Secret Sharing and Multi-user Authentication: From Visual Cryptography to RRAM Circuits

Md Tanvir Arafin
Department of Electrical and Computer Engineering
University of Maryland, College Park, USA
marafin@umd.edu

Gang Qu
Department of Electrical and Computer Engineering
University of Maryland, College Park, USA
gangqu@umd.edu

ABSTRACT

In this era of Internet of Things (IoT), connectivity exists everywhere, among everything (including people) at all times. Therefore, security, trust, and privacy become crucial to the design and implementation of IoT devices [12]. However, it is challenging to build security into IoT devices because most of them are constrained by extremely limited resources such as the battery, memory, and computation power *etc*. Inspired by the concept of visual cryptography [4] that requires the least amount of computation and a recent work on pure hardware-based single-user authentication [6], we present a novel solution to the secret sharing and multi-user authentication problem. Our solution is built on the observation that non-volatile resistive memories display nice monotonic and additive properties during resistive state transitions. We demonstrate how to design a hardware dependent multi-user authentication protocol using resistive random access memory (RRAM)-based hardware and provide the necessary circuits for the application. Finally, we simulate the proposed circuit to understand the nature of the operation and practical problems that these designs encounter during operation.

CCS Concepts

• Security and Privacy → Security in Hardware → Hardware Security Implementation → Hardware-based security protocol.

Keywords

Secret Sharing; Visual Cryptography; Multi-User Authentication; Resistive Memory.

1. INTRODUCTION

With the advent of the Internet of Things (IoT), where more and more devices are connected by the Internet infrastructure, there is an increased demand for ultra-low power for computing and communication. Meanwhile, the pervasive nature of the IoT devices in modern life ranges from medical devices to critical infrastructures, and the fact that these devices often need to collaborate to accomplish the desired mission, make security, trust, and privacy new important design objectives for the IoT devices [12]. However, solutions based on modern cryptography are computationally expensive and not suitable for the extremely resource constrained IoT devices. As a result, security and privacy are becoming the *Achilles' heel* for IoT system design and implementation. For example, most of the wearable and implantable medical devices (such as pacemaker and insulin pump) communicate without any encryption or protection. Therefore, research on novel low-power security primitive is imperative for a safe and secure realization of IoT applications.

In [6], we have proposed a couple of resistive random access memory (RRAM) based lightweight user authentication protocols. In this work, we consider the problem of how to design and implement simple and lightweight secret sharing and multi-user authentication scheme for IoT devices. Secret sharing is a well-studied problem, which seeks to distribute pieces of information (or called the secret) to multiple parties in a way such that the information (the secret) can be revealed when all or a sufficiently large subset of the parties contribute their shares. Shamir's secret sharing algorithm [1] defined the concept of *threshold scheme* for secret sharing. This so-called *(k,n) threshold scheme* (where $k \leq n$) can be described as follows:

> Assume that a secret S needs to be shared by n parties. The secret is divided into n pieces such that, the knowledge of k or more pieces would be sufficient to reconstruct S. However, if the knowledge of any $k-1$ pieces or less is available, it would be impossible to reconstruct S.

A direct application of this problem is the multi-user authentication problem, where at least k authentic users must present to gain the access to a system. One example of this is the "*two person rule*" originally designed to prevent the accidental or malicious launch of nuclear weapons by a single individual [13] and later on enforced by NSA for the access to sensitive operation and information after the infamous Snowden case [14]. One may argue that authenticating each individual user and counting the authenticated users can trivially solve the multi-user authentication problem. This does solve the problem but has two significant drawbacks in scalability and user privacy. First, the expensive authentication protocol has to be applied at least k times and some mechanism to check duplicate users must be implemented. Second, unlike the traditional secret share-based approaches [1-4], this method will reveal the identity of each authenticated user, creating user privacy concerns.

Our goal is to perform multi-user authentication without using any computation intensive cryptographic operations so it can be applied to IoT applications. Our approach is inspired by the *Visual Cryptography scheme* proposed by Naor and Shamir [4], which also completely avoids cryptographic calculations but is limited to secret image information. We design resistive random access

GLSVLSI '16, May 18-20, 2016, Boston, MA, USA

© 2016 ACM. ISBN 978-1-4503-4274-2/16/05 $15.00

DOI: http://dx.doi.org/10.1145/2902961.2903039

memory (RRAM) based circuits to demonstrate the concept of hardware-based multi-user authentication.

The rest of this paper is organized as follows. In the next section, we survey the classic solutions, which unfortunately are computationally expensive, and review the fundamentals of RRAM as well as the RRAM based authentication protocols proposed in [6]. In Section 3, we discuss the basic ideas of secret sharing, then elaborate our multi-user authentication protocol, and describe the proposed RRAM circuit. We report simulation results in Section 4 before concluding in Section 5.

2. RELATED WORK

In this section, we provide the background on classical secret sharing schemes, RRAMs and recently proposed RRAM-based single user authentication protocols.

2.1 Classical Secret Sharing Schemes

Shamir [1] and Blakley [2] independently proposed solutions for the aforementioned problem of *(k,n)-threshold scheme*. More specifically, Blakley used techniques from finite geometry to provide a solution for safeguarding and sharing cryptographic keys. Shamir's solution is based on the polynomial interpolation over a finite field. It has since become a widely acceptable solution for secure secret sharing and distribution of cryptographic keys. A detailed explanation of the scheme and the impact of subsequent works can be found in [3].

One of the drawbacks of these early solutions is their high computational cost. For example, Shamir's secret sharing algorithm requires calculations over the finite field during both secret share generation phase and secret reconstruction phase. This, in turn, requires complex digital circuitry for hardware implementation, which is known to be more efficient than the software implementation. The *Visual Cryptography scheme* proposed by Naor and Shamir [4] is an alternative lightweight scheme for secret sharing to Shamir's original scheme. In visual cryptography, the secret shared among multiple parties is simply an image. The secret image is broken into *n* pieces and each piece is printed on a transparency. When *k* or more pieces of these transparencies are placed as a stack, the secret image is revealed and can be comprehended by human eyes. In this scheme, the secret share generation needs only very simple calculations and the revelation of the secret image does not involve any mathematical computation. In essence, this is an example of how inherent physical properties of hardware can be used in designing lightweight security primitives to avoid complex mathematical computation and formal cryptography protocols.

One of the many interesting applications of secret sharing is the *simultaneous multi-user authentication* problem. Naor and Pinkas first discussed the application of visual cryptography for authentication and identification [11]. When an entity (say Alice) tries to simultaneously authenticate *k-out-of-n*-other entities (B_1, … B_k), the problem turns into simultaneous verification of *k-out-of-n* cryptographic keys (*i.e.,* passwords). If these keys are generated from a secret that Alice knows or possesses, then, instead of checking *n*-keys separately, Alice can use the keys provided by B_1, … B_k to regenerate her secret and authenticate all entities at once. Note that, as Alice uses a *k-out-of-n* secret sharing scheme, Alice can reconstruct the secret from any of the k-users shares. To avoid collusion among the users, this procedure requires Alice to have some interference in key/password generation process during the registration of the users.

2.2 Hardware Dependent Secret Sharing and Properties of RRAM

In this work, we discuss hardware dependent secret sharing and multi-user authentication ideas that require minimum computational effort during authentication. We explore on how to create hardware-dependent or hardware-biased secrets that can only be revealed by that hardware used during authentication. Physical variations in electronic devices due to fabrication variations can be exploited to create hardware specific secret-sharing process. Therefore, computational hardware would not only be the engine for computation, but also an active party in the secret sharing and authentication purposes. We have demonstrated our ideas using multivalued RRAM-based circuits.

Resistive random access memory (RRAM) is a promising candidate for non-volatile solid-state memory. RRAM devices have a very simple geometry- in most cases, a metal-insulator-metal (MIM) structure. In this work, we are focusing on HfO_x-based RRAMs. These devices have high packing density, multibit storage capacity, large ON/OFF ratio, and good switching endurance (>10^6 cycles) [10]. Device properties of RRAMs are dependent on the dynamics of the conductive thin-film formation in the HfO_x layer. This is a highly non-linear process, which leads to non-linear properties of these devices. Recent works have shown that unique properties of resistive non-volatile memories can be useful in designing security primitives. A survey of these works can be found in [5].

For this work, we have considered the 1T1R-configuration of RRAM devices, where an access transistor is used for controlling the voltage (and current) across the RRAM device as shown in Figure 1.

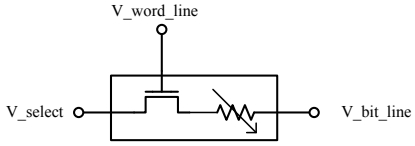

Figure 1. One-transistor-one-resistor (1T1R) configuration of an RRAM cell.

Non-volatile memory elements such as RRAMs can be used to remember different resistive states. By applying voltage/current pulses through the memory elements these resistive states can be changed. We define ON pulse as a small duration voltage pulse (applied to the gate of the 1T1R configuration) that can reduce the resistance of a given RRAM. The duration of an ON pulse is a magnitude lower than the write-time of the device, therefore, a single ON pulse cannot produce a noticeable change in the resistance of a device that is in a high resistive state. However, with proper V_select and grounded V_bit_line, a collection of ON pulses can put an RRAM from a higher resistive state (HRS) to a lower resistive state (LRS). It has been demonstrated that using pulse train- a sequence of ON pulses, one can control the HRS to LRS transition [6]. This pulse-controlled transition of the RRAM's resistive state has two interesting properties:

P1. Changes due ON pulses are "*additive*" in nature. For example, two consecutive ON pulses would put the device in a lower resistive state than a single ON pulse.

P2. The memory elements are *monotonic*, *i.e.,* applying an ON pulse will always decrease the resistance and its effect cannot be undone without resetting the device.

After applying input pulses, logical decisions can be made based on whether an element's resistance is above or below a given threshold. This is the key idea of the threshold detector circuit described in later sections. The "*additive*" property along with the *monotonicity* can be useful for designing cryptographic primitives. Since the properties described above are also common for memristive devices, this work can be extended to other designs of resistive memories.

2.3 RRAM-based Authentication

Recently, we have demonstrated the application of RRAM-based circuits and protocols for single user authentication in [6]. For RRAM-based single user authentication, an entity Alice tries to authenticate Bob by using a voltage profile that Bob possess. Bob's voltage profile (*i.e.*, password) consists of a random sequence of ON pulses. A straightforward way for authenticating Bob is to digitize his voltage profile and save it the RRAM-based memory. Each time Alice wants to authenticate, she will read the provided voltage profile, digitize it and match it with the one she stored in the RRAM-based memory. However, this approach suffers when an adversary tries attacks such as password stealing from the database. To address such issues, we have developed device dependent authentication protocols in [6].

In hardware dependent single user authentication, Alice can only authenticate Bob using the exact device she used for registering him. Physical properties of the given hardware play the role of non-linear function that "*encrypts*" Bob's password. To design such protocols, we notice that along with the "*additive*" and *monotonic* properties, RRAM-based circuits also show unobservability property of the intermediate resistance during resistive state transition. The state-transition of an RRAM is dependent on the initial conditions, applied voltage profile or the physical construction of the device. Therefore, without knowing the device, the required initial conditions, and the correct voltage profile to apply, it is difficult to put a given RRAM in a certain intermediate state. Based on this key concept, we have designed simple authentication protocols in [6].

For RRAM-based authentication, during the registration phase, Alice preconditions the device to a certain state using her key. Then, Bob applies a segment of his voltage profile to the RRAM. This would put the RRAM in a certain intermediate state. Alice then *reads out* the intermediate state and stores it for later authentication. During authentication, Alice preconditions the corresponding device and let a user apply his voltage profile. If the user can successfully put the RRAM in the correct intermediate state, Alice recognizes the user as Bob. This scheme is secure from password stealing and guessing attacks since it requires exact hardware to be present, and the data stored (*hashed* value of Bob's password) in Alice's memory is dependent on the RRAM used.

This RRAM-based single-user authentication protocol is the basis for our solution to the multi-user authentication problem. However, one cannot apply this multiple times to authenticate each user for the following reasons: (1) scalability: For the *k-out-of-n* user authentication problem, this will require running the single-user authentication, at least, *k* times and additional circuit or memory to check for duplicate users. So it will not scale well for large *k* and n. (2) privacy: The *k-out-of-n* user authentication problems demand *k* authentic users, but it does not need the identification of each user. Simply applying the single-user authentication protocol *k* times will reveal the identity of the users and create privacy concerns.

3. MULTI-USER AUTHENTICATION PROTOCOL

3.1 Motivational Example

To address the problems regarding multi-user authentication (as discussed in the previous section), we have designed a *Visual Cryptography* inspired RRAM-based multi-user authentication scheme. A small motivational example for a 2-out-of-3-user authentication using RRAM-based threshold detector circuit is given below to illustrate the key concepts.

Assume that an entity Alice wants to authenticate three users B_1, B_2, and B_3. Alice chose a random key *K*, and for each bit of the key, for each user, Alice randomly choose a row from the following matrix C0 and C1 using the given rules and gives the corresponding 3-bits to the users. The matrices are given as:

C0 = all the matrices obtained by permuting the columns of $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$;

C1 = all the matrices obtained by permuting the columns of $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$;

and the distribution rules are:

- If the *i'th*-bit of the key is 0, distribute the rows of C0 to B_1, B_2, and B_3.
- If the *i'th*-bit of the key is 1 distribute the rows of C1 to B_1, B_2, and B_3.

Note that for a single user, it would be impossible to guess the key bit. For authentication, Alice let any 2 or more users to access the RRAM for the duration of three pulses. The users apply their keys in each pulse. The keys are taken through an OR gate. An ON pulse is applied at a cycle if the result of the OR gate is 1 otherwise no pulse is applied. If the *i'th* bit of the key is zero, then for only two of the cycles the users will give an ON pulse but there will be no ON pulse for one cycle. However if the *i'th* bit is 1, there will be three ON pulses. As a result, after the duration of three pulses, the device's resistance should be lower if the key was 1 and higher if the key was 0. A threshold detector can easily detect this and reconstruct the key. This scheme can easily be modified to an *n-out-n* or a *k-out of-n* authentication scheme.

One weakness with the above example is that to reconstruct a bit of the key, the users do not necessarily need Alice. If any two of the valid users come together, they can reconstruct the bits. If we want a hardware dependent authentication, where the participating hardware should also be a factor during authentication, this might cause a collusion problem. However, this problem does not exist if Alice has some interference in key generation. We have elaborated on this in later sections.

3.2 Authentication Protocol

In this section, we will discuss an RRAM-based multi-user authentication protocol. This hardware dependent protocol uses the core ideas of *Visual Cryptography* to ensure its security and completeness.

3.2.1 Key Generation and Registration

For generating and distributing hardware dependent keys, Alice first chose a random key X and a global bit-line voltage V_{BL}. For each bit of X (x_i), Alice uses distinct an RRAM R_i. Since Alice owns the hardware and controls the V_{BL}, Alice knows the number of ON pulses (NP_i) required for pushing the resistance of the

RRAM R_i from a fixed HRS state to the reference resistance (R_{in}). Therefore, R_i would be perceived to be in an LRS iff the number of applied pulses is greater than NP_i. Due to the fabrication variations, each RRAM would have different write-time, which will lead to different (but of same order) values of NP for different devices. Since Alice owns the device, Alice knows the value of NP_i for any given R_i at a given V_{BL}.

To generate keys for each user, Alice needs to construct the matrices C0 and C1 similar to the ones presented in the motivational example. For a *k-out-of-n* visual secret sharing these two collections of $n \times m$ Boolean matrices (*i.e.,* C0 and C1) is required. These two metrics can be constructed by the construction rules given in [4,7]. Here we have provided a rule from [4] for completeness of the discussion:

For a *k-out-of-k* scheme, let us consider a ground set $W=\{e_1,e_2,\ldots, e_k\}$ consisting of k elements. All subsets of W with even cardinality is denoted by $p_1, p_2,\ldots, p_2^{k-1}$ and all the subsets of W of odd cardinality is denoted by $q_1, q_2, \ldots, q_2^{k-1}$. To design a *k-out-of-k* scheme, Boolean metrics S0 and S1 of dimensions $k \times 2^{k-1}$ are required. From the constructions given in [4], we define, S0[i,j]=1iff $e_i \in p_j$ and S1[i,j]=1iff $e_i \in q_j$, where $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant 2^{k-1}$. Then, by permuting all the column of S0 and S1 metrics we can derive collections of C0 and C1 respectively [4]. This *k out-of-k* scheme can easily be expanded into a k-out-of-n scheme as discussed in [4] and [7].

Since different RRAM needs different numbers of ON pulses, additional 1s need to be padded in each row of C0 and C1. A simpler solution is to *precondition* each RRAM with this additional 1s. Since Alice knows the value of NP_i for any given R_i at a given V_{BL}, Alice can precondition each RRAM so that each requires the same number of additional ON pulses to reach LRS. Another way is to use the block-length defined constructs presented at [7]. Block-length is the number of ones resulted by *or*-ing all the columns of C1. Therefore, Alice can easily design block length adjusted matrices C0' and C1' depending on the number of ON pulses required for a state transition.

To share a bit (x_i) of the key X, Alice would follow the rules R1 and R2 which are given below:

R1. If the *i'th*-bit of the key is 0, distribute the rows of C0' to the users

R2. If the *i'th*-bit of the key is 1 distribute the rows of C1' to the users.

To make a hardware dependent authentication scheme, Alice can choose smaller block length or to pad less number of 1s at the end of C1' for some of the random bits of the key X. Since the users do not own the hardware, they do not know the exact value of NP_i's for a given R_i. Therefore, Alice can share a 0 by sharing contents of C1' for cases where C1' is not properly generated from C1. Thus, it would be impossible for *k*-users (or even all the users) to collude and guess the secret key (X).

To complete the registration phase, Alice would provide each user distinct rows of either C0' or C1' (based on R1 and R2) for every corresponding bit of X along with the respective id of the RRAM used.

3.2.2 Authentication

The authentication process is trivial. Alice would simultaneously accommodate *k* or more users to apply their keys in the circuit shown in Figure 2. For each x_i, Alice will first RESET the corresponding R_i and then accommodate the users to apply their keys. If the total number of ON pulses applied by the users is correct, corresponding device's resistance should be lower than the reference resistance if the key was 1 or higher if the key was 0 (as designed in the registration phase). Alice can check the generated bit with her secret and simultaneously authenticate *k*-users.

3.3 Authentication Hardware

The user keys consist of voltage pulses (ON pulses), which are applied simultaneously to the gate of the respective 1T1R cell as shown in circuit presented in Figure 2. During authentication, Alice keeps V_BL_ALICE to the bit-line voltage used at registration.

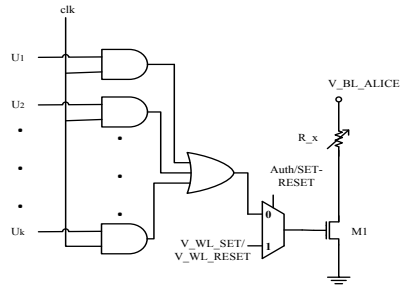

Figure 2. Circuit for applying the pulsed input from k-users.

To reconstruct a key, after the entire user input is received, the authenticator (*e.g,* Alice) applies another ON pulse to V_WL and activates V_BL_REF. We define it as the READ pulse. If R_x is sufficiently lower, then the drain voltage of M1 falls below a certain threshold. This can be sensed using the voltage-mode sense-amplifier-buffer circuitry as shown in Figure 3. Alice can examine the output at V_out during the READ pulse and reconstruct the key bit.

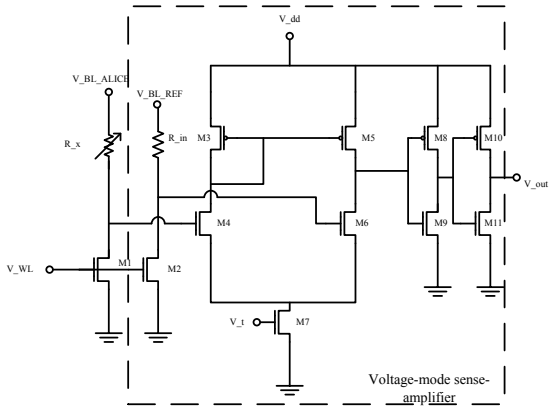

Figure 3. Voltage-mode sense amplifier-buffer circuit for reconstructing the secret.

4. EXPERIMENTS

We have simulated the proposed circuits to understand the effect of physical variations and hardware dependence of the scheme. For these simulations, we have used PTM's 65nm MOSFET models [8] and RRAM models proposed by Guan *et al*. [9]. We have used HSPICE platform for the simulations and MATLAB for calculation and post-processing. Common parameters used for the experiments are presented in table 1.

Table 1. Common parameters used for the experiments presented in this work.

Parameter Name	Value
Clock frequency (f_{clk})	50MHz
V_BL_SET	1.8V
V_WL_SET	1.0V
V_WL_RESET	2.7V
V_BL_RESET	1.9V
R_in	250kΩ
Gap_ini	1.367nm
ΔGap_0	0.05nm

For the given parameters, we find that it requires eight ON pulses (*i.e.,* SET operation) to move the device in question from a high resistive state to the low resistive state. Therefore, after 8 consecutive ON pulse to R_x, one would observe logic 1 at V_out. Since we want Alice to reveal the secret by applying the final ON pulse, a system with block-length of 7 would be required for multi-user authentication. If we consider a 3-out-of-3 user authentication system, then using the constructions provided in [7], we have C0' and C1' respectively as:

$$C0' = \begin{pmatrix} 000\,1\,111 \\ 0110\,011 \\ 011\,1\,100 \end{pmatrix}; \quad C1' = \begin{pmatrix} 0001111 \\ 0110011 \\ 1010101 \end{pmatrix}$$

In Figure 4, we have shown how Alice verifies a 1 and a subsequent 0 shared to the participating entities.

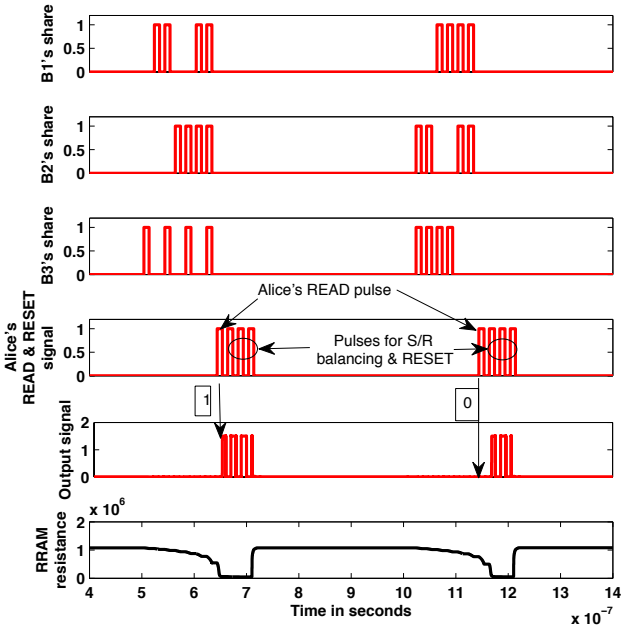

Figure 4. Simultaneous multi-user authentication using RRAM based hardware. Alice verifies a 1 and a 0 shared to the participating entities. The first pulse from Alice is the READ pulse and the three subsequent pulses are for ensuring balanced SET/RESET operation.

Note that, after authentication Alice still needs to apply two consecutive SET pulses for overcoming the SET-RESET balancing problem. SET/RESET unbalancing is a common problem for resistive memory-based circuits and properly balanced operations are required to mitigate this effect. Also, consecutive authentication cycles without properly resetting the device can aggravate the SET/RESET balancing problem as shown in Figure 5. Furthermore, a longer pattern of zeros or ones in the secret can also cause SET/RESET balancing problem.

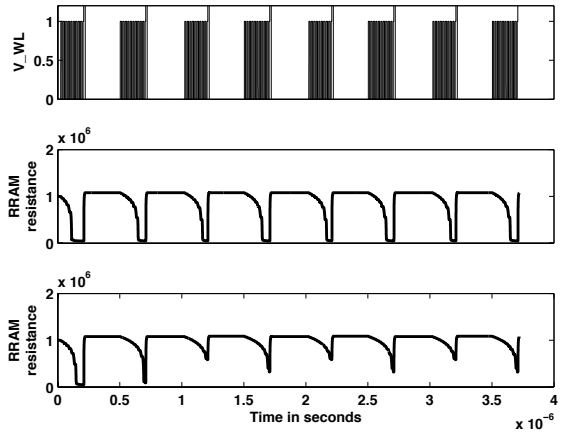

Figure 5. SET/RESET unbalance problem after consecutive authentication cycles. Here, the effect of unbalanced SET voltage in V_BL_SET line is presented. The top plot represents the word line voltage for consecutive authentication cycles. The plot in the middle shows the changes in the RRAM resistance for balanced condition (V_BL_SET=1.8V). The bottom plot shows the effect of unbalanced biasing where the V_BL_SET voltage is lowered by 12mV.

Another weakness of this authentication protocol comes from the adverse effects of noise on the RRAM-based circuits. As noticed in Figure 5, small deviations from SET/RESET voltages can produce significant errors in the reconstructed output. Therefore, the voltage sources that dictate SET and RESET operations should be carefully designed.

As discussed in [6], random filament formation and spatial variations in filament formation can create reliability issues during authentication. Since random fluctuation in the resistive values will void any reliable operation of the device, careful fabrication and testing are required before deploying these devices in a security application. Furthermore, experimental evidence suggests that RESET operations in RRAM devices are more stable and reliable than the SET operations. Therefore, one can choose either HRS-LRS transition or HRS-LRS transition based on the device to ensure more reliable performance. RRAMs with lower endurance values should be eligible for the authentication protocol discussed since each authentication process with *k*-users only corresponds to one read-write cycle. Finally, common factors such as temperature, aging *etc.* that threaten the reliability of RRAMs should be taken into consideration before deploying such hardware.

5. CONCLUSIONS

Non-volatile memory based devices and circuits are monotonic in nature. Exploiting this monotonicity can be useful in designing secure circuits and security protocols. In this work, we have connected the "*additive*" and *monotonic* nature of RRAM devices with secret sharing and simultaneous multi-user authentication

ideas. We have reported the designed protocol and the necessary circuits required for multi-user authentication using RRAM based hardware. This robust, hardware dependent multi-user authentication scheme can be useful in designing security primitives in the extremely resource constrained IoT applications.

ACKNOWLEDGMENT

The authors would like to thank Dr. Swarup Bhunia and Dr. Ajay Joshi for their helpful comments. This work was supported by AFOSR MURI under award number FA9550-14-1-0351.

6. REFERENCES

[1] Shamir, Adi. "How to share a secret." *Communications of the ACM* 22.11 (1979): 612-613.

[2] Blakley, George Robert. "Safeguarding cryptographic keys." *AFIPS*. IEEE, 1979.

[3] Stinson, Douglas R. "An explication of secret sharing schemes." *Designs, Codes and Cryptography* 2.4 (1992): 357-390.

[4] Naor, Moni, et al. "Visual cryptography." *Advances in Cryptology—EUROCRYPT'94*. Springer Berlin/Heidelberg, 1995.

[5] Arafin, M. T., et al. "A survey on memristor modeling and security applications." *Quality Electronic Design (ISQED), 2015 16th International Symposium on*. IEEE, 2015.

[6] Arafin, Md Tanvir, and Gang Qu. "RRAM based lightweight user authentication." *Computer-Aided Design (ICCAD), 2015 IEEE/ACM International Conference on*. IEEE, 2015.

[7] Verheul, Eric R., and Henk CA Van Tilborg. "Constructions and properties of k out of n visual secret sharing schemes." *Designs, Codes and Cryptography* 11.2 (1997): 179-196.

[8] Predictive technology model (PTM), http://ptm.asu.edu/

[9] Guan, Ximeng, et al. "A SPICE compact model of metal oxide resistive switching memory with variations." *IEEE electron device letters* 33.10 (2012): 1405-1407.

[10] Wong, H-S. Philip, et al. "Metal–oxide RRAM." *Proceedings of the IEEE* 100.6 (2012): 1951-1970.

[11] Naor, Moni, and Benny Pinkas. "Visual authentication and identification." *Advances in Cryptology—CRYPTO'97*. Springer Berlin Heidelberg, 1997. 322-336.

[12] Qu, Gang, and Lin Yuan. "Design THINGS for the Internet of Things—An EDA perspective." *Computer-Aided Design (ICCAD), 2014 IEEE/ACM International Conference on*. IEEE, 2014.

[13] "Two-person integrity" available online at http://informationtechniciantraining.tpub.com/14222/css/14222_85.htm, pp. 3-9 & 3-10

[14] "How the NSA is preventing another Snowden (and why you should do the same) ", Computer World, Jan. 7, 2014. Available online at http://www.computerworld.com/article/2474270/cybercrime-hacking/how-the-nsa-is-preventing-another-snowden--and-why-you-should-do-the-same-.html