StreamSanitize: Secure Deletion In Solid State Drive Using Stream Cipher

Devesh Kumar
Department of Computer Science & Engineering
Indian Institute of Technology
Ropar, India
2022csm1022@iitrpr.ac.in

T.V. Kalyan
Department of Computer Science & Engineering
Indian Institute of Technology
Ropar, India
kalyantv@iitrpr.ac.in

***Abstract*—The secure deletion of data from Solid-State Drives (SSDs) is a crucial security concern due to their inherent characteristics, such as out-of-place update and flash memory storage. Unlike traditional hard disk drives, when a file is deleted from an SSD, the data is not actually destroyed but tagged as invalid and kept on the drive until the garbage collection process erases it. This makes sensitive data vulnerable to malicious attacks and deleting invalid data securely from flash storage is critical to protect user data privacy. This paper proposes StreamSanitize, a secure sanitization technique combing normal data deletion with secure data deletion using Stream Cipher ChaCha20, the more cost-effective one to delete invalid data securely. The proposed method significantly reduces the secure deletion cost while guaranteeing complete security by deleting all invalid SSD pages, as demonstrated through real-world benchmark experiments.**

***Index Terms*—Data Sanitization, Secure Deletion, SSD, SSD-Sim, Stream cipher, ChaCha20**

I. Introduction

Solid State Drives are now commonly used in computer systems, increasing the demand for an effective way to securely delete data from the drives. Due to their out-of-place update property and large data migration requirements, traditional data sanitization techniques and encryption-based methods are ineffective for SSDs. To address this challenge, a new proposed scheme called StreamSanitize is presented in this paper. The scheme combines normal data sanitization with secure data deletion and selects stream cipher chacha20 as the more cost-effective method to delete invalid SSD pages securely. This reduces the read-write operations and enhances the SSD's performance.

The paper also refers to the ErasuCrypto [1] framework, which integrates erasure-based and cryptography-based schemes to delete invalid data while reducing the deletion cost. Another scheme, Workload Aware Secure Deletion (WAS-Deletion) [2], is also referred to in this paper that reduces the overhead of secure deletion for SSDs. The scheme effectively separates invalid and valid pages into blocks or chunks based on their update frequencies and request sizes that have been accumulated over time. However, the AES-128 encryption scheme used in WAS-Deletion can be replaced with a stream cipher that requires fewer keys and enhances endurance for lesser read/writes for keys. In figure 1, ChaCha20 is proposed as an encryption scheme that is much faster than AES-128 on microcontrollers and safer in the context of padding attacks.

Thanks to IIT Ropar for providing the infrastructure for this research work.

The newly proposed technique, StreamSanitize is a cost-effective and efficient solution for secure data deletion in SSDs. The scheme combines different secure deletion methods to minimize the deletion cost and reduce the overhead of secure deletion for SSDs. The paper also presents experimental results compared to previous studies and offers insights into the potential impact of these schemes on future SSD development. These proposed schemes are critical for protecting sensitive data in computer systems.

The paper is structured as follows: Section 2 discusses the background and related work on secure deletion. Section 3 presents the proposed scheme called StreamSanitize. The experimental results are presented in Section 4 and are compared to those of previous studies. Lastly, Section 5 shows future work and Section 6 presents the conclusions drawn from the study.

II. Background And Related Work

Various basic secure deletion methods, such as Temperature-aware [3], Scrubbing-aware [4], Selectively secure deletion [5], and Fast sanitization [6], are available to address data security concerns. In the paper, the Challenges and Designs for Secure Deletion in Storage Systems [7], the authors have evaluated these methods based on various criteria, including key management, encryption, memory, and performance overheads. However, these methods must be improvised to be more effective, verifiable, and portable for flash-based solid-state storage.

One framework that combines erasure-based and encryption-based data deletion techniques is ErasuCrypto [1]. It selects the most cost-effective method to remove invalid data securely and presents a problem formulation to limit data migrations and block erasures, thereby reducing deletion costs. In this paper, A chunk is composed of several blocks. Its size is denoted as $Chunk_{size}$, which integrates erase-based and cryptography-based approaches to determine the lowest cost of secure deletion (as represented in Eq. (1)) by utilizing either of the schemes.

The goal is to minimize the cost by adding the number of erases (#Era) and migrations (#Mgr) required for secure deletion, multiplied by a coefficient k representing the ratio between the erasing and migrating data cost.

$$\textbf{Goal} : \textbf{Minimize}(\#Mgr + k \times \#Era) \quad (1)$$

This approach can be implemented at the controller level but has key management-related overheads.

Another novel approach is the Workload-Aware Secure Deletion scheme [2], which minimizes secure deletion overheads by using vertical encryption, adaptive scheduling, and efficient data block splitting. However, it needs to undergo a thorough evaluation of alternative cutting-edge secure deletion techniques.

III. StreamSanitize scheme

My proposed solution offers a secure approach to sanitizing data in SSDs with minimal impact on performance. The primary cause of overhead in secure data deletion arises from accessing the SSD keys or relocating valid blocks to a new location. To address this issue, we have focused on implementing an encryption scheme that uses fewer keys while maintaining data security. Furthermore, We have ensured that the encryption and decryption operations are smooth to avoid additional performance-related overhead. We have resolved the migration problem using region classification based on page frequency. We have selected ChaCha20, a stream cipher-based encryption algorithm efficient enough to execute on low-processing hardware such as SSD controllers. This algorithm 1 uses only the fewest keys possible to encrypt data, unlike other encryption methods like AES-128.

To achieve this, We propose a four-step process involving accessing the flash translation layer (FTL), identifying blocks and keys, migrating valid data, obtaining keys, and finally erasing and deleting selected blocks and keys.

1) **Accessing the FTL** - The first step in securely deleting data is accessing the FTL to collect information on the page's state. The FTL maps logical addresses to physical addresses on the SSD. By accessing the FTL, it is possible to determine which pages must be deleted and their location on the SSD.
2) **Identifying Blocks and Keys** - The second step identifies each chunk with the block(s) and group key(s) that need to be erased and deleted. Once identified, these blocks and keys can be migrated to new locations, ensuring the availability of valid data. The valid pages from the chosen blocks or groups are migrated and encrypted with new keys. They are then written to the corresponding SSD plane's active block and the FTL is updated with new page mapping information.
3) **Obtaining Keys** - The third step involves obtaining the keys from the designated blocks in the SSDs. This is done only when there is a valid page to encrypt. No extra or fixed-size key blocks are read from the SSD. Obtaining keys is critical to ensuring that data is encrypted with the latest keys and that no keys are left behind in blocks that have been deleted.
4) **Erasure and Key Deletion** - The final step in securely deleting data is the erasure and key deletion stage. The deletion engine communicates with the garbage collection engine and the SSD to erase the chosen blocks and delete the selected keys. This step ensures that all data is deleted securely and cannot be accessed by unauthorized parties.

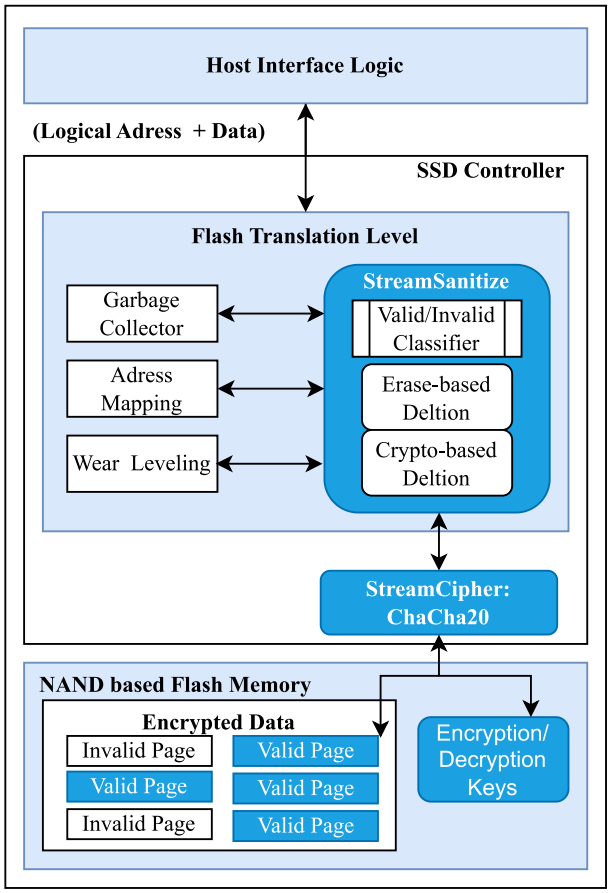

Fig. 1. SSD controller upgraded with StreamSanitize

IV. Experimental Results

A. Simulation Setup

We used the SSDSim simulator that operates based on pre-recorded traces. The simulated SSD is 512GiB in capacity and includes 8 NAND-Flash channels. Each element of the SSD has an independent event queue, allowing for separate handling of various accesses. In SSDSim, a new flag 'crypto' is added in the SSD structure and a node is inserted in the linked list for a fixed location(0x1000) is read keys. The encryption algo is selected with a new flag passed as a parameter to the SSDSim. The latency for ChaCha20 and AES128 is calculated based on the results demonstrated in Security and Performance in IoT [8] and added in the SSDSim for encryption and decryption of pages.

Algorithm 1 Secure Deletion with ChaCha20

procedure SECURE DELETION(a, b)
  Initialize Key Block & $queue_{block}$ for ChaCha20
  Initialize $cost_{crypto}$ & $cost_{erase}$ to 0
  while $queue_{block} \neq NULL$ **do**
    Compute $cost_{erase}$ for a whole block
    Compute $cost_{crypto}$ for multiple rows
    if Region is nearest **then**
      Erase-based sanitization for this region
    else if Region is farthest **then**
      Crypto-based sanitization for this region
      Delete used keys from SSD
    else
      if $cost_{erase} \leq cost_{crypto}$ **then**
        Cryptography deletion for this chunk
        Delete used keys from SSD
      else
        Erase-based deletion for this chunk
      end if
    end if
  end while
end procedure

TABLE I
SSD CONFIGURATION

Parameter	Value	Parameter	Value
Capacity	512gb	**No. of Channels**	8
Page Size	4kb	**No. of Chips**	64
Block Size	4096kb	**Erase Latency**	2.5ms
Pages/block	256	**Reserved Space for keys**	512MB

Table I shows the configuration where SSD comprises 1 plane per element, with each plane containing 4096 blocks. Each block consists of 256 pages, with a page size of 4KiB. These modifications to the SSDSim simulator enable the simulation of secure data deletion in a controlled and accurate manner.

TABLE II
BENCHMARKS: 1-WEEK BLOCK I/O MSR TRACES OF ENTERPRISE SERVERS AT MICROSOFT [9]

S.No.	File Name	Description	I/O (M)	Size
1	rsrch_0.csv	Research	1.43	75M
2	src1_2.csv	Source	1.91	97M
3	src2_0.csv	Source	1.56	79M
4	src2_2.csv	Source	1.16	61M
5	stg_0.csv	Stage	2.03	101M
6	stg_1.csv	Stage	2.2	111M
7	ts_0.csv	Test	1.8	89M
8	usr_0.csv	User	2.24	113M
9	wdev_0.csv	Web Dev	1.14	58M
10	web_0.csv	Web Dev	2.03	103M

We have selected ten real-world from the Microsoft I/O server trace set and summarized their characteristics in Table II. This high workload creates a significant challenge for the storage system's performance and generates a considerable amount of invalid data. For example, web staging servers may contain numerous temporary files and cookies that users may consider deleted.

B. Simulation Results

The stream sanitization process is evaluated using two metrics: the number of page migrations and read & write operations. Figure 2 shows the number of page migrations during sanitization, while Figure 3 displays the number of reads and writes among ErasuCrypto [1], WAS-Delete [2] and the proposed StreamSanitize schemes.

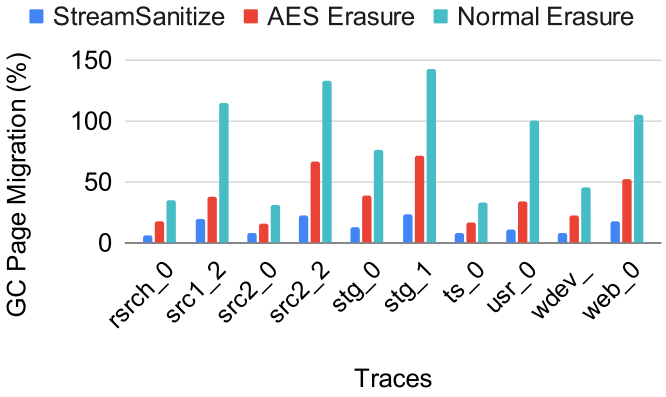

Fig. 2. Number of page migration from one block to another due the intrinsic properties of SSD.

The garbage collector(GC) identifies blocks containing mostly invalid pages for erasure and transfers any valid pages. The frequency of garbage collection and the number of migrations performed during garbage collection can be impacted by the deletion process.

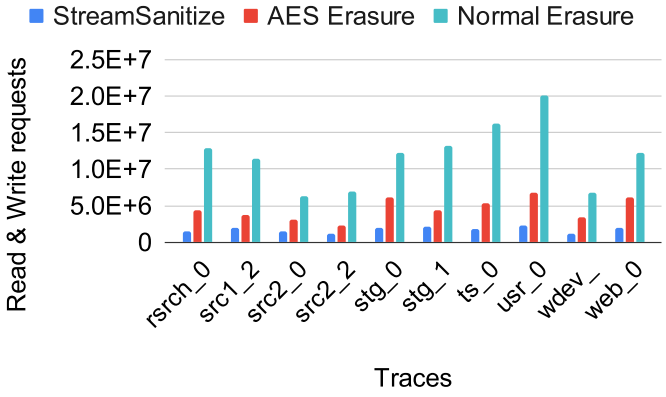

Fig. 3. Number of read and write operations including keys read for ChaCha20.

The StreamSanitize scheme, as shown in Figure 2, significantly reduces the overhead of the page migration compared to the other two schemes. Furthermore, as seen in Figure 3, the

StreamSanitize scheme has the lowest number of reads and writes due to its use of ChaCha20, which employs optimal keys for encryption and decryption, unlike other schemes that use AES-128.

V. Future Work

By analyzing the access patterns and write request sizes of previous requests, a machine learning classifier can be trained to recognize patterns that can be used to predict the appropriate region for new requests. This approach will allow for developing highly targeted data security protocols that can be customized to specific use cases and data access patterns.

However, even with the use of machine learning, there remains a need for robust encryption schemes to protect data against potential threats such as quantum computer attacks. One such example is NewHope [10] from Google, a post-quantum encryption algorithm designed to resist attacks from quantum computers. By developing and implementing such strategies, sensitive information protection and data security can be ensured.

VI. Conclusion

In this paper, the StreamSanitize technique is proposed which combines normal data sanitization with secure data deletion techniques using Stream Cipher ChaCha20, the more cost-effective one to delete invalid pages securely. This technique involves accessing and analyzing data, migrating valid data to secure locations, encrypting and writing data to designated blocks, and erasing and deleting keys to ensure unauthorized parties cannot access the data. This approach is essential for protecting sensitive information and maintaining privacy, making it an indispensable tool for individuals and organizations. Following this process can ensure that sensitive data is safely and securely deleted, providing peace of mind and mitigating the risk of data breaches or leaks.

Acknowledgement

Dr. T.V. Kalyan, Assistant Professor, CSE department, IIT Ropar, guided this work under the ”Memory System and Architecture” course. Also, Waqar Hassan Mir, Research Scholar, CSE department, IIT Ropar motivated to use different simulators to carry out experiments in this paper.

References

[1] C. Liu, H. A. Khouzani, and C. Yang, “Erasucrypto: A light-weight secure data deletion scheme for solid state drives,” *Proceedings on Privacy Enhancing Technologies*, vol. 2017, pp. 132 – 148, 2017.

[2] B. Li and D. Du, “Was-deletion: Workload-aware secure deletion scheme for solid-state drives,” in *2021 IEEE 39th International Conference on Computer Design (ICCD)*, 2021, pp. 244–247.

[3] B. Li and D. H.-C. Du, “Tasecure: Temperature-aware secure deletion scheme for solid state drives,” *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, 2019.

[4] W.-C. Wang, C.-C. Ho, Y.-H. Chang, T.-W. Kuo, and P.-H. Lin, “Scrubbing-aware secure deletion for 3-d nand flash,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2790–2801, 2018.

[5] H. A. Khouzani, C. Liu, and C. Yang, “Architecting data placement in ssds for efficient secure deletion implementation,” in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018, pp. 1–6.

[6] 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 *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018, pp. 1–8.

[7] W.-C. Wang, C.-C. Ho, Y.-M. Chang, and Y.-H. Chang, “Challenges and designs for secure deletion in storage systems,” in *2020 Indo – Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN)*, 2020, pp. 181–189.

[8] L. E. Kane, J. J. Chen, R. Thomas, V. Liu, and M. Mckague, “Security and performance in iot: A balancing act,” *IEEE Access*, vol. 8, pp. 121 969–121 986, 2020.

[9] SNIA, “Snia:the storage networking industry association,” http://iotta.snia.org/traces/block-io/388, 2023, accessed: April 10, 2023.

[10] Wikipedia, “Newhopein:post-quantum cryptography,,” https://en.wikipedia.org/wiki/NewHope, 2023, accessed: April 10, 2023.