A REVIEW ON STUDY OF DEFECTS OF DRAM-ROWHAMMER AND IT'S MITIGATION

[1] **Sonia Sharma** , [2] **Debdeep Sanyal**, [3] **Arpit Mukhopadhyay**, [4] **Ramij Hasan Shaik**

[1] PROFESSOR & HEAD, DEPARTMENT OF CSE, M.I.M.I.T, MALOUT, INDIA

[2,3,4] **Student,** DEPARTMENT OF CSE, **Malout Institute of Management & Information Technology(MIMIT),Punjab**

----------***----------

ABSTRACT

Row hammer is a key specimen of how a circuit-level failure process can cause a practical and extensive system security susceptibility. It is the circumstance that frequently accessing a row in a latest DRAM chip causes bit flips in physically adjacent rows at invariably foreseeable bit locations. DRAM disturbance errors that is a demonstration of circuit level cell-to-cell interference in a spanned memory technology, is a hardware failure method which causes Row hammer. The truculent memory density ascending gives rise to modern DRAM devices to tolerate from Row hammer, an occurrence where speedily activating (i.e., hammering) and deactivating/discharging a DRAM row can produce bit-flips in physically close by rows. It is observed that the un-authorized users can utilize Row hammer bit-flips to infallible mount system-level strikes to surge prerogative and leaked one's personal data. It is condemnatory to make sure Row hammer safe functioning on all DRAM-based structures, as they set off progressively more unsafe to Row hammer. In this paper The author will be going through the Experimental Methodology as discussed in [4] and two mitigation techniques with exploiting time window counter and block hammer and recent row hammer attacks.

Keyword-DRAM, hammering, bit-flips, refresh interval, aggressor row,TWICE

1. PROBLEMS

Isolating memory is a main feature of a sealed, secure and reliable computing system. It should be maintained that accessing one memory address should not show undesired side effects on the information stored in other addresses. It is shown that memory chips become more endangered to the inconvenience when process technology is cut down to comparatively smaller dimensions. There is an occurrence of interference between the operations of different memory cells. Frequently accessing from the exact same address might corrupt data of the neighbouring addresses. Most significantly, when a DRAM is activated and deactivated again and again, one or more bits in physically adjacent DRAM rows can be changed to the different values. Increasing memory density of DRAM chips generates Row Hammer. The solution of this problem requires knowledge of DRAM internals.

2. INTRODUCTION

Memory is a key unit of all latest computing systems, often determining the global performance, energy ability, and reliability characteristics of the total system. The push for expanding the density of latest storage technologies via technology scaling, which has ensured in greater capacity (i.e. density) memory and storage at reduced cost, has enabled big leaps in the performance of latest computers developments have been made to the technology of developing DRAM have increased DRAM storage compactness by diminishing DRAM cell size and cell-to-cell spacing for decennium. The author identifies the root cause of DRAM disturbance errors as voltage fluctuations on an internal wire called the word line. DRAM comprises a two-dimensional array of cells, where each row of cells has its own word line. To access a cell within a particular row, the row's word line must be enabled by raising its voltage — i.e., the row must be activated. When the same row is activated many times, it forces the word line to switch between on and off, again and again. According to our observations, such voltage fluctuations on a row's word line have a disturbance effect on nearby rows, inducing some of their cells to leak charge at an accelerated rate. If a cell suffers the loss of too much charge before it is refreshed (restoration of original values), a disturbance error is produced. These optimizations negatively affect DRAM reliability as these refine chip's cost per bit. The latest DRAM chips are open to the

Row hammer occurrence, where opening and closing a DRAM row (i.e. aggressor row) at a large rate (i.e. hammering) can cause bit-flips in physically nearby rows (i.e., victim rows). Therefore, saving DRAM against all types of Row hammer attacks is crucial for the security and reliability of current and future DRAM-based computing systems. DRAM merchants presently implement in-DRAM Row hammer mitigation processes. The chips become further exposed to Row hammer, most state-of-the-art processes of all 4 approaches either cannot comfortably adapt because they are based on fixed design points, or their performance, energy, and/or area overheads become increasingly significant. (i) Increasing the refresh rate further in order to prevent all Row hammer bit-flips is prohibitively expensive, even for existing DRAM chips, due to the large number of rows that must be refreshed within a refresh window. (ii) Physical desolation processes must provide greater isolation (i.e. boosts the physical distance) between sensitive data and a potential attacker's storage space as DRAM chips become denser and more exposed to Row hammer.

3. METHODOLOGY

DRAM operation basics- Fundamentals of DRAM Operation: In DRAM memory technology, MOS technology is at the heart of its design, manufacturing, and operation. Looking at how DRAM works, a basic dynamic RAM or DRAM memory cell uses a capacitor to store each bit of data and a transfer device (MOSFET) as a switch. The charge level on the memory cell capacitor determines whether that bit is a logic '1' or '0'. It indicates a logic '1' when there is charge on the capacitor and a logic '0' when there is no charge. ". The basic format of dynamic DRAM cell is given in the figure. The format is so simple. It can be arranged densely on a silicon chip. These results lower the cost.

There are two lines connected to each dynamic RAM cell - a word line (W/L) and a bit line (B/L) are connected as shown so that the desired cell in the array is a data line. can read and write.

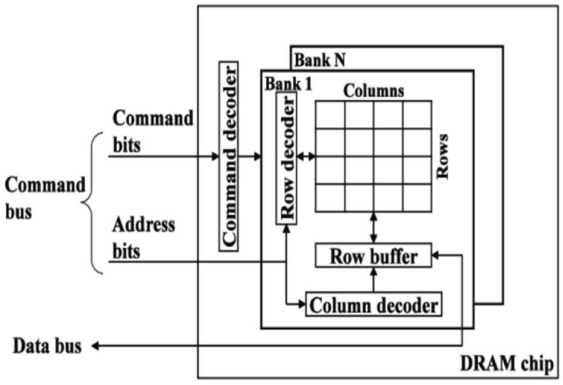

Fig 1-Architecture of DRAM Rows

The following algorithm represents the characteristics and effects of ROW HAMMER.

3.1 PSEUDO CODE FOR ROW HAMMER

```
DRAM_RowHammer_Characterization()
{
        for DP in [Data Patterns]
                write the DP to all cells of DRAM
                for row in DRAM
                        victimRow = row
                        aggressorRow1 = victimRow -1
                        aggressorRow2 = victimRow + 1
                 for HC in [HC sweep]
                        Disable DRAM refresh
                        Refresh victimRow
                 for n = 1->HC
                        start aggressorRow1
                        start aggressorRow2
                        Enable DRAM Refresh
        Record Row hammer Bit flips to storage
        Restore bit flips
```

It is an overhead of having DRAM refresh in the test case.
First, DRAM refresh is disabled.
Intentionally, fully charged row is induced to generate bit flips.
In the test loop, the adjacent rows are accessed.
1 Hammer Count = 2 accesses
DRAM refresh is enabled to avert further control failures.
Bit Flips are recorded for further analysis.

3.2 SYSTEM DEMONSTRATION ON ROW HAMMER

3.2.1 Disturbance Error-As technologies have advanced, the cell density of DRAM has increased and the cells are no longer isolated, causing them to electrically interact with each other. Therefore, a refresh operation to restore charge in a DRAM cell caused by charge leakages affects the neighbouring cells, causing disturbance errors. Recently, an attack called a row hammer attack has been reported that applies a bit flip attack by deliberately generating such disturbance errors. If disturbance errors can be induced, the row hammers can be exploited in system-level attacks to escalate privileges, leak confidential data, and cause a denial of service. Accordingly, hardware manufacturers have recently adopted a target row refresh (TRR) to protect against row hammer attacks and reduce the vulnerability of DRAM chips. In addition, memory controller and system manufacturers have implemented countermeasures such as increasing the refresh rate.

The mechanism of a row hammer attack involves deliberately generating a disturbance error in DRAM, as described earlier. The steps involved in the row hammer attack mechanism are as follows [4]:

Step 1: Two mov instructions (codes 1 and 2) read data from the DRAM at addresses X and Y and load the data into the register and cache.

Step 2: Two "clflush" instructions (codes 3 and 4) evict the data that were loaded in the cache, enabling data to be read directly from DRAM rather than from the cache.

Step 3: Finally, the iteration of these instructions allows repeated hammering, a process of memory reading from DRAM, and thereby enables bit flips to be applied by causing disturbance errors.

To generate a disturbance error by accessing addresses X and Y as above, addresses X and Y must be mapped to the same bank and to unlike rows instantaneously. To purposely generate a disturbance error by retrieving the desired address, the address in the adjacent row of the same bank must be known in advance. Accordingly, the virtual address needs to be mapped to the physical address. Therefore, a predictable method or a probabilistic method is used to identify the corresponding addresses. It is also necessary to avoid the cache because a quick activation of the rows in each bank of DRAM is vital to create the bit flips. To avoid the cache, it needs to clear the cache line using the clflush instruction

```
1 mov(X),%eax //read values of address X and Y
2 mov(Y),%ebx
3 clflush(X)      //evict data on cache
4 clflush(Y)
5 jmp code      // repeat above stage
```

Assembly code generated on INTEL/AMD machine

Storage isolation is a key property of a consistent and secure computer system. An access to 1 memory address shouldn't have unintentional side effects on data stored in other Addresses. Though, as process technology scales down to lesser dimensions, memory chips become more susceptible to disturbance, a phenomenon in which different memory cells interfere with each other's' operation. In ISCA 2014 paper [4], the existence of disturbance errors in commodity DRAM chips that are sold and used in the field. Frequently reading from the same address in DRAM could corrupt data in nearby addresses. Precisely, when a DRAM row is activated and pre-charged frequently (i.e., hammered), enough times within a DRAM refresh interval, one or more bits in physically adjacent DRAM rows can be flipped to the wrong value. This DRAM failure mode is now commonly called Row-hammer. Using an FPGA-based experimental DRAM testing infrastructure, which we originally developed for testing retention time issues in DRAM, it is tested 129 DRAM modules developed by three major developers (A, B, C) in seven recent years (2008–2014) and found that 110 of them exhibited Row hammer errors, the earliest of which dates to 2010. This is illustrated in the given figure, which shows the error rates found in all 129 modules we tested where modules are categorized based on manufacturing date.2 In particular, all DRAM modules from 2012–2013 were vulnerable to Row hammer, indicating that Row hammer is a recent phenomenon affecting more advanced process technology generations (as also demonstrated repeatedly by many works that come after our ISCA 2014 paper [3].

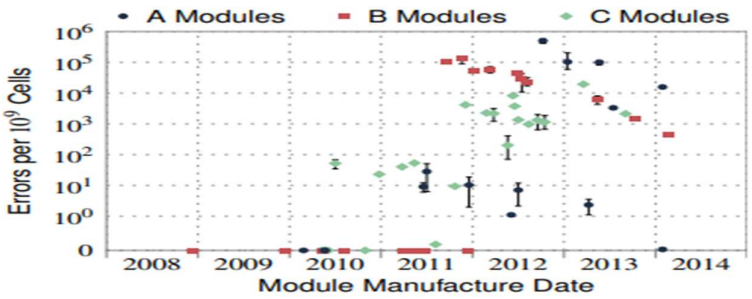

Fig 2: Normalized number of errors vs. manufacture date

3.2.2 RH DETECTION TECHNIQUE-As the density of DRAM increases the chances of having more than 2 victim rows per aggressor row also increases .RH attacks has both direct and indirect impact on DRAM.The bit flips comes under the Direct attack.The row hammer is malicious attack which is activating the aggressor row.The DRAM bank can at the same time of the attack been shared by other operations,and it increases the blockage time.There are two types of RH detection –(1)Deterministic Detection (2)Probabilistic Detection

Naïve Deterministic Approach:

- use a counter for each row in MC and count the total number of activations per row
- If any counter reaches its threshold value, then refresh it's neighbouring rows
- After every refresh reset the corresponding counter values

The disadvantages of Naïve's approach is performance area and energy overhead

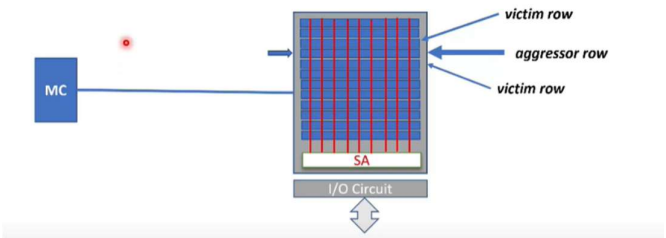

Fig 3: View of Row hammer Attack

4. ROW HAMMER MITIGATION

4.1 Exploiting Time Windows Couter-

- Performing one ACT needs tRC times
- Total ACTS possible per refresh period: tREFW /tRC
- Total aggressor rows per refresh period : tREFW/tRCx N_{th}
- Total victim rows per refresh period (assuming 2 victim rows per aggressor row): 2 X tREFW/Trc x N_{th}

Only up to 20 rows can be exposed to the RH attack from a bank in the duration of tREFW

4.1.1Time Window Couter-TWICE guarantees protection against RH attacks with minimum number of counters. They remove the entry of a row which is not activating frequently during a Window Period called Pruning Interval (PI).TWiCE consists of a counter Table called and a counter logic as shown in the given figure. As each row is refreshed once every refresh window (tREFW), the number of ACTS to a row must

exceed th_{RH} within tREFW for a successful RH attack.Thus, the average number of ACTS to an aggressor row over a refresh interval (tREFI) must exceed $\frac{thRH}{tREFW/tREFI}$

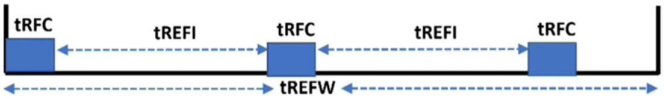

Fig 4: Demonstration of Terms by TWICE

4.1.2 Operations-TWICE receives a DRAM command and address pair. For each DRAM ACT, TWiCE allocates an entry in the counter table if the entry for the row does not already exist, and increments the counter (*act_cnt*) by one.If *act_cnt* reaches th_{RH}, TWiCE refreshes the adjacent rows of the entry and deallocates the entry. After each pruning interval (*PI=tREFI*), each entry in the TWiCE table is checked and removed if act_cnt <thpi x life.(In other words, a row is considered to be an aggressor candidate only if the average number of ACTS over tREFI is equal to or greater than th_{PI}).This step enables the counter table size to be bounded. For the remaining entries, life is incremented by one[5].

4.1.3 Proof of RH Prevention-Consider the maximum number of ACTS to a row over tREFW when the row is not tracked by the TWiCE table is: count $_{not\text{-}tracked}$

- TWiCE keeps a row in its counter table if act_cnt >= thpt x life
- Hence count not-tracked must be less than th_{PI} x life
- Given the maximum value of life over the refresh window is tREFW /REFI theH and th_{PI} is $\frac{thRH}{tREFW/tREFI}$

$$\text{count}_{\text{not-tracked}} < \text{thpi X tREFW /tREFI} = \text{thRH}$$

In other words, if a row is activated the times or more within a refresh window, it will be in the counter table.Consider ACT count of a row while considered as aggressor is:$count_{tracked}$ which must be less than th_{RH} attack is detected as an aggressor is $count_{combined}$=$count_{not\text{-}traced}$+$count_{tracked}$<2.thRH).A row needs to experience 139K or more ACTs on it's neighbour rows with tREFW to have a bit flip**(Nth=139K).**Considering that a row has two adjacent rows in general(double-side RH),the actual threshold to detect an aggressor is it's half,69K.In order to ensure that $count_{combined}$ does not exceed threshold,69K, th_{RH} should be less than half of 69K.In this TWiCE,the value of th_{RH} is set as 32,768.

4.1.4 Couter Table Size- TWICE consider one counter table per bank. The maximum number of ACTS in a DRAM bank during tREFI max_{act}=(tREFI-tRFC)/tRC.(**With tREFI of 7.8 µs and tRC of 45 ns, max_{act} is 165**). The table size should be set based on the worst case when the table has the largest number of valid entries.The valid entries fall into two categories(1) Entries newly inserted in the current Pl, and (2) Entries identified as aggressor candidates in the previous Pls. The number of new entries is bounded by max_{act}. The number of surviving entries is maximized when the counter entries with the smallest life survive the most. The maximum number of entries with life = 2 is maxact/1xth_{PI}. Similarly, the maximum number of entries whose life is n can be calculated as max_{act} /(n-1)xth_{PI} .

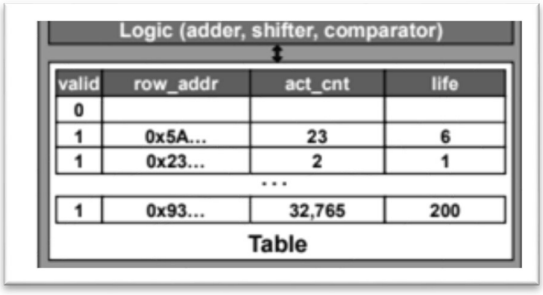

Fig 5: The organization of TWiCE. Each table entry holds valid bit, row addr, act cnt, and li f e. An entry is inserted when a new row is activated and invalidated when pruned or refreshed after thRH is reached.

Table 1: Defination & Type of values by TWiCE

TERM	DEFINATION	TYPICAL VALUE
tREFW	Refresh window	64 ms
tREFI	Refresh interval	7.8us
tRFC	Refresh command time	350ns
tRC	ACT to ACT interval	45ns
tH_{RH}	RH detection threshold	32.768
tH_{PI}	Pruning interval threshold	4
max_{ACT}	Max # Of ACTs during PI	165
max_{life}	Max life of row in PI	8.192
(N_{th}=139K)		

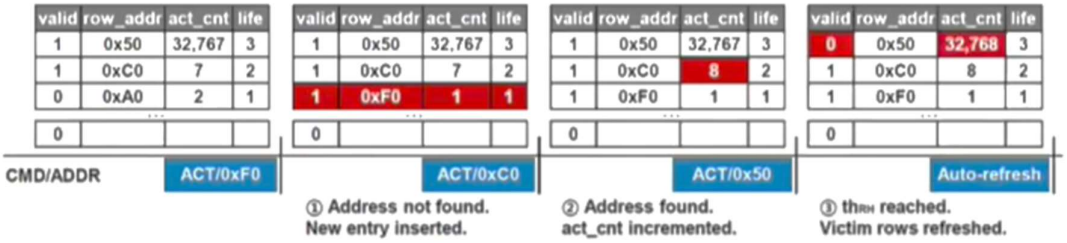

Fig 6:TWiCe operation example.The DRAM command/address changes in TWiCE are coloured blue and red respectively.1-When the target address of ACT is not found, a new entry is inserted 2.When found act_cnt is incremented by 1. 3.If act_cnt reaches thRH, the victim rows are refreshed and entry is invalidated.4.During an auto refresh, the table is updated; the aggressor candidates life is increased by 1,while others are pruned.

4.2 Block-Hammer- BlockHammer[1] is designed to (1) scale efficiently as DRAM chips become increasingly vulnerable to Row hammer and (2) be compatible with commodity DRAM chips. BlockHammer consists of two components. The first component, RowBlocker , prevents any possibility of a Row hammer bit-flip by making it impossible to access a DRAM row at a high enough rate to induce Row hammer bit-flips. RowBlocker achieves this by efficiently tracking row activation rates using Bloom filters and throttling the row activations that target rows with high activation rates. We implement RowBlocker entirely within the memory controller, ensuring Row hammer-safe operation without any proprietary information about or modifications to the DRAM chip. Therefore, RowBlocker is compatible with all commodity DRAM chips. The second component, AttackThrottler , alleviates the performance degradation a Row hammer attack can impose upon benign applications by selectively reducing the memory bandwidth usage of only threads that AttackThrottler identifies as likely Row hammer attacks (i.e., attacker threads). By doing so, AttackThrottler provides a larger memory bandwidth to benign applications compared to a baseline system that does not throttle attacker threads. As DRAM chips become more vulnerable to Row hammer, AttackThrottler throttles attacker threads more aggressively, freeing even more memory bandwidth for benign applications to use. By combining RowBlocker and AttackThrottler, BlockHammer achieves both of its design goals.

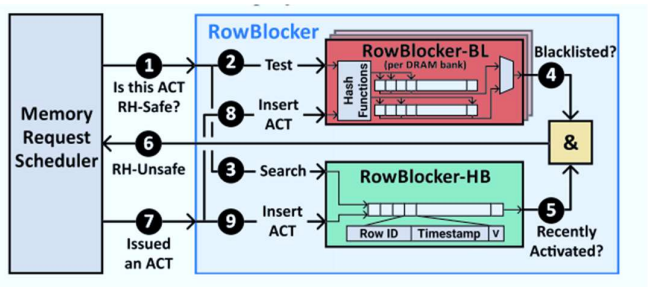

Fig 7: Block Hammer

4.3 Mitigating Techniques on Hardware

The hardware-based techniques used for the mitigation of row hammer attacks are as follows:

- **CRA and PRA**: Counter-based row activation (CRA) and probabilistic row activation (PRA) have been planned to reduce row hammer attacks. CRA employs a counter to compute the no. of activations of every row, and if the corresponding counter exceeds the hammer threshold, a dummy activation is actively transmitted to refresh the info. PRA is used to decrease the overhead that is incurred throughout CRA, that facilitates the generation of a dummy activation probabilistically for all memory accesses. CRA and PRA need an additional counter to count the amount of hammerings of the victim rows. For instance, if there is an 8 Giga-Bytes storage system with one million rows, 2 Mega-Bytes are needed for the total counter size. Because this requires additional memory consumption, mitigation methods using CRA and PRA can downgrade the performance.
- **GuardION**: GuardION guards Against direct memory access (DMA)-based attacks in an ARM situation. This technique allows buffers to be physically isolated by adding 2 empty rows known as guard rows. It removes the choice of a bit flip because the victim rows can be arranged in more than one row away from the attacked rows. However, GuardION consumes additional DRAM memory of the users device to add the Guard-rows, which can affect the device performance.

- **ECC memory**: To Defend against row hammer attacks, error correcting code (ECC) storage is announced in each rank of DRAM. This method Adds a parity bit for sleuthing errors occurring in data bits and a control bit for transmitting whether an error has occurred. ECC memory was used to detect & correct the errors that could occur owing to the external environment of DRAM, and the introduction of ECC memory became an obstacle for misusing during a row-hammer attack. However, it has been reported that an attack called ECC-ploit can be used to conduct a row-hammer attack even in ECC memory-added DRAM.
- **TRR**: In reply to row-hammer Attacks, a target row refresh (TRR) has been implemented in a DDR4 model to refresh adjacent rows when an access is tried beyond the threshold value. Though, even in TRR-applied DRAM, row-hammer attacks can be reactivated by enabling bit flip attacks through new hammering patterns.

The methods that use hardware to reduce row-hammer attacks require extra hardware resources to detect or monitor errors, which can downgrade the performance of present devices. Also, the row-hammer may be reactivated despite the application of row-hammer attack reduction techniques to DRAM. summarizes the hardware-based techniques used for reducing row hammer attacks.

Table 2: Analysis of mitigating row hammer attacks based on hardware

Mitigation Technique	Description	Disadvantages
CRA & PRA	Counters are added to calculate the number of activations of each row to apply a data refresh based on threshold of the number of hammerings	Degradation of memory performance from memory overhead and additional counters
Guard Ion	Guard rows are added to reduce the possibility of rowhammer occurrence through physical isolation of buffer	Degradation of memory performance from the use of memory resources in DRAM
ECC memory	A parity bit that locates errors in a data bit is appended to detect and correct memory errors in DRAM	Degradation of memory performance from additional bit adoption and calculation and the possibility of bypassing through ECC-pilot attack.
TRR(target row refresh)	The number of row activations is calculated and if the corresponding count exceeds the threshold, a data refresh is used	The possibility of bypassing through new hammering patterns such as TRR-pass.

4.4 Mitigating Techniques Based on Software

The software-based techniques used for mitigation of row hammer attacks are as follows:

- **ANVIL**: ANVIL, a software-based method for reducing row hammers, detects row hammer attacks by tracking the access location of DRAM using an existing hardware performance counter. Subsequently, victim rows within the vicinity are selectively refreshed to prevent hammering operations on the victim rows that are frequently accessed through row hammer attacks. However, because this technique uses current hardware counters to dynamically perceive and respond to attacks, an added system overhead may occur, which also needs modification in the Linux kernel.
- Technique using tilted Hash tree: The amount of bit flip is detected through a safe Hash algorithm-3 (SHA-3) Keccack Hash function-based dynamic integrity tree structure and a sliding window. This has been introduced to cause minimum problems and make it cost effective. But it demands a memory controller (MC) that stores the root hash and is not at risk of alteration.
- Technique using deep learning: Another study accepted a convolutional neural network (C.N.N.) model, which is a deep-learning model, to analyse the access patterns of DRAM to predict the rows where row hammer attacks can arise in advance. However, it requires ill-using various row hammers in advance for analysis and training of the access patterns of DRAM.

As such, the techniques that use software to reduce row-hammer attacks require the process of monitoring DRAM to adjust the refresh rate or expecting the rows in which an Attack may occur by learning the DRAM access pattern.

Table 3: Analysis of mitigating row hammer attacks based on hardware

Mitigation Technique	Description	Disadvantages
ANVIL	A rowhammer attack can be detected through the location of rows frequently accessed by DRAM using the existing hardware performance counters	System overhead may be incurred and modification of Linux Kernel is required
Technique using skeThe authord hash tree	Bit flips can be detected through dynamic integrity tree structure and sliding window	Memory controller is needed to store the root hash and safety of the MC should be ensured
Technique using Deep Learning	Could occur, can be predicted by learning the access patterns of DRAM	The training of patterns exploiting various rowhammers is required for deep learning

5. RECENT ROW HAMMER ATTACKS

RAMPage Attack- Every Android device from the last 6 years may be at risk to RAMPage susceptibility. It has been found practically that all Android-powered devices freed in the last six years–basically, from the Ice-Cream Sandwich (4.0) era to now–are susceptible to a variant of the Row hammer memory bout called RAM page. The RAM Page attack (CVE-2018-9442) relies on the behaviour of an Android component called ION, the paper noted, which was introduced in 4.01 as a spare for merchant-specific memory running interfaces that had formerly been employed by device constructors. ION was also intended to work as an intermediary between memory allocations between the core Android OS and user space apps. While not impossible, RAM Page is harder to practically attack on end-user devices, partially as merchant-specific or device-specific problems make it more difficult to reliably generate the situations that allow for mistreatment. Because of the degree of accuracy intricate, it would theoretically be possible that the same model phone with DRAM from different vendors would have different avenues to attack, or that certain optional hardware protections of LPDDR4, if added at developing time, would partially mitigate the attack, the paper noted.

GLitch Attack: the 1st remote Row-hammer Attack Against Android phones. The scientists from the VUSec Lab at Vrije University Netherlands sustained their analysis of the Row-hammer Attack method and demonstrated how to influence Graphics Processing Units (G.P.U.) to target Android phones. The experts started with the utmost limitation of the D-rammer Attack that was signified by the need to have a spiteful application being installed on the target phones. Now for the initial time, the similar team of experts has planned a technique dubbed GLitch to conduct the Row hammer attack against an Android phone remotely. The GLitch technique influences embedded Graphics processing units (G.P.U.) to launch the attack. “We demonstrate that G.P.U. , already extensively employed to accelerate a variety of applications such as photo rendering, can also be used to “accelerate” micro-architectural Attacks on product platforms,” [3].“, we show that an Attacker can build all the necessary primitives for performing effective GPU-based micro-architectural attacks and that these primitives are all exposed to the web through consistent browser extensions, allowing side-channel and Row-hammer attacks as of JavaScript.”

The term GLitch comes from a widely used browser-based graphics code library known as WebGL for rendering graphics to activate a known glitch in D.D.R. storages. The experts published a GLitch proof-of-concept attack and demonstrated that it is possible to conduct a Row hammer attack by tricking victims into visiting a website hosting a malicious JavaScript code. By with this Attack system, it is likely to hack an Android phone in just 3 minutes remotely. The spiteful script runs only within the privileges of the internet

browser, which means that the attacker can harvest users' credentials and spy on user's browsing activity. The GLitch Attack could not permit threat performers to gain the complete control over the victim's Device. GLitch rather than influence the CPU like other application for the Row hammer technique uses the Graphics processing units (GPU).

"While powerful, these G.P.U. primitives aren't easy to execute owed to undocumented hardware options. We tend to describe novel reverse engineering techniques for glancing into the formerly unknown cache design and replacement policy of the Adreno 330, an integrated GPU found in several common mobile platforms," continues the paper. Affected phones run the Snapdragon-800 & 801 schemes on a chip; this infers that the Glitch attack only works only on grown-up Android devices, including LG Nexus 5, HTC One M8, or LG G2. The PoC code works against both Firefox and Chrome; the video demo researchers demonstrate the GLitch attack on a Nexus 5 running over Mozilla's Firefox browser.[4] Tactlessly, it is dreadful to solve the GLitch attack with a software patch because it influences hardware vulnerabilities. Experts warn of potential effects of Row-hammer attacks on a big scale; they are presently helping Google to solve the attacks.

Throw-hammer – Row hammer attack on a system in a LAN

With the GLitch attack experts demonstrated how to leverage graphics processing units (GPUs) to launch a remote Row hammer attack against Android smartphones, they also devised a variant of the Row hammer attack dubbed Throw hammer to target a system in a LAN. The technique was invented by the same team of researchers that proposed the earlier ones, a group of experts from the Vrije University Amsterdam and the University of Cyprus. This time the researchers demonstrated that sending malicious packets over LAN it is possible to implement a Row hammer attack on systems running Ethernet network cards equipped with Remote Direct Memory Access (RDMA). Such kind of configuration is widely adopted in cloud infrastructure and data centres.

6.CONCLUSIONS

Thus far, Row hammer has been commonly perceived as a dangerous hardware bug that allows attackers capable of executing code on a machine to escalate their privileges. In this paper we have gone through different aspects of Row hammer attack along with its characterization and affects.

7. REFERENCES

[1] Giray Yaglıkçı , Minesh Patel, Jeremie S. Kim, Ataberk Olgun, Roknoddin Azizi,Hasan Hassan, Lois Orosa, Jisung Park, Taha Shahroodi, Konstantinos Kanellopoulos, Saugata Ghose, Onur Mutlu **,** **"**BlockHammer: Preventing Row hammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows", arXiv:2102.05981v1 [cs.CR] ,11 Feb 2021

[2] Jeremie S. Kim, Minesh Patel, A. Giray Yağlıkçı,Hasan Hassan,Roknoddin Azizi,Lois Orosa, Onur Mutlu,ETH Zürich, "Revisiting Row hammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques", 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)

[3] Onur Mutlu,Jeremie S. Kim,ETH Zurich, "Row hammer: A Retrospective", Volume 39Issue 8Aug. 2020 pp 1555–1571

[4] Yoongu Kim,Ross Daly Jeremie Kim, Chris Fallin Ji Hye Lee,Donghyuk Lee,Chris Wilkerson,Konrad Lai Onur Mutlu "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors" kim-ISCA -2014

[5] Eojin Lee, Ingab Kang, Sukhan Lee, G. Edward Suh, Jung Ho Ahn "TWiCE: Preventing Row-hammering by Exploiting Time Window Counters" ISCA '19, June 22–26, 2019, Phoenix, AZ, USA

[6] Minkyung Lee and Jin Kwak, "Detection Technique of Software-Induced Row hammer Attacks" ,Computers, Materials & Continua.