Підвищення ефективності Row-Sampling методів для захисту від атак типу Row-Hummer

Improving the effectiveness of Row-Sampling methods to protect against Row-Hummer attacks

Валентин Мазурок [A]

Corresponding author: аспірант кафедри кібербезпеки, e-mail: valentin.mazurok@gmail.com, ORCID: 0009-0006-2174-0800

Володимир Луценко [A]

к.тех.н., старший науковий співробітник, доцент кафедри кібербезпеки, e-mail: lutsenkovn@ukr.net, ORCID: 0000-0001-7632-1730

Valentyn Mazurok [A]

Corresponding author: Postgraduate Student of the Department, e-mail: valentin.mazurok@gmail.com, ORCID: 0009-0006-2174-0800

Volodymyr Lutsenko [A]

Candidate of Technical Sciences, Senior Researcher, Associate Professor of the Department, e-mail: lutsenkovn@ukr.net, ORCID: 0000-0001-7632-1730

[A] Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", м. Київ, Україна

[A] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

Received: December 17, 2024 | Revised: December 22, 2024 | Accepted: December 31, 2024

DOI: 10.33445/sds.2024.14.6.7

Мета роботи: провести аналіз захисту систем від атак типу RowHammer на основі методу вибірки рядків та запропонувати її покращення використовуючи більш реалістичну модель пам'яті та модель атаки.

Результати дослідження: показано недоліки в представленні пам'яті в захисних механізмах на основі вибірки рядків та показано більш коректні способи обчислення порогу вибірки. Результати представлено на реальних прикладах покращення захисту пам'яті DRAM.

Практична цінність дослідження: Знайдені формули можуть значно покращити захищеність нових видів пам'яті DRAM від атак типу Rowhammer.

Цінність дослідження: представлено дані тестування нових чіпів пам'яті кількох виробників DRAM. Також представлено нові види обрахунку порогових значень для захисту від RowHammer.

Майбутні дослідження: це дослідження відкриває шляхи для покращення програмних методів захисту від RowHammer а також пропонує розробку технічних засобів для комбінацій з програмними рішеннями.

Тип статті: аналітична.

Purpose: to analyze the protection of systems against RowHammer attacks based on the row sampling method and to propose its improvement using a more realistic memory model and attack model.

Findings: shortcomings in the representation of memory in protection mechanisms based on row sampling are shown and more correct methods for calculating the sampling threshold are shown. The results are presented on real examples of improving DRAM memory protection.

Practical implications: The formulas found can significantly improve the protection of new types of DRAM memory against Rowhammer attacks.

Value: testing data of new memory chips from several DRAM manufacturers is presented. New types of threshold calculation for protection against RowHammer are also presented.

Future research: this research opens up ways to improve software methods for protection against RowHammer and also suggests the development of technical means for combinations with software solutions.

Papertype: analytical.

Ключові слова: RowHummer, Row-Sampling, RAM, DRAM, атаки на пам'ять.

Key words: RowHummer, Row-Sampling, RAM, DRAM, memory attacks.

Вступ

Захист Rowhammer на основі вибірки рядків є одним із найпростіших і найстаріших методів захисту [1], які можна застосувати до контролера пам'яті. Під час активації кожного рядка контролер пам'яті генерує випадкове значення з певним відхиленням. З низькою ймовірністю p << 1 цей рядок потрапляє до вибірки і розглядається як атакуючий. Після контролер пам'яті виконує захисні дії, наприклад, оновлює рядки жертви на відстані 1 від атакуючого. Високий поріг p запобігає атаці Rowhammer, оскільки це гарантує, що ряд агресора не зможе уникнути вибірки з високою ймовірністю. Перші статті про Rowhammer запропонували варіанти захисту на основі вибірки, зокрема "імовірнісну активацію сусідніх рядків" (PARA) [2] та "Активацію імовірнісного рядка" (PRA)[3]. Основною перевагою Row-Sampling є його простота: контролеру пам'яті не потрібно зберігати жоден стан, що різко відрізняється від інших методів захисту Rowhammer, які вимагають збереження і відстеження великих таблиць рядків [3, 4].

Ці переваги роблять Row-Sampling дуже привабливим для великих компаній, які розглядають можливість його використання у своїх контролерах пам'яті. Так у старіших версіях

процесорів Intel була реалізована форма вибірки рядків для захисту DDR3 DRAM, від RowHammer, яка називається pTRR [5]. Через те, що пам'ять DDR4 має вищу частоту оновлення розробники Intel, відмовилася від підтримки pTRR бо думали що цього достатньо для захисту від RowHammer. Однак тепер виявилося, що DDR4 все ще вразлива [4] і нещодавня робота показує, що новіші чіпи DRAM потребують навіть менше звернень до пам'яті, щоб біти почали змінювати значення. Враховуючи ці тенденції, ми очікуємо відновлення інтересу до методів вибірки рядків. Таким чином, постає важливе питання: яке значення має бути встановлено поріг *p*, щоб забезпечити адекватний рівень захисту? Відповідь на це запитання має надаватися цілісно для всієї системи протягом усього терміну її експлуатації (а не лише для окремого блоку чи окремої частоти оновлення).

Результати

Модель атак

Rowhammer — добре відома вразливість DRAM, яка спричиняє зміну значення бітів [2]. Це виникає через частий запис "атакуючих" рядків, що через фактичне розміщення роблять наведення на рядках сусідах ("жертвах"). Як ми показували в минулій статті [6], бітові зміни можна спостерігати у всіх комерційних DDR4 DRAM після того, як атакуючі рядки записують лише 20 тисяч разів з достатньою частотою.

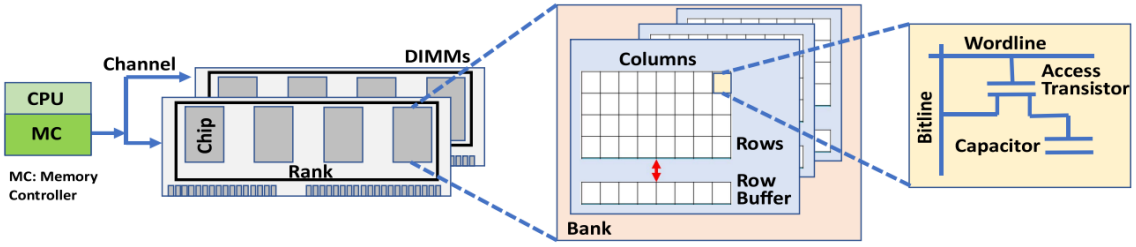

Рисунок 1 – Будова чіпу пам'яті DRAM

Існуючі варіанти атаки Rowhammer включають однолокаційну (атакується лише один ряд), односторонню (атакується рядок r і один із рядків r ± θ, де θ > 2), двосторонню (атакується рядок r і один із рядків r ± 2), і більш загальні N-сторонні атаки (де N означає кількість рядів агресора в блоці пам'яті) [1]. Обидві ці атаки в першу чергу призводять до зміни значення бітів у рядках, які безпосередньо прилягають до рядків-агресорів. Нещодавно дослідники з Google продемонстрували атаку "Half-double", коли вони змогли змінити біти за два ряди від основного ряду агресора (тобто Rowhammer на відстані) [7]. Ці нові атаки можуть обійти багато існуючих засобів захисту. Також проблема Rowhammer, ймовірно, погіршиться з часом, оскільки комірки DRAM в новіших чіпах розміщуються ще ближче одна до одної, при зменшенні технічного процесу.

Вибір DRAM моделі

Більшість засобів захисту Rowhammer передбачає просту й уніфіковану модель DRAM. Після запису кожен рядок створює наводки в сусідніх рядах. Ступінь наведення на ряд жертви залежить лише від його відстані від ряду агресора. Наприклад, ряд агресора *K* впливає однаково на двох сусідніх жертв – рядки *K±1*. При цьому *K* впливає *K±2* меншою мірою, ніж *K±1*, *K±3* ще меншою, і так далі. Швидкість, з якою збурення зменшується з відстанню, називається *коефіцієнт ослаблення* (КО) а *радіус дії* (РД) вказує на відстань між рядом агресора та його найдальшою жертвою. Модуль DRAM із радіусом дії 2 означає що *K* збурює тільки чотири ряди: *K±1* і *K±2*.

Через це засоби захисту Rowhammer, придатні для запису в контролер пам'яті, перш за все намагаються ідентифікувати ряди агресора. Їх мета полягає в тому, щоб ніколи не

дозволяти йому отримувати більше активацій рядка, ніж фіксоване порогове значення, яке називається пороговим значенням Rowhammer (TH_{RH}), в межах інтервалу оновлення (64 мс у DDR4 та 32 мс у DDR5). Як тільки кількість активацій досягне TH_{RH}, доступ блокується до наступного оновлення. Передбачається, що такий підхід нейтралізує все збурення, створене рядом агресора. Схеми вибірки рядків ж налаштовують поріг *p* так, щоб імовірність, що кількість запитів будь якого рядка дійде до TH_{RH} буде дуже низькою. Проблема ж, що не дуже зрозуміло, якою має бути ця дуже низька ймовірність, хоча в деяких нещодавніх роботах описується значення 10^{-15} за годину безперервних записів [2]. Попередні схеми для Row Sampling [2], [3] припускали, що при досягнення TH_{RH} контролер пам'яті також оновлює і рядків жертв. На жаль, внутрішня топологія рядків DRAM залишається комерційною таємницею постачальників DRAM. Тож контролер пам'яті нездатен ідентифікувати рядки жертв, на які впливає конкретний рядок агресора.

На практиці DRAM не поводиться так, як пропонує ця спрощена та уніфікована модель. Деякі рядки вимагають менше запитів, щоб викликати зміну бітів, ніж інші. Це відповідає необхідності мати індивідуальне TH_{RH} для кожного рядка, а не єдине постійне значення для всієї DRAM у системі. Крім того, збурення DRAM не є рівномірними: деякі комірки мають більше шансів бути зміненими, ніж інші, навіть якщо їх відстань до ряду агресорів однакова. Нарешті, радіус дії змінюється в залежності від ряду атакуючих; деякі ряди мають більший радіус дії, ніж інші. Саме тому справжня модель DRAM має налічувати всі ці змінні щоб аналізувати захист Rowhammer з математичним підходом та реальними результатами в існуючих системах. При цьому така система може бути спрощена і до глобальних значень, але потрібно обирати найвищий поріг *p* з усіх можливих для чіпу пам'яті і найнижчий TH_{RH} після тестувань всіх рядків.

Модель захисту

Для аналізу удосконаленої схеми вибірки рядків ми припускаємо найгіршу, але реалістичну модель загрози. Зловмисник знає модель DRAM і реалізацію схеми вибірки рядків, включаючи значення **p**. Зловмисник може активувати будь-який рядок у будь-якому порядку, але не порушуючи таймінги та коректність шини DRAM. DRAM налаштовано для роботи з нормальною частотою оновлення: контролер пам'яті видає 8192 команди для кожного вікна в 64 мс для DDR4 та 32 мс для DDR5 до оновлення. Такі припущення відповідають сценарію, за якого зловмисник може запустити довільний код на хост-системі, але не може змінити апаратне забезпечення, мікропрограму або налаштування BIOS/ UEFI. В попередній роботі [6] ми вже бачили результати такої атаки для трьох основних виробників пам'яті і знаємо що цей тип атаки може обходити захист від Rowhammer.

Одна з оригінальних статей [2] про Row-Sampling включає виведення формули для знаходження ймовірності невдалої Rowhammer атаки для даного p, TH_{RH}, а також термін під назвою "раунди" (скорочено *r*). Раунди вказують на час, необхідний зловмиснику для досягнення TH_{RH} активації рядків один за одним в незалежності від вікна оновлення.

$$P_{\text{невдачі}} = 1 - (1 - e^{-p \times TH_{RH}})^k$$

На жаль, ця формула дуже применшує значення порогу вибірки. Його визначення ґрунтується на припущенні, що кожен із раундів є незалежним і жодні збурення не передаються від одного раунду до наступного. Але на практиці, атака, яка активує половину рядків наприкінці раунду, а іншу половину одразу на початку наступного, має високий шанс уникнути вибірки. Тобто ці події потрібно розглядати як взаємозалежні.

Для обчислення порогу в PARA [3] використовують такі формули:

$$P(e_N) = P(e_{N-1}) + p\left(1 - \frac{1}{2}p\right)^{TH_{RH}} (1 - P(e_{N-TH_{RH}-1})) \quad (1)$$

$$P(e_N) = 0 \text{ при } N < TH_{RH} \quad (2)$$

де e_N випадок успішної атаки. Умова 2 тут тривіальна і означає що атака не сталась при кількості запитів меншій пороговій на контролері. Ймовірність поломки Rowhammer дорівнює нулю.

Як описано в розділі 3 – атака Rowhammer вимагає виконання двох умов:

- TH_{RH} активації рядків що не потрапили до вибірки
- відсутність автоматичного оновлення рядків жертви.

Так як ми не приймаємо за даність автооновлення рядків жертв, то можемо ще додати параметр $P(v_{TH})$ що показує таку ймовірність. Ця ймовірність пропорційна відношенню двох часових інтервалів: частина вікна оновлення, яка виходить за межі TH_{RH} ($t_{атак}$) та часового інтервалу вікна оновлення ($t_{онов}$).

$$\mathrm{P}(v_{TH}) = \frac{t_{онов} - t_{атак} \times TH_{RH}}{t_{онов}} \quad (3)$$

Оскільки дві ймовірності є незалежними то ймовірність невдачі в даному випадку є їх добутком.

Додатково формула ймовірності вдалої атаки Rowhammer має враховувати всю систему (тобто не лише окремий блок пам'яті) і тривалість атаки. Таким чином, ми вводимо два додаткові параметри: загальна кількість блоків *b* у всій системі, які можуть бути одночасно атаковані Rowhammer та *A*, загальна максимальна кількість активацій рядків у блоці протягом атаки. Отримавши ймовірність успішної атаки для окремого блоку, ми можемо масштабувати її до всієї системи, бо навіть один скомпрометований блок пам'яті вже є загрозою для нас. Таким чином, якщо успішну атаку на один блок позначити P_1 то для всієї системи формула стає $P_{системи} = 1 - (1 - P_1)^b$ А розписавши для незалежних вищеописаних ймовірностей отримаємо:

$$P_{системи} = 1 - \left(1 - P(e_A) \times P(v_{TH})\right)^b$$

Де $P(e_A)$ знаходиться з системи (1-2) а $\mathrm{P}(v_{TH})$ з формули (3).

При цьому радіус дії не фігурує в формулі безпосередньо, але після калькуляції саме це порогове значення ми використовуємо для всіх рядків в глобально зазначеному системою радіусі дії навколо атакуючого рядка.

Приклади використання

У цьому розділі представлено частоту відмов Rowhammer для двох порогів активації для різних апаратних конфігурацій. Базове значення p відповідає початковому стану чіпів «з коробки», а кореговане p відповідно розраховане в розділі 4. Конфігурація тестової машини відповідає одному серверу, подібному до тих, які можна знайти в хмарних центрах обробки даних або ж звичайних робочих умовах: подвійний сокет із 8 каналами DDR5 або ж DDR4 на сокет, 2 модулі DIMM на канал (DPC) і дворангові модулі DIMM. Конфігурація налічує 3 найбільші виробники DIMM та дві конфігурації DDR4 та DDR5 для кожного з них відповідно. Таблиця 1 узагальнює результати наших тестів.

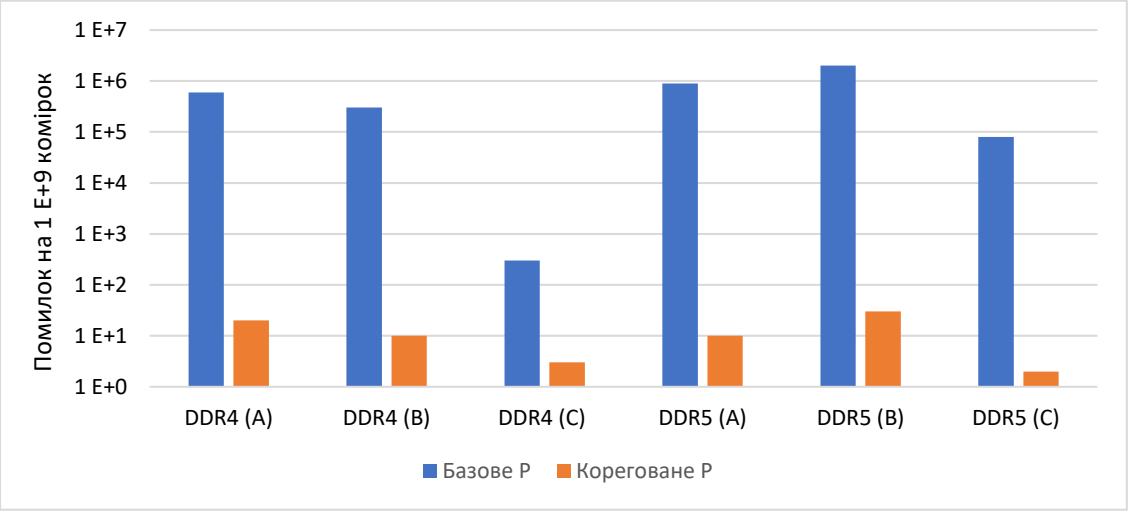

Рисунок 2 – Порівняння кількості RowHammer помилок для різних *p*.

Як бачимо з порівняння ймовірність відмови Rowhammer для різних порогів p для конфігурацій A та B. Для частот дискретизації ми використовували зворотні величини ступеня двійки (тобто 1 з 32, 1 з 64), оскільки ми очікуємо, що такі частоти дискретизації будуть легкими реалізувати в контролері пам'яті. Однак наша формула та код можуть працювати з будь-якими значеннями частоти дискретизації. Ми також використовували низькі TH_{RH} значення, що відповідають останнім тенденціям, які показують, що нові комірки DRAM, що вимагають менше звернень до пам’яті, поки біти не почнуть змінюватись [5]. Графік ілюструє, що при правильній регуляції порогу вибірки залежно від апаратних конфігурацій, можливо значно підвищити захищеність системи від атак типу RowHammer. Наприклад для DDR5 розробника C поріг був підвищений достатньо, щоб вдалось зменшити кількість спотворених бітів на 89.5%, при цьому не перевищивши вимоги по енергоефективності.

Обговорення

Наш аналіз припускає, що зловмисник не може контролювати або викликати відкладення оновлення комірок пам’яті. На сьогоднішній день ми не знаємо про програмні атаки, які дозволяють контролювати розклад оновлення контролерів пам’яті, але ми не можемо виключити цю можливість. Хоча відстрочка оновлення не впливає на формули, наведені в рівняннях (1)–(2), вона може зменшити ймовірність оновлення рядка жертви, як показано в рівнянні (3). У гіршому випадку зловмисник, який контролює розклад оновлення, може зменшити кількість команд оновлення, щоб збільшити успішність атаки. Наша модель загроз припускає, що атаку можна масово розпаралелювати на всі блоки пам’яті в системі (або в групі). Ми розуміємо що це екстримальний випадок, що може бути не реалістичним через обмеження на паралелізм, які накладають таймінги шини DDR. Наприклад, pTRR обмежує швидкість активації рядків для різних блоків у групі пам’яті або в межах рангу. Так само $t_{атак}$ це вікно часу, яке обмежує кількість активацій рядків для одного рангу до чотирьох. На жаль, включення цих часових обмежень у рівняння (3) не є тривіальним. Контролер пам'яті не має можливості визначати додаткові активації рядків, виконані додатковим оновленням сусідніх рядків за допомогою системи захисту. На жаль, це обмеження є фундаментальним, і його можна лише вирішити знаючи внутрішню топологію DRAM. Цей вектор атак не можна урегулювати шляхом зміни порогу вибірки. Іншим важливим фактором є здатність контролера пам’яті генерувати справжні випадкові числа при кожній активації рядка. На практиці ж використовують генератор псевдовипадкових чисел, який також в теорії можливо

скомпрометувати. Багато пристроїв DRAM мають вбудовані засоби захисту Rowhammer, такі як pTRR [5]. На жаль, засоби захисту як pTRR є запатентованими (тобто покладаються на захист через невідомість) і неповними [7]. Ці недоліки змушують постачальників процесорів, хмарних технологій і мобільних пристроїв розглядати можливість розгортання власних засобів захисту Rowhammer у контролерах пам'яті чи програмному забезпеченні. Ці засоби захисту частково збігаються з pTRR, що призводить до повторних оновлень і збільшень енерговитрат в експоненційних масштабах. А поки засоби захисту DRAM залишаються в таємниці, їх важко включити в нашу модель.

Висновки

Тож RowHammer є серйозним викликом для систем пам'яті і Row-Sampling є одним з основних факторів захисту на базі контролера. Вибірка рядків є привабливою технікою оскільки вона проста у застосуванні, ефективна та може забезпечити надійний захист за умови правильного налаштування. В даній статті ми провели ретельний аналіз того, як налаштувати реалізацію вибірки рядків. Також ми представили реалістичну модель DRAM, щоб зменшити неоднозначність і підвищити ясність припущень, зроблених під час математичного аналізу. Додатково ми описуємо більш реалістичну модель загроз, ніж ті, що використовувалися в попередній роботі. Ми розширили формули наведені в попередніх працях на мему, щоб отримати остаточну формулу, яка включає нашу модель загроз. Нарешті, ми представляємо розразунок правильних параметрів для захисту Rowhammer на основі вибірки рядків, що було протестовано на одному сервері з різними конфігураціями DRAM чіпів генерацій 4 та 5. В найкращому випадку вдалось зменшити кількість помилок спричинених атакою RowHammer на 89.5%.

Фінансування

Це дослідження не отримало конкретної фінансової підтримки.

Конкуруючі інтереси

Автори заявляють, що у них немає конкуруючих інтересів.

Список використаних джерел

1. Lin, J. and Garrett, M. "Handling Maximum Activation Count Limit and Target Row Refresh in DDR4 SDRAM," Patent No. US 2015/0200002 A1, 2015
2. Kim, Y., Daly, R., Kim, J., Fallin, C., Lee, J. H., Lee, D., Wilkerson, C., Lai, K. and Mutlu, O. "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," in ISCA, 2014. https://doi.org/10.1109/ISCA.2014.6853210.
3. Kim, D.-H., Nair, P. J. and Qureshi, M. K. "Architectural Support for Mitigating Row Hammering in DRAM Memories," CAL, 2015. https://doi.org/10.1109/LCA.2014.2332177.
4. Frigo, P., Vannacci, E., Hassan, H., V. van der Veen, O. Mutlu, C. Giuffrida, H. Bos, and K. Razavi, "TRRespass: Exploiting the Many Sides of Target Row Refresh," in S&P, 2020 https://doi.org/10.1007/978-3-031-64171-8_24.
5. Kaczmarski, M. "Thoughts on Intel Xeon E5-2600 v2 Product Performance Optimisation," 2014
6. Мазурок, В., & Луценко, В. (2024). Аналітичний огляд та аналіз трендів вразливості RowHummer для різних виробників DRAM. *Social Development and Security*, *14*(3), 238-244. https://doi.org/10.33445/sds.2024.14.3.16.
7. Kim, M., Choi, J., Kim, H. and Lee, H.-J. "An Effective DRAM Address Remapping for Mitigating Rowhammer Errors," in TC, 2019. https://doi.org/10.1109/TC.2019.2907248.

References

1. Lin, J. and Garrett, M. “Handling Maximum Activation Count Limit and Target Row Refresh in DDR4 SDRAM,” Patent No. US 2015/0200002 A1, 2015
2. Kim, Y., Daly, R., Kim, J., Fallin, C., Lee, J. H., Lee, D., Wilkerson, C., Lai, K. and Mutlu, O. “Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors,” in ISCA, 2014. https://doi.org/10.1109/ISCA.2014.6853210.
3. Kim, D.-H., Nair, P. J. and Qureshi, M. K. “Architectural Support for Mitigating Row Hammering in DRAM Memories,” CAL, 2015. https://doi.org/10.1109/LCA.2014.2332177.
4. Frigo, P., Vannacci, E., Hassan, H., V. van der Veen, O. Mutlu, C. Giuffrida, H. Bos, and K. Razavi, “TRRespass: Exploiting the Many Sides of Target Row Refresh,” in S&P, 2020 https://doi.org/10.1007/978-3-031-64171-8_24.
5. Kaczmarski, M. “Thoughts on Intel Xeon E5-2600 v2 Product Performance Optimisation,” 2014
6. Mazurok, V., & Lutsenko, V. (2024). An analytical overview and trend analysis of RowHummer vulnerabilities for various DRAM vendors. *Social Development and Security*, *14*(3), 238-244. https://doi.org/10.33445/sds.2024.14.3.16
7. Kim, M., Choi, J., Kim, H. and Lee, H.-J. “An Effective DRAM Address Remapping for Mitigating Rowhammer Errors,” in TC, 2019. https://doi.org/10.1109/TC.2019.2907248.