A Reliability-Physics-Based Approach for Data Tampering Detection in Commercial 3D-NAND Flash Memory

Yuhan Wang*, Jian Huang*§, Ruibin Zhou*, Yao Liu*, Haotian Ye*, Xianping Liu*
* School of Microelectronics Science and Technology, Sun Yat-sen University, Zhuhai, 510275, China
§ Corresponding author, E-mail: huangj573@mail.sysu.edu.cn

***Abstract*—Solid-state drives (SSDs) and flash-based storage devices are vulnerable to covert data tampering, presenting a major obstacle to their secure adoption in critical sectors such as government, finance, and critical infrastructure. Currently, research on detecting data tampering in solid-state drives (SSDs) remains inadequate and some existing system-level measures can even be bypassed. To address this challenge, we propose a novel tamper-detection framework based on intrinsic physical properties of commercial 3D NAND flash memory. Central to our approach is the use of Rewrite Detection Bits (RDBs), which intentionally introduces controlled errors into stored data. The framework further integrates device-level reliability metrics alongside RDBs to improve robustness. In contrast to conventional solutions, our method requires no hardware modifications and functions entirely through standard read/write commands. Tampering attempts are identified by analyzing error bit locations and error rate distributions. Experimental results indicate that the framework can provide high security with negligible resource consumption: allocating only 6 RDBs per block (≈0.000005% of storage capacity) reduces the detection error rate to below below 1.4761×10^{-7}, while remaining fully compatible with existing flash memory architectures.**

***Index Terms*—tamper-detection, flash memory, rewrite detection bit, bit error rate, lightweight**

I. INTRODUCTION

Solid-state drives (SSDs) and USB flash drives have become essential tools for data transfer, driven by their steadily increasing storage capacities and physical portability. These devices are particularly valuable in scenarios where network connectivity is unavailable, unreliable, or deliberately restricted, such as within secured internal networks, air-gapped systems, or remote field operations. However, despite their widespread utility, they also introduce considerable security risks when used to transport sensitive information. The sensitive data generated through extensive research, operational investment, and specialized infrastructure often represents high-value assets that are highly attractive targets for cyberattacks [1], [2].

One of the most concerning threats is stealthy data tampering, in which adversaries secretly alter legitimate data without the user's knowledge [2]–[4]. Such attacks compromise data integrity and may lead to misinterpretation, operational disruption, or even the deliberate spread of misinformation. Commercial 3D NAND flash memory chips, which serve as the foundation of most modern SSDs and high-capacity USB drives, lack inherent capabilities to detect or record such malicious modifications, leaving stored data vulnerable to forgery and stealthy changes. As a result, the inability of current storage devices to detect and document tampering attempts constitutes a major barrier to their secure adoption in sensitive domains such as government, healthcare, finance, and critical infrastructure.

While anomaly detection techniques have been extensively studied in domains like network transmission and sensor data monitoring [5]–[8], their application to detecting data tampering at the SSD device level remains a relatively under-explored area. Storage chips with built-in encryption capabilities have not seen widespread market adoption. Additionally, a significant challenge stems from the inherent limitations of SSD controller architectures. Specifically, even when controller-based detection mechanisms (e.g., tamper-proof logging or version counters [9]) are employed, adversaries with physical access can potentially bypass the Flash Translation Layer (FTL) and associated security protocols to directly access and manipulate data within NAND flash chips without triggering any alerts. This phenomenon exposes a potential vulnerability: current storage systems lack effective mechanisms to detect such device-level tampering activities, enabling malicious modifications to occur covertly and severely compromising the integrity and trustworthiness of stored data.

To address this challenge, we propose a novel tamper-detection approach that leverages the physical characteristics of commercial 3D NAND flash memory. Unlike conventional methods that rely on external security modules or complex controller logic, our approach makes use of the inherent properties of flash memory cells to identify secretly modifications. The core of this method is the Rewrite Detection Bit (RDB), a lightweight yet effective technique that embeds intentional errors into stored data. By exploiting the natural randomness of Program Disturb (PD) and Data Retention (DR) errors in 3D NAND flash, RDBs remain indistinguishable from normal bit errors, making them resistant to adversary detection. We further combine RDBs with device-level reliability effects, such as threshold voltage drift and error-rate variations, to build a robust tamper-detection framework. Importantly, this method employs only standard read/write operations and requires no hardware modifications, ensuring full compatibility with existing flash-based storage systems. Detection of tampering

978-3-9826741-1-7/DATE26/© 2026 EDAA

operations can be efficiently performed via streamlined test and analysis of raw bit error rates. Experimental results validate that our approach accurately identifies effective stealthy data modifications, providing a practical and deployable solution for securing data in modern flash devices.

II. BACKGROUND

In this section, we will discuss the structure and behavior of NAND Flash memory cells and reliability characteristics.

A. Organization and Operations of 3D-NAND Flash Memory

a) Structure of a 3D-NAND Flash Memory Cell: Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) with a charge trap layer form the core structure of NAND flash memory cells [10] (shown in Fig. 1a). The charge trap layer is sandwiched between tunnel oxide and blocking oxide layers, and the charge stored in the charge trap layer determines the cell's threshold voltage. When a voltage is applied to the control gate, electrons can either enter or exit the floating gate through the tunnel oxide, depending on the polarity of the voltage. The charge then resides in the charge trap layer between the oxide layers. This structure enables data retention without a power supply, as the charge in the charge trap layer does not easily leak through the insulating oxide layers, thus ensuring non-volatile storage.

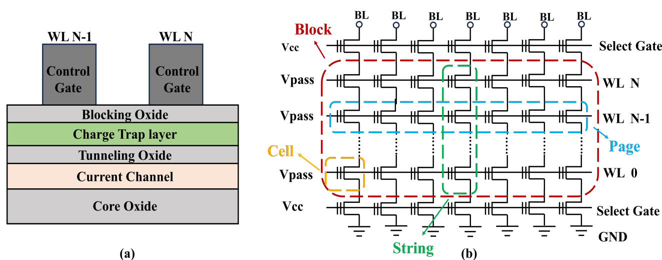

Fig. 1. (a) The schematic of 3D Charge-Trap NAND flash memory cell structures. (b) The planar structure of the 3D Charge-Trap NAND flash memory array.

b) 3D-NAND Array Architecture and Operation: 3D NAND flash memory organizes multiple memory cells in series, with dozens to hundreds of cells connected through a channel and sharing a bit line (BL) [11], [12], as shown in Fig. 1b. These cells are further grouped through their gates to form a word line (WL), which serves as the basis for data storage and manipulation [13], [14]. Programming occurs on a per-WL basis, so in multi-level storage, multiple pages are programmed simultaneously. The read operation is performed on a per-page basis, where cells are accessed using different read voltages to distinguish between 0 and 1. Erase is performed on a block basis. Erase operations set all the cells to a data value of 1, and it must be performed before any new data can be programmed.

c) Threshold Voltage and Gray Code: Threshold voltage (Vth) is a key parameter for the performance of a NAND flash memory cell, which directly affects the read, program and erase operations of the memory cell. High-density flash memory often adopts gray code to encode multiple bits in one cell to represent different threshold voltage states [14]–[16]. Figure 2 shows a fundamental mathematical model of voltage distribution based on the gray code. Gray code maps 3-bit values to 8 voltage level states differently. Precise control of these states is critical to ensure that data is read and written correctly. Once all three pages have been programmed, the pages can be read by applying corresponding read voltages. In the TLC NAND, the LSB page requires two single read voltages, the CSB page can be read by applying three read voltages, and retrieving information from the MSB page requires two read voltages [17]. If any of the bit's threshold voltages cross the read voltage and spill over into adjacent states, read errors will occur.

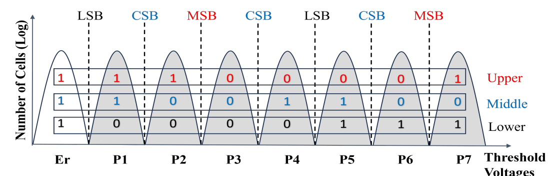

Fig. 2. Schematic of TLC threshold voltage distribution with the gray code.

B. Traditional Reliability Issues of Flash Memory

a) Data Retention (DR): Data retention is the ability of a flash memory cell to retain charge. It is a crucial metric for evaluating the data storage capability of non-volatile memory. The charge in flash memory cells may leak through the tunnel oxide layer via mechanisms such as Trap-Assisted Tunneling (TAT) and Stress-Induced Leakage Current (SILC) [17], [18]. As the feature size of NAND flash devices continues to shrink, the number of electrons stored in each memory cell decreases. Even a small gain or loss of electrons can significantly change the threshold voltage of the memory cell. The State with a higher initial Vth, exhibit a downward shift as retention time increases. Since the read reference voltage used to distinguish between logic '1' and '0' remain fixed, this downward shift may cause Vth to cross the read reference voltage, resulting in retention-induced errors. In addition, as flash memory undergoes program/erase cycles, the error rate tends to increase. This is due to the fact that erasure of flash memory cells requires the addition of high erase voltages, which can form defects in the oxide layer. These defects can cause accidental charge capture or accelerated charge leakage in the charge storage layer, leading to shifts in the cell's threshold voltage [19].

b) Program Disturb (PD): Due to the array configuration, the Vth of unselected memory cells can be unintended modification during the programming process in flash memory. When a flash memory cell is being programmed, the high voltage applied to the selected cell can induce an undesired charge in unselected cells (see Figure 3), especially in triple-level cells (TLC), or quad-level cells (QLC). Program disturb leads to unavoidable initial errors in the data programmed into the flash memory. The erroneous locations caused by program interference exhibit significant randomness originated from the quantum tunneling process, inherent physical parameter variations and P/E cycles across different cells [20].

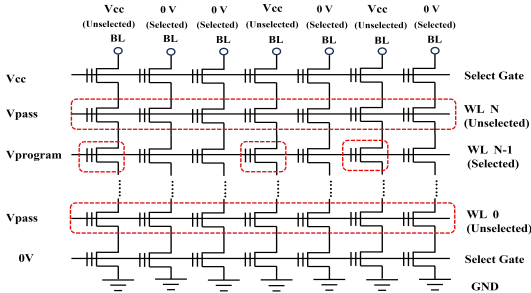

Fig. 3. The voltage configuration of the block during programming. The unselected bits (cycle in red) will be disturbed by the applied high voltage.

III. METHODOLOGY

In this section, based on the reliability mode described above, we propose two metrics that exploit device-level reliability characteristics to detect write and read operations in 3D NAND flash memory. To ensure the integrity of data received by the destination, we then demonstrate a scheme to detect data tampering access during transportation.

A. Principle of RDB-Based Tampering Detection

The main idea of this technique is to embed a small number of intentionally error bits, referred to as Rewrite Detection Bits (RDBs), into the user data. These bits are used to detect data tampering by checking whether they have been flipped. To prevent adversaries from identifying the RDBs, the method leverages the inherent randomness of PD and DR errors in 3D NAND flash memory. Since most of the bit error positions caused by PD and DR are unpredictable, it extremely difficult for adversaries to distinguish RDBs from naturally occurring errors. Additionally, Vth verification errors caused by Random Telegraph Noise (RTN) during programming verify stage in ISPP can result in random under-programming or over-programming, which introduces further randomness into the error pattern. Figure 4 illustrates the error bit distributions

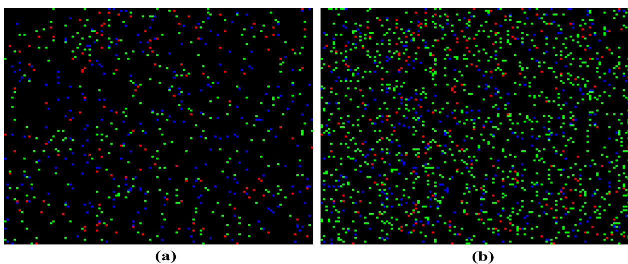

Fig. 4. The 2-dimentional error bitmap image of (a) after programming and (b) after 85°C 3 hours bake. The error bit addresses are from the first 2kB of 36 pages. The top left corner of the image is the bit 0 and the bottom right corner is the 16384 bit. The address number increase from left to right and top to bottom. The green, blue and red dots represent the errors of 1st programming, 2nd programming and the overlay of two programming.

across 36 pages within the same block for two separate programming cycles. In Figure 4a, the error pattern reflects PD-induced errors immediately after programming. Figure 4b shows the error distribution after a retention period. In both figures, blue marks the error locations from the first programming, green indicates those from the second programming, and red highlights overlapping error positions. It can be observed that a significant number of error bits differ between the two instances, confirming the spatial randomness of error occurrences. This randomness provides natural camouflage for the RDBs, significantly increasing the difficulty for adversaries to identify or manipulate them.

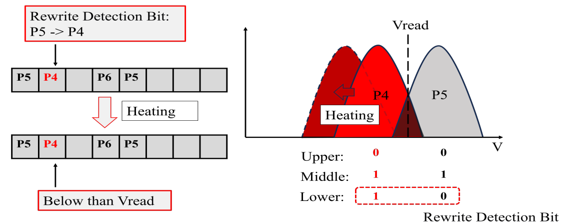

Fig. 5. Rewrite Detection Bit. We intentionally change the lower page of a P5 state cell from 0 to 1. Heating will not change this bit from 0 back to 1.

B. RDBs-Based Write Detection Scheme

Below are the details of this technique. First, we briefly introduce the Error Correction Code (ECC) commonly used in NAND Flash memory. Since NAND Flash programming inevitably introduces bit errors, and these errors tend to increase over time due to data retention effects, ECC algorithms are employed in Flash or SSD controllers to ensure data integrity. In programming stage, the ECC engine generates a set of check bits from a fixed-length user data block, and both the user data and its corresponding check bits are written into the flash memory. During data retrieval, the controller reads both the user data and check bits and uses the ECC algorithm to reconstruct the correct original data. In the RDB-based approach, the data sender intentionally introduces a few bit errors into the data as rewrite detection bits (RDBs) into the post-ECC data by flipping bit value at selected addresses and write to the memory cells. These intentionally embedded errors are effectively concealed by the numerous inherent random errors present in flash memory. Here is an example of a specific implementation. The sender and receiver agree on a set of bit addresses according to a specific rule. These addresses physically retain the data corresponding to the threshold voltage of the P5 state. In other words, for a given 3 data bits in a memory cell, the three pages (Page i, Page i+1, and Page i+2) should correspond to bit values ”1”, ”1”, and ”0,” respectively. Sender can intentionally write these bits as ”0”, ”1”, and ”0”. According to the Gray code used in our chip, the threshold voltage of this physical unit will shift to the Vth range of the P4 state, which is adjacent to P5. This introduces an error bit on Page i. This error bit can effectively conceal in the errors caused by PD and DR. Due to the charge leakage during data retention, the threshold voltage will drift toward lower values. Since the P4 state is lower than the P5 state, this ensures that these error bits remain consistently below the distinguishing voltage between the P4 and P5 states and remain as error bits, as shown in Fig. 5. Here, P5 is provided as an illustrative example. In practical applications, the specific state or combination of states designated as the RDB can be

determined through mutual agreement between the sender and the receiver. Algorithm 1 outlines the specific steps.

Algorithm 1: RDBs Encoding and Decoding Algorithm

Encoding:
1: Encode user data and generate ECC codeword
2: Choose a set of bit addresses in user data area
3: Flip the bit in the user data to adjust the memory cell's Vth to its lower neighboring state for each chosen address
4: Program the modified user data with the ECC codeword directly to NAND Flash

Decoding:
1: Read the user data through ECC to get error-free user data
2: Read the user data bypass ECC directly from NAND Flash
3: Compare the targeted bits with the error-free data to verify whether the RDBs remain in the error state

C. Security Analysis of RDB-Based Tampering Detection

Here we explain why this method enables tamper detection. It leverages two fundamental hardware properties of 3D-NAND flash memory chips.

First, programming operations can only shift a cell's threshold voltage from a low to a high level. Therefore, to modify data on a specific bit line, the flash memory must follow the "erase-before-write" hardware rule. Direct programming would cause massive data errors, exceeding the correction capability of Error Correction Codes (ECC). Thus, the erase operation required for rewriting inevitably removes all previous error characteristics, including the intentionally embedded RDB errors.

Second, due to hardware constraints in 3D-NAND, erase operations must be performed on entire blocks which contain multiple pages (e.g., a 96-layer 3D-NAND TLC block contains 1152 pages), while write operations are carried out per page. This allows RDB and other error features to be distributed across the whole block, making it much harder for an adversary to interpret or tamper with the data.

As a result, an adversary must read the entire block before rewriting any data. There are two ways to read data from the flash: with or without using the ECC module. Reading with ECC returns error-free, corrected data, while reading without ECC returns "raw" data that contains errors caused by PD, DR, and the injected RDB errors.

We first consider the case where error-free data is read using the ECC module, which is the standard reading method for devices like SSDs and USB drives. Fig. 6 illustrates the entire process of ECC decoding and encoding. As long as the number of errors does not exceed the ECC correction capability, the adversary obtains fully corrected data, including the original RDB errors. If an adversary obtains data through the ECC module and alters some or all of it, regardless of whether the data is rewritten via the ECC module, the RDB bits will no longer function as error bits. The threshold voltages at the RDB locations are likely to be reprogrammed. It should be noted that reliability issues such as PD and DR can also introduce errors. Therefore, multiple RDBs can be placed in one block, and a majority voting mechanism can be used for verification. The effectiveness of this approach will be discussed in the Evaluation section.

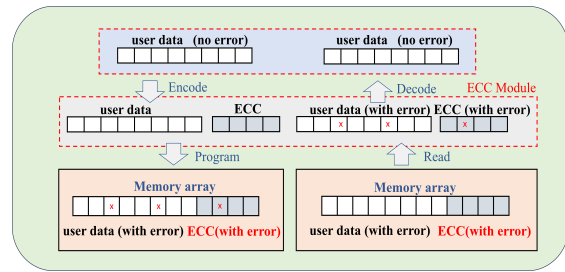

Fig. 6. ECC encode and decode process. For the encoding, the ECC module use user data to generate ECC codeword. The user data is combined with ECC codeword and programmed into flash memory. For the decoding process, the ECC module correct the errors of user data. All the data store in the TLC flash memory contains errors. The red cross indicates the error.

If an adversary bypasses the ECC module through hardware means to directly access the chip, raw data can be obtained from the NAND. In this scenario, the retrieved data preserves the original RDB information. When writing back modified data, the adversary may adopt one of the following two strategies:

- The adversary modifies the data and subsequently reprograms it through the ECC module. This entails recalculating the ECC check bits based on the altered user data and writing both into the NAND. In this case, the RDB values will present as valid and error-free upon subsequent decoding. This is because the data decoded by the receiver using the updated ECC check bits matches the data written by the adversary.
- The adversary makes minor modifications to the raw data that was read out and rewrites it combined with the original ECC check bits. This approach constitutes an ineffective attack. Although the original error characteristics of the RDB are preserved, the unmodified ECC check bits cause the ECC algorithm to identify the intentionally altered bits as errors and correct them back to their original state. Consequently, the data eventually decoded by the receiver remains identical to the sender's original data, and the RDB pattern is also retained. If the receiver wishes to further verify the unauthorized access, the block bit error rate (BER) can be used as a metric for assessment.

Although adversaries can bypass the ECC module and directly overwrite the original erroneous data back into flash memory, the risk of the receiver utilizing false data remains absent as long as the RDB scheme is present. However, if the adversary modifies the data, recalculates the corresponding ECC check bits based on the new content, and deliberately reintroduces bit errors at the exact locations where errors originally occurred prior to writing the updated codeword, the RDB-oriented integrity verification could potentially be circumvented. However, due to the PD and DR errors introduced during the sender's preprocessing stage, the adversary' reprogramming process, error injection and any associated thermal manipulation will cause overlapping of error-prone regions. This leads to a measurable increase in steady-state block-level BER, as depicted in Fig. 7. The increase of BER provides

a reliable test metric for detecting stealthy reprogramming. The sender and receiver can perform pre-characterization of the reliability metrics specific to the flash chip model in use, thereby establishing an acceptable range for the error rate. An excessively high error rate indicates that the data has been rewritten. In practice, this type of attack is highly risky because the error correction capability of NAND flash memory is quite limited. Typical soft-decision ECC mechanisms can generally correct only around 10,000 errors per block [21], most of which are already consumed by inherent reliability-related errors. As a result, any attempt at tampering by an adversary is highly likely to cause ECC decoding failure.

In summary, by verifying whether the RDB pattern matches the expected values and auxiliary detection based on block-level BER, we can ensure the integrity and correctness of the data retrieved from 3D-NAND flash memory after ECC correction in most cases. As long as the RDB remains normal, the data can be considered unaltered.

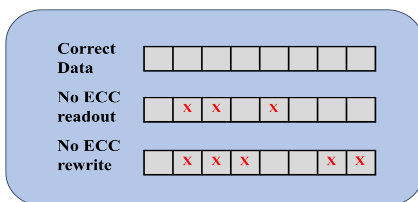

Fig. 7. The schematic of error accumulation if no ECC. The red cross indicates the error.

IV. EVALUATION

To validate our detect scheme, we conducted tests based on four types of commercial 3D-NAND chips. Detailed parameters of tested flash chips which come from three vendors are listed in Table 1. We used a commercial flash memory tester to perform erasure, programming, and data reading operations, followed by the test of the BER.

A. Detection of Tampering through ECC

The first scenario to validate is when the adversary obtains error-free data through ECC and then reprograms the data. Since the RDBs are generated after ECC, they will be corrected by ECC during error correction process. These bits will be corrected to the correct value and reprogrammed to the memory by adversary. In Fig. 8, we randomly selected a unit that should have been in the P5 state and programmed it to P4, as indicated by the yellow box. We overlay the error location from sender's initial programming and adversary's reprogramming. The error bits and RDB in sender's initial programming include the green and red points, while the error bits of adversary's reprogramming include the blue and red points. The results in Fig. 8(a) show that the RDB disappears after the reprogramming. Because DR will introduce more errors in flash memory, we further validate this scenario by introducing DR-related errors via high-temperature baking. The corresponding results are shown in Fig. 8(b). We randomly selected 8 positions for simulating replay experiment, and all produced consistent results.

TABLE I
CHIP TYPE

Type	Vendor	Chip Type	Capacity
1	Vendor 1	64Layers TLC	64Gb
2	Vendor 2	96Layers TLC	256Gb
3	Vendor 2	112Layers TLC	256Gb
4	Vendor 3	64Layers TLC	256Gb

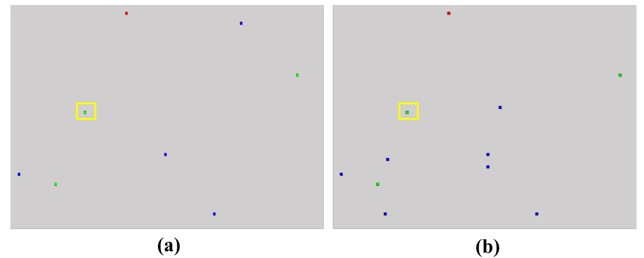

Fig. 8. The 2-dimentional error bitmap images of attacker's ECC-based reprogramming (a) immediately after original programming and (b) after 85°C 3 hours bake overlay with the sender's initial programming. The green dots represent the errors of original programming with an RDB (in yellow rectangle). The blue dots represent the errors of adversary's 2nd programming which is reprogrammed with the data after ECC. The red dots represent the overlap errors.

To assess the stability of the single RDB existence scenario, we injected 400 RDBs with seven threshold voltage states into 20 blocks across four types chips. The stability of the algorithm was evaluated based on the detection rate. By monitoring the RDBs both in the initial state and a long period of storage, it was observed that the detection rate consistently exceeded 99.5%, as illustrated in Fig. 9. Notably, the detection rate exhibited further improvement after prolonged storage. These results demonstrate that the RDB scheme maintains high stability and delivers robust practical performance.

B. Failure Rate Assessment

However, as mentioned earlier, flash memory errors exhibit a certain level of randomness due to PD. Meanwhile, after the adversary applies heating, even if the RDB changes to the correct bit by ECC after reprogramming, the threshold voltage may still decrease due to the heating, resulting in a chance that the bit may flip back to an error. To ensure the security of the scheme, we recommend selecting the number of RDBs based on the BER and security requirements and applying the majority voting principle. Assuming the worst-

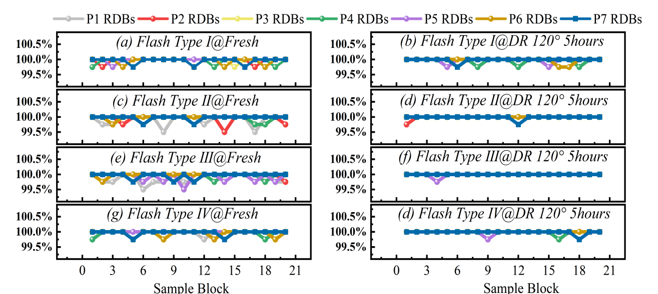

Fig. 9. Retention Status of four hundred RDBs after initial programming and after 120°C 5 hours bake in four types of flash chip.

case scenario where all errors occur randomly, we can estimate the false positive probability. If we select n RDBs and use the criterion that more than half of them remain in the error state to determine data security, then the false positive probability can be estimated as:

$$\text{probability} = \sum_{x>n/2}^{n} C_n^x \cdot (\text{BER})^x \cdot (1 - \text{BER})^{n-x}, \quad x \in \mathbb{Z} \tag{1}$$

where C_n^x is the number of combinations of x elements out of n elements.

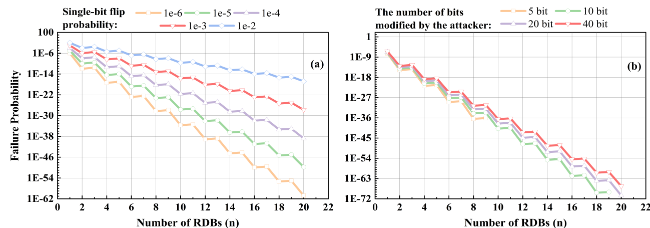

Fig. 10. Relationship between the false positive probability and the number of RDBs: (a) under different single-bit flip probabilities, and (b) under different numbers of modified bits.

To investigate whether random bit errors caused by PD and DR affect the majority voting principle, we measured the false alarm probability at different flip probabilities. Fig. 10(a) shows how the false positive probability changes as the number of RDBs increases under different flip probabilities. When the probability of a single bit flip is held constant, the failure probability drops exponentially, as more RDBs are added. Quantitative analysis using the proposed probabilistic model indicates that for a true BER of 0.01, when the number of RDBs used increased from 4 to 6, the authentication failure probability decreased significantly from 3.97×10^{-6} to 1.4761×10^{-7}. This underscores the critical role of RDB quantity in enhancing verification robustness. Fig. 10(b) illustrates the probability that an adversary will precisely hit an RDB, causing it to fail, when multiple RDBs are placed within a block. Even when adversaries interfere by randomly flipping bits in the RDBs, which could lead to incorrect outcomes, using more RDBs still substantially mitigate the risk of failed verification. In fact, adversary tampering rendering the RDB invalid only occurs in scenarios where the ECC check bits remain unmodified. Attacks in such scenarios like data remaining unchanged after ECC are ineffective. Nevertheless, we still consider this possibility to alert recipients that the data has been accessed. These results highlight that increasing the number of RDBs can significantly strengthen the system's defense mechanism against both errors and attacks.

C. Detection of Tampering without ECC

Another scenario arises when the adversary reads an entire page of data without using the ECC module and subsequently reprograms the page into the storage array, either without regenerating the ECC check bits or by deliberately injecting errors. Both cases retain all the errors from the previous programming and the errors introduced by heating, while also adding new programming errors from the reprogramming and additional errors from the baking process, resulting in a significant increase in the bit error rate. As illustrated in Fig. 11, the block BER exhibits a substantial increase following the implementation of tampering, reaching a level at least 200% higher than the baseline error rate. This pronounced elevation in BER serves as a key reliability-based indicator for detecting unauthorized data modifications.

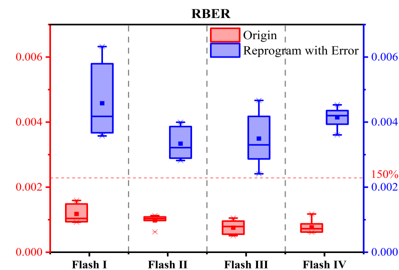

Fig. 11. Block BER comparison before and after reprogramming with inherited error.

Based on the above test results, we set the reference standard as follows:

- Block BER exceeds the safe range of 150% of the normal BER.
- At least more than half of the RDBs have undergone a reversal.

By leveraging the reference standard along with the pre-transmitted measured RBER and RDB rules from the sender, accurate determination can be achieved. The increase in RBER attributable to retention time can be derived analytically through established formulas. However, any excess RBER beyond this predicted level may be identified as indicative of data tampering operations. It should be noted that the acceptable range for block BER may need to be adjusted based on actual measurement data, as performance fluctuations can occur during different flash memory operation.

V. CONCLUSION

In this paper, we presented a novel detection approach that exploits the inherent reliability-related physical characteristics of 3D NAND flash memory to detect covert data tampering. The core of the proposed approach lies in combining RDBs with the effects of DR and PD, enabling reliable detection of unauthorized data alterations. Experimental results demonstrate that these metrics exhibit measurable responses under attack conditions and achieves a high-level detection rate exceeding 99.9999% with employing 6 RDBs. Furthermore, combining block-level BER with RDB deployment can identify diverse covert attack behaviors, further enhancing system robustness.

REFERENCES

[1] L. Mashayekhi and M. E. Kuhl, "Simulation of Low Earth Orbit Satellite Communication Data for Cyber Attack Detection," presented at the 2024 Winter Simulation Conference (WSC), 2024.

[2] F. Zhang and Y. Ding, ”Research on Anti-tampering Simulation Algorithm of Block Chain-based Supply Chain Financial Big Data,” presented at the 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 2021.

[3] S. Simonthomas and R. Subramanian, ”Detection and Prevention of Cyber-Attacks in Cyber-Physical Systems based on Nature Inspired Algorithm,” presented at the 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), 2023.

[4] Xingquan Fu, Mengfei Niu and Guanghui Wen, ”Detection of Stealthy Cyber-attack in Distributed DC Microgrids Based on LSTM Neural Network,” presented at the 2021 International Conference on Neuromorphic Computing (ICNC), 2021.

[5] Aditya Kumar Pathak, Saguna Saguna, Karan Mitra and Christer Åhlund, ”Anomaly Detection using Machine Learning to Discover Sensor Tampering in IoT Systems,” presented at the ICC 2021 - IEEE International Conference on Communications, 2021.

[6] P. D. Rosero-Montalvo, Z. István, P. Tözün and W. Hernandez, ”Hybrid Anomaly Detection Model on Trusted IoT Devices,” IEEE Internet of Things Journal, 2023.

[7] M. S. H. Onim and H. Thapliyal, ”Detection of Physiological Data Tampering Attacks with Quantum Machine Learning,” presented at the 2025 IEEE International Symposium on Circuits and Systems (ISCAS), 2025.

[8] Sushree Padhan and Ashok Kumar Turuk, ”POSTER: Defense against False Data Injection Attack in a Cyber-Physical System,” presented at the 19th ACM Asia Conference on Computer and Communications Security, 2024.

[9] Jinwoo Ahn, Junghee Lee, Yungwoo Ko, Donghyun Min, Jiyun Park, Sungyong Park and Youngjae Kim, ”DISKSHIELD: A Data Tamper-Resistant Storage for Intel SGX,” presented at the 15th ACM Asia Conference on Computer and Communications Security, 2020.

[10] X. Zhao et al., ”Error Bits Recovering in 3D NAND Flash Memory: A Novel State-Shift Re-Program (SRP) Scheme,” presented at the 2023 IEEE International Con-ference on Integrated Circuits, Technologies and Appli-cations (ICTA), 2023.

[11] Y. Xi et al., ”Bits Mapping in Triple-level-cell (TLC) Charge-trap (CT) 3D NAND Flash Memory and its Applications to IoT Security,” presented at the 2021 IEEE International Conference on Integrated Circuits, Tech-nologies and Applications (ICTA), 2021.

[12] X. Zou, L. Jin, D. Jiang, Y. Zhang, G. Chen, and Z. Huo, ”Investigation of Cycling-Induced Dummy Cell Disturbance in 3D NAND Flash Memory,” IEEE Electron Device Letters, vol. 39, no. 2, pp. 188-191, 2018, doi: 10.1109/led.2017.2785843.

[13] W. Liu, F. Wu, S. Meng, X. Chen and C. Xie, ”Error Generation for 3D NAND Flash Memory,” presented at the 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, doi: 10.23919/DATE54114.2022.9774514.

[14] H.-W. Hu et al., ”A 512Gb In-Memory-Computing 3D-NAND Flash Supporting Similar-Vector-Matching Oper-ations on Edge-AI Devices,” presented at the 2022 IEEE International Solid- State Circuits Conference (ISSCC), 2022.

[15] B. Chen, Y. Kong, Z. Sun, X. Fang, X. Zhan, and J. Chen, ”High-to-Low Flippling (HLF) Coding Strategy in Triple-levell-cell (TLC) 3D NAND Flash Memory to Con-struct Reliable Image Storages,” presented at the 2022 6th IEEE Electron Devices Technology & Manufactur-ing Conference (EDTM), 2022.

[16] D. Wei, Z. Piao, H. Feng, L. Qiao, C. Hu, and X. Peng, ”TCSE: A Target Cell States Elimination Coding Strate-gy for Highly Reliable Data Storage Based on 3D nand Flash Memory,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 12, pp. 5299-5312, 2022, doi: 10.1109/tcad.2022.3155380.

[17] H. -T. Lue et al., ”Radically extending the cycling endurance of Flash memory (to ¿ 100M Cycles) by using built-in thermal annealing to self-heal the stress-induced damage,” presented at the 2012 International Electron Devices Meeting (IEDM), 2012.

[18] S. Liu and X. Zou, ”QLC NAND study and enhanced Gray coding methods for sixteen-level-based program algorithms,” Microelectronics Journal, vol. 66, pp. 58-66, 2017, doi: 10.1016/j.mejo.2017.05.019.

[19] J. Y. Lim et al., ”Analysis of Intrinsic Charge Loss Mechanisms for Nanoscale nand Flash Memory,” IEEE Transactions on Device and Materials Reliability, vol. 15, no. 3, 2015.

[20] B. Lou, Q. Liu, Z. Zeng, Y. Zhou and J. Zhong, ”A Review of Program disturb of 3D NAND Flash Memory,” presented at the 2023 24th International Conference on Electronic Packaging Technology (ICEPT), 2023.

[21] Toshiki Nakamura, Shun Suzuki and Ken Takeuchi, ”Data Pattern & Memory Variation Aware Fine-Grained ECC Optimized by Neural Network for 3D-TLC NAND Flash Memories with 2.0x Data-Retention Time Extension and 30% Parity Overhead Reduction,” presented at the 2019 IEEE 11th International Memory Workshop (IMW), 2019.