Attacking ReRAM-based Architectures using Repeated Writes

Biresh Kumar Joardar*, Krishnendu Chakrabarty†

*Deptartment of ECE, University of Houston, Houston, TX, 77204, bjoardar@central.uh.edu
†School of ECEE, Arizona State University, Tempe, AZ 85287, krishnendu.chakrabarty@asu.edu

***Abstract*—Resistive random-access memory (ReRAM) is a promising technology for both memory and for in-memory computing. However, these devices have security vulnerabilities that are yet to be adequately investigated. In this work, we identify one such vulnerability that arises from the write mechanism in ReRAMs. Whenever a cell/row is written, a constant bias is automatically applied to the remaining cells/rows to reduce sneak current. We develop a new attack (referred as *WriteHammer*) that exploits this process. By repeatedly exposing a subset of cells to this bias, *WriteHammer* can cause noticeable resistance drift in the victim ReRAM cells. Experimental results indicate that *WriteHammer* can cause up to 3.5X change in cell resistance by repeatedly writing to the ReRAM cells for a duration of 4 ms.**

Keywords—ReRAM, Rowhammer, Write, Hardware security

I. Introduction

DRAM and SRAM have been the traditional choice of memory technology for computing systems. However, as the scaling of traditional memory technologies is approaching the physical limit, it is difficult to continue providing sufficient computing and storage capacity for future data-intensive applications [1]. Moreover, large cell size and high leakage power of traditional memory lead to large design area and energy consumption [1]. Non-Volatile Memory (NVM) demonstrates excellent scaling and near-zero leakage power. This relatively new memory technology has emerged as a promising candidate for future memory applications. NVMs tend to be smaller in dimension, which enables higher storage density. Moreover, NVMs can also be used for in-memory computing, which makes it an attractive choice for future use. A comparison of various NVM technologies, conventional DRAMs, and SRAMs can be found in [2].

Among the various NVM technologies, resistive random-access memory (ReRAM) is a popular choice for both memory and in-memory computing applications due to its small size and high ON/OFF ratio [3]. ReRAM prototypes have been developed by TSMC and CEA-Leti [4][5]. These prototypes demonstrate the viability of ReRAMs for both memory and in-memory computing. ReRAMs are a popular choice for accelerating deep learning training and inferencing [6][7]. Recent work shows that a speed-up of more than an order of magnitude is obtained using ReRAMs, compared to traditional GPUs, for deep learning applications [6][7]. The efficiency of ReRAMs can be attributed to the massive data parallelism enabled by its crossbar structure. ReRAM crossbars are composed of several thousands of resistors (cells); each cell can store data and also implement a multiplication operation [6].

However, despite these advantages, existing ReRAM-based architectures must overcome reliability challenges. ReRAMs are susceptible to different types of defects and noise [8]. These non-idealities hamper the widespread adoption of large-scale deep learning algorithms on ReRAM crossbar-based accelerators. These shortcomings can also be exploited by an adversary to launch targeted attacks and compromise normal operations. The security concerns associated with these emerging devices and the attack surface have not received much attention in the literature. We focus on this problem in this paper.

Security vulnerabilities in ReRAMs have been demonstrated in [9] and [21]. In [9], the authors demonstrate Rowhammer attacks in ReRAMs. Rowhammer attacks exploit a vulnerability in traditional DRAMs, where an attacker can cause bit flips by simply accessing its neighboring rows [10]. However, it is shown in [9] that a Rowhammer attack in ReRAM requires extremely high temperature (>100°C). Such high temperature is detrimental to the device itself and therefore may not be practical. In [21], the authors identify a few possible ways an attacker can inhibit the use of ReRAMs for neuromorphic computing. An attacker can suppress neuron firing, interfere with communication between the neurons or exploit sneak currents to cause a neuron to misfire. However, their findings are specific to neuromorphic computing and may not be applicable when ReRAM is used as memory or for other in-memory computing purposes.

In this work, we demonstrate another attack on ReRAM crossbars using repeated write operations. The new attack (we refer to it as *WriteHammer*) results in a permanent loss of data stored on the ReRAM cells. However, unlike Rowhammer attacks, the new attack does not rely on the parasitic coupling between adjacent rows. The *WriteHammer* attack exploits the constant voltage bias that is applied to the ReRAM rows to prevent sneak current during write operations. By repeatedly writing to a subset of aggressor ReRAM rows, the attacker can cause a noticeable resistance drift on the victim rows. *WriteHammer* will affect normal operation of ReRAMs irrespective of its use as memory or for in-memory computing. Unlike in conventional Rowhammer attacks, the bit flips (data loss) in ReRAMs can happen in non-adjacent rows/cells to the aggressor row/cell. Moreover, the effectiveness of the attack is also a function of the data stored on the cell and the operating temperature.

We list below the key contributions of this paper:

- We develop a new *WriteHammer* attack on ReRAM crossbars. The attack can be implemented by simply writing to a handful of rows repeatedly.

This research was supported in part by the Semiconductor Research Corporation (Task ID 2994.001) and NSF grant CNS-2011561. Biresh Kumar Joardar was also supported in part by NSF Grant # 2030859 to the Computing Research Association for the CIFellows Project.

978-3-9819263-7-8/DATE23/© 2023 EDAA

- We demonstrate the effectiveness of *WriteHammer* using two open-source ReRAM Verilog models. Our experiments indicate that *WriteHammer* can cause up to 3.5X change in the resistance of victim ReRAM cells.
- We show how the effectiveness of *WriteHammer* varies based on temperature, input conditions, etc. These factors make it more difficult to detect these attacks unlike traditional Rowhammer attacks.

The rest of the paper is organized as follows. Section II presents relevant prior work related to ReRAM-based architectures and associated reliability issues. Section III presents the proposed *WriteHammer* attack in detail. We analyze the proposed attack experimentally in Section IV. Finally, we conclude this paper by summarizing our findings in Section V.

II. Related Prior Work

ReRAMs store data using variable resistance (or conductance) [6]. An ReRAM cell consists of a metal oxide layer (e.g., Ti, Ta, and Hf) sandwiched by two metal (e.g., Pt) electrodes [11]. The electronic behavior of metal/oxide interfaces depends on the oxygen vacancy concentration of the metal oxide layer. The behavior of the ReRAM cell can be modulated by the application of a suitable voltage. The cell can be switched ON (SET operation) or OFF (RESET operation) through voltage control. Prototypes from TSMC and CEA-Leti [4][5] show promising properties of fast switching speed and low energy consumption. Prior work has shown that these devices can also be repurposed for computing [6] because ReRAM cells can perform high-throughput matrix multiplication operations. Due to this dual nature (both memory and computing), ReRAM-based architectures have become popular in recent years. ReRAM-based in-memory computing systems have been proposed for machine learning, graph analytics, and bioinformatics applications [7][12][18]. ReRAMs can also be used for other mathematical operations such as addition, subtraction, division, etc. [13].

However, ReRAMs tend to suffer from various non-ideal effects such as sneak current, thermal noise, etc. [8]. These non-ideal effects not only affect normal ReRAM behavior but they can also be exploited for hardware attacks by adversaries. Several hardware vulnerabilities, such as Rowhammer [10], have been discovered in recent years in traditional CMOS-based systems. However, the security vulnerabilities in emerging non-volatile memories (such as ReRAMs) have not received significant attention. While a Rowhammer attack has been presented for ReRAMs [9], this attack requires an extremely high temperature (>100°C) for successfully causing a bit flip, which may not be practical. The system may get damaged by the extreme heat before the attacker can cause a bit flip. Another attack on ReRAMs exploits the sneak current to cause undesired neurons to misfire in neuromorphic computing applications [21]. However, this attack is specific to neuromorphic computing only. Moreover, sneak current can be minimized by applying constant bias during write operations [21]. This approach can greatly reduce the severity of this attack. In this work, we present the *WriteHammer* attack, which exploits the mechanisms of write operations to cause data loss at lower temperatures. The new attack exploits the voltage applied to prevent sneak current in ReRAM crossbars. Typically, NVM technologies (including ReRAMs) have a retention time of higher than 10 years [11]. However, *WriteHammer* can significantly reduce the retention time of ReRAMs and damage stored data.

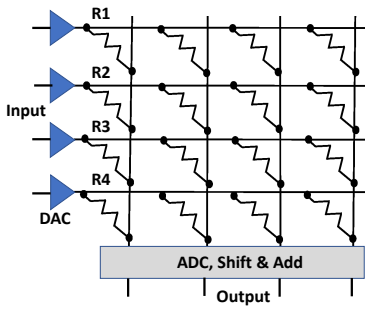

Fig. 1: Illustration of a typical ReRAM crossbar-based architecture

III. WriteHammer-ing ReRAMs

In this section, we present the details of the proposed *WriteHammer* attack on ReRAM-based systems.

A. Normal ReRAM operations: Background

Fig. 1 shows a typical ReRAM crossbar structure that can be used as both memory and for in-memory computing. Some ReRAM-based crossbars utilize a MOS access transistor for each cell (commonly referred as 1T1R configuration). However, ReRAM crossbars can also be designed without access devices (commonly referred as 0T1R configuration). Conventionally, in the MOS-accessed 1T1R structure, memory cell arrays are isolated by MOS access devices, which results in low leakage current. However, the cell size is dominated by the large MOS access device that is necessary to drive enough write current; the ReRAM cell itself is much smaller [11]. In addition, the 1T1R configuration requires more complex wire routing to control each MOS access transistor. In the 0T1R setting, an ReRAM crossbar can be accessed without any extra access devices. The removal of MOS access devices leads to a memory cell size of only $4F^2$, where F is the process feature size [11]. Therefore, the 0T1R configuration is popular in in-memory computing, especially for accelerating deep-learning applications [6][7]. Fig. 1 shows an ReRAM crossbar with the 0T1R cell configuration.

As shown in Fig. 1, an ReRAM cell stores data as a resistance value. To 'read' this data, a read voltage is applied at the input. Following Ohm's law, a proportional current (I=V/R) is observed at the output, which is then converted to a digital value using an ADC. To 'write' a new value to the ReRAM cell, a higher write voltage is applied to the ReRAM cell, which changes the resistance of the cell (and hence the stored value). Note that the read voltage (typically 0.5V [14]) is often much lower than the write voltage (2V [14]). Recall that the resistance of an ReRAM cell can be varied by applying an input voltage. However, a read operation should not alter the stored data; hence, the read voltage is lower than the write voltage to prevent any change in cell resistance.

The application of a higher voltage for write operations in a 0T1R configuration can result in high sneak current [9]. Due to the crossbar structure of ReRAMs, applying a voltage to one cell/row (R_1) results in sneak current through the other cells/rows (R_i, where $i \neq 1$). The sneak current results in high power consumption, especially for ReRAM crossbars that contain 128×128 cells [9]. This problem is not present in the 1T1R configuration as the MOS access transistor prevents any sneak current. However, as mentioned earlier, the 1T1R

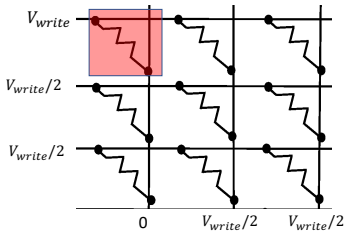

Fig. 2: Setup for the *WriteHammer* attack. When the red cell is being written, the other cells/rows are biased to reduce sneak current. Repeatedly exposing these cells to the bias voltage can cause their resistance to drift, which results in loss of data.

configuration is associated with higher area overhead and more complex wiring. To reduce sneak currents in the 0T1R setting, a constant voltage of $V_{write}/2$ is applied to the other rows/cells every time a row/cell is written [9][15]. For instance, if one ReRAM crossbar row/cell (R1) is being written, all the other rows/cells (R_i, where $i \neq 1$) are biased at $V_{write}/2$. This significantly reduces the amount of sneak current through the other ReRAM cells during write operations [9][15].

Under normal circumstances, the $V_{write}/2$ bias does not have any noticeable effect on the cell's resistance. However, the repeated application of this voltage can result in significant resistance drift (and thereby loss of data). This phenomenon is exploited by *WriteHammer* to corrupt stored data on ReRAM cells without ever accessing it. We discuss the details of the attack setup in the next sub-section.

B. Attacking ReRAMs using write operations

Fig. 2 shows the setup for the attack. For the sake of demonstration, we assume a 3×3 ReRAM crossbar. However, the principles described here are valid for any ReRAM crossbar size/shape. As shown in Fig. 2, the threat model assumes that the attacker has read/write access to cell R_1 (shown in red) but does not have access to the other cells/rows R_i, where $i \neq 1$. This is similar to how DRAM memory is shared by multiple processes running concurrently on the computing units. Each process has read/write access only to a certain allocated part of the memory. We assume a similar memory allocation in ReRAM-based memory systems. The attacker aims to corrupt the data stored on cells R_i ($i \neq 1$) using the *WriteHammer* attack. *WriteHammer* repeatedly writes data to R_1 (aggressor row, marked in red in Fig. 2). The attacker does not care about the actual data being written on R_1. The goal of the attacker is to simply write to R_1 multiple times. Naturally, to prevent sneak current, a voltage $V_{write}/2$ is applied to cells R_i for every write. By writing to cell R_1 multiple times, the attacker forces the $V_{write}/2$ voltage on the other cells for a prolonged duration of time. This can cause resistance drift in the cells R_i, resulting in permanent data loss. Here, it should be noted that ReRAMs suffer from relatively low write endurance. As a result, the cell being written repeatedly may get damaged in the process. However, recall that the attacker does not care about the value being written; their target is to damage the value stored on the other cells. Therefore, the attacker will sacrifice cell R_1 in Fig. 2 to cause data loss in R_i, where $i \neq 1$. As we show later, *WriteHammer* can cause up to 3.5X change in the resistance of victim cells. In addition, the amount of resistance drift due to *WriteHammer* is also a function of the current state of the ReRAM cell, temperature, and attack pattern as we show next.

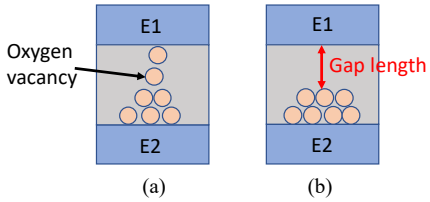

Fig. 3: Illustration of (a) ON and (b) OFF states in ReRAM cell

Current state of the cell: The operation of ReRAM is associated with the conductive filament growth due to the movement of oxygen ions. Fig. 3 shows an example to explain how the resistance drift varies with the current state of the ReRAM cell (i.e., the gap length); here, we define gap length as the distance between the tip of the filament and the opposite electrode. Fig. 3(a) shows a scenario where the vacancies form a bridge between the two electrodes E1 and E2 (i.e., gap length of zero). Under this condition, the resistance is low (ON state), and current can flow easily, Fig. 3(b) shows the case where the vacancies are clustered towards E2 (i.e., gap length > 0). In this condition, the resistance is high (OFF state) and less current can flow. Applying a negative voltage to E1 will push the vacancies towards E2 in both cases. However, the vacancies will move more easily in the scenario of Fig. 3(a) compared to the scenario of Fig. 3(b). This happens as the vacancies are already clustered around E2 in Fig. 3(b). Therefore, it will be more difficult for the vacancies to be pushed further towards E2 i.e., it becomes progressive more difficult to increase the resistance of an ReRAM cell. This example shows how the change in resistance will vary depending on the current state of the ReRAM cell. Since data is stored as resistance in ReRAM cells, this observation also implies that the effect of *WriteHammer* will depend on the data stored on each cell.

Temperature: The average rate of growth and the variation amplitude of the filament have a strong dependence on temperature. It has been observed that the temperature increases significantly during SET and RESET operations (i.e., write operations), which assists the growth and rupture of the filament [16]. However, in an ReRAM cell, the resistance fluctuates when the cell is heated to higher than room temperature. This property can be exploited by an adversary to cause higher resistance drift using *WriteHammer*.

Attack pattern: *WriteHammer* relies on repeated writes to a handful of aggressor rows/cells. However, the severity of *WriteHammer* also depends on the type of writes (i.e., SET or RESET). Consecutive SET (or RESET) pulses have the maximum effect. However, to prevent detection, an attacker can also combine both SET and RESET pulses. For instance, an attacker may apply one RESET pulse after every ten SET operations. Note that opposite polarity voltages are used for SET and RESET operations. Hence, having one RESET after every ten SET partially offsets some of the resistance drift. This reduces the severity of the *WriteHammer* attack; the severity is highest when only SET (or RESET) pulses are applied consecutively. However, by combining both SET and RESET pulses, an attacker can avoid being detected easily.

C. WriteHammer vs Rowhammer

The *WriteHammer* attack shares many similarities with Rowhammer attacks in DRAM-based memory systems. Both attacks require repeatedly accessing a few target aggressor rows. Both attacks result in loss of data in the victim rows. However, there are several key differences between

Table 1: Change in resistance with different coupling capacitance values during repeated write operations.

Capacitance (fF)	Temperature (°C)	Attack condition	ΔR after 4 ms (%)
0	25	Bias	0.15%
1	25	Bias+Coupling	0.15%
10	25	Bias+Coupling	0.15%
100	25	Bias+Coupling	0.15%
1000	25	Bias+Coupling	3.69%

WriteHammer and traditional Rowhammer attacks as we discuss next.

Unlike Rowhammer, which is due to parasitic coupling between the rows, the proposed attack relies primarily on the voltage applied to prevent sneak current. Our experiments indicate that the amount of voltage change induced by Rowhammer is often not sufficient to cause any noticeable change in resistance in the victim ReRAM cells. This happens as the voltage necessary for changing the resistance is not achievable with parasitic coupling only. Next, traditional Rowhammer attack on DRAMs can be done using both read and write operations. However, ReRAM crossbars use two different voltages for read and write operations. The read voltage is significantly lower than the write voltage and does not result in high sneak current. As a result, the $V_{write}/2$ voltage is not applied during read operations. Hence, *WriteHammer* can only be implemented using write operations. In addition, the victim rows need not be adjacent to the aggressor in *WriteHammer*. Bit flips happen most commonly in rows that are adjacent to the aggressor row in Rowhammer attacks in DRAMs. However, the $V_{write}/2$ voltage is applied to every row in a ReRAM crossbar (besides the one where the write operation is being performed). Hence, the victim rows are independent of the location of the aggressor rows. This can cause resistance drift far from the aggressor row(s). As a result, an attacker can compromise any cell within the crossbar by simply accessing a single aggressor row. Finally, the effect of the attack is also different in ReRAM cells compared to Rowhammer in DRAMs. *WriteHammer* causes resistance drift, which is different from bit flips in Rowhammer.

IV. Exerimental Results

In this section, we present experimental validation of the new *WriteHammer* attack.

A. Experimental setup

To assess the effectiveness of *WriteHammer*, we use a widely-used ReRAM Verilog-based compact model [16]. The model is fitted to the experimental data of HfO_x -based ReRAMs, which is the one of the most common type of ReRAM cell [16]. This model can reproduce both the transient behavior and the statistical characteristics of the ReRAM. The ReRAM cells are arranged in a 128×128 sized crossbar. The crossbars use a 0T1R configuration, i.e., there is no access transistor. Note that 128×128 is a common ReRAM crossbar size adopted in prior work and also for manufacturing [6][7]. We assume a 50 MHz operating frequency for the ReRAM crossbars, and the read and write voltages are 0.5V and 2V respectively [14]. The separation between two ReRAM rows is assumed to be 100 nm [19]. We vary the operating temperature from 25°C to 100°C to ensure thorough analysis. We use HSpice simulations for all our experiments. We apply *WriteHammer* for 4 ms as an example in every case to showcase how quickly the resistance of the victim cell can be changed. Recall that ReRAM cells are non-volatile and ideally have a retention time of higher than 10 years [11]. We simulate *WriteHammer* for 4 ms to show that the resistance (and hence the stored data) can be damaged in a significantly shorter period of time.

B. Effect of WriteHammer

Eliminating possibility of Rowhammer: Write operations in ReRAMs can affect neighboring cells due to both the constant bias voltage and the capacitive coupling between rows [9]. However, we first show that capacitive coupling (and hence Rowhammer) has little effect on the victim cell's resistance under normal operating conditions. Recall that Rowhammer happens due to the coupling between the two neighboring rows. By repeatedly accessing the aggressor row, the attacker can result in a non-zero voltage across the victim row due to the capacitive coupling. The capacitance depends on the physical dimensions of the crossbar, spacing between the cells, etc. However, this information is proprietary and not disclosed by the manufacturer. Hence, we study the effect of repeated writes at different capacitance values.

In the case of ReRAM crossbars, each write is also accompanied by the $V_{write}/2$ bias voltage. Hence, we study the combined effect of both the voltage bias and the capacitive coupling in Table 1. We also study a hypothetical case where there is no capacitive coupling (hence no Rowhammer). Table 1 shows the change in resistance (normalized with respect to the initial resistance) at different capacitance values. For this experiment, we apply repeating RESET pulses to the aggressor cell (simulating a repeated WRITE operation), and we observe the effect on the victim cell. Here, we assume an initial gap length of 0.8 nm for the victim ReRAM cell, and an operating temperature of 25°C; gap length is defined as the distance between the tip of the filament and the opposite electrode in an ReRAM cell. Here, we choose the operating temperature and gap length values as an example to demonstrate the effect of repeated accesses. We show the effect of *WriteHammer* at other operating temperatures and gap length conditions later. As shown in Table 1, the resistance of the victim cell remains mostly unaffected when the coupling capacitance is below 1 pF. The amount of change due to coupling is the same as without coupling, which indicates that Rowhammer has little to no effect. Beyond 1 pF, we start to observe the effects of capacitive coupling. This is expected as higher coupling allows for more voltage across the victim cell, which causes resistance drift. For instance, we see a 3.69% drift at 1 pF. However, note that such large coupling capacitance values are generally not feasible as it would require very large crossbars and/or extremely low separation between adjacent cells.

Typically, a challenge in security vulnerability assessment is that the information about physical dimensions of a crossbar, and the distance between each cell, are not disclosed by manufacturers. From prior studies on DRAM [19], we estimate the parasitic coupling of a typical 128×128 sized crossbar to be around 10 fF. For all experiments henceforth, we shall use 10 fF as the coupling capacitance (unless otherwise specified). Similar observations are made with SET pulses and at relatively higher temperature conditions (with 10 fF capacitance). Overall, Table 1 shows that Rowhammer is only effective at extremely high temperatures (>100°C) and at

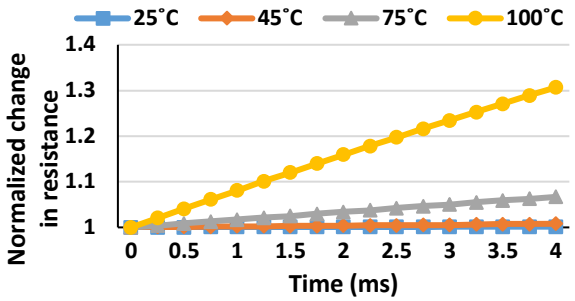

Fig. 4: Gradual change in resistance over time when *WriteHammer* is applied at different temperatures.

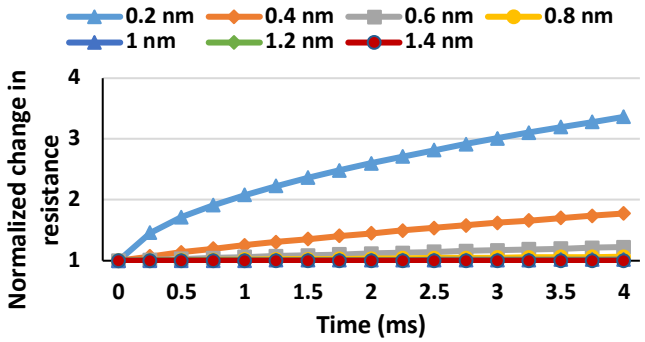

Fig. 5: Gradual change in resistance at different gap length of victim cell when *WriteHammer* attack is used.

very high capacitive coupling, both of which are unrealistic. Hence, we can eliminate the possibility of resistance drift due to Rowhammer for all the remaining sets of experiments.

Role of temperature: Next, we investigate the effect of repeated writes at different operating temperatures due to *WriteHammer* attacks. As mentioned in Section 3, the average rate of growth and the amplitude of variation of the conducting filament in an ReRAM cell have a strong dependence on temperature. Fig. 4 shows the change in resistance (normalized with respect to the initial resistance) when *WriteHammer* is applied at different operating temperature. For this experiment, we also assume a gap length of 0.8nm and apply RESET pulse for 4 ms duration. As shown in Fig. 4, there is no noticeable change in the victim cell's resistance at room temperature (25°C). However, we start to see resistance drift at relatively higher temperatures. For instance, at 75°C, we see a 6.7% drift, which increases to 30% at 100°C. This happens as high temperature assists in the formation (or rupture) of the conducting filament of the ReRAM cell. Fig. 4 clearly shows that the severity of *WriteHammer* increases with temperature. We note that the temperature required for *WriteHammer* is significantly lower than in the case of Rowhammer (which requires >100°C temperature [9]). *WriteHammer* can cause resistance drift at temperatures well below 100°C. This observation can be exploited by an attacker to launch effective *WriteHammer* attacks.

Effect of gap length: Next, Fig. 5 shows the effectiveness of *WriteHammer* at different victim cell gap length. As shown in Fig. 3, it is more difficult to push the vacancies towards an electrode if all the vacancies are already clustered near it (and vice-versa). We can confirm this from Fig. 5, which shows that it is easier for resistance to drift when the gap length is lower. For this experiment, we choose an operating temperature of 75°C and apply repeated RESET pulses. The RESET pulses are applied consecutively without any pause in between for a 4 ms duration. The gap length is varied from 0.2 nm to 1.6 nm, which represents the minimum and maximum gap length for the Verilog model [16]. Note that for each case, we normalize the change in resistance to the initial resistance; since the gap lengths are different, the initial resistance values for all the different cases here are not the same.

As shown in Fig. 5, *WriteHammer* causes an almost 3.5× change in the resistance of the victim cell compared to its initial resistance at gap length of 0.2 nm. However, at a gap length of 1.6 nm, *WriteHammer* fails to cause any noticeable change in the victim cell's resistance. This can be attributed to the fact that the gap length is already at its maximum value (determined by the physical dimensions of the ReRAM cell). The application of any RESET voltage pulses cannot increase the gap length any further; hence there is no impact on the victim cell when gap length is 1.6 nm. The observations are reversed when SET pulses are used (instead of RESET), i.e., resistance drift is highest when gap length is maximum and vice-versa.

These observations are interesting as they show that the effectiveness of *WriteHammer* depends on the initial state of each device. Moreover, since the data is represented as resistance, which in turn is governed by the gap length parameter, this observation implies that the effect of *WriteHammer* is dependent on the stored data. If an adversary is aware of the values stored on the victim cells, this observation can be used to target specific cells/rows using suitable *WriteHammer* attacks. As an example, graph data tends to be extremely sparse. Hence, if graph data is stored on ReRAM crossbars, many cells will have '0' stored on them i.e., high gap length. If an attacker is aware of this information, they can target these cells using repeated SET pulses to maximize their chances to cause resistance drift.

Effect of attack pattern: So far, we have studied the effectiveness of *WriteHammer* assuming that the attacker only applies the same type of pulse (SET or RESET) for an extended duration of time. However, the attacker can also choose to include both types of pulses to camouflage the attack and avoid detection. Next, we investigate the effect of *WriteHammer* if an attacker uses a mix of SET and RESET pulses in different proportions. Fig. 6 shows the change in resistance (compared to the starting resistance) when different input patterns are applied to the aggressor cell. 'nR-mS' in Fig. 6 represents n RESET pulses followed by m SET pulses. As shown in Fig. 6, combining both SET and RESET reduces the resistance drift caused by *WriteHammer*. The application of SET pulse partially nullifies the resistance drift due to a RESET pulse as a voltage of opposite polarity is used for SET (compared to RESET). As shown in Fig. 6, the effect is highest when there is a higher imbalance between the number

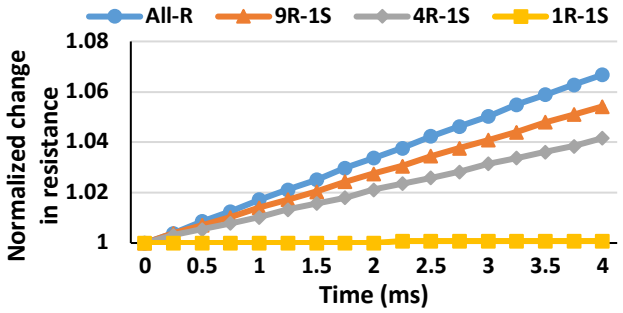

Fig. 6: The effect of *WriteHammer* attack with respect to different input (SET/RESET) pattern; 'nR-mS' indicates *n* RESET followed by *m* SET operations

Table II: Change in resistance with Ti-based ReRAM cells during *WriteHammer* attack under different settings.

Input type (SET/RESET)	Temperature (℃)	Gap length (nm)	ΔR after 4 ms (%)
SET	45	6	28%
SET	45	8	4%
RESET	45	6	18%
RESET	75	6	19%
RESET	45	8	17%
RESET	75	8	18%

of SET and RESET operations. There is almost no effective resistance drift when the number of SET and RESET pulses are uniform (the 1R-1S configuration in Fig. 6). This observation can be utilized by an adversary to camouflage attacks, making it significantly more difficult to detect *WriteHammer* attacks.

C. Effect of WriteHammer on other types of ReRAM cells

Finally, we demonstrate *WriteHammer* with another type of ReRAM device. For this purpose, we use the compact model from [17]. This ReRAM model is based on TiO_2-TiO_{2-x} memristors, following the classic ion transportation theory [17]. The model can simulate real-time switching, which is a critical feature in memristor-based analog circuit design. TiO2-TiO2–x is another promising technology for ReRAMs and it has received significant attention [20]. The behavior of the TiO2-TiO2–x device is modeled based on the ion/vacancy motion driven by the electric field [17]. The model behavior matches the measurements of a real TiO2-TiO2–x device on the static I-V curve and dynamic pulse programming [17].

Our evaluation on two different types of ReRAM models is aimed at showing that the *WriteHammer* attack is not specific to one type of ReRAM model. Table II shows the resistance change in the victim cell due to *WriteHammer* at different operating conditions using this model. Here, we consider different scenarios for a thorough analysis: we consider both SET and RESET pulses, two different operating temperatures, and two gap length configurations. Following similar setting assumed for the HfO_x-based ReRAMs, we assume a write voltage of 2V, operating frequency of 50 MHz. As usual, every write operation requires the bias voltage to be applied across the other cells (besides the one being written) to reduce sneak current. From Table II, we can see that this results in a significant amount of resistance drift on the victim cells. We can observe resistance drift under all operating conditions, with a maximum of 28% drift in resistance at 45℃ and 6 nm gap length. Table II clearly shows that *WriteHammer* is significant for this type of ReRAM cell too.

V. Conclusion

ReRAM cells are one of the primary choices for future memory and computing purposes. ReRAM offers low energy dense storage and is non-volatile in nature. However, the security vulnerabilities in these architectures have not received much attention. We have identified one such vulnerability in ReRAM-based architectures. We refer to it as *WriteHammer*. *WriteHammer* is implemented using typical write operations and takes advantage of the constant bias that is applied to reduce sneak current via adjacent ReRAM cells. We have shown how the effectiveness of *WriteHammer* attacks varies with the gap length, temperature, and input pattern. These properties make it significantly more difficult to detect these attacks compared to traditional Rowhammer attacks. Overall, these attacks can cause up to 3.5X change in resistance of an ReRAM cell; therefore, countermeasures are needed to address this security vulnerability.

References

[1] M. Patel, J. S Kim, and O. Mutlu, “The reach profiler (reaper): Enabling the mitigation of dram retention failures via profiling at aggressive conditions,” in ACM SIGARCH Comp. Arch. News, 45(2), pp. 255–268, 2017.

[2] B. Li, B. Yan, and H. Li, “An Overview of In-memory Processing with Emerging Non-volatile Memory for Data-intensive Applications,” in GLSVLSI , New York, NY, pp. 381–386, 2019.

[3] F. Zahoor, T. Z. A. Zulkifli, and F. A. Khanday, “Resistive Random Access Memory (RRAM): an Overview of Materials, Switching Mechanism, Performance, Multilevel Cell (mlc) Storage, Modeling, and Applications, “ in *Nanoscale Res Lett* 15, 90, 2020.

[4] M. Giordano et al., "CHIMERA: A 0.92 TOPS, 2.2 TOPS/W Edge AI Accelerator with 2 MByte On-Chip Foundry Resistive RAM for Efficient Training and Inference," in Symp. on VLSI Circuits, pp. 1-2, 2021.

[5] A. Grossi et al., "Resistive RAM Endurance: Array-Level Characterization and Correction Techniques Targeting Deep Learning Applications," in IEEE Trans. on Electron Devices, vol. 66, no. 3, pp. 1281-1288, 2019.

[6] A. Shafiee et. al., “ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars,” in SIGARCH Comp. Arch. News 44, 3, 14–26, 2016.

[7] B. K. Joardar, et. al., "AccuReD: High Accuracy Training of CNNs on ReRAM/GPU Heterogeneous 3D Architecture," in IEEE TCAD, vol. 40, no. 5, pp. 971-984, 2021.

[8] A. Chaudhuri and K. Chakrabarty, "Analysis of Process Variations, Defects, and Design-Induced Coupling in Memristors," in ITC, pp. 1-10, 2018.

[9] F. Staudigl, et. al., “NeuroHammer: inducing bit-flips in memristive crossbar memories,” in DATE, Leuven, Belgium, 1181–1184, 2022.

[10] Y. Kim et al., “Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors,” in ISCA, pp. 361–372, 2014.

[11] X. Dong, C. Xu, Y. Xie and N. P. Jouppi, "NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory," in IEEE TCAD, vol. 31, no. 7, pp. 994-1007, 2012.

[12] L. Song, Y. Zhuo, X. Qian, H. Li and Y. Chen, "GraphR: Accelerating Graph Processing Using ReRAM," in HPCA, pp. 531-543, 2018.

[13] D. Fujiki, S. Mahlke, and R. Das, “In-Memory Data Parallel Processor,” in ASPLOS, New York, NY, 1–14, 2018.

[14] P. -Y. Chen, X. Peng and S. Yu, "NeuroSim: A Circuit-Level Macro Model for Benchmarking Neuro-Inspired Architectures in Online Learning," in IEEE TCAD, vol. 37, no. 12, pp. 3067-3080, 2018.

[15] C. Xu et al., "Overcoming the challenges of crossbar resistive memory architectures," in HPCA, pp. 476-488, 2015.

[16] P. -Y. Chen and S. Yu, "Compact Modeling of RRAM Devices and Its Applications in 1T1R and 1S1R Array Design," in IEEE Trans. on Electron Devices, vol. 62, no. 12, pp. 4022-4028, 2015.

[17] Lu Zhang et. al., “A compact modeling of TiO2-TiO2–x memristor,” in Appl. Phys. Lett. 102, 153503, 2013.

[18] A. I. Arka, B. K. Joardar, J. R. Doppa, P. P. Pande, and K. Chakrabarty, "ReGraphX: NoC-enabled 3D Heterogeneous ReRAM Architecture for Training Graph Neural Networks," in DATE, pp. 1667-1672, 2021.

[19] M. Redeker, B. F. Cockburn and D. G. Elliott, "An investigation into crosstalk noise in DRAM structures," in IEEE Intl. Workshop on Memory Technology, Design and Testing, pp. 123-129, 2002.

[20] F. Miao, J. J. Yang, J. P. Strachan, D. Stewart, R. S. Williams, and C. N. Lau, "Force modulation of tunnel gaps in metal oxide memristive nanoswitches," in Applied Physics Letters, vol. 95, p. 113503, 2009.

[21] C. Rajamanikkam, J. S. Rajesh, S. Roy, K. Chakraborty, “Understanding Security Threats in Emerging Neuromorphic Computing Architecture,” in Journ. Hardware System Security 5, 45–57, 2021.