CIS 5370
Plausibly Deniable SSD Through Data Layout Permutation

Prathyusha Batchalakuri
A Parameswara Reddy

Introduction

In today's digital era, the protection of sensitive data against unauthorized access and sophisticated forensic analysis is critical. Solid State Drives (SSDs) are prevalent for storage, known for their speed and reliability, yet they face unique security challenges due to their opaque operation systems. These challenges are intensified by advanced forensic methods that aim to uncover hidden or deleted data.

The research at the Florida Institute for Cybersecurity introduces "Plausibly Deniable SSD Through Data Layout Permutation," a novel approach to enhance data privacy through plausible deniability. This method allows users to convincingly deny the existence of sensitive data under scrutiny, protecting against both digital and physical threats.

Background and Related Work

The field of data privacy and security in solid-state drives (SSDs) has been thoroughly explored, yet the incorporation of plausible deniability through data layout permutation presents a novel contribution to the discourse. *Plausible Deniability in Digital Storage*: The concept of plausible deniability in digital contexts has evolved significantly over time. Early works like TrueCrypt pioneered methods for deniable encryption, allowing users to deny the existence of encrypted volumes [2]. These methods typically involve creating hidden volumes within visible ones, which cannot be confirmed without the correct authentication credentials [3]. However, these approaches are largely limited to encrypting data at rest and do not address the nuances of data layout manipulation at the hardware level. *Advanced Forensic Analysis Techniques*: Research in forensic analysis has shown that data remnants can be recovered from SSDs even after rigorous deletion protocols are followed [7]. These studies underscore the limitations of traditional deletion methods and highlight the necessity for more robust mechanisms that can withstand forensic scrutiny. *Emerging Techniques in SSD Security*: Recent studies have proposed various methods to enhance SSD security, such as the use of Write-Once-Read-Many (WORM) devices to prevent data tampering [4]. Additionally, Oblivious RAM (ORAM) has been suggested as a way to obfuscate access patterns, further securing data against both physical and digital threats [6] [5]. *Novel Data Permutation Techniques*: The technique of data layout permutation, as proposed by Chen et al [1], involves rearranging data blocks within the SSD in a non-linear and unpredictable pattern. This method not only encrypts data but also scatters it across the storage medium, which makes it extremely challenging for unauthorized entities to reconstruct the original information.

Methodology

This section presents the comprehensive methodology used in our project, which focuses on the integration of plausible deniability into SSDs through data layout permutation. It details the threat model that the project addresses and the specific technical approaches, including the use of the GNU Multiple Precision Arithmetic Library (GMP), ranking and

unranking permutations, and multithreading techniques, to answer the research questions effectively.

1. Threat Model: Our threat model accounts for realistic adversarial scenarios affecting flash-based SSDs, encompassing both digital and physical threats. We consider adversaries capable of forensic analysis and physical access, skilled in extracting and analyzing SSD data. The model specifically addresses the risk of coercive attacks that employ physical force or legal pressure to breach data integrity and confidentiality. This model is crucial for devising robust security strategies to mitigate a wide range of security threats effectively.

2. Methodology/Technical Approach

The technical approach of this project includes multiple detailed steps and methodologies designed to effectively implement a plausibly deniable SSD system:

- *Utilization of the GMP Library*: We use the GNU Multiple Precision Arithmetic Library (GMP) for complex number computations necessary in our data permutation processes. This library supports operations with large integers, rational numbers, and floating-point numbers, ensuring precision in our algorithms that manage data layout permutations.

- Algorithm Development for Ranking and Unranking: The core of our methodology involves the development of ranking and unranking algorithms. Ranking converts a bit stream into a permutation of data blocks, thereby obfuscating the original data arrangement. Unranking reverses this process, reconstructing the original data from the permuted layout without leaving interpretable traces for forensic recovery.

- Integration of Multithreading: To enhance our SSD system's performance with the proposed security features, we employ multithreading techniques. This enables concurrent execution of multiple operations, such as converting bit streams to permutations and vice versa, ensuring efficient system operation without notable delays in data access or processing.

- Performance Evaluation: We perform thorough performance evaluations to benchmark our plausibly deniable SSD system against conventional SSD architectures, focusing on latency, CPU utilization, and system responsiveness across different operational loads.

- Iterative Testing and Refinement: The system undergoes iterative testing to identify and rectify any inefficiencies or vulnerabilities within the ranking and unranking algorithms or the multithreading implementation. This stage is critical to refining the system to meet stringent security and performance standards.

This methodology enhances the privacy and security of SSDs while maintaining practicality for daily use, ensuring no compromise on performance or user experience. By thoroughly addressing both theoretical and practical aspects of plausible deniability in SSDs, our project significantly advances data security technology.

Experimental Results

Experiment Setup: Three file types — .txt, .docx, and .pdf — served as inputs. Results:

- All input files were successfully converted into a uniform text format and completed the ranking and unranking processes required for the Plausibly Deniable SSD without compromising data integrity.

- Latency for key functions in the system: load_text_to_gmp: $\tilde{0}.003$ seconds get_rank_from: $\tilde{0}.000023$ seconds get_permutation: $\tilde{0}.000098$ seconds Total execution time: 5.655267 seconds

Analysis and Discussion

Computational Efficiency and Multi-threading: Our multi-threading approach leverages modern CPUs' concurrent processing capabilities to speed up file conversion. The thread pool architecture reduces

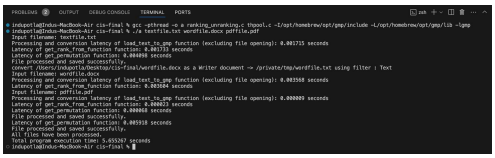

Figure 1: Execution

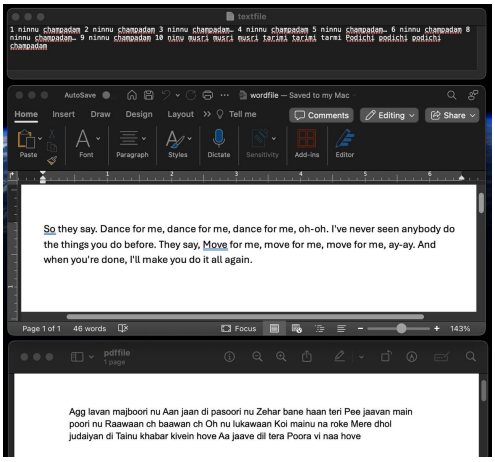

Figure 2: Input files

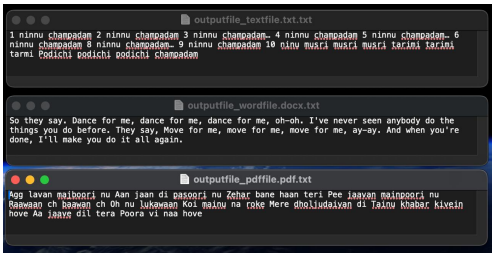

Figure 3: Output Files

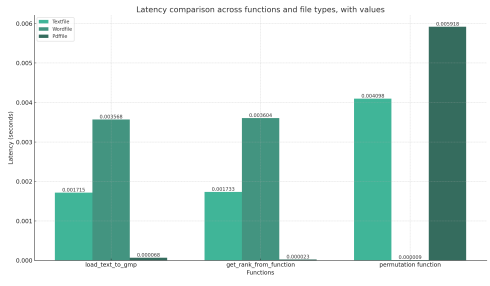

Figure 4: Latency Comparision

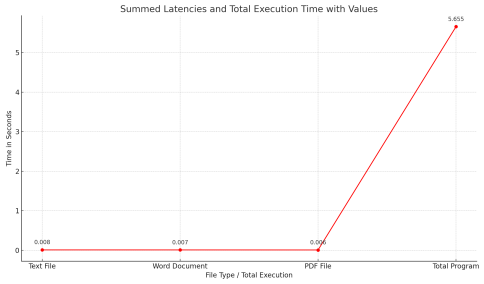

Figure 5: Summed Latency

overhead by minimizing thread creation, mirroring the efficiency of advanced systems that utilize concurrent processing for complex tasks.

Security and Reliability in Plausibly Deniable SSDs: Our SSD's security hinges on complex data layout permutation, scrambling data to thwart unauthorized decryption attempts, thus fortifying protection against intruders. Additionally, our system integrates robust error handling mechanisms, swiftly addressing conversion failures to maintain integrity and reliability.

Precise System Performance Across File Types: Our system demonstrates rapid response and precision in processing different file formats, evidenced by low summed latencies and swift total execution. Detailed latency metrics across critical functions further highlight our capacity to maintain operational integrity with minimal delay, ensuring high efficiency and security during complex data handling tasks.

Limitations and Areas for Enhancement: Our system currently relies on external utilities for file conversions, a notable limitation. Future improvements will focus on integrating native conversion algorithms to lessen this dependency and enhance security. Additionally, dynamically allocating worker threads based on system load could optimize resource utilization.

Future Work: The aim is to refine permutation algorithms for dynamic adaptation to input data, bolstering security, and integrate encryption pre-permutation for enhanced protection. The research will rigorously test against varied attack vectors, especially those targeting plausibly deniable encryption schemes.

References

[1] Chen Chen, Anrin Chakraborti, and Radu Sion. {PEARL}: Plausibly deniable flash translation layer using {WOM} coding. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1109–1126, 2021.

[2] Alexei Czeskis, David J St Hilaire, Karl Koscher, Steven D Gribble, Tadayoshi Kohno, and Bruce Schneier. Defeating encrypted and deniable file systems: Truecrypt v5. 1a and the case of the tattling os and applications. In *HotSec*, 2008.

[3] Clemens Fruhwirth. New methods in hard disk encryption. 2005.

[4] Tanmaya Mishra, Thidapat Chantem, and Ryan Gerdes. Survey of control-flow integrity techniques for real-time embedded systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 21(4):1–32, 2022.

[5] Wendy Myrvold and Frank Ruskey. Ranking and unranking permutations in linear time. *Information Processing Letters*, 79(6):281–284, 2001.

[6] Emil Stefanov, Marten van Dijk, Elaine Shi, T-H Hubert Chan, Christopher Fletcher, Ling Ren, Xiangyao Yu, and Srinivas Devadas. Path oram: an extremely simple oblivious ram protocol. *Journal of the ACM (JACM)*, 65(4):1–26, 2018.

[7] Michael Wei, Laura Grupp, Frederick E Spada, and Steven Swanson. Reliably erasing data from {flash-based} solid state drives. In *9th USENIX Conference on File and Storage Technologies (FAST 11)*, 2011.