www.ijaceeonline.com ISSN: 2456 - 3935

International Journal of Advances in Computer and Electronics Engineering
Volume: 8 Issue: 6, June 2023, pp. 1 – 8

Secured Data Storage in Cloud Environment Using Deduplication

Muthurajkumar S
Associate Professor, Department of Computer Technology,
MIT Campus, Anna University, Chennai – 600044, Tamil Nadu, India
Email: muthurajkumarss@gmail.com

Karthikeyan V
Student, Department of Computer Technology,
MIT Campus, Anna University, Chennai – 600044, Tamil Nadu, India
Email: karthik11072000@gmail.com

Kowsi N
Student, Department of Computer Technology,
MIT Campus, Anna University, Chennai – 600044, Tamil Nadu, India
Email: kowsinatarajan2001@gmail.com

Barath S
Student, Department of Computer Technology,
MIT Campus, Anna University, Chennai – 600044, Tamil Nadu, India
Email: barathdon82@gmail.com

Abstract: *Customer-side data deduplication attains decrease in storage as well as bandwidth in cloud storage, which results in decreased operating cost and more customer satisfaction. But the deduplication checking ushers the attacker to check the file's extant status by creating a new side channel. File existence information is revealed from the binary response sent for the deduplication checks. By exploiting this behavior further attacks can be launched that may result in revealing sensitive file contents. While the existing works provides a weak privacy, we propose a Randomized Response approach to achieve a solid privacy. In this paper, we establish a result based on a Randomized response technique, which prevents the attacker from obtaining existence of the chunks from the duplicate scans. With minimal modification to the ordinary duplicated mechanism the above approach can be implemented which in turn yields us additional privacy and performance guarantee.*

Keyword: *Cloud Computing; Data Storage; Deduplication; Cloud Security.*

1. INTRODUCTION

Cloud storage services like OneDrive and iCloud have revolutionized the way we store and access our data. They promise limitless storage capacity and seamless synchronization across multiple devices. However, this convenience comes with some drawbacks, including increased power consumption, complex data management, and the wastage of network resources due to data duplication. To address these issues, cloud storage providers have turned to data deduplication technology, a crucial tool in reducing redundant data copies.

Data deduplication is a technique that identifies and eliminates duplicate data, thus optimizing storage efficiency. Cloud storage providers, especially those offering commercial services, employ various strategies to maximize deduplication advantages. One prominent approach is cross-user consumer-side data deduplication with a fixed chunk size. This method treats the cloud storage as a virtual shared disk among users, deduplicating files across multiple users, thus maximizing deduplication opportunities.

Consumer-side data deduplication is enabled when users submit the hash of a file as a duplicate scan request (DS request). Cryptographic hash functions are used to check a file's existence, and a binary duplication scan response is returned to the user, preventing the upload of duplicate files. However, this DS response can potentially serve as a new side channel for privacy violations, creating a challenging dilemma in developing countermeasures.

Cite this paper:

Muthurajkumar S, Karthikeyan V, Kowsi N, and Barath S, "Secured Data Storage in Cloud Environment Using Deduplication", International Journal of Advances in Computer and Electronics Engineering, Vol. 8, No. 6, pp. 1-8, June 2023.

Developing a countermeasure to address this privacy concern presents conflicting requirements. On one hand, it is essential to determine a predetermined response from the cloud regarding the file upload requirement, which might leak privacy. On the other hand, any deterministic response can be interpreted as a privacy breach.

To address this privacy challenge, a proposed solution introduces randomized response in the data deduplication process. This approach forces the cloud user to upload two chunks simultaneously for deduplication requests. The randomized response provided by the cloud is carefully designed to preserve both the benefits of deduplication gain and minimize privacy leakage. Key Aspects of the Proposed Solution:

1. **Preserving Privacy**: The randomized response mechanism ensures that user privacy is maintained while still allowing for effective data deduplication. By introducing randomness into the response, it becomes exceedingly difficult for malicious actors to exploit the DS response as a privacy violation.
2. **Efficiency:** The proposed solution maintains the advantages of data deduplication, leading to considerable bandwidth and storage savings. This results in lower communication costs for both cloud service providers and users.

The remaining of this research is organized as follow: Section 2 provides the related work of the proposed system. Section 3 explains the proposed system architecture and modules. Section 4 explains the necessary details of the proposed mechanism. Section 5 provides the results and performance analysis of the proposed system. Finally, Section 6 provides the conclusion and future work.

2. LITERATURE SURVEY

This section explains the cloud storage environments, data deduplication in cloud and privacy challenges.

Hyungjune Shin et al. [1] proposed an approach that does secure data deduplication using the Message-Locked Encryption (MLE) which overcomes the result of different ciphertexts in encryption even when the original messages are the same. MLE is suggested as a solution to this problem and shows that it is secure even with an unexpected message set. However, MLE techniques are susceptible to brute force attacks if the message set is predictable.

Waghmare, Milind B., and Suhasini V. Padwekar [2] proposed a data deduplication compression technique that maintains unique data while removing duplicates from the cloud in order to optimise storage and reduce bandwidth usage. While doing data deduplication in the cloud, the preserving of data privacy component is disregarded. There are many literature-survey about the cloud data storage.

Sarada Prasad Gochhayat, Chia-Mu Yu, Mauro Conti [3] proposed techniques for privacy-conscious deduplication called ZEUS and ZEUS+ which are proposed to ensure two-sided privacy with no extra requirement for hardware. By using a framework called zero knowledge about the deduplication feedback, the exploiter is prevented to learn the results of Duplicate Scans regarding their existence. The proposed work preserves the deduplication benefits and also eliminates data deduplication-based side channel. Though ZEUS is more efficient when compared cost wise it provides a weaker privacy and on the other hand ZEUS+ provides a guaranteed higher privacy but with an increased communication cost. There are many literature-survey about the cloud data security [11]-[15].

D.Viji, Dr.S.Revathy [4] classified the data deduplication techniques and drawbacks of each basic storage technology. Current research questions and unclear design elements of primary storage deduplication algorithms are acknowledged and examined. While the backup storage system mostly contains immutable data, which is data that is never altered regularly, the primary storage system contains mutable data, which is data that is frequently added, deleted, or updated.

Dhanaraj Suresh Patil, R. V. Mane, V.R.Ghorpade [5] discussed the multiple cloud vendors (cloud-of-clouds). To distribute data There are replication and erasure code schemes. Data deduplication is a technique used to eliminate cloud system redundancy and prevent data duplication. Replication produces several copies, expanding the cloud system's storage capacity. Using this approach to update a file is more difficult than using the erasure code scheme. There are many literature-survey about the cloud data environment [15-20].

Armknecht Frederik et al [6] discusses the security concerns of client-side deduplication, which removes redundant copies of files or data blocks stored on the cloud, but introduces potential side-channel attacks that can leak user information. The authors provide formal definitions for deduplication strategies and their security, and propose a criterion for designing effective strategies. They also prove a bound that characterizes the necessary trade-off between security and efficiency. They review existing side-channel attacks on cloud storage and countermeasures and discusses their security model and optimality of defenses.

Bellare Mihir et al. [7] formalized a cryptographic primitive called Message-Locked Encryption (MLE), where the key used for encryption and decryption is derived from the message. MLE enables secure deduplication, which is a goal pursued by cloud storage providers. They provided definitions for privacy and tag consistency, a form of integrity, in MLE schemes.

They conducted ROM security analyses of deployed MLE schemes and explored connections with deterministic encryption, hash functions, and the sample-then-extract paradigm. The paper highlighted the widespread deployment and application of MLE schemes for secure deduplication but noted the lack of a theoretical treatment. It also mentioned the vulnerability of currently deployed schemes to duplicate faking attacks.

Chen, Rongmao et al. [9] proposed a new approach called block-level message-locked encryption (BL-MLE) to achieve more efficient deduplication for large files in secure cloud storage. BL-MLE enables file-level and block-level deduplication, block key management, and proof of ownership simultaneously using a small set of metadata. The BL-MLE scheme can also be extended to support proof of storage, making it multi-purpose for secure cloud storage. Their research formalizes the security definitions for BL-MLE schemes, considering the unpredictability and privacy of block encryption.

Douceur, John R. et al [10] presented a mechanism to reclaim space from duplicate files in a distributed file system by using convergent encryption and a Self-Arranging Lossy Associative Database (SALAD). The mechanism allows duplicate files to be coalesced into a single file, even if they are encrypted with different users' keys, and aggregates file content and location information in a decentralized and fault-tolerant manner. Large-scale simulation experiments show that the duplicate-file coalescing system is scalable, highly effective, and fault-tolerant. By setting a threshold on the minimum file size eligible for coalescing, the message traffic and fingerprint database sizes can be substantially reduced. A target redundancy factor of L 2.5 achieves nearly all possible space reclamation.

Hovhannisyan, Hermine et al [11] investigates the threat of establishing a covert channel over cloud storage services, specifically focusing on data deduplication schemes. The authors design a more powerful deduplication-based covert channel that can transmit a complete message, using a synchronization scheme and a novel coding scheme. They implement the covert channel and conduct extensive experiments in different cloud storage systems to evaluate its effectiveness. Their research highlights the existence of a more severe security threat in cloud storage services due to the potential for covert communication.

Liu, Jian et al. [12] presented a secure cross-user deduplication scheme that supported client-side encryption without the need for additional independent servers. The scheme was based on using a PAKE (password authenticated key exchange) protocol and provided better security guarantees compared to previous efforts. The effectiveness and efficiency of the scheme were demonstrated through simulations using realistic datasets and an implementation. Their research also discussed different deduplication strategies, such as file-level and block-level deduplication, and extended the basic protocol to address various issues.

Li, Jin et al. [13] addressed the challenge of achieving efficient and reliable key management in secure deduplication in cloud storage. It introduces a baseline approach where each user holds an independent master key for encrypting the convergent keys and proposes a new construction called Dekey to mitigate the key management overhead and provide fault tolerance guarantees for key management. They present the system model and security requirements of deduplication, and discusses the limitations of the baseline approach in key management. It also defines the cryptographic primitives used in secure deduplication, including convergent encryption, and presents the Dekey construction as a solution to the key management problem in secure deduplication.

Lee, Seungkwang, and Dooho Choi [14] proposed a new cross-user source-based deduplication method that significantly enhances security in cloud storage. The authors revisit a previous solution by Harnik et al. and show that it does not provide sufficient security against side channel information leakage. They improve upon it and demonstrate that their proposed solution offers outstanding security compared to existing alternatives. Their research emphasizes the importance of a large "d" value, which steeply decreases the probability of information leakage through deduplication. The proposed solution ensures that the attacker cannot determine whether a copy of a specific data item was previously uploaded, except in cases where deduplication occurs after uploading exactly "d" copies of the item. However, the probability of this event occurring is negligible with larger "d" values.

May, Alexander [15] discussed the challenge of handling huge data spaces in cryptanalysis and the need for technical solutions that traverse the search space without storing elements. It focused on the subset sum problem, which involves finding a subset of integers that sums to a given target. The research surveyed memory-less algorithms for solving the subset sum problem, including the van Oorschot-Wiener technique, the representation technique, and the two-layered collision finding algorithm. These algorithms achieved faster running times than the trivial memory-less algorithm, with the best algorithm achieving a running time of 2^(0.65n).

Ramanathan, Saravanan et al [16] reviewed the current state-of-the-art resource allocation techniques for the cloud continuum, with a focus on time-sensitive applications. It highlights the challenges in managing computation and communication resources in the cloud and edge platform to meet performance and latency guarantees. Their research presented a taxonomy to classify existing literature on resource allocation and identifies potential research gaps in this area. The design of resource allocation techniques is complex due to the heterogeneity of cloud resources, variable cost structure, and unpredictable workload patterns.

Godhrawala, Husain et al. [17] focused on the problem of resource allocation in cloud computing, particularly in a multitenant environment. Existing approaches to resource allocation were found to be inadequate in optimizing revenue without negatively impacting resource utilization rates. They proposed a solution for a pool of cloud service providers using a hybrid method, which involved optimizing fair prices of resources through stock market-based technical analysis and optimizing resource utilization using Stackelberg output volume. The findings of the paper provided insights into improving resource allocation in cloud computing, leading to better results for both service providers and subscribers.

Würdemann, Nick [18] proposed a Distributed Synthesi problem of automatically generating correct controllers for individual agents in a distributed system, and Petri games model this problem by a game between two teams of players on a Petri net structure. Petri games can be solved by reducing them to a two-player game under certain restrictions. The concept of symmetries in Petri nets is closely related to high-level representations of Petri games. Applying symmetries to the states in the two-player game leads to a significant reduction in the state space.

Berani Abdelwahab, Erzana, and Martin Fränzle [20], Delays in feedback dynamics of coupled dynamical systems were a common occurrence, particularly in embedded control systems where the physical plant and the controller interacted through digital networks. Formal analysis often relied on simplified, delay-free substitute models, which neglected the adverse impact of delay on control performance. Their research demonstrated that for continuous systems like delay differential equations, a significant portion of the complexity induced by delay could be effectively reduced by adding natural constraints to the model of the delayed feedback channel. These constraints included transporting a band-limited signal and implementing a non-punctual, distributed delay. The reduction in complexity was achieved through a sampling approach, which was applicable when the specified conditions on the feedback were met.

Secured data storage in a cloud environment using deduplication is an important aspect of modern cloud computing and data management. Let's break down this concept step by step: (1) Secured Data Storage: Secured data storage refers to the practice of storing data in a manner that ensures its confidentiality, integrity, and availability. In a cloud environment, this involves implementing robust security measures to protect data from unauthorized access, tampering, and data loss. (2) Cloud Environment: Cloud environments provide on-demand access to computing resources and services over the internet. They are characterized by scalability, flexibility, and cost-effectiveness. Cloud providers like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud offer various storage options. (3) Data Deduplication: Data deduplication is a technique used to reduce storage space and bandwidth requirements by eliminating redundant copies of data. Deduplication identifies and stores unique data once, while subsequent copies only reference the original data. This helps in optimizing storage utilization.

3. SECURED DATA DEDUPLICATION ARCHITECTURE

When a file is needed to be uploaded in the cloud, instead of uploading it as a whole it is broken into chunks of preset size. The Chunked Uploading technique provides a way to easily upload large files into the cloud by chunking them into segments of small parts that can be uploaded individually. So instead of uploading an entire application only a single part needs to be uploaded.

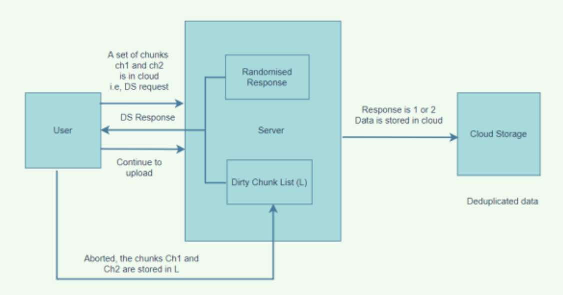

Figure 1 Secured Data Deduplication Architecture.

Two of the chunks are compared simultaneously as opposed to just one. Since the naive implementation of double chunk comparison can only be viewed as performing the standard duplicate scan twice, it is ineffective in preventing privacy leaks. It becomes significantly more difficult for the attacker to distinguish between the chunk's existence and nonexistence when this strategy is used in conjunction with a randomized response method because of the uncertainty in the DS answer.

By controlling numerous Sybil accounts, the attacker using Sybil can carry out independent duplicate scans. To prevent abuse from free cloud accounts, a requested piece of chunk of a file which is not uploaded is labelled as a dirty piece of chunk or dirty chunk. For every request with dirty chunks, a DS response of 2 is delivered automatically. Because of the possibility for attacker exploitation of dirty chunks, all subsequent duplicate scans pertaining to dirty chunks will never cause the deduplication to occur. The aforementioned guideline is put into practice by using a dirty chunk list (L) that contains every dirty chunk hash values. The cloud checks to see if the hash is present in L whenever a DS request is received. If that's the case, the cloud returns 2, else it provides the correct value in accordance with the DS response Table II.

4. PROPOSED SOLUTION

This section explains the necessary fundamental details of the proposed method.

4.1 Chunk Uploading

When a file is needed to be uploaded in the cloud, instead of uploading it as a whole it is broken into chunks of predefined size. The Chunked Upload technique provides a way to reliably upload large files into the cloud by chunking them into a sequence of parts that can be uploaded individually. So instead of uploading entire document only a single part needs to be uploaded.

TABLE II DS RESPONSE.

Chunk 1 (Ch1) Existence	Chunk 2 (Ch2) Existence	Response
0	0	-, -
0	1	-, +
1	0	+, -
1	1	+, +

4.2 Double Chunk Comparison with Randomized Response

Two pieces are compared simultaneously rather than just one. Because the natural double chunk comparison can only be viewed as performing the standard Duplicate Scan (DS) twice, it is ineffective in preventing privacy leaks (shown in Table I). This strategy makes it more challenging for the attacker to discern between a chunk's existence and nonexistence when used in conjunction with a randomized response mechanism due to the randomness in the DS response.

TABLE II DS RESPONSE FOR RANDOMIZED RESPONSE.

Chunk 1 (Ch1) Existence	Chunk 2 (Ch2) Existence	Response
0	0	2(x1)
0	1	1(x2)
1	0	1(x3)
1	1	1 or 2(x4)

In random response method, the values of x2, x3 and x4 takes a random value of 1 or 2 (shown in Table II). If neither piece (for example, ch1 and ch2) is in the cloud, the user must upload both parts separately. If not, the user must submit either two separate pieces or the exclusive-or (XOR) ch1⊕ ch2 of two chunks, each with the probability of 0.5. A chunk in possession can be used to derive another chunk in such a design.

4.3 Proposed Algorithm

```
Step 1: Take a file F which contains size of each
        chunks CS, dirty chunk list L
Step 2: Partition F into chunks Ch1, Ch2, Ch3, ...,
        Chn and set n1=n.
Step 3: If n! = CS, then user carries out padding
        to Chn
Step 4: If n is an odd number, then user selects a
        random piece of chunk as Cn+1
        and assigns n1=n+1
Step 5: User then perform a Duplicate Scan on
         <H(Chi), H(Chi+1)> for values of i=1, 3,
         ...., n1-1.
Step 6: If H(Chi) and H(Chi+1) do not belong to
        L, then cloud feedbacks 1 or 2 as in the
        table or else cloud feedbacks DS response
         as II.
Step 7: If the user receives back DS response as 1
        then user uploads Chi xor Chi+1 to the
        cloud and assuming the cloud not receive
        Chi xor Chi+1 then L = L U {Chi, Chi+1}
Step 8: If the user receives back DS response 2
        then user uploads Chi and Chi+1 in the
        cloud and assuming the cloud not receive
        Chi and Chi+1 then L = L U {Chi, Chi+1}
```

4.4 Dirty Chunk List

By using the Sybil accounts, the attacker might use duplicate scans on their own. We designate the requested chunk as a dirty chunk to prevent further misuse of free cloud accounts. The DS response for queries containing dirty chunks will thereafter be 2. The reasoning for this is that since filthy chunks could potentially be exploited by an attacker, deduplication will never be triggered by any duplicate scans that are relevant to dirty chunks. Maintaining a dirty chunk list (L) which contains every hash value of dirty piece of chunks will allow you to put the aforementioned policy of employing dirty chunks into practice. The cloud initially determines if the chunk hashes are present in L when the DS queries are received. If that's the case, the cloud returns 2, else it provides the correct value in accordance with the DS response table.

5. RESULT ANALYSIS

Here, we first briefly discuss the statistic we used to validate our approach across a range of chunk sizes before describing the dataset we utilized for analysis. Communication cost, the metric we employ in our approach, is quantity of the bits needed for the complete uploading of the chunk procedure, which includes the duplicate scan (i.e., the duplicate scan request and its feedback) and uploading of the chunks.

Additionally, we define a "dirty chunk" as a chunk for which no deduplication is implemented and for which none of the DS queries will result in deduplication. In other words, by continually sending DS requests for the same chunks, the attacker is prevented from learning the existence of those chunks. Therefore, the benefit of deduplication is compromised by the use of dirty chunk lists. As a result, each set of dirty chunks receives the identical set of assessments.

For our analysis, we have used the Enron Email Dataset (as shown in Figure 2). Regular users can also backup their emails to cloud storage using this dataset, which serves as the actual programme. We removed all of the dataset's files that were smaller than 5KB.

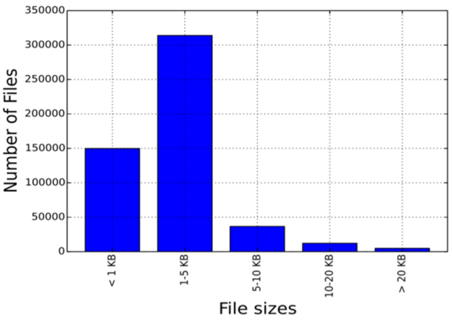

Figure 2 Statistics of Enron Email Dataset.

Then, 1000 files were randomly chosen and uploaded to the cloud. Then, we performed the duplicate scans on 200 randomly selected files and, if necessary, explicitly uploaded the chunks. Due to the fact that this can only be viewed as doing the Duplicate Scan twice, the naive implementation of just the double chunk uploading comparison is ineffective in stopping privacy leakage.

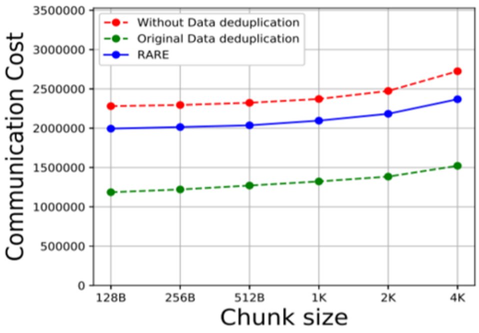

Figure 3 No Dirty Chunk.

The Randomized Response system is then applied to the Enron Email dataset with a range of chunk sizes, and the average communication cost is evaluated as a result. The communication cost versus chunk size for the original data deduplication, Randomized Response, and without data deduplication methods is shown in Figure 2, Figure 3, and Figure 4, respectively. The dirty chunk lists, 10% dirty chunks, and 25% dirty chunks are also taken into consideration. According to these calculations, the quantity of dirty chunks would increase as duplicate scan requests grew.

The Table III shows the transmission cost difference for the various file sizes between the initial data deduplication and without data deduplication. The minimal degrees of communication cost decrease.

TABLE III COMMUNICATION COST.

Chunk Size	Without Data Deduplication	Original Data Deduplication
128B	2500000	1900000
256B	2600000	2000000
512B	2700000	2100000
1K	2800000	2200000
2K	2900000	2300000
4K	3000000	2400000

The Table IV depicts a file storage level between the initial data deduplication and without data deduplication for the various file sizes. The file storage level improves the minimum percentage levels. Then, we randomly choose a predetermined amount of chunks to be dirty chunks in order to examine the effect of dirty chunk count on communication cost. We can observe that the more transmission expenses are incurred, the more dirty chunks are used. It's because the cloud cancels the deduplication for the request when it identifies any of the chunks in the duplicate scan request as dirty chunks, which results in higher communication costs.

TABLE IV FILE STORAGE LEVEL.

File Size	Without Data Deduplication (%)	Original Data Deduplication (%)
128B	73	76
256B	78	81
512B	81	84
1K	85	88
2K	86	89
4K	91	93

We can draw the conclusion that if no dirty chunk is used, users will occasionally experience unexpected disconnections, their vulnerability will be increased, and an attacker could be hesitant to set up a side channel based on deduplication. There would be a very small number of filthy bits in the cloud as a result. It is evident from the chart that initial data deduplication offers the cheapest communication cost compared to the other methodologies.

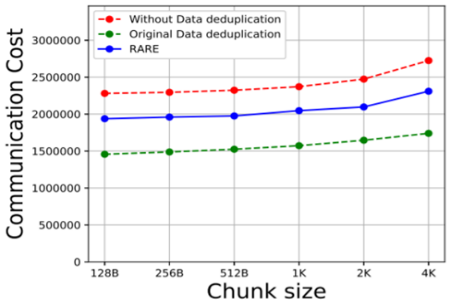

Figure 4 25% Dirty Chunk.

This is due to the fact that this method has a larger likelihood of data deduplication than the Randomized Response method and may reduce communication costs much more. Despite the low communication costs of this method, our algorithm, Randomized Response, provides the privacy of the absence and the weak presence. Furthermore, the difference in communication costs between the Randomized Response method and the original data duplication actually relies on the data element.

The comparison analysis of the proposed and current state of art method is shown in Table V.

Table IV explains that the proposed method is perform better than the current state-of-art methods. The proposed scheme is taking less complexity and offers a high-level security. Also, provides a real-world application in cloud deduplication technique.

TABLE IV COMPARISON ANALYSIS

Schemes	**[21]**	**[22]**	**[23]**	**Proposed Method**
Offline brute-force attack	No	Yes	Yes	Yes
Data validity	No	No	Yes	Yes
Global deduplication	No	Yes	Yes	Yes
Without additional servers	Yes	No	Yes	Yes
Real world application	No	No	No	Yes

6. CONCLUSION

A distinct data-based channel construction poses a genuine and serious danger to privacy because commercial cloud storage providers frequently use client-side data. In this research, we propose a method which prevents the exploiter from obtaining state of file information from repeated tests by using a randomized response structure. In addition to its privacy and speed guarantees, this approach's implementation necessitates a small modification of the traditional deduplication method, which results in decreased work. The data extraction method offers both privacies namely the existence and non-existence of privacy. The deduplication method saves a lot of bandwidth and storage thus providing lesser communication costs.

The future direction of secured data storage in a cloud environment using deduplication is likely to evolve in response to several emerging trends and challenges in the technology landscape. Here are some key directions and considerations for the future of this field:

1. **Enhanced Privacy-Preserving Deduplication Techniques**: As privacy concerns continue to grow, there will be a focus on developing advanced privacy-preserving deduplication techniques. These methods will further protect user data from potential privacy breaches while still benefiting from deduplication.
2. **Distributed and Edge Computing**: With the increasing adoption of edge computing and the proliferation of IoT devices, data storage and deduplication will extend beyond centralized cloud environments. Deduplication techniques will need to adapt to decentralized and edge computing infrastructures.
3. **Machine Learning and AI Integration**: Machine learning and AI algorithms can play a significant role in identifying redundant data and optimizing deduplication processes. Expect further integration of AI to enhance the efficiency of data deduplication.
4. **Blockchain for Data Integrity**: Blockchain technology may be employed to ensure the integrity of deduplicated data. Immutable and decentralized ledgers can be used to track and verify the authenticity of data throughout its lifecycle.

REFERENCES

[1] Shin, Hyungjune, Dongyoung Koo, Youngjoo Shin, and Junbeom Hur. "Privacy-preserving and updatable block-level data deduplication in cloud storage services." *In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 392-400. IEEE, 2018.

[2] Waghmare, Milind B., and Suhasini V. Padwekar. "Survey on techniques for authorized deduplication of encrypted data in cloud." In *2020 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-5. IEEE, 2020.

[3] Yu, Chia-Mu, Sarada Prasad Gochhayat, Mauro Conti, and Chun-Shien Lu. "Privacy aware data deduplication for side channel in cloud storage." *IEEE Transactions on Cloud Computing* 8, no. 2, pp. 597-609. 2018.

[4] SureshPatil, Dhanaraj, R. V. Mane, and V. R. Ghorpade. "Improving the availability and reducing redundancy using deduplication of cloud storage system." In *2017 International*

Conference on Computing, Communication, Control and Automation (ICCUBEA), pp. 1-5. IEEE, 2017.

[5] Viji, D. and Revathy, S., 2019, July. Various data deduplication techniques of primary storage. In *2019 International conference on communication and electronics systems (ICCES)*. pp. 322-327. IEEE. 2019.

[6] Armknecht, Frederik, Colin Boyd, Gareth T. Davies, Kristian Gjøsteen, and Mohsen Toorani. "Side channels in deduplication: Trade-offs between leakage and efficiency." In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 266-274. 2017.

[7] Bellare, Mihir, Sriram Keelveedhi, and Thomas Ristenpart. "Message-locked encryption and secure deduplication." In *Annual international conference on the theory and applications of cryptographic techniques*, pp. 296-312. 2013.

[8] Chen, Rongmao, Yi Mu, Guomin Yang, and Fuchun Guo. "BL-MLE: Block-level message-locked encryption for secure large file deduplication." *IEEE Transactions on Information Forensics and Security* 10, no. 12, pp. 2643-2652., 2015.

[9] Broder, Andrei, and Michael Mitzenmacher. "Network applications of bloom filters: A survey." *Internet mathematics* 1, no. 4, pp. 485-509. 2004.

[10] Douceur, John R., Atul Adya, William J. Bolosky, P. Simon, and Marvin Theimer. "Reclaiming space from duplicate files in a serverless distributed file system." In *Proceedings 22nd international conference on distributed computing systems*, pp. 617-624. IEEE, 2002.

[11] Hovhannisyan, Hermine, Kejie Lu, Rongwei Yang, Wen Qi, Jianping Wang, and Mi Wen. "A novel deduplication-based covert channel in cloud storage service." In *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6. IEEE, 2015.

[12] Liu, Jian, Nadarajah Asokan, and Benny Pinkas. "Secure deduplication of encrypted data without additional independent servers." In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 874-885. 2015.

[13] Li, Jin, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick PC Lee, and Wenjing Lou. "Secure deduplication with efficient and reliable convergent key management." *IEEE transactions on parallel and distributed systems* 25, no. 6, pp. 1615-1625, 2013.

[14] Lee, Seungkwang, and Dooho Choi. "Privacy-preserving cross-user source-based data deduplication in cloud storage." In *2012 International conference on ICT convergence (ICTC)*, pp. 329-330. IEEE, 2012.

[15] May, Alexander. "Solving subset sum with small space–handling cryptanalytic big data." *it-Information Technology* 62, no. 3-4, 181-187. 2020.

[16] Ramanathan, Saravanan, Nitin Shivaraman, Seima Suryasekaran, Arvind Easwaran, Etienne Borde, and Sebastian Steinhorst. "A survey on time-sensitive resource allocation in the cloud continuum." *it-Information Technology* 62, no. 5-6, pp. 241-255. 2020.

[17] Korkan, Ege, Sebastian Kaebisch, and Sebastian Steinhorst. "Streamlining IoT system development with open standards." *it-Information Technology* 62, no. 5-6, pp. 215-226. 2020.

[18] Godhrawala, Husain, and R. Sridaran. "A dynamic Stackelberg game based multi-objective approach for effective resource allocation in cloud computing." *International Journal of Information Technology* 15, no. 2, pp. 803-818, 2023.

[19] Würdemann, Nick. "Exploiting symmetries of high-level Petri games in distributed synthesis." *it-Information Technology* 63, no. 5-6, pp. 321-331, 2021.

[20] Berani Abdelwahab, Erzana, and Martin Fränzle. "A sampling-based approach for handling delays in continuous and hybrid systems." *it-Information Technology* 63, no. 5-6, pp. 289-298, 2021.

[21] Jiang, Tao, Xiaofeng Chen, Qianhong Wu, Jianfeng Ma, Willy Susilo, and Wenjing Lou. "Secure and efficient cloud data deduplication with randomized tag." *IEEE transactions on information forensics and security* 12, no. 3, pp. 532-543, 2016.

[22] Nayak, Sanjeet Kumar, and Somanath Tripathy. "SEDS: secure and efficient server-aided data deduplication scheme for cloud storage." *International Journal of Information Security* 19, no. 2, pp. 229-240. 2020

[23] Lin, Yu, Yunlong Mao, Yuan Zhang, and Sheng Zhong. "Secure deduplication schemes for content delivery in mobile edge computing." *Computers & Security* 114, pp. 102602, 2022.

Authors Biography

Dr. S. Muthurajkumar, is Associate Professor, Department of Computer Technology, Madras Institute of Technology (MIT) Campus, Anna University, Chennai. His research interests are Cloud Computing, Network Technologies, artificial intelligence, and Network security.

Karthikeyan V, is a B.E (CSE) student, Department of Computer Technology in Madras Institute of Technology (MIT) Campus, Anna University, Chennai.

Kowsi N, is a B.E (CSE) student, Department of Computer Technology in Madras Institute of Technology (MIT) Campus, Anna University, Chennai.

Barath S, is a B.E (CSE) student, Department of Computer Technology in Madras Institute of Technology (MIT) Campus, Anna University, Chennai.

Cite this paper:

Muthurajkumar S, Karthikeyan V, Kowsi N, and Barath S, "Secured Data Storage in Cloud Environment Using Deduplication", International Journal of Advances in Computer and Electronics Engineering, Vol. 8, No. 6, pp. 1-8, June 2023.