

# DEDUCT: A SECURE DEDUPLICATION OF TEXTUAL DATA IN CLOUD/ SERVER ENVIRONMENTS

Kali.Naga Prasad<sup>1</sup>, Mr.S.Muni Kumar <sup>2</sup>

<sup>1</sup>student, Mca 2<sup>nd</sup> Year Kmmips, Tirupati, Affiliated To S.V. University, Tirupati, A.P, India

<sup>2</sup>Associate professor, Dept Of Mca, Kmmips, Tirupati, Affiliated To S.V. University, Tirupati, A.P, India

\*\*\*

**ABSTRACT** - The exponential growth of textual data in Vision-and-Language Navigation tasks poses significant challenges for data management in large-scale storage systems. Data deduplication has emerged as a practical strategy for data reduction in large-scale storage systems; however, it has also raised security concerns. This paper introduces DEDUCT, an innovative data deduplication method for textual data. DEDUCT employs a hybrid approach that combines cloud-side and client-side deduplication mechanisms to achieve high compression rates while maintaining data security. DEDUCT's lightweight preprocessing and client-side deduplication make it suitable for resource-constrained devices like IoT devices. It has also been designed to resist side-channel attacks. Experimental evaluations on the Touchdown dataset, consisting of human-written navigation instructions for routes, demonstrate the effectiveness of DEDUCT. It achieves compression rates of nearly 66%, significantly reducing storage requirements while preserving the confidentiality of textual data. This substantial reduction in storage demands can lead to significant cost savings and improved efficiency in large-scale data management systems.

**Key Words:** Cloud service provider, compression, secure data deduplication, textual data deduplication

## 1. INTRODUCTION

Vision-and-Language Navigation (VLN) tasks are becoming increasingly important due to their significant impact on advancing autonomous vehicles and intelligent systems. VLN technology empowers agents to navigate real world environments, enhancing human-robot interactions and safeguarding safety in autonomous vehicle operations. Beyond navigation, VLN applications extend to diverse domains, including robotics, virtual assistants, and smart homes, making human-machine interactions more intuitive and user-friendly. The significance of textual data in VLN cannot be overstated, as it is the foundation for communication between humans and autonomous agents. Users convey detailed navigational commands through natural language instructions, and autonomous systems rely heavily on the accurate interpretation and execution of these textual directives. Efficient data management has become critical to meet the increasing demands of VLN and its associated applications. Data deduplication is a highly

effective technique for reducing storage space consumption by eliminating the need for storing identical files or data blocks multiple times. Instead, only one copy of each unique data is stored, and references are used to point to the original copy. This method is particularly beneficial in cloud environments where vast amounts of data are typically stored. In backup applications, deduplication can reduce storage needs by up to 90–95% while in standard file systems, it can lead to a reduction of up to 68%. There are three main categories of data deduplication techniques based on granularity file level, fixed-size block, and variable-sized block. File-level deduplication finds and removes entire duplicate files. Fixed-size block deduplication divides a file into fixed-size blocks and eliminates duplicate blocks. Variable-sized block deduplication utilizes various sizes of chunks to identify redundant data, but it may create more metadata and lead to hash collisions. Block level deduplication is typically more efficient as it can detect duplicates even if they are stored across different files or portions of the storage system. Deduplication techniques can also be categorized based on place: server-based and client-based. Server based deduplication identifies and eliminates duplicate data on the server. Server-based deduplication eliminates the need for users to perform deduplication tasks locally. However, server-side deduplication may only partially mitigate communication overhead. On the other hand, client-side deduplication takes place on the user's device before uploading data to the cloud. It involves collaboration between the client and server to find redundant data. This can significantly reduce bandwidth consumption by sending only unique data. However, client-side deduplication raises concerns regarding side-channel attacks and data leakage. Finally, deduplication can be classified based on time: inline and offline. Inline deduplication eliminates duplicate data before or as it is being stored. Offline deduplication deals with deduplication after data is stored on a storage device. Classic Deduplication (CD) methods primarily focus on identifying and removing duplicate files, which can lead to inefficient storage when files share similar content but are not identical. Generalized Deduplication (GD) has emerged as a more comprehensive approach to address this limitation. GD expands the scope of traditional methods by recognizing and eliminating nearly identical or similar data chunks. This reduces storage requirements significantly, eliminates data redundancy, and improves data

management efficiency. Textual data deduplication, while receiving less attention than other data types, has become increasingly crucial due to the exponential growth in textual information generation. There may be more efficient approaches than cloud-based solutions, and clients can significantly contribute by performing initial data preprocessing. Nevertheless, existing client-side deduplication methods face security challenges, particularly in cross-user deduplication scenarios. The risk of side-channel attacks, where unauthorized access to files uploaded by other users is possible, underscores the need for an enhanced system model for textual data deduplication. Our motivation stems from addressing the limitations of current client-side deduplication approaches.

stored in each user's outsourced data. Yang et al. proposed a data deduplication scheme using Bone-Goh-Nissim cryptosystem and bloom filters. Their approach aims to achieve tag client and cloud-side ( $\tau < 6$ ).

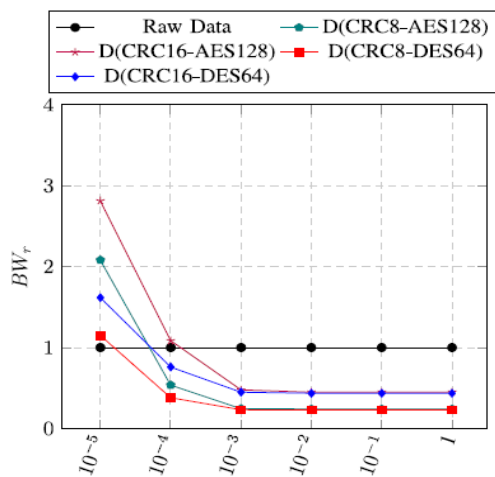


FIGURE1. Client's storage Ratio

## 2. RELATED WORKS

This section investigates related works in four categories: Deduplication and privacy, popularity-based encrypted deduplication, mitigating side-channel risks in deduplication systems, and generalized reduplication and privacy concerns.

### 2.1. DE DUPLICATION AND PRIVACY

Classic data deduplication methods have raised concerns about data privacy. Traditional encryption algorithms pose challenges for deduplication since they make encrypted data indistinguishable from random bits, making it difficult to identify identical messages. Convergent encryption (CE) was introduced as a solution to achieve encrypted deduplication by deriving encryption keys from the data content. This enables deterministic encryption and ensures that identical messages produce identical ciphertexts. However, CE only offers confidentiality guarantees for unpredictable data, leaving predictable data vulnerable to offline brute-force attacks. Additionally, given sufficient time and resources, the cloud service provider could break the encryption and gain access to the exact information

Client's Storage Ratio	CRC	Encryption	$G_r$		$BW_r$	
			$\tau < 2$	$\tau < 6$	$\tau < 2$	$\tau < 6$
0.0001	CRC-8	AES-128	0.3492	0.3495	0.537	0.5350
0.001	CRC-8	AES-128	0.3492	0.3495	0.246	0.2464
0.01	CRC-8	AES-128	0.3492	0.3495	0.242	0.2427
0.1	CRC-8	AES-128	0.3492	0.3495	0.242	0.2427
0.0001	CRC-16	AES-128	0.3509	0.3511	1.0848	1.0885
0.001	CRC-16	AES-128	0.3509	0.3511	0.4764	0.4760
0.01	CRC-16	AES-128	0.3509	0.3511	0.4494	0.4499
0.1	CRC-16	AES-128	0.3509	0.3511	0.4494	0.4499

TABLE1. The effect of the threshold value

Consistency, confidentiality, access control, and resistance to brute-force attacks in cloud storage. However, latency remains a concern in its implementation. To further address the challenges of deduplication and privacy, introduces a secure data-sharing scheme that integrates data deduplication and sensitive information hiding. Wildcard substitution is employed in electronic medical records to enhance privacy and deduplication efficiency. Moreover, multiple key servers are utilized to mitigate the risk of brute-force attacks and single-point-of-failure scenarios.

Recent advancements in the data deduplication have focused on optimizing Maximum Likelihood Estimation (MLE) specifically for deduplicating file chunks rather than entire files, enhancing deduplication efficiency. Password-Authenticated Key Exchange (PAKE)-based protocols have also been introduced to facilitate secure key sharing and determination on the client-side. Alternative privacy-enhancing mechanisms, such as Multi-Key Revealing Encryption (MKRE), have been proposed to address the challenges of deduplication while preserving privacy. By using MKRE, the encryption scheme becomes more resistant to attacks attempting to break the encryption. However, the security claims of MKRE have only been proven in the programmable random oracle model, which may not accurately represent real-world scenarios. A secure cloud auditing scheme that supports data deduplication with efficient ownership management was proposed by Wang et al. in. This scheme employs a lazy update strategy to efficiently manage data ownership changes. The cloud maintains a flag determining whether an update is necessary, effectively reducing update frequency and computation overhead.

## 3. POPULARITY-BASED ENCRYPTED DEDUPLICATION

Data deduplication systems often face a trade-off between data security and storage efficiency. To address this challenge, researchers have explored popularity-based encrypted deduplication schemes. These schemes treat

popular data, such as widely shared songs or movies, differently from unpopular data, like medical records or scientific research results. In popularity-based encrypted deduplication schemes, only popular data is encrypted using convergent encryption and subjected to deduplication, while unpopular data is randomly encrypted to ensure semantic security. Existing schemes often rely on a trusted third party to store deterministic tags that record data popularity. However, this reliance on a trusted third party introduces a security vulnerability. If the trusted third party is compromised, the deterministic tags become accessible, enabling offline brute-force attacks to reveal data content. Reference utilizes a double-layer encryption approach for less popular data, allowing Cloud Storage to verify the correctness of inner-layer convergent ciphertext. The outer-layer encryption employs PRP, symmetric encryption, and XOR, leading to reduced computational costs for users and cloud storage.

#### 4. MITIGATING SIDE-CHANNEL RISKS

Harnik et al. were the first to propose a method to safeguard against side-channel attacks by malicious users. They introduced the concept of employing a random threshold for uploads to prevent attackers from inferring the presence or absence of a file on the cloud. The server randomly selects a threshold for each data chunk, and client-side deduplication is only enabled when the number of file uploads surpasses this threshold. This approach prevents an attacker from determining the non-existence of a file. Alternative methods involve randomization of thresholds during operation or their determination based on game theoretic optimization. Armknecht et al. studied the trade-offs between security and efficiency, proposing a randomized response technique to preserve privacy. Another approach, known as ZEUS, necessitates the client simultaneously request the storage of two chunks. The server's response to these requests is deterministic, meaning that if one or both chunks already exist, the user will be prompted to upload a combination of the two. While this indicates the server's possession of at least one chunk, it does not reveal which one specifically. Building upon this concept, RARE further enhances protection by randomly requesting the user to upload either both chunks or a combination of the two whenever at least one chunk is detected on the server. As a result, attackers face more significant challenges in accurately determining whether a particular chunk is stored in the cloud. To further address client-side deduplication, CIDER extends the principles introduced by RARE to encompass the simultaneous storage of more than two chunks. This method enables users to request the storage of multiple chunks simultaneously. Before enabling client-side deduplication, users must include two fingerprints in each storage request to facilitate proper randomization of responses and ensure that no individual chunk can undergo client-side deduplication.

#### 5. CONCLUSIONS

This paper presents DEDUCT, a textual deduplication technique that leverages generalized deduplication and client-side preprocessing to significantly enhance cloud storage efficiency and data security. DEDUCT demonstrates notable improvements in these key areas compared to existing state-of-the-art methods. DEDUCT achieves a compression ratio of 66% which translates to direct cost savings and improved scalability for cloud storage solutions, offering increased capacity and reduced financial burden. Moreover, DEDUCT's design is well-suited for resource-constrained devices commonly found in the Internet of Things (IoT). This adaptability addresses crucial needs in resource-limited environments where efficient data handling is critical. While the evaluation focused on the Touchdown dataset, DEDUCT's applicability extends to broader domains. Its strengths in efficiently deduplicating large textual datasets make it highly relevant to IoT, mobile, and embedded systems, where storage and bandwidth are often limited. DEDUCT's flexibility and resource-friendly approach offer valuable solutions for these areas.

#### 6. REFERENCES

- [1] P. Anderson, Q. Wu, D. Tenney, J. Bruce, M. Johnson, N. Sunehra, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in Proc. IEEE/CVF Conf. Compute. Vis. Pattern Recognit., Jun. 2018, pp. 3674–3683, doi: 10.1109/CVPR.2018.00387.
- [2] W. Xia, H. Jiang, D. Feng, F. Douglas, P. Shalane, Y. Hua, M. Fu, Y. Zhang, and Y. Zhou, "A comprehensive study of the past, present, and future of data deduplication," Proc. IEEE, vol. 104, no. 9, pp. 1681–1710, Sep. 2016, doi: 10.1109/JPROC.2016.2571298.
- [3] P. Prajapati and P. Shah, "A review on secure data deduplication: Cloud storage security issue," J. King Saud Univ. Compute. Inf. Sci., vol. 34, no. 7, pp. 3996–4007, Jul. 2022, Doi: 10.1016/j.jksuci.2020.10.021.
- [4] D. T. Meyer and W. J. Bolosky, "A study of practical deduplication," ACM Trans. Storage, vol. 7, no. 4, pp. 1–20, Jan. 2012, Doi: 10.1145/2078861.2078864.
- [5] Opened. (2023). Opened. Accessed: Aug. 6, 2023. [Online]. Available: <http://openedup.org/>
- [6] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupes: Server-Aided encryption for deduplicated storage," in Proc. 22nd USENIX Secure. Symp. (USENIX Secure.), 2013, pp. 179–194.
- [7] Liu, N. Asokan, and B. Pinkas, "Secure deduplication of encrypted data without additional independent servers," in

- Proc. ACM SIGSAC Conf., Oct. 2015, pp. 874–885, Doi: 10.1145/2810103.2813623.
- [8] K. Ghassemi, P. Pahlavan, and D. E. Lucani, “Deduplication of textual data by NLP approaches,” in Proc. IEEE 97th Veh. Technol. Conf. (VTC-Spring), Florence, Italy, Jun. 2023, pp. 1–6, Doi: 10.1109/vtc2023-spring57618.2023.10199538.
- [9] K. Jin and E. L. Miller, “The effectiveness of deduplication on virtual machine disk images,” in Proc. Israeli Exp. Syst. Conf., May 2009, pp. 1–12, Doi: 10.1145/1534530.1534540.
- [10] S. Lee and D. Choi, “Privacy-preserving cross-user source-based data deduplication in cloud storage,” in Proc. Int. Conf. ICT Converge. (ICTC), Oct. 2012, pp. 329–330, doi: 10.1109/ICTC.2012.6386851.
- [11] B. Wang, W. Lou, and Y. T. Hou, “Modeling the side-channel attacks in data deduplication with game theory,” in Proc. IEEE Conf. Commun. Netw. Secur. (CNS), Sep. 2015, pp. 200–208, doi: 10.1109/CNS.2015.7346829.
- [12] F. Armknecht, C. Boyd, G. T. Davies, K. Justen, and M. Torani, “Side channels in deduplication,” in Proc. ACM Asia Conf. Compute. Commun. Secur., Apr. 2017, pp. 266–274, doi: 10.1145/3052973.3053019.
- [13] H. Chen, A. Suhr, D. Misra, N. Snaveley, and Y. Artzi, “TOUCHDOWN: Natural language navigation and spatial reasoning in visual street environments,” in Proc. IEEE/CVF Conf. Compute. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12530–12539, doi: 10.1109/CVPR.2019.01282.
- [14] R. Vestergaard, Q. Zhang, and D. E. Lucani, “Generalized deduplication: Bounds, convergence, and asymptotic properties,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2019, pp. 1–6, doi: 10.1109/GLOBECOM38437.2019.9014012.
- [15] H. Sehat, E. Pagnin, and D. E. Lucani, “Yggdrasil: Privacy-aware dual deduplication in multi client settings,” in Proc. IEEE Int. Conf. Commun., Jun. 2021, pp. 1–6, doi: 10.1109/ICC42927.2021.9500816.
- [16] L. Nielsen and D. E. Lucani, “Hekate a tool for gauging data deduplication performance,” in Proc. IEEE 6th Int. Conf. Smart Cloud (Smart Cloud), Nov. 2021, pp. 67–72, Doi: 10.1109/SmartCloud52277.2021.00019.
- [17] Z. Pooran Ian, K.-C. Chen, C.-M. Yu, and M. Conti, “RARE: Defeating side channels based on data-deduplication in cloud storage,” in Proc. IEEE Conf. Compute. Commun. Workshops (INFOCOM WKSHPS), Apr. 2018, pp. 444–449, doi: 10.1109/INFOCOMW.2018.8406888.
- [18] R. Vestergaard, Q. Zhang, and D. E. Lucani, “CIDER: A low overhead approach to privacy aware client-side deduplication,” in Proc. GLOBECOM IEEE Global Commun. Conf., Dec. 2020, pp. 1–6, doi: 10.1109/GLOBECOM42002.2020.9348272.
- [19] J. R. Douceur, A. Adya, W. J. Bolosky, P. Simon, and M. Theimer, “Reclaiming space from duplicate files in a serverless distributed file system,” in Proc. Int. Conf. Diatribe. Compute. Syst., Jun. 2003, pp.