IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 29, 2020, accepted January 13, 2021, date of publication January 18, 2021, date of current version January 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052353

Forensics and Anti-Forensics of a NAND Flash Memory: From a Copy-Back Program Perspective

NA YOUNG AHN[1] AND DONG HOON LEE[1,2], (Member, IEEE)
[1]Institute of Cyber Security & Privacy, Korea University, Seoul 02841, South Korea
[2]Graduate School of Cybersecurity, Korea University, Seoul 02841, South Korea

Corresponding author: Dong Hoon Lee (donghlee@korea.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government Ministry of Science and ICT (MSIT) under Grant NRF-2017R1A2B3009643.

ABSTRACT This paper proposes a safe copy-back program operation in a NAND flash memory, which is targeting digital forensics for a variety of reasons. Due to the background management operation of the NAND flash memory, the original data is highly likely to remain without truly being deleted. We have carefully investigated the possibility of data exposure due to a copy-back program operation, among, frequently used management operations as such data exposure increases the possibility of privacy invasion. We propose a safe copy-back program operation that lowers the possibility of privacy invasion. And we additionally introduce various techniques for solving the reliability problem of adjacent cells caused by the proposed copy-back program operation. For example, when deleting the original data in a copy-back program operation, overwriting is performed to minimize program disturbance. Also, after acquiring the victim cell information of the adjacent cell before proceeding with overwriting, program prohibition is determined on each page buffer based on the victim cell information. Our research results are meaningful for forensics and anti-forensics issues to be raised regarding NAND flash memories. We look forward to the development of NAND flash memories that guarantee privacy in subsequent studies.

INDEX TERMS Forensic, anti-forensic, NAND flash memory, copy-back program, deletion operation, privacy, hacker, overwriting, program disturbance, victim cell, program prohibition.

I. INTRODUCTION

Largely composed of identification, collection, preservation, analysis, submission, and verification of evidence [1]–[3], digital forensics is classified into disk forensics, memory forensics, and network forensics depending on the target [4], [5]. Disk forensics analyzes the structure of the disk and recovers deleted files that once existed in the hardware [6], [7]. Recently, since the main storage devices of computers mainly use SSD or flash memory, analysis techniques for these are the principal focus of research [8]. Disk forensics include USB Stick, CF Card, SD Card, eMMC, and UFS using NAND flash memory. In general, memory forensics analyzes computer RAM and is used in various fields of information protection, such as malicious code analysis, network security, threat information collection, and incident response [9], [10]. Network forensics collects and analyzes data related to computer networks [11], which mainly refers to digital forensics that analyzes and monitors network connection information or packets. Our main concern is disk forensics using NAND flash memory.

In NAND flash memory, even if the data is deleted, the original data is known to remain. Researchers have introduced various techniques to solve this problem recently. It is known that NAND flash memory cannot be overwritten in general, but in 2017, Ahn and Lee first proposed a complete deletion technique that overwrites random data using a programmable state [12]. Other researchers have also proposed programming multi-bit programs as single-bit programs [13]–[15]. In 2019, Ahn and Lee proposed a down-level programming technique and an erasure pulse application technique [16]. These studies are currently ongoing. While previous papers have mainly discussed complete deletion, NAND flash memory runs a copy-back program

The associate editor coordinating the review of this manuscript and approving it for publication was Mervat Adib Bamiah.

that internally backs up the original data regardless of this deletion operation [17]–[19]. In this case, the original data is divided into managed original data and unmanaged original data.

Our paper is concerned with the original, unmanaged data. In Section II, we expand on forensics and anti-forensics for unmanaged data according to the copy-back program operation. Section III describes the forensic possibility of copy-back program operation, and in Section IV, we propose a safe copy-back program operation from an anti-forensic point of view in which forensic analysis is impossible. The proposed copy-back program operation suggests various deletion techniques to reduce program disturbance. In Section V, we compare the performance of the proposed copy-back programming schemes and the conventional copy-back programming schemes.

This paper is the first to discuss the forensic technique of NAND flash memory's background operations. In particular, it proposes an anti-forensic technique based on a copy-back program operation, which is a major copy-back program operation among these background operations. As NAND flash memories are increasingly integrated, data reliability problems continue to arise. For this reason, background operations are frequently performed inside the NAND chip, which makes it easy to predict that the neglect of the original data (or, data remanence) according to the background operations deepens. We are also the first to raise the issue of neglect of the original data and present ways to resolve this. In addition, it was revealed that the methods we propose not only achieve anti-forensics but are also more effective in terms of data reliability. Below, we explain in more detail the contributions of our paper.

II. FORENSICS AND ANTI-FORENSICS OF NAND FLASH MEMORIES

Today we live in a flood of digital information. With the Internet of Things, digital information is being shared and spread more rapidly. At the same time, issues regarding the security of digital information are emerging. Most information devices use NAND flash memory. NAND flash memory is a non-volatile memory, and is advantageous in terms of power compared to the conventional volatile memory. In addition, since NAND flash memory is preferential for integration, a relatively large amount of data can be stored in a smaller area than the conventional DRAM.

NAND flash memories are widely used as storage devices in electronic devices/data centers and are generally known as a memory type that cannot be overwritten [20]–[22]. Due to the unique structure of a data block, it is better to use a deletion operation that logically destroys the mapping relationship rather than one that performs a physical deletion (or erasure) operation. These features increase the possibility of NAND flash memories becoming forensics targets. Many studies have been conducted to perform digital forensics by restoring metadata, which is management data of NAND flash memory [23], [24].

To compensate for this metadata vulnerability, studies proposed a file wiping technique [25], [26]. File wiping is a technique that completely deletes files as an alternative to solving the vulnerabilities remaining in the data area while the metadata of the file system is modified when files are deleted. Wiping is to overwrite 0 or 1, a specific pattern data, and random number data several times in the data area of the same logical address as the data in order to ensure complete deletion. The number of times the other data is overwritten by wiping varies from 1 to 35 times depending on the algorithm. Representative wiping tools include Both Eraser, WipePro+, BCWipe, Secure Data Wiper, Delete Files Permanetly, and Secure Delete, and are divided into Windows-based wiping tools and Linux-based wiping tools depending on the operating system environment. In addition, it is divided into an entire area, a sector range, and a file according to the execution target in the device.

However, even if file wiping is performed in the operating system, overwriting is not physically performed in the existing NAND flash memory unit alone. File wiping only leaves the original data and its children extensively in NAND flash memory. This is because changing a file in the operating system does not overwrite NAND flash memory, but meaning essentially that the new overwritten files are created in NAND flash memories. The host overwrites the file, but the data remains. To solve this problem, Ahn and Lee introduced complete data deletion techniques [16], and they raised the question of data retention due to garbage collection and suggested how to handle the original data. As with garbage collection, a copy-back program is an important management operation for handling NAND flash memories. Such a copy-back program is generally performed in the background, and in this case, the original data is often left as is. This can be an anti-forensic target. Next, we examine the concept of copy-back program operation and discuss in detail how this operation leaves data.

III. FORENSICS IN COPY-BACK PROGRAM

Is the data once programmed permanent? In the case of NAND flash memory, data corresponds to the amount of charge stored in the charge storage layer. However, the charges in the charge storage layer react sensitively to changes in the environment (e.g., temperature, time, and voltage), and thus have deterioration characteristics [27]. Typically, this causes charge loss, thereby destroying the stored data. In order to prevent this problem, a copy-back program operation is used in NAND flash memory. The degree of deterioration of the memory cell is monitored, and when a specific condition is satisfied, a copy-back program operation is internally performed [17]–[19], [28]. In the copy-back program operation, as shown in FIG. 2, the original data stored in the source page of the first block BLK1 is programmed in the destination page of the second block BLK2. In general, NAND flash memory performs such a copy-back program operation in response to a copy-back command.

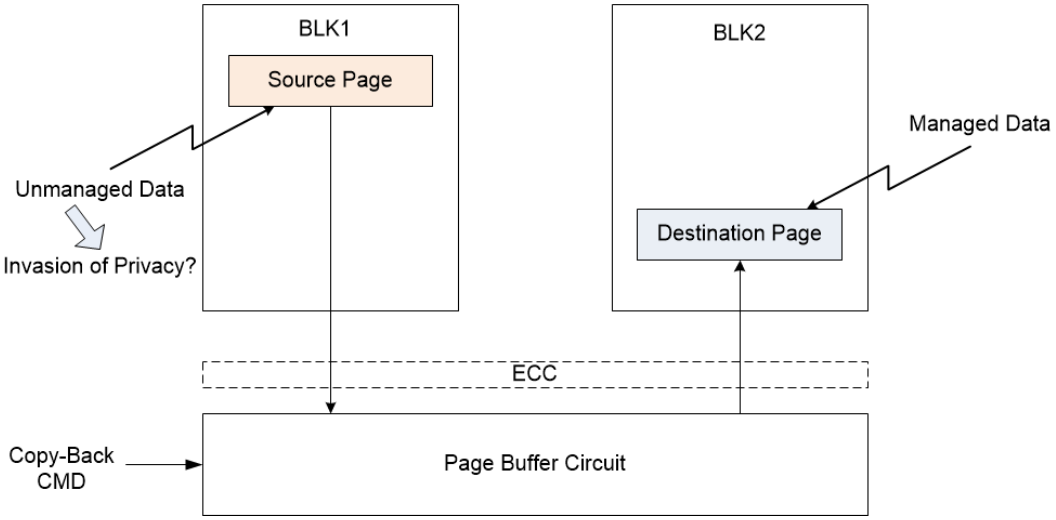

FIGURE 1. A typical copy-back program operation is shown. Source page data from the first block is transferred to the destination page of the second block according to the copy-back program command. After the copy-back program operation, the data stored in the source page becomes unmanaged data.

As shown in FIG. 1, this copy-back program operation performs error correction to improve the reliability of data read from a source page. Such error correction is performed by a controller external to NAND flash memory (Off-chip ECC) or by itself inside NAND flash memory (On-chip ECC).

When the copy-back program operation is completed, the original data from the source page of the first block BLK1 becomes unmanaged data, and the data from the destination of the second block BLK2 is manageable, and thus becomes validate data. Our concern in this study is unmanaged data. If the unmanaged data is personal information, personal information may exist in NAND flash memory not only in a managed data form but also an unmanaged data form, meaning that unmanaged data can be hacked by illegal users. That is, there is a high possibility of privacy exposure through this kind of unmanaged data.

Is the data in NAND flash memory safe? We can confirm that the possibility of data exposure is high under a set of extreme assumptions. It is assumed that the copy-back program operation is performed internally, and it is also assumed that the original data is personal information. The host transmits a deletion request regarding personal information to the storage device (SSD/USB/Card) as needed. According to the host request, the controller of the storage device may transmit an erase command to the second block BLK2 to NAND flash memory. NAND flash memory performs an erase operation on the second block BLK2 as per the command. Accordingly, personal information stored in the destination page of the second block BLK2 is deleted. When the erasing operation has completed, NAND flash memory outputs the erasing operation completion information to controller. Controller outputs deletion completion on the host in response to the erasure completion information, and the host subsequently recognizes that the personal information is no longer stored on the storage device. But, as described in FIG. 1, personal information stored as unmanaged data remains in NAND flash memory.

Referring to FIG. 2, potential hackers can exploit the weaknesses of this cache-back program behavior. We assume that the hacker has powerful abilities, can access the management information of controller, has sufficient access to the block management information of controller, and can access NVM by changing the unmanaged block to a manageable block. From this point, a hacker can access unmanaged data and send a read request to the storage device in order to access an address for unmanaged data like any other normal user. In response to this type of read request, controller may transmit a read command and allow access to the unmanaged data on NAND flash memory. In response to the read command, NAND flash memory may then perform a read operation from the source page of the first block BLK1, where the unmanaged data is stored, and transmit the result to controller. Controller may, as a result, transmit the read data to the hacker as per the read operation. In this way, the privacy of NAND flash memory is only violated in simple terms.

To look at this in a positive light, potential hackers can inadvertently help simplify digital forensics as related to criminal activity. On the other hand, from a negative point of view, potential hackers have the ability to seriously invade personal privacy.

IV. PROPOSED SECURE COPY-BACK PROGRAM

In our research, we added an enhanced secure program operation to the conventional copy-pack program operation. As shown in FIG. 3, the secure copy-back program operation includes a read operation for a physical page corresponding to a source address, a program operation for a physical page corresponding to a destination address, and a physical page corresponding to the source page. It may also include an enhanced secure program operation.

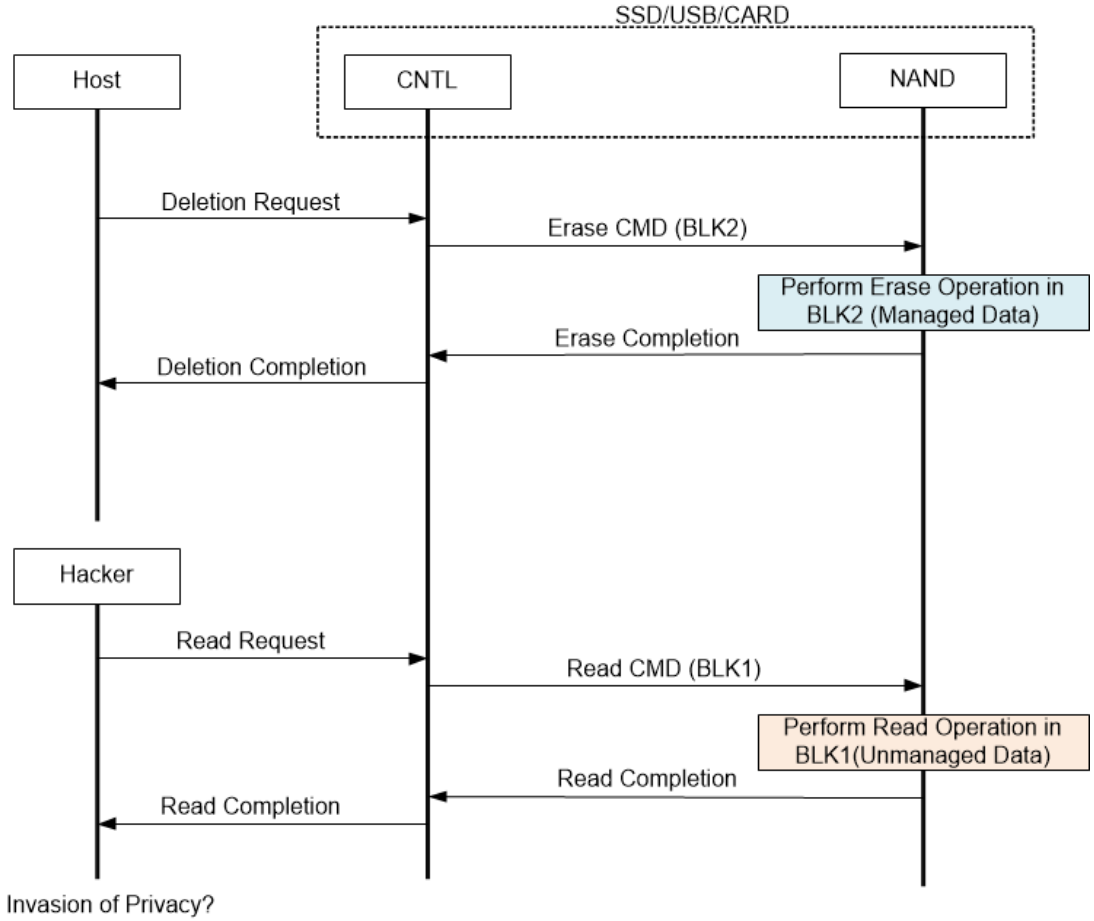

FIGURE 2. When performing a copy-back program operation in NAND flash memory, the process shows unmanaged data being targeted by hackers and how the hacker can access such unmanaged data.

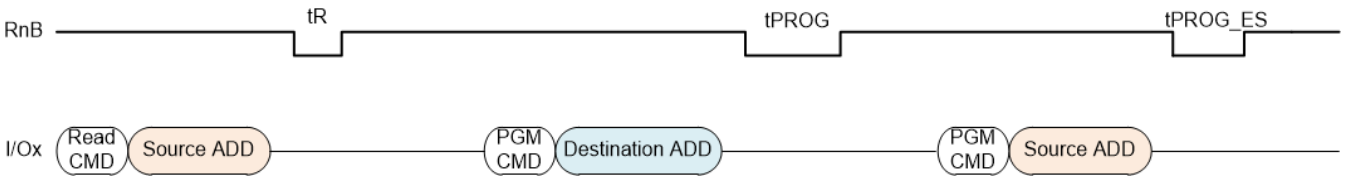

FIGURE 3. The timing of the proposed copy-back program operation is shown. It consists of one read command and two program commands.

This read operation may be performed on a corresponding source page via a read command and a source address in NAND flash memory during read time tR. This program operation may be performed on a destination page corresponding to a program command and a destination address in NAND flash memory during program time tPROG.

Thereafter, the enhanced secure program operation may be performed in NAND flash memory according to the program command and the source address during the enhanced secure program operation time tPROG_ES. After the existing copy-back program operation, the program operation proposed in the source page is further performed. As a kind of overwrite, the original data stored in the source page is changed.

A. OVERWRITING SCHEMES IN SECURE COPY-BACK PROGRAM

The proposed secure copy-back program operation further performs an enhanced secure program operation after the existing copy-back program operation. This enhanced secure program operation can be implemented in two main forms. The first reinforced secure program operation means an operation of programming random data in the source page, but the random data does not need to be fully programmed into the source page as the goal is to destroy the data stored in the source page, referring to FIG. 4 (a). The second enhanced secure program operation refers to an operation that applies a plurality of deletion pulses to the source page, referring to FIG. 4 (b). Data stored on the source page is changed

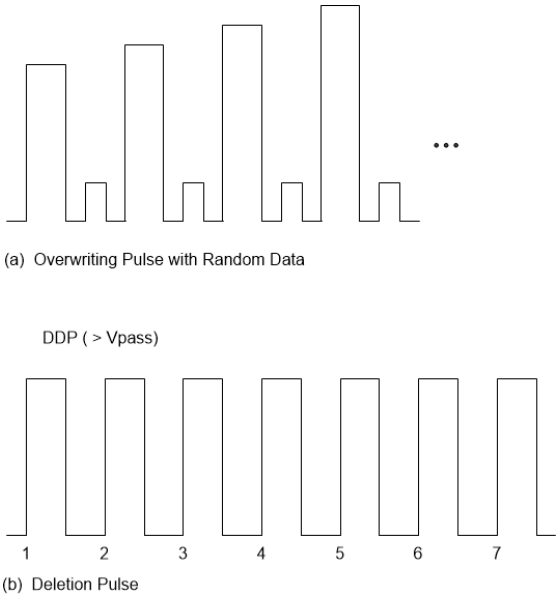

FIGURE 4. The proposed deletion techniques are shown. (a) is an overwriting technique using random data, and (b) is a technique that applies a plurality of deletion pulses.

into an unknown form in conjunction with the application of the deletion pulses. Here, the application of the erasure pulse means application to the word line corresponding to the source page.

B. PROGRAM DISTURBANCE ISSUE

On the other hand, a secure copy-back program operation must be performed to minimize damage to any data connected to another word line when performing an enhanced secure program operation. This is because it is easy to cause program disturbance due to the reinforced secure program operation [29], [30]. In general, other researchers have already introduced various methods to reduce program disturbance. For example, a channel boosting operation may be performed in advance to reduce program disturbance caused by an enhanced secure program operation. In particular, in order to reduce program disturbance, a reinforced secure program operation can be performed when the channel is initially charged.

Before performing an enhanced secure program operation, the channel is precharged with power supply voltage (VDD). The channel is either all-precharged or partially precharged depending on the data stored in the word line [31], [32]. Pass voltage Vpass can be applied to the unselected word lines WLi − 2, WLi − 1, WLi + 1, and WLi + 2, and a deletion pulse or a program pulse can then be applied to the selected word line, referring to FIG. 5. Accordingly, the source data of the memory cell connected to selected word line WLi becomes invalid data.

C. MAINTAINING THE RELIABILITY OF ADJACENT CELLS

The overwriting technique inevitably affects the deterioration of memory cells connected to adjacent word lines. A boosting operation is performed to minimize the effect of word line coupling, but this is still insufficient. Although the operator may attempt to delete unnecessary data, in reality this action reduces the reliability of necessary data. Accordingly, a data recovery read operation may be performed on word lines adjacent to the word line using the overwriting technique.

The data recovery read operation refers to changing a read level by reflecting program states of memory cells connected to an adjacent word line when reading data of memory cells connected to a selected word line [33]. The data recovery read operation largely involves reading data from an attack cell connected to an adjacent unselected word line, changing the read condition of a memory cell connected to the selected word line based on the read data of the attack cell as well as the changed read condition. In addition, reading data of a memory cell connected to the selected word line is read on the basis of the read condition. Here, the changed read condition may include a read level, a development time, or a precharge time.

D. OVERWRITING USING VICTIM CELL INFORMATION

Among the memory cells connected to the adjacent word line, there are cells that are heavily affected and cells that are not. For example, a cell programmed as a sub-state is likely to be a victim cell. On the other hand, cells programmed to the higher state are less likely to be affected by overwriting. Therefore, prior to performing a deletion operation according to the overwriting technique, it is possible to first read the victim cell group information in the adjacent word line and decide whether to program random data using the read victim cell group information. For example, if the big team cell group information of the adjacent word line indicates the existence of a big team cell, the corresponding page buffer becomes controlled to maintain the program prohibition state.

V. PERFORMANCE COMPARISON

From our research, we found that techniques such as copy-back programs among conventional background-ground operations cause the object of forensics. In addition, we proposed a secure copy-back program to solve this problem. While our techniques are proven safe against forensics on the one hand, they have the potential to cause degradation in overall system performance on the other. Therefore, by comparing system performance, we can confirm the superiority of the proposed techniques as follows.

First, the prior art can be divided into ECC on-chips and ECC-off chips, and the technology that we propose corresponding to this can also be divided generally into secure ECC-on chips and secure ECC-off chips as well as Integrity & Secure ECC-on chips and Integrity & Secure ECC-off chips that do not prevent program disturbance. For convenience of explanation, it is assumed that the integrity and secure copy-back program operation performs at least two read operations on word lines adjacent to the selected word line in order to obtain the victim cell information.

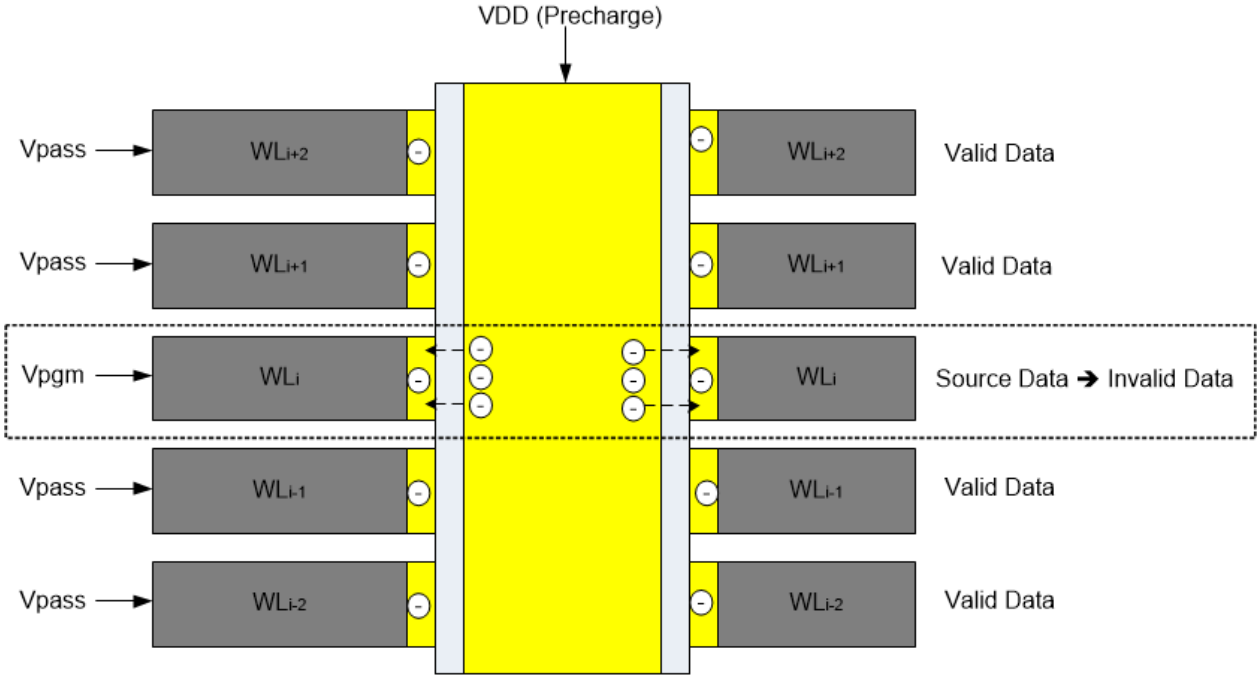

FIGURE 5. The structure of a general 3D NAND flash memory is shown. The plate-shaped word lines are stacked. When the erasing technique is applied to a memory cell connected to word line WLi, a program disturbance may be induced in a memory cell connected to the adjacent word line. Various deletion techniques are applied to minimize such a program disaster.

TABLE 1. Performance comparison of copy-back program operation according to chip types.

Chip Type	Total Operation Time	Adjacent Victim Prevention	Anti-Forensic
Prior ECC On-Chip	$T_R + T_{PGM}$	X	X
Prior ECC Off-chip	$T_R + T_{ECC} + T_{PGM}$	X	X
Secure ECC On-Chip	$T_R + 2T_{PGM}$	X	O
Secure ECC Off-Chip	$T_R + T_{ECC} + 2T_{PGM}$	X	O
Integrity& Secure ECC On-Chip	$3T_R + 2T_{PGM}$	O	O
Integrity& Secure ECC Off-Chip	$3T_R + T_{ECC} + 2T_{PGM}$	O	O

Here, T_R is read operation time, T_{PGM} is program operation time, and T_{ECC} is ECC operation time. In general, the ECC on-chip structure can hide the ECC operation time from the normal copy-back program operation. Therefore, only in the ECC-off chip structure does T_{ECC} exists. For example, page program T_{PGM} is approximately 600 μs, page read time T_R is approximately 25 μs, and internal ECC time T_{ECC} is approximately 100 μs [34]. Since T_{ECC} and T_R are small compared to T_{PGM}, the overall time is less affected. The proposed schemes consume as much time as programming but have the added benefit of being able to ward off forensics. Moreover, the existing copy-back program method does not prevent neighbor-cell victims, but the integrity and secure copy-back program method that we propose in this paper can achieve this.

As described above, we compared proposed copy-back program operations with conventional operations and performance metrics such as total operation time, victim cell prevention, and anti-forensics. Meanwhile, in NAND flash memory, the copy-back program operation is not the only background operation. A read reclaim related to data healing of a read operation can also be developed similarly to our secure copy-back program.

VI. CONCLUSION

Targets of digital forensics have been revealed according to NAND flash memory's background operations like a garbage collection, a copy-back program operation, and so on. In particular, in the copy-back program operation, we confirmed that unmanaged original data was the main target of forensics. We proposed a safe copy-back program operation that safely deletes such unmanaged data processing, and the copy-back program operation we proposed uses various techniques to minimize the influence of adjacent victim cells. For example, in order to minimize program disturbance, the initial boosting operation is preferentially performed on the channel. Alternatively, when overwriting is performed by acquiring information about adjacent victim cells, the prohibition of the program is first determined. Such secure copy-back program operation can minimize traces of original data even if the background operations are internally performed in NAND flash memories. For data center or personal electronics that are highly germane to privacy invasion issues, the secure

copy-back program operation is perfectly applicable. In the future, further studies on privacy issues of unmanaged data using management data are needed.

ACKNOWLEDGMENT

The authors are so grateful to Prof. Sang-Jin Lee of the Graduate School of Information Security, Korea University, for his interest in the concept of digital forensics.

REFERENCES

[1] S. Grigaliunas, J. Toldinas, A. Venckauskas, N. Morkevicius, and R. Damasevicius, "Digital evidence object model for situation awareness and decision making in digital forensics investigation," *IEEE Intell. Syst.*, early access, Aug. 27, 2020, doi: 10.1109/MIS.2020.3020008.

[2] A. Jolfaei, P. Ostovari, M. Alazab, I. Gondal, and K. Kant, "Guest editorial special issue on privacy and security in distributed edge computing and evolving IoT," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2496–2500, Apr. 2020, doi: 10.1109/JIOT.2020.2980103.

[3] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-based camera model fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, 2020, doi: 10.1109/TIFS.2019.2916364.

[4] P. Joseph and J. J. Norman, "An analysis of digital forensics in cyber security," in *Proc. 1st Int. Conf. Artif. Intell. Cogn. Comput.*, in Advances in Intelligent Systems and Computing, vol. 815, R. Bapi, K. Rao, and M. Prasad, Eds. 2019, doi: 10.1007/978-981-13-1580-0_67.

[5] M. Rychlý and O. Ryšavý, "TARZAN: An integrated platform for security analysis," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Prague, Czech Republic, Sep. 2017, pp. 561–567, doi: 10.15439/2017F280.

[6] J. Vieyra, M. Scanlon, and NA. Le-Khac, "Solid state drive forensics: Where do we stand?" in *Digital Forensics and Cyber Crime* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 259, F. Breitinger and I. Baggili, Eds. Cham, Switzerland: Springer, 2019, pp. 149–164, doi: 10.1007/978-3-030-05487-8_8.

[7] S. Tomer, A. Apurva, P. Ranakoti, S. Yadav, and N. R. Roy, "Data recovery in forensics," in *Proc. Int. Conf. Comput. Commun. Technol. Smart Nation (IC3TSN)*, Gurgaon, India, Oct. 2017, pp. 188–192, doi: 10.1109/IC3TSN.2017.8284474.

[8] M. Gibson, N. Medina, and Z. Z. Nail, "SSD forensics: Evidence generation and analysis," in *Digital Forensic Education* (Studies in Big Data), vol. 61, X. Zhang and K. K. Choo, Eds. Cham, Switzerland: Springer, 2020, pp. 203–218, doi: 10.1007/978-3-030-23547-5_11.

[9] D. C. D'Elia, E. Coppa, F. Palmaro, and L. Cavallaro, "On the dissection of evasive malware," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2750–2765, 2020, doi: 10.1109/TIFS.2020.2976559.

[10] A. Kazim, F. Almaeeni, S. A. Ali, F. Iqbal, and K. Al-Hussaeni, "Memory forensics: Recovering chat messages and encryption master key," in *Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS)*, Irbid, Jordan, 2019, pp. 58–64, doi: 10.1109/IACS.2019.8809179.

[11] S. P. Utomo, B. Pramudiono, and A. Muharram, "Method to uncover IP spoofing attack on network forensics using NFAT and IP correlation as combined approach," in *Proc. Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Yogyakarta, Indonesia, Jul. 2019, pp. 302–305, doi: 10.1109/ICOIACT46704.2019.8938476.

[12] N.-Y. Ahn and D. Hoon Lee, "Duty to delete on non-volatile memory," 2017, *arXiv:1707.02842*. [Online]. Available: http://arxiv.org/abs/1707.02842

[13] D. Chang, W. Lin, and H. Chen, "FastRead: Improving read performance for multilevel-cell flash memory," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 9, pp. 2998–3002, Sep. 2016.

[14] D. Wang, J. Tang, M. Jia, Z. Xu, and H. Han, "Review of NAND flash information erasure based on overwrite technology," in *Proc. 39th Chin. Control Conf. (CCC)*, Shenyang, China, Jul. 2020, pp. 1150–1155, doi: 10.23919/CCC50068.2020.9189542.

[15] W.-C. Wang, C.-C. Ho, Y.-M. Chang, and Y.-H. Chang, "Challenges and designs for secure deletion in storage systems," in *Proc. Indo–Taiwan 2nd Int. Conf. Comput., Anal. Netw. (Indo-Taiwan ICAN)*, Punjab, India, Feb. 2020, pp. 181–189, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181335.

[16] N.-Y. Ahn and D. H. Lee, "Schemes for privacy data destruction in a NAND flash memory," *IEEE Access*, vol. 7, pp. 181305–181313, 2019, doi: 10.1109/ACCESS.2019.2958628.

[17] H. Nakamura, K. Imamiya, T. Himeno, T. Yamamura, T. Ikehashi, K. Takeuchi, K. Kanda, K. Hosono, T. Futatsuyama, K. Kawai, R. Shirota, N. Arai, F. Arai, K. Hatakeyama, H. Hazama, M. Saito, H. Meguro, K. Conley, K. Quader, and J. Chen, "A 125 mm^2 1Gb NAND flash memory with 10 MB/s program throughput," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2002, pp. 106–450, doi: 10.1109/ISSCC.2002.992961.

[18] G. Xin, D. Zibin, L. Wei, and F. Lulu, "Design and implementation of a NAND flash controller in SoC," in *Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits*, Tianjin, China, Nov. 2011, pp. 1–2, doi: 10.1109/EDSSC.2011.6117658.

[19] S. Jin Kwon, H.-J. Cho, and T.-S. Chung, "Fast responsive flash translation layer for smart devices," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 52–59, Feb. 2014, doi: 10.1109/TCE.2014.6780925.

[20] H.-L. Li, C.-L. Yang, and H.-W. Tseng, "Energy-aware flash memory management in virtual memory system," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 8, pp. 952–964, Aug. 2008, doi: 10.1109/TVLSI.2008.2000517.

[21] G. Xu, M. Wang, and Y. Liu, "Swap-aware garbage collection algorithm for NAND flash-based consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 60–65, Feb. 2014, doi: 10.1109/TCE.2014.6780926.

[22] P.-H. Lin, Y.-M. Chang, Y.-C. Li, W.-C. Wang, C.-C. Ho, and Y.-H. Chang, "Achieving fast sanitization with zero live data copy for MLC flash memory," in *Proc. Int. Conf. Comput.-Aided Design*, San Diego, CA, USA, Nov. 2018, pp. 1–8, doi: 10.1145/3240765.3240773.

[23] L. Caviglione, S. Wendzel, and W. Mazurczyk, "The future of digital forensics: Challenges and the road ahead," *IEEE Secur. Privacy*, vol. 15, no. 6, pp. 12–17, Nov. 2017, doi: 10.1109/MSP.2017.4251117.

[24] J. Hou, Y. Li, J. Yu, and W. Shi, "A survey on digital forensics in Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 1–15, Jan. 2020, doi: 10.1109/JIOT.2019.2940713.

[25] M. Sahri, S. N. H. S. Abdulah, M. F. E. Senan, N. A. Yusof, N. Z. B. Z. Abidin, N. S. Bin Shaiful Azam, and T. J. Bin Tajul Ariffin, "The efficiency of wiping tools in media sanitization," in *Proc. Cyber Resilience Conf. (CRC)*, Putrajaya, Malaysia, Nov. 2018, pp. 1–4, doi: 10.1109/CR.2018.8626824.

[26] J. Zheng, Y.-A. Tan, X. Zhang, C. Liang, C. Zhang, and J. Zheng, "An anti-forensics method against memory acquiring for Android devices," in *Proc. 7 IEEE Int. Conf. Comput. Sci. Eng. (CSE) IEEE Int. Conf. Embedded Ubiquitous Comput. (EUC)*, Guangzhou, China, Jul. 2017, pp. 214–218, doi: 10.1109/CSE-EUC.2017.45.

[27] W. Lee, M. Kang, S. Hong, and S. Kim, "Interpage-based endurance-enhancing lower state encoding for MLC and TLC flash memory storages," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 9, pp. 2033–2045, Sep. 2019, doi: 10.1109/TVLSI.2019.2912228.

[28] A. Khanbadr, M. B. Marvasti, S. A. Asghari, and S. Khanbadr, "The SBM: A victim block selection method based on min-heap priority queues," in *Proc. CSI/CPSSI Int. Symp. Real-Time Embedded Syst. Technol. (RTEST)*, Tehran, Iran, Jun. 2020, pp. 1–8, doi: 10.1109/RTEST49666.2020.9140084.

[29] S.-K. Lu, S.-C. Yu, C.-L. Hsu, C.-T. Sun, M. Hashizume, and H. Yotsuyanagi, "Fault-aware dependability enhancement techniques for flash memories," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 3, pp. 634–645, Mar. 2020, doi: 10.1109/TVLSI.2019.2957830.

[30] D. Wei, L. Qiao, S. Wang, and X. Peng, "Fixation ratio of error location-aware strategy for increased reliable retention time of flash memory," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 10, pp. 3145–3155, Oct. 2016, doi: 10.1109/TVLSI.2016.2537846.

[31] S. Sakib, M. T. Rahman, A. Milenkovic, and B. Ray, "Flash memory based physical unclonable function," in *Proc. SoutheastCon*, Huntsville, AL, USA, Apr. 2019, pp. 1–6, doi: 10.1109/SoutheastCon42311.2019.9020567.

[32] M.-C. Yang, Y.-H. Chang, T.-W. Kuo, and F.-H. Chen, "Reducing data migration overheads of flash wear leveling in a progressive way," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 5, pp. 1808–1820, May 2016, doi: 10.1109/TVLSI.2015.2495252.

[33] C. Adnan Aslam, Y. L. Guan, and K. Cai, "Detector for MLC NAND flash memory using neighbor-a-priori information," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 9, pp. 2827–2836, Sep. 2016, doi: 10.1109/TVLSI.2016.2523759.

[34] *4Gb, 8Gb, and 16Gb x8 NAND Flash Memory Features*, Mircron, Boise, ID, USA, 2009.

NA YOUNG AHN received the B.S. and M.S. degrees from the Department of Electrical Engineering, Korea University, and the Ph.D. degree in cyber security. He is currently a Postdoctoral Researcher with the Institute of Cyber Security & Privacy, Korea University, South Korea. Since 2005, he has been a Patent Engineer with patent and law firms. His articles have been published in journals, including IEEE Access and *Ad-Hoc and Sensor Wireless Networks*. His research interests include physical layer security in vehicular communications, biometric authentications, PoN based blockchain, and anti-forensics in flash memories.

DONG HOON LEE (Member, IEEE) received the B.S. degree in economics from Korea University, Seoul, South Korea, in 1985, and the M.S. and Ph.D. degrees in computer science from The University of Oklahoma, Norman, OK, USA, in 1988 and 1992, respectively. Since 1993, he has been with the Faculty of Computer Science and Information Security, Korea University. His research interests include the design and analysis of cryptographic protocols in key agreement, encryption, signatures, embedded device security, and privacy-enhancing technology.