LPN-based Device Authentication Using Resistive Memory

Md Tanvir Arafin
ECE Department
University of Maryland
College Park, Maryland 20742
marafin@umd.edu

Haoting Shen
ECE Department
University of Florida
Gainesville, Florida 32611
htshen@ece.ufl.edu

Mark M. Tehranipoor
ECE Department
University of Florida
Gainesville, Florida 32611
tehranipoor@ece.ufl.edu

Gang Qu
ECE Department
University of Maryland
College Park, Maryland 20742
gangqu@umd.edu

ABSTRACT

Recent progress in the design and implementation of resistive memory components such as RRAMs and PCMs has introduced opportunities for developing novel hardware security solutions using unique physical properties of these devices. In this work, we utilize the faults in HfOx-based resistive RRAMs to design secure, lightweight device authentication protocols. To detail our design, first, we introduce the device breakdown problem due to high bias conditions in resistive memory and the physics behind non-recoverable resistive states. Then, using the concepts of learning with parity noise (LPN) based authentication protocols, we demonstrate that simple READ and WRITE operations on resistive memory cells with defects can perform necessary calculation required for LPN-based authentication schemes. Next, we design two simple authentication protocols using resistive memory based hardware and provide a detailed security analysis for these protocols. We find that these authentication mechanisms can offer significant improvement against its CMOS counterpart regarding the area and power budget. Finally, we provide detailed physical design requirements for the memory components. The resistive memory components that are capable of performing the proposed authentication protocols have also been designed and fabricated. From our analysis, we find that these memory dependent authentication protocols are lightweight, resistant to learning attacks from active and passive adversaries, and reliable under normal changes in operating conditions.

CCS CONCEPTS

•Security and privacy → Hardware-based security protocols;

KEYWORDS

Resistive Random Access Memory (RRAM), Learning Parity in the Presence of Noise (LPN), One-time Writable Memory, Device Authentication.

ACM Reference format:
Md Tanvir Arafin, Haoting Shen, Mark M. Tehranipoor, and Gang Qu. 2019. LPN-based Device Authentication Using Resistive Memory. In *Proceedings of Great Lakes Symposium on VLSI 2019, Tysons Corner, VA, USA, May 9–11, 2019 (GLSVLSI '19),* 6 pages.
DOI: 10.1145/3299874.3317970

GLSVLSI '19, Tysons Corner, VA, USA

DOI: 10.1145/3299874.3317970

1 INTRODUCTION

Resistive memory primitives such as phase change memory (PCM), resistive random access memory (RRAM) and memristors promises a significant breakthrough in next-generation computing by replacing volatile DRAMs with faster non-volatile main memory components. High speed, simple device geometry, high density, large ON/OFF ratio, low-power of operation, in-memory computation capability, and application in neuromorphic computation are driving current research in improving the yield, reliability, and fabrication of existing resistive memory designs. Non-volatile memory elements also offer unique physical properties useful in designing next-generation hardware dependent security primitives. Such hardware-based security and trust designs can provide low-power yet provably secure cryptographic schemes essential for progressing towards next-generation Internet-of-trusted-things.

Modern efforts on the design and implementation of the Internet-of-things (IoT) envision a future of wide-scale deployment of interconnected smart objects. Intelligence in such network of *things* is derived from observations made using a large number of end-node components such as sensors and data collection hardware. The computation power is mostly budgeted for information acquisition and communication, and security is often an afterthought for these end-node components. Common cryptographic protocols usually require a substantial amount of energy, and thus, become prohibitive for most of the resource-constrained devices. Therefore, the IoT infrastructure remains dependent on a large number of untrusted, insecure components that jeopardize the security of the entire network [19].

One of the primary challenges in securing low-power end-nodes in IoT is authenticating the devices used for data acquisition. In this work, we develop solutions for secure authentication of low-power devices using resistive memory components and cryptographic primitives derived from learning parity in the presence of noise (LPN) problems. LPN-based authentication protocols were first proposed by Hopper and Blum [10] that inspired further investigation of the hardness of LPN-problems, and design of authentication protocols effective against active and passive attackers [4, 16]. One attractive feature of LPN-based techniques is that they require straightforward computation. For example, the Hopper and Blum (HB) authentication protocol [10] and its derivatives HB+, HB++, *etc* discussed in [5, 14, 16] require only vector inner-product calculation, parity computation, and counting operation. Thus,

LPN-based protocols can offer lightweight authentication solutions with provable security even in a post-quantum scenario. However, in extreme resource-constrained system, such protocols need to be optimized and redesigned to balance achievable security and the power-budget. This work presents research on optimized implementation of LPN-based shared-key authentication using the resistive memory components of a system. The main contributions of this work are detailed below:

(1) We study the breakdown mechanisms of resistive memory components in section 2 and explore the opportunity of intentionally introducing device failure for secure key storage.
(2) We demonstrate a non-conventional method of purposefully corrupting (one-time) memory components in designing LPN-based device authentication scheme in Section 3.
(3) We present fabrication and design details of resistive memory components capable of executing the authentication protocols presented in this work in section 4, and discuss the security analysis, opportunities and implementation challenges of these protocols in section 5.

It should be noted that the authentication scheme presented in this work apply not only to resistive memory primitives but also to any other memory technologies that can support the coexistence of one-time programmable components in the memory systems.

2 PHYSICS OF RESISTIVE MEMORY

Changes in the resistive states in memories such as RRAMs depend on the conductive filament formation in the metal-oxide thin films. These devices have a simple metal-insulator-metal (MIM) structure, with nanometer-thin metal-oxide layer within the top and bottom metal electrode.

2.1 Device Model

The current-voltage relation of an RRAM is determined by - (1) the internal state variable g that represents the spatial distance between the conductive filament in the oxide and the metal boundary, and (2) the electron tunneling in metal-insulator boundary. The analytical model of the thin-film evolution in a HfO_x-based RRAM can be described using the following equation as [8]:

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

where , q is the electron charge, L is the device filament thickness, V is the applied voltage, T is the device temperature, Ea, m, γ, v_0 are device dependent physical parameters. Then, the current-voltage relationship is given by [8]:

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

where I_0, V_0, g_0 are device dependent physical parameters. For this work, we assume the binary memory operation of a resistive memory unit, *i.e.,* it can stay in a high resistive state (HRS) or a low resistive state (LRS). This state transition is determined by the conductive filament formation in the oxide and metal boundary.

2.2 Device Failure

From section 2.1 it is evident that RRAMs exhibit resistive switching and non-volatility due to the reversible nature of the conductive filament formation in the insulator film. High voltage biasing can break this reversible process and put the device in a fixed resistive state.

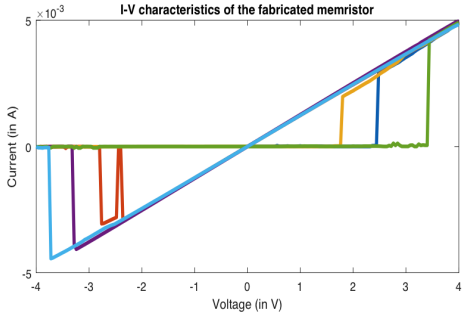

Figure 1: Sample I-V curve of the fabricated memristor. Multiple READ-WRITE cycle of the memristor is provided. The violet, sky-blue and red lines represent a LRS to HRS transition when the voltage across a memristor at a low resistive state is varied from +4 to -4V. The green, yellow and dark-blue lines represent a HRS to LRS transition when the voltage across a memristor at a high resistive state is varied from -4 to +4V. Normal operating condition demonstrates the unbalanced SET-RESET voltage at around $\pm 3V$.

Recent studies in Pt/TiOx/Pt/Cr-based devices suggest that such failure occurs due to either electrical or thermally assisted dielectric breakdown [6]. Experimental evidence of full crystallization oxide along with intrusion of the electrode metal (Pt) into the insulator layer has been found in the non-programmable RRAM devices [6]. It is evident that electrical or local thermal stressing of the insulator layer irreversibly breaks the programming properties of this devices [17]. This breakdown can be viewed as hard-failures of the memory elements. In this work, we utilize intentional high-voltage biasing to set a device in a non-recoverable resistive state.

Another mode of failure can occur due to the unbalanced SET-RESET operation of the resistive memories. Unbalanced SET-RESET causes accumulation (or depletion) of conductive filaments, and after multiple cycles, the devices gets stuck in a low (or high) resistive state. These failures are reversible in a sense that sufficient high voltage biases or current flow can *cure* the device by depleting (or accumulating) the conductive filaments. However, if there is no hardware support for *curing* the devices (*i.e.,* balancing the SET/RESET operation) in practice, the devices become non-programmable after a certain number of SET-RESET operation.

We have observed both modes of failure in an HfO_x-based memory unit as shown in Figure 1 and 2. Cycle-to-cycle variation in the SET-RESET voltage illustrates the need for proper READ/WRITE balancing for correctly writing a device to an HRS or LRS. Furthermore, Figure 2 depicts the irreversible hard-breakdown of the memory cell at a high RESET voltage applied with faster RESET time. After the hard breakdown, the device is stuck at the high resistive state. Further switching back has not been observed with SET voltages as large as 40V with 10mA currents through the device for multiple cycles. This represents a permanent breakdown of the reversible state-transition mechanism for the device.

3 AUTHENTICATION PROTOCOL DESIGN

3.1 Notations

In this work, the set of integers modulo an integer $q \geq 1$ is denoted by $\mathbb{Z}_q$. Matrices, vectors, and single elements over $\mathbb{Z}_q$ are represented by consecutively upper case bold letter, lower case bold

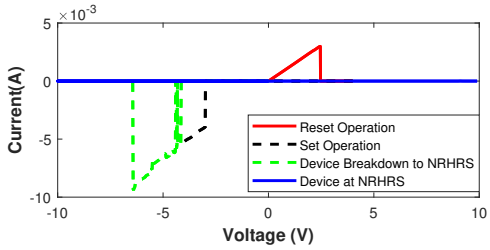

Figure 2: Sample I-V curve for the SET/RESET operation and hard breakdown of the fabricated RRAM for this work. Hard dielectric breakdown occurs for the device when a larger RESET voltage with fast ramping is applied along the devices. After the breakdown, the device becomes incapable of switching to LRS even with larger SET voltages.

letters and lower case letters such as $\mathbf{X}$, $\mathbf{x}$ and x. For a vector $\mathbf{x}$, the length of the vector is denoted by $|\mathbf{x}|$, i^{th} element is represented by $\mathbf{x}[i]$ and $wt(\mathbf{x})$ denotes the Hamming weight (*i.e.*, the number of indices for which $\mathbf{x}[i] \neq 0$) of the vector $\mathbf{x}$. The Hamming distance between two binary matrices are denoted by $hd(\mathbf{A}, \mathbf{B})$ (i.e., the number of indices for which $\mathbf{A}[i][j] \neq \mathbf{B}[i][j]$). $hdp(\mathbf{A}, \mathbf{B})$ represents the parity of $hd(\mathbf{A}, \mathbf{B})$ (i.e, $hdp(\mathbf{A}, \mathbf{B}) = 0$ for even parity of $hd(\mathbf{A}, \mathbf{B})$ and $hdp(\mathbf{A}, \mathbf{B}) = 1$ for odd parity of $hd(\mathbf{A}, \mathbf{B})$). $\mathbf{0}^{\ell \times n}$ represents a $\ell \times n$ null matrix. The Hadamard product of two matrices $\mathbf{A}, \mathbf{B}$ is given by $\langle A \circ B \rangle$. Given two vectors $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{z} = \mathbf{x} \oplus \mathbf{y}$ represents bitwise XOR operation of $\mathbf{x}$ and $\mathbf{y}$. $c \xleftarrow{\$} \{\mathbf{x} \in \mathbb{Z}\}$ represents a random sampling of $\mathbf{x}$. We denote *probabilistic polynomial time* (PPT) algorithms with upper case calligraphic alphabets such as $\mathcal{A}$. Therefore, if $\mathcal{A}$ is probabilistic, then for any input $\mathbf{x} \in \{0,1\}^*$ there exists a polynomial $p(.)$ such that the computation of a terminates in at most $p(|\mathbf{x}|)$ steps. All the physical quantities such as voltage, current and device names are denoted with upper case letters.

3.2 Assumptions

Recent works have demonstrated that resistive memory hardware can assist simple authentication and secure key storage facilitates [1, 2]. To harness this potential, we present two learning parity in the presence of noise based authentication schemes that is provably secure against passive attacks. These schemes are derived from the Hopper and Blum (HB) authentication protocol that provides simple yet efficient authentication from hard learning problems [10, 16]. For our authentication protocol design, we will assume that an RRAM can be put into a non-recoverable high resistive state (NRHRS) using high current flow through the device as shown in Figure 2.

Theorem 1: Writing one bit to an RRAM, where the physical state of the RRAM can be either writable or NRHRS, is the same as performing the 2-input logic AND operation.

Proof: Assume that an RRAM in a high resistive state represents the storage of a binary '0' and the low resistive state represents '1', and the physical state describing whether the RRAM can be written is denoted by y (*i.e.*, $y = 1$ meaning there will be successful state change if the data to be written is 1, $y = 0$ means there will not be a state change and the RRAM will stay at NRHRS). The existence of nor-recoverable high resistive state ensures that there will be RRAMs with $y = 0$. Then, if the incoming bit is represented by x and the actual value stored after the WRITE operation is z, one can draw the truth table as shown in Table 1

Table 1: Truth table describing the relation between the resistive state and the data stored

x (Incoming Bit)	y (Resistive State)	z (Memory Content)
0	0(NRHRS)	0
0	1(Writeable)	0
1	0(NRHRS)	0
1	1(Writeable)	1

From the truth table, it is evident that, $z = x \wedge y$.

For device authentication protocols discussed in this work, we will also assume an interactive protocol between a single prover $\mathcal{P}$ (*i.e.,* the device) and a verifier $\mathcal{V}$. Both the prover and the verifier has some knowledge about a shared secret $\mathbf{x}$. The secret is generated through a key-generation procedure $KeyGen(1^\lambda)$,where λ is a security parameter. The authentication protocol responds with the outputs accept or reject after a successful run of the protocol.

3.3 Description of Protocol I

The protocol uses an RRAM crossbar M of size $\ell \times n$. The secret $\mathbf{X} \in \mathbb{Z}_2^{\ell \times n}$ for authentication is distributed using Algorithm 1. For authentication, multiple round of interactive authentication (as presented in table 2) is performed.

Algorithm 1 Key Generation and Storage in RRAM Crossbar for LPN based Authentication

procedure $\mathbf{X} \leftarrow KeyGen(1^\lambda)$
  Sample $\mathbf{X} \xleftarrow{\$} \mathbb{Z}_2^{\ell \times n}$
  return $\mathbf{X}$
end procedure
procedure $KeyStorage(\mathbf{X})$
  for all $i \in \{1, \ldots, \ell\}$ **do**
    for all $j \in \{1, \ldots, n\}$ **do**
      if $\mathbf{X}[i][j] = 0$ **then**, **RESET** $M[i][j]$ to NRHRS.
      end if
    end for
  end for
end procedure

Enrollment: The verifier saves $\mathbf{X}$ for later authentication, the prover keeps the crossbar M, that is the device ultimately contains the crossbar for later authentication.

Verification The authentication is performed in t rounds. The verifier finally authenticates the device if the response of the prover was wrong for fewer than $t\tau$ times.

3.4 Reduction of Protocol I to an LPN-problem

Hopper and Blum first proposed a simple authentication protocol (HB) [10] using the learning parity in the presence of noise (LPN) problem. The authentication protocol described in table 2 can be derived from the HB protocol. The reduction of Protocol I to an LPN problem is discussed below:

Definition 1 (LPN Problem) Assume $\tau \in \mathbb{R}$ is a constant noise parameter where $0 < \tau < 0.5$, $t \in \mathbb{N}$ is the number of samples, $\mathbf{e}$ is a random binary vector such that $\mathbf{e} \xleftarrow{\$} \{\mathbf{x} \in \mathbb{Z}_2^t : wt(\mathbf{x}) \leq \tau t\}$ and $\mathbf{s}$ be a ℓ-bit binary vector (i.e., $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^\ell$). Given a random binary

Table 2: Protocol I: Single round interactive authentication

Prover(M, τ)		Verifier($\mathbf{X}$)
		$\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{\ell \times n}$
	$\xleftarrow{\mathbf{R}}$	
$-\ e \in \{0,1 \| Prob[e=1] = \tau\}$ – Write $\mathbf{R}$ in M using the following scheme: If $\mathbf{R}[i][j]$=0, **RESET** $M[i][j]$; else **SET** $M[i][j]$ – Read back the corrupted value $\mathbf{C}$ from M using the following scheme: If $M[i][j]$=HRS, $\mathbf{C}[i][j] = 0$ else $\mathbf{C}[i][j] = 1$ $-\ z := e \oplus hdp(\mathbf{C}, \mathbf{0}^{\ell \times n})$		
	$\xrightarrow{z}$	
		$-\mathbf{P} = \langle \mathbf{R} \circ \mathbf{X} \rangle$ –If $z = hdp(\mathbf{P}, \mathbf{0}^{\ell \times n})$ accept

Table 3: Protocol I+: Single round interactive authentication

Prover $(M, \mathbf{Q_v} \in \mathbb{Z}_2^{\ell \times n}, Q_z \in \{0,1\}, \tau)$		Verifier $(\mathbf{X}, \mathbf{Q_v} \in \mathbb{Z}_2^{\ell \times n}, Q_z \in \{0,1\})$
		$\mathbf{V} \xleftarrow{\$} \mathbb{Z}_2^{\ell \times n}$
	$\xleftarrow{\mathbf{V}}$	
$-\ e \in \{0,1 \| Prob[e=1] = \tau\}$ $-\ \mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{\ell \times n}$ $-\ \mathbf{Y} = \mathbf{Q_v} \oplus \mathbf{V}$ – Write $\mathbf{R}$ in M using the following scheme: If $\mathbf{R}[i][j]$=0, **RESET** $M[i][j]$; else **SET** $M[i][j]$ – Write $\mathbf{Y}$ in M – Read back the corrupted value $\mathbf{C}$ from M using the following scheme: If $M[i][j]$=HRS, $\mathbf{C}[i][j] = 0$ else $\mathbf{C}[i][j] = 1$ $-\ z := Q_z \oplus e \oplus hdp(\mathbf{C}, \mathbf{0}^{\ell \times n})$		
	$\xrightarrow{(z,\mathbf{R})}$	
		$-\mathbf{Y'} = \mathbf{Q_v} \oplus \mathbf{V}$ $-\mathbf{P} = \langle \mathbf{R} \circ \langle \mathbf{X} \circ \mathbf{Y'} \rangle \rangle$ –If $z = Q_z \oplus hdp(\mathbf{P}, \mathbf{0}^{\ell \times n})$ accept

matrix $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{t \times \ell}$, and $\mathbf{z} = \langle \mathbf{R}, \mathbf{s} \rangle \oplus \mathbf{e}$, find an ℓ-bit vector $\mathbf{x'}$ such that $wt(\langle \mathbf{R}, \mathbf{x'} \rangle \oplus \mathbf{z}) \leq \tau t$.

Theorem 2 Assume $\mathbf{r} = vec(\mathbf{R})$ denotes vectorization operation that generates a vector $\mathbf{r} \in \mathbb{Z}_2$ of a matrix $\mathbf{R}$. Then, the $hdp()$ calculation by the prover $\mathcal{P}$ returns the parity of $\langle \mathbf{r}, \mathbf{s} \rangle$ where $\mathbf{r} = vec(\mathbf{R})$, $\mathbf{s} = vec(\mathbf{S})$, $\mathbf{R}$ is the random challenge sent by the verifier and $\mathbf{S}$ is the matrix representing the writabilty of each RRAM in the crossbar (i.e., if $\mathbf{S}[i][j] = 0$, $M[i][j]$ is at NRHRS).

Proof: From Theorem 1, we can see that, after a writing operation the memory crossbar contains the corrupted value $\mathbf{C}$ which is equal to the binary Hadamard product of $\mathbf{R}$ and the physical state ($\mathbf{S}$) of the crossbar (i.e., $\mathbf{C} = \langle \mathbf{R} \circ \mathbf{S} \rangle$). Then, if we consider $\mathbf{c} = vec(\mathbf{C})$, $\sum_i \mathbf{c}[i] \bmod 2$ will be equal to the inner product of $\mathbf{r}$ and $\mathbf{s}$ (i.e., $\sum_i \mathbf{c}[i] \bmod 2 = \langle \mathbf{r}, \mathbf{s} \rangle$). Since $hdp(\mathbf{C}, \mathbf{0})$ defines the parity of the Hamming distance between $\mathbf{C}$ and $\mathbf{0}$, it effectively calculates the parity of the Hamming weight of $\mathbf{c}$ which is equal to $\sum_i \mathbf{c}[i] \bmod 2$. Therefore, $hdp(\mathbf{C}, \mathbf{0}) = \langle \mathbf{r}, \mathbf{s} \rangle$.

From theorem 2, it is evident that XORing an error e to the response of the prover ($hdp(\mathbf{C}, \mathbf{0})$), one can construct an LPN problem with the shared secret $\mathbf{S}$ representing the physical state matrix of an RRAM crossbar as shown in the authentication protocol at table 2 .

We have shown that, Protocol I is secure against passive attackers in the Discussions section. However, an active adversary who pretends to be the verifier and provides chosen adaptive non-random $\mathbf{R}$ can learn the secret using polynomial time algorithms presented in [13]. Thus, the protocol I is insecure against active attacks. So, we improved the protocol in a similar fashion proposed at [16] in Table 3.

3.5 Description of Protocol I+

Enrollment: In this protocol, the prover and the verifier both shares additional information $\mathbf{Q_v} \xleftarrow{\$} \mathbb{Z}_2^{\ell \times n}, Q_z \in \{0,1\}$ in the enrollment phase.

Verification The protocol is executed in t rounds. The verifier finally authenticates the device if the response of the prover was wrong for fewer than $t\tau$ times.

4 HARDWARE DESIGN

The authentication scheme discussed in the previous section uses an $\ell \times n$ RRAM crossbar M. The key-storing process described by the procedure $KeyStorage(\mathbf{X})$ puts the RRAM element $M[i][j]$ to NRHRS for $\mathbf{X}[i][j] = 0$. This operation makes later LPN-based computation simpler to achieve with the resistive memory hardware. Experimentation on intentional device failure is performed using fabricated RRAM devices. The details of the fabrication for the *HfOx* based resistive memory components used for the experimental results presented in section 2.2 is given next.

4.1 Fabrication Details

The fabrication is performed on oxide cover Si wafer. The first metal layer (bottom) is prepared by lift-off photo-lithography processing, including Chromium (Cr, 10 nm), copper (Cu, 100 nm) and platinum (Pt, 10 nm) deposited subsequently, serving as the adhesion, conducting, and interface adjusting layer, respectively. Then, HfO_2 (10 nm) is coated by atomic layer deposition (ALD) at 200 ^oC. After that, the second metal layer (top) is prepared by lift-off again, including Pt (10 nm) and Cu (100 nm), serving as interface adjusting and conducting layer, respectively.

4.2 Supporting Hardware

Protocol I is an extremely lightweight protocol that only requires the prover to calculate $hdp()$ function and generate biased random number with a given τ. Since, the READ/WRITE mechanism is

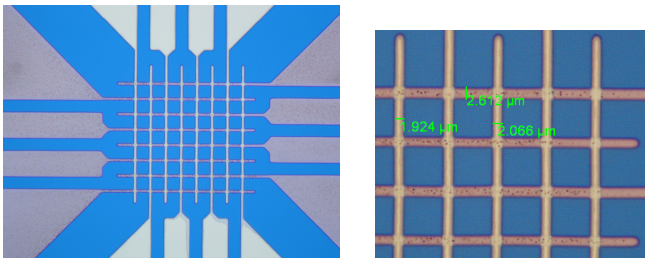

Figure 3: Fabricated memristor and its dimensions. The top (light colored) and bottom (dark colored) electrode are seen as the gray crossbars. The width of the top and bottom crossbars are $2\mu m$ on average. In between the top and bottom crossbars, a 10 nm thin film of HfO_x is deposited using atomic layer deposition.

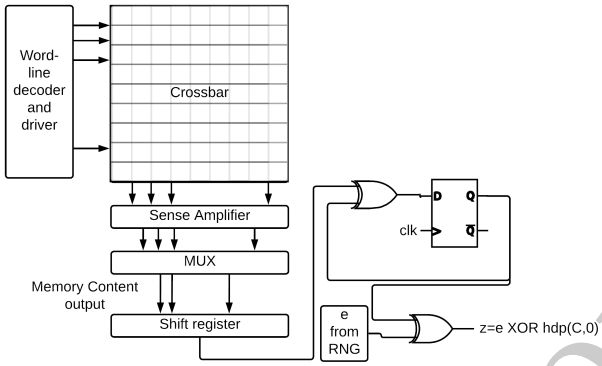

Figure 4: Structure of RRAM crossbar with additional hardware for LPN-based authentication.

fundamental to the memory systems, therefore, the driver design for READ/WRITE is trivial. The *hdp*() calculation can be supported in software, or can be implemented in hardware where *hdp*() function can easily be calculated using a finite state machine built with an ℓ-bit shift register, a D-flip-flop and a two input XOR gate as shown in Figure 4.

Additionally, there have been several resistive memory-based true RNGs reported in the current literature [11, 12] that have passed NIST's RNG test-suites. These designs depend on the random resistance fluctuation of the memory elements. A construction of a biased-RNG with a fixed τ is given in Figure 5. It should be noted that, the sampling of e in the protocols can also be performed in software where the system contains a TRNG source.

5 DISCUSSIONS

From our experiments with the fabricated devices and hardware design, we observe that the proposed LPN-based authentication schemes can be optimally implemented using the one-time programmable properties of resistive memory components. In this section, we discuss the physical implication of the design and our experimental observations.

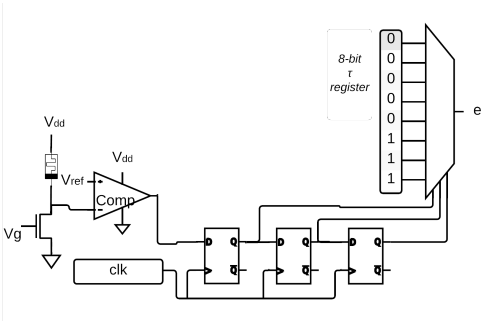

Figure 5: Implementation of a biased RNG with $\tau = 3/8$. The value of τ is fixed by modifying the values of the τ-register.

5.1 Security Analysis

Protocol I is secure against random guessing attacks. It can be seen that for a single round of operation, a random guess on z has a probability of $(1/2)$ to be correct. As the number of rounds t increases, the chance of success (for correctly answering z) becomes $(1/2)^t \sum_{j=(1-\tau)t}^{t} \binom{t}{j} \leq e^{-t(3-2\tau)^2/6}$. Thus, the success of random guessing attack diminishes exponentially for Protocol I.

Furthermore, protocol I is secure against passive attackers, Let us assume that an attacker ($\mathcal{A}$) tries to learn the secret $\mathbf{X}$ by eavesdropping $(\mathbf{R}, z)$ over multiple rounds. It is shown in Section 3 that this learning problem for the attacker can be reduced to an LPN problem. LPN is an NP-hard problem [9], and it has been shown that in statistical query model, parity is not efficiently learnable in the random case [15]. The LPN problem is also known as the Syndrome decoding problem that tries to find the closest vector to a random linear error-correcting code, which is believed to be exponentially hard[3]. For $\tau > 0$, the BKW algorithm described in [4] gives a subexponential time algorithm that solves the LPN problem in $2^{O(\ell/log\ell)}$ time. Optimized implementation modified the BKW algorithm reported at [7] demonstrates how the efficient computation required for breaking LPN problems are memory-bounded, and it required about 15 days to solve for LPN instances with key-size $k = 243$ and $\tau = 1/4$. With $k \geq 790$ and $\tau = 1/4$ LPN can achieve 256-bit security on classical computers and 162-bit security on quantum computers with memory constrained to 2^{80}-bits [7] which is sufficient for achieving NIST's post quantum call for security [18].

Protocol I+ also requires a solution for the LPN problem, and thus, it remains secure for random guessing attack and attacks from the passive adversaries with only eavesdropping capabilities. Therefore, let us assume an active attacker $\mathcal{A}$ who can query the prover $\mathcal{P}$ multiple times to learn about the shared secret, and then correctly answers the verifier $\mathcal{V}$ with finite probability ϵ. Then to break protocol I+, the attacker needs to solve for $\mathbf{X}, \mathbf{Q_v}$, and Q_z with given instances of $\mathbf{V}, \mathbf{R}, z$. This requires the attacker solving for $\mathbf{P}$. However, $\mathbf{Y}'$ sample the random challenge $\mathbf{V}$ using $\mathbf{Q_v}$ and use this for computing $\mathbf{P}$. Hence, solving for $\mathbf{P}$ and $\mathbf{X}$ essentially reduced to solving subset-LPN (SLPN) problem which is also a hard problem. Furthermore, since a chosen cipher-text by $\mathcal{A}$ always sampled using $\mathbf{Q_v}$ and the prover has control over the randomness of $\mathbf{R}$, the overall response of each queries from an honest provider will not leak any information about the secret $\mathbf{X}$. A detailed proof of security of such SLPN-based authentication can be found in [16]. Since the protocol does not leak any information about the secret

even in chosen-ciphertext attacks, the protocol is secure against both active and passive attackers.

5.2 Storage, Power and Area Overhead

The authentication protocols presented in this work requires permanent faults in a subset of the memory components of a given device. Therefore, the storage overhead for the secret key and error correction techniques must be taken into consideration in designing this authentication. For example, let us consider in protocol I, the shared secret is given by an n-bit vector (*i.e.,* $n \times 1$ matrix). Then, if this secret is stored in an m-bit memory array, there will be $(m-n)$-bit usable memory cells in the array after the execution of Algorithm 1. However, as noted in the previous subsection, a key-length $n = 790$-bits can provide sufficiently secure implementation, and therefore, the storage overhead for the implementation will be extremely low on practical systems. Furthermore, the $(m-n)$-bit usable memory cell occurs in the worst case situation when all the secure bits are 0, requiring all n -memory location to be set to NRHRS. However, on average $(n/2)$- memory location will be in writable state and additional $n/4$ memory location will contain a correct data. Thus, in average cases $(m-n/4)$-bits can be correctly stored in the an m-bit memory array. As m grows, the adverse effect of the length of secret-key size n diminishes for regular memory operations. If the system allows for approximate memory, with small memory error acceptance, then, one-time writing based key storage essentially would have zero overhead.

The main contribution in power consumption for protocol I and I+ comes from the writing operation of the resistive memory component. In practice, the WRITE operation requires significantly larger power than the READ operation ($E_{READ} \approx 1-5nJ$, whereas $E_{WRITE} \approx 10-20nJ$) [20]. Fabricated RRAMs reported in this work also shows similar properties, however, the READ/WRITE power are a magnitude larger due to the thicker filament, and larger low resistive state ($1k\Omega$). Therefore, for t-rounds, the approximate energy consumption for the protocols would be $tE_{Write} + tE_{peripheral}$.

The proposed design is compact due to it's application of cross-point structure for memory design. The crossbar is controlled by external driver circuit, and thus, the area overhead is contributed mainly from the control circuits. If we assume that the crossbar is primarily used for memory operation, then from the hardware design in Section 4.2, it can be seen that the only area overhead results from the additional shift-register and D-filp-flops. Given the complete area requirement for the crosspoint memory driver circuits, the additional hardware requires a very small area overhead.

5.3 Noise Tolerance

It should be noted that, the authentication protocols relies on the correct memory operation of the underlying RRAM devices used for storing th secret key. Physical variation due to biasing, temperature *etc.* will not affect the authentication techniques as long as these variation does not push the RRAMs to the non-recoverable resistive state. On the other-hand, permanent fault due to aging or improper SET/RESET operation will lead a failure of the device authentication technique. However, it should be noted that, this is a trivial condition for strong security constructs where a single bit-flip should render the secure key void.

6 CONCLUSIONS

In this work, we have presented a secure and lightweight application of LPN-based device authentication technique in resistive memory based hardware. The constructs discussed are not bound only to the RRAM memory technology, rather, it is applicable in any memory system containing a mix of one-time and multiple-time programmable memory components. We have demonstrated that, integrating simple physical properties in simple yet secure cryptographic technique can successfully yield extremely lightweight authentication solutions useful for securing low-power nodes in the Internet-of-things.

7 ACKNOWLEDGEMENT

This work was supported by AFOSR MURI under award number FA9550-14-1-0351. We also thank the reviewers for their valuable comments and feedback.

REFERENCES

[1] Md Tanvir Arafin, Carson Dunbar, Gang Qu, N McDonald, and L Yan. 2015. A survey on memristor modeling and security applications. In *Quality Electronic Design (ISQED), 2015 16th International Symposium on*. IEEE, 440–447.
[2] Md Tanvir Arafin and Gang Qu. 2018. Memristors for Secret Sharing-Based Lightweight Authentication. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 99 (2018), 1–13.
[3] ER Berlekamp, RJ McEliece, and HCA van Tilborg. 1978. On the inherent intractability of certain coding problems. (1978).
[4] Avrim Blum, Adam Kalai, and Hal Wasserman. 2003. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)* 50, 4 (2003), 506–519.
[5] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. 2006. HB++: a Lightweight Authentication Protocol Secure against Some Attacks. In *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2006. SecPerU 2006. Second International Workshop on*. IEEE, 28–33.
[6] Daniela Carta, Peter Guttmann, Anna Regoutz, Ali Khiat, Alexander Serb, Isha Gupta, Adnan Mehonic, Mark Buckwell, Steven Hudziak, AJ Kenyon, and others. 2016. X-ray spectromicroscopy investigation of soft and hard breakdown in RRAM devices. *Nanotechnology* 27, 34 (2016), 345705.
[7] Andre Esser, Robert Kübler, and Alexander May. 2017. LPN decoded. In *Annual International Cryptology Conference*. Springer, 486–514.
[8] Ximeng Guan, Shimeng Yu, and H-S Philip Wong. 2012. A SPICE Compact Model of Metal Oxide Resistive Switching Memory With Variations. *IEEE Electron Device Letters* 33, 10 (2012), 1405–1407.
[9] Johan Håstad. 2001. Some optimal inapproximability results. *Journal of the ACM (JACM)* 48, 4 (2001), 798–859.
[10] Nicholas J Hopper and Manuel Blum. 2001. Secure human identification protocols. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 52–66.
[11] Chien-Yuan Huang, Wen Chao Shen, Yuan-Heng Tseng, Ya-Chin King, and Chrong-Jung Lin. 2012. A contact-resistive random-access-memory-based true random number generator. *IEEE Electron Device Letters* 33, 8 (2012), 1108–1110.
[12] H Jiang, D Belkin, SE Savel'ev, S Lin, Z Wang, Y Li, S Joshi, R Midya, C Li, M Rao, and others. 2017. A novel true random number generator based on a stochastic diffusive memristor. *Nature communications* 8, 1 (2017), 882–882.
[13] Ari Juels and Stephen A Weis. 2005. Authenticating pervasive devices with human protocols. In *Annual international cryptology conference*. Springer, 293–308.
[14] Jonathan Katz and Ji Sun Shin. 2006. Parallel and concurrent security of the HB and HB+ protocols. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 73–87.
[15] Michael Kearns. 1998. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)* 45, 6 (1998), 983–1006.
[16] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. 2011. Efficient authentication from hard learning problems. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 7–26.
[17] SB Lee, DH Kwon, K Kim, HK Yoo, S Sinn, M Kim, B Kahng, and BS Kang. 2012. Avoiding fatal damage to the top electrodes when forming unipolar resistance switching in nano-thick material systems. *Journal of Physics D: Applied Physics* 45, 25 (2012), 255101.
[18] NIST. 2017. Post-Quantum Cryptography. (2017). https://csrc.nist.gov/projects/post-quantum-cryptography
[19] Gang Qu and Lin Yuan. 2014. Design things for the internet of things: an EDA perspective. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 411–416.
[20] Cong Xu, Xiangyu Dong, Norman P Jouppi, and Yuan Xie. 2011. Design implications of memristor-based RRAM cross-point structures. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*. IEEE, 1–6.