift away with time. An example of this effect is illustrated in Figure 4 where we show that due to unbalanced reset, the resistance of the RRAM drifts to HRS over multiple cycles.

To examine the reliability of the circuit, we define the a parameter called ***SET/RESET mismatch*** as the following: Let's assume that for a given pure LRS it takes a v_o volt pulse with τ ns duration to RESET it to a pure/defined HRS. We define τ as the ideal reset time. Now, a mismatch can be quantified as the difference between the ideal reset time (τ) and the actual reset time (τ') while the bias v_o remains constant.We can find out the reliability of a circuit (in our case the authenticator unit) by measuring the number of reliable operations it performs before erroneous bit flip is seen at the output (V_out) due to SET/RESET mismatch. In Figure 5, we have shown the number of reliable operation that can be performed with the authenticator unit as the ***SET/RESET mismatch*** increases.

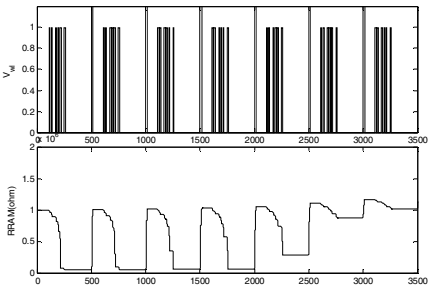

Fig 4. Demonstration of the cycle-to-cycle resistance drift due to unbalanced SET/RESET. The upper figure shows the gate voltage applied at V_WL node (for the circuit shown in figure 4) over multiple cycle for performing set and reset operation and the lower figure shows the resistance of the RRAM tend to settle at HRS with time. The x-axis of both the figure is in nanoseconds unit.

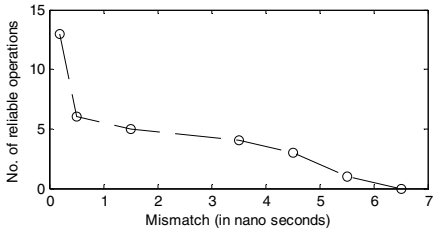

Figure 5. SET/RESET mismatch (in ns) vs number of reliable cycles for fast (~15ns) reset pulse.

To fight error from unbalanced SET/RESET, the authenticator can perform a full SET/RESET operation after every authentication and RESET operation and verify whether the total number of pulses for a SET remains constant.

Another error correction technique is, if the authenticator sees the loss/gain of 1 pulse for SET operation after *n'* operation, then a correction factor 1 can be subtracted/added to the pre-conditioning or the hashed value. However, after a bit flip is observed, the best solution would be to recalibrate the device using a reference resistance and the Copy State utility function given in section 4.2.

It can be seen that with fast reset the number of reliable operations before reset correction is small. Therefore, slower reset with longer pulses and smaller bias voltage is preferable.

Temperature dependence of filament formation: Experimental evidences report that the conductive filament formation in a given device depends on the device temperature [**Error! Reference source not found.**]. This can also be seen from equations (1) and (2).

Fast SET/RESET operation and on-chip temperature can increase the operating temperature of the device and thus change the response of the unobservability function discussed in this work. We have presented the effect of increasing temperature on the reliability of the authentication unit in Figure 6. The reliability is calculated as the ratio of the correct output bit flips to the total number of input bits.

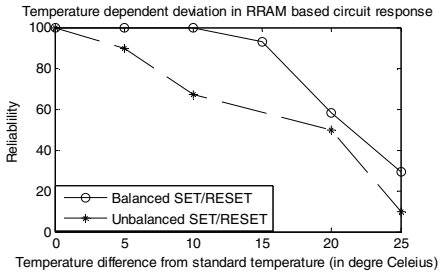

Fig 6. Circuit reliability with the variation in temperature. For the figure, we have computed the reliability of the circuit over 10 consecutive programming cycles with each cycle containing SPC=10.

It was found that a device that goes through balanced SET/RESET usually is more reliable with the variance in temperature than the one having unbalanced operation. Therefore maintaining proper operating temperature for the circuit with balanced SET/RESET will be very critical for reliable operation of the proposed protocols. It should be noted that the result shown in Figure 6 is for 10 consecutive programming cycles; and therefore one should be careful to use this result to measure the reliability in cycle-only operation.

Spatial variations in filament formation and random filament formation: During SET/RESET process lattice temperature of the filament increases abruptly which gives rise to random non-uniform filament generation. This non-uniformness of filament generation will introduce random fluctuations resistive values during the intermediate states. Overall this would result cycle-to-cycle write time variability in the device. To reduce this cycle-to-cycle write-time variability, one can use the same techniques discussed for unbiased SET/ RESET case.

Noise: The primary feature of our proposed design is to use the bias dependent write-time variability. However, noise in the writing signal can cause cycle-to-cycle variations in the number of pulses required for performing a given write. Noise on the SET or RESET line can profoundly impact the reliability of the circuit therefore; care should be taken to remove the bit-line noise.

Aging: Proper model of aging of RRAMs is still unavailable. However, with aging the response of the circuit would certainly change and proper aging model and error correction mechanism is required to combat this phenomenon.

From our observation on the effect of variation in simple resistive memory based security hardware, we recognize the opportunities in the following areas for future research.

1. Novel circuit designs are required for reducing the supply voltage noise across a memristor/RRAM used in security hardware.
2. New simple error correction techniques need to be developed to fight predictable variation.
3. Novel algorithmic techniques and supporting hardware are required for developing better state read-out and copy operation of memristive devices for hardware security.

ACKNOWLEDGMENT

The authors would like to thank Mr. Nathan McDonald and Dr. Lok Yan for their helpful discussion and comments. This project was supported in part by Air Force Research Laboratory under agreement number FA8750-13-2-0115 and AFOSR MURI under award number FA9550-14-1-0351.

REFERENCES

[1] H.-S. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. Chen, and M.-J. Tsai, "Metal #x2013;oxide rram," Proceedings of the IEEE, vol. 100, no. 6, pp. 1951–1970, June 2012.

[2] K.-H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, and W. Lu, "A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications," Nano Letters, vol. 12, no. 1, pp. 389–395, 2012.

[3] S. H. Jo, K.-H. Kim, T. Chang, S. Gaba, and W. Lu, "Si memristive devices applied to memory and neuromorphic circuits," in Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. IEEE, 2010, pp. 13–16.

[4] L. O. Chua, "Resistance switching memories are memristors," Applied Physics A, vol. 102, no. 4, pp. 765–783, 2011.

[5] Memristor. [Online]. Available: http://en.wikipedia.org/-wiki/Memristor

[6] M. Arafin, C. Dunbar, G. Qu, N. McDonald, and L. Yan, "A survey on memristor modeling and security applications," in Quality Electronic Design (ISQED), 2015 16th International Symposium on, March 2015, pp. 440–447.

[7] J. Rajendran, R. Karri, J. B. Wendt, M. Potkonjak, N. R. McDonald, G. S. Rose, and B. T. Wysocki, "Nanoelectronic solutions for hardware security." IACR Cryptology ePrint Archive, vol. 2012, p. 575, 2012.

[8] G. S. Rose, N. McDonald, L.-K. Yan, and B. Wysocki, "A write-time based memristive puf for hardware security applications," in IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE, 2013, pp. 830–833.

[9] W. Che, J. Plusquellic, and S. Bhunia, "A non-volatile memory based physically unclonable function without helper data," in Computer-Aided Design (ICCAD), 2014 IEEE/ACM International Conference on, Nov 2014, pp. 148–153.

[10] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, Handbook of applied cryptography. CRC press, 1996.

[11] X. Guan, S. Yu, H.-S. Wong et al., "A SPICE compact model of metal oxide resistive switching memory with variations," IEEE Electron Device Letters, vol. 33, no. 10, pp. 1405–1407, 2012.

[12] Predictive technology model [online]. Available: http://ptm.asu.edu/ .