ENHANCING DATA DEDUPLICATION TECHNIQUES OF STORAGE OF BIG DATA IN CLOUD

S. Suruthi[1], P.Chitra Devi[2], D.Kanthasamy[3], U.Sundhar[4]

[1]P.G student, Department of CSE, Thiruvalluvar College of Engineering and Technology, Vandavasi.

[2]Assistant Professor, Department of CSE, Thiruvalluvar College of Engineering and Technology Vandavasi

[3]Head of the Department, Department of CSE(AI&ML),Thiruvalluvar College of Engineering and Technology Vandavasi.

[4]Head of the Department, Department of CSE, Thiruvalluvar College of Engineering and Technology, Vandavasi.

ABSTRACT - *Data deduplication is a crucial data compression method aimed at removing duplicate instances of recurring data, and it has been extensively utilized in cloud storage to minimize storage requirements and conserve bandwidth. To ensure the confidentiality of sensitive information while facilitating deduplication, the convergent encryption method has been introduced to encrypt data prior to outsourcing. In an effort to enhance data security, this project represents the initial attempt to systematically tackle the issue of authorized data deduplication. Unlike conventional deduplication systems, this approach takes into account the varying privileges of users during the deduplication check, in addition to the data itself. We also introduce several innovative deduplication frameworks that support authorized duplicate checks within a hybrid cloud environment. Security evaluations confirm that our approach is secure according to the criteria outlined in the proposed security model. As a proof of concept, we have developed a prototype of our authorized duplicate check scheme and conducted experimental tests using this prototype. The findings indicate that our authorized duplicate check scheme results in minimal overhead when compared to standard operations.*

Key Words: Cloud, Big Data, Data Deduplication Techniques

1.INTRODUCTION

Cloud computing offers users seemingly limitless "virtualized" resources as services throughout the Internet, concealing platform and implementation specifics. Current cloud service providers deliver both highly reliable storage and extensive parallel computing capabilities at comparatively low prices. As cloud computing gains traction, a growing volume of data is being stored in the cloud and shared among users with designated privileges that outline the access rights to the stored information. A significant challenge faced by cloud storage services is the management of the continuously expanding data volume. To enhance data management scalability in cloud computing, deduplication has emerged as a well-recognized technique that has garnered increasing attention recently.

The technique is utilized to improve storage efficiency and can also be employed in network data transfers to minimize the amount of data that needs to be transmitted. Rather than maintaining multiple copies of data with identical content, deduplication removes redundant data by retaining only one physical copy and referencing other duplicate data to that copy. Deduplication can occur at either the file level or the block level. In the case of file-level deduplication, it removes duplicate copies of the same file. Deduplication can also happen at the block level, which eliminates duplicate data blocks found in non-identical files. Despite the numerous advantages of data deduplication, security and privacy issues emerge as users' sensitive information is vulnerable to both internal and external threats. While traditional encryption ensures data confidentiality, it is not compatible with Data deduplication is a process that addresses the challenges of traditional encryption, which necessitates that different users encrypt their data using their own keys. Consequently, identical data copies from various users result in different cipher texts, rendering deduplication unfeasible. To tackle this issue, convergent encryption has been introduced to

maintain data confidentiality while enabling deduplication. This method encrypts and decrypts a data copy using a convergent key, which is generated by calculating the cryptographic hash value of the data copy's content. Once the key is created and the data is encrypted, users keep the keys and transmit the cipher text to the cloud. Because the encryption process is deterministic and based on the data content, identical data copies will produce the same convergent key and, therefore, the same cipher text. To safeguard against unauthorized access, a secure proof of ownership protocol is necessary to demonstrate that the user genuinely owns the file when a duplicate is detected. Following this proof, subsequent users with the same file can receive a pointer from the server without needing to upload the duplicate file again. A user can then download the encrypted file using the pointer from the server, which can only be decrypted by the respective data owners using their convergent keys.

Thus, convergent encryption enables the cloud to conduct deduplication on the cipher texts, while the proof of ownership safeguards against unauthorized access to the file. Nevertheless, existing deduplication systems lack support for differential authorization duplicate checks, which are crucial for many applications. In an authorized deduplication system, each user is assigned a set of privileges during the system's initialization. Each file uploaded to the cloud is also associated with a specific set of privileges that dictate which users are permitted to perform duplicate checks and access the files. Before a user submits a duplicate check request for a particular file, they must provide this file along with their own privileges as inputs. The user is able to find aduplicate for this file if and only if there is a copy of this file and a matched privilege stored in cloud. For example, in a company, many different privileges will be assigned to employees. In order to save cost and efficiently management, the data will be moved to the storage server provider (SSP) in the public cloud with specified privileges and the deduplication technique will be applied to store only one copy of the same file.

Due to privacy concerns, certain files will be encrypted and only employees with designated privileges will be permitted to perform duplicate checks to enforce access control. Traditional deduplication systems that utilize convergent encryption, while offering some level of confidentiality, do not accommodate duplicate checks with varying privileges. In essence, the deduplication process based on convergent encryption does not take into account differential privileges. This presents a contradiction when attempting to achieve both deduplication and a differential authorization duplicate check simultaneously.

Need For the Study

Data deduplication methods are extensively used to back up data and reduce network and storage costs by identifying and removing redundancy within the data.

Objective of the Paper

The primary objective is to facilitate deduplication and the distributed storage of data across several storage servers.

2.LITERATURE REVIEW

A literature survey is a crucial step in the software development process. Prior to tool development, it is essential to assess factors such as time, budget, and the company's capabilities. Once these criteria are met, the next phase involves selecting the appropriate operating system and programming language for the tool's development. As programmers

begin to create the tool, they will require significant external assistance. This support can come from experienced programmers, literature, or online resources. Before constructing the system, the aforementioned considerations must be taken into account to develop the proposed system effectively. The paper development sector primarily focuses on thoroughly surveying all necessary requirements for the paper development. For every project, the literature survey remains a vital component of the software development process. Before creating the tools and their associated designs, it is important to evaluate and survey the time constraints, resource needs, workforce, budget, and company capabilities.

Once these parameters are thoroughly assessed, the next step is to identify the software specifications required for the system, including the type of operating system needed and all necessary software to advance to the next phase, such as tool development and related operations.

In this paper [1], the authors introduced an architecture that offers secure deduplication storage capable of resisting brute force attacks, implemented in a system named dupLESS. This system allows clients to encrypt their data using an existing service. The encryption method for deduplicated storage can achieve performance and space savings that are nearly equivalent to those obtained when using the storage service with plaintext data.

There exists a mechanism [2] to recover space from incidental duplication, making it available for controlled file replication. This mechanism employs convergent encryption, which allows duplicate files to be merged into a single space file, even if the files are encrypted with different user keys.

This approach [3] serves as a baseline where each user possesses an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, this baseline key management scheme results in a significant number of keys as the user base grows, necessitating that users diligently protect their master keys.

In this paper [4], a private deduplication protocol based on standard cryptographic assumptions is constructed, presented, and analyzed. The authors demonstrate that the private data deduplication protocol is likely secure, provided that the underlying hash function is collision-resilient, the discrete logarithm problem is difficult, and the erasure coding algorithm can recover a substantial fraction of the bits.

In this paper [5], the authors design an encryption scheme that ensures semantic security for less popular data while offering weaker security and improved storage and bandwidth advantages for more popular data. This approach allows data deduplication to be effective for popular data, while semantically secure encryption safeguards less popular content. The authors show that their scheme is secure under the Symmetric External Decisional Diffie-Hellman Assumption.

3.IMPLEMENTATION

The system architecture defines the fundamental structure of the system, outlining the key design features and components that form the framework for the system. It offers architects a perspective on the users' vision regarding the system's requirements and functionalities, as well as the directions in which it should develop. Additionally, it aims to

preserve the integrity of that vision throughout the detailed design and implementation phases.

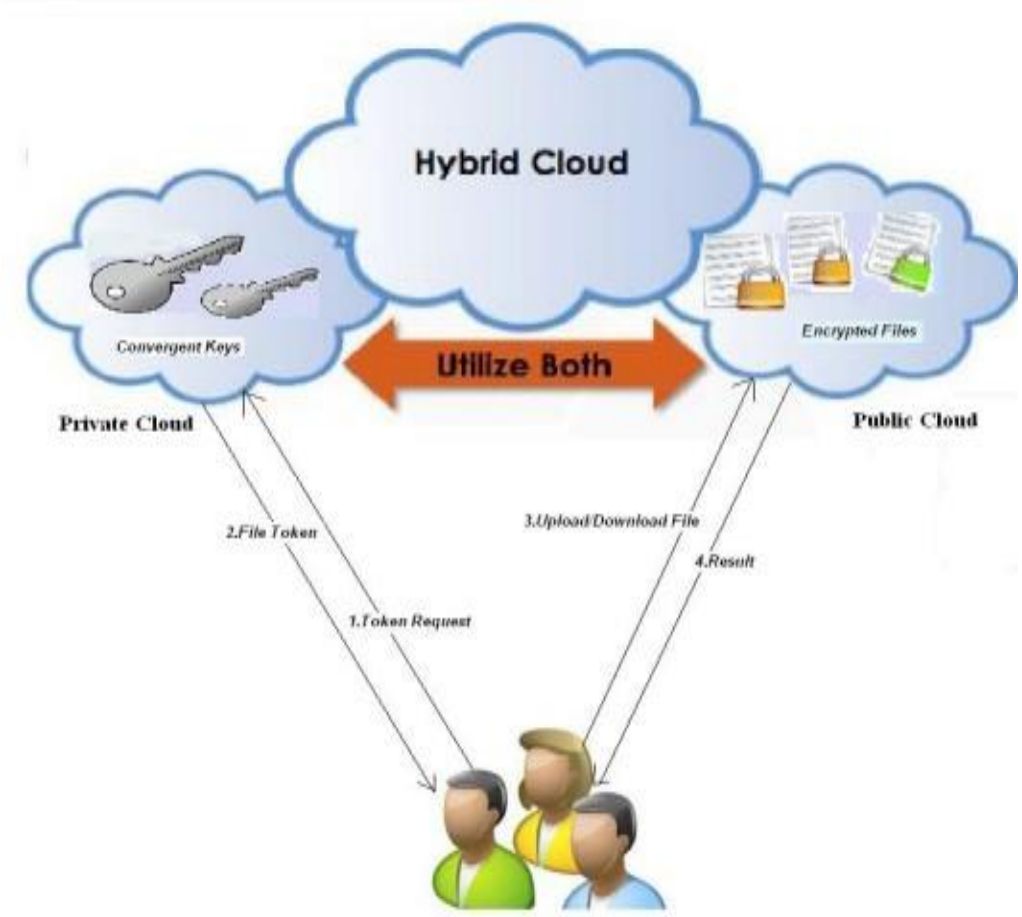

Fig. 3.1Architecture of the system

In this module, users are provided with authentication and security measures to access the information presented in the ontology system. Before users can access or search for details, they must have an account; otherwise, they need to register first. At a minimum, users are required to provide an email address, username, password, display name, and any other profile fields that have been marked as mandatory. The display name will be utilized by the system whenever it needs to show the user's proper name.

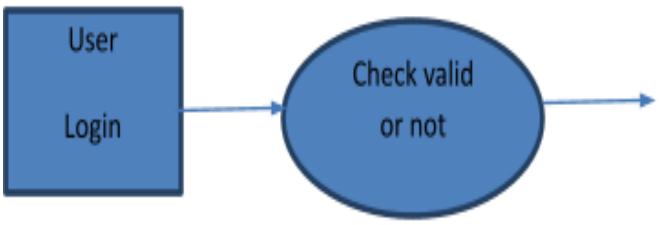

Fig. 3.2 user module

Server starts up and upload file

The user can startup the server after cloud environment is opened. Then the user can upload the file to the cloud.

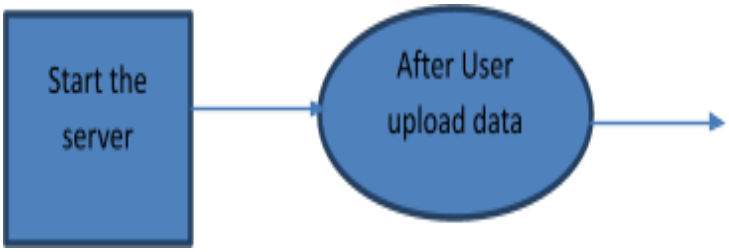

Fig. 3.3 server start up and upload file

Encryption

M3 encryption:

- The algorithm is highly intricate and secure, yet its usage is straightforward.
- The fundamental concept of this algorithm involves character remapping that relies on key self-mutation.
- The lifespan of a key corresponds to its length. This indicates that any state of the key will only encrypt a segment of the clear-text that matches the length of that key version before it undergoes self-mutation into a new version. This new version will then encrypt the subsequent segment of the clear-text, and so on.
- Throughout the entire process, a clear-key provided by the user is split into four distinct "threads" of various and continuously self-mutating keys. These four keys work in unison to convert the clear-text letters into cipher text one letter at a time using two different techniques: array remapping and a type of dynamic "substitution cipher." This entire procedure is repeated multiple times, re-encrypting everything several times before finalizing the cipher-text.
- For a potential attacker to reverse the process, they would need to determine the end state of four different keys simultaneously, working backwards through one mutation version at a time. Since two of these keys are utilized for array remapping, it is essential to decode the entirety of these keys for each letter in the clear-text.

Decryption algorithm

Data Encryption Standard (DES)

This refers to the Data Encryption Standard, which was created in 1977. It was the initial encryption standard endorsed by NIST (National Institute of Standards and Technology). DES features a key size of 64 bits and a block size of 64 bits. Over the years, numerous attacks and techniques have revealed vulnerabilities in DES, rendering it an insecure block cipher.

Algorithm:

functionDES_Encrypt(M,K) where M = (L, R)

M←IP (M)

Forround←1to16do K←SK (K, round) L←Lxor F(R,Ki) swap(L, R)

end
swap (L, R)
M←IP-1 (M)
Return M End

Chunking Technique for Deduplication

Chunking refers to the method of dividing a file into smaller segments known as chunks. This technique is crucial in various applications, including remote data compression, data synchronization, and data deduplication, as it significantly influences the system's ability to detect duplicates. Content-defined chunking (CDC) is a technique that segments files into chunks of variable lengths, with the cut points determined by specific internal characteristics of the files. In contrast to fixed-length chunks, variable-length chunks offer greater resistance to byte shifting. This enhances the likelihood of identifying duplicate chunks both within a single file and across multiple files. However, CDC algorithms necessitate additional computational resources to identify the cut points, which can be resource-intensive for certain applications. In our earlier research (Widodo et al., 2016), we found that the hash-based CDC algorithm implemented in the system required more processing time than other operations within the deduplication framework. The current study introduces a high-throughput hash-less chunking method. Rather than relying on hashes, this approach utilizes the byte values to establish the cut points. The algorithm employs both a fixed-size window and a variable-size window to identify the maximum-valued byte, which serves as the cut point. This maximum-valued byte is incorporated into the chunk and positioned at the chunk's boundary. This setup enables the RAM to perform fewer comparisons while maintaining the properties of CDC. We conducted a comparison of RAM against existing hash-based and hash-less deduplication systems. The experimental findings indicate that our proposed algorithm achieves superior throughput and a higher number of bytes saved per second when compared to other chunking methods.

Input Design

The input design serves as the connection between the information system and its users. It includes the specifications and procedures necessary for data preparation, ensuring that transaction data is transformed into a usable format for processing. This can be achieved by having the computer read data from written or printed documents, or by having individuals enter the data directly into the system. The focus of input design is on managing the volume of input required, minimizing errors, preventing delays, eliminating unnecessary steps, and maintaining simplicity in the process. The design is crafted to ensure security and user-friendliness while preserving privacy.

Output Design

A high-quality output is one that fulfills the end user's requirements and presents information in a clear manner. In any system, the results of processing are conveyed to users and other systems through outputs. Output design determines how information will be displayed for immediate needs, as well as the format for hard copy outputs. It serves as the most crucial and direct source of information for the user. Effective and thoughtful output design enhances the system's ability to support user decision-making. The output format of an information system should achieve one or more of the following objectives.

Summary

In this proposal, we are ensuring high security for the data stored in cloud storage, protecting it from attackers through the use of an encryption technique (Blowfish algorithm) that involves a key sharing process between the user and the client.

Data deduplication can be achieved using a chunking technique, where the data is divided into fragments and compared with existing data. If a duplicate copy of the data is found, there is no need to store it again, and access is restricted to only authorized users of that data copy. This process helps us achieve data confidentiality, tag consistency, and data reliability.

The primary advantage of the proposed system is its support for both file-level and block-level operations. A key feature of this proposal is the assurance of data integrity, which includes tag consistency, and the ability to perform deduplication on both the client and server sides. The deduplicated data is then transferred to the cloud storage library. This approach is more efficient, reduces space requirements, and is cost-effective. When an attempt is made to transfer a duplicate copy of data, deduplication occurs at the source, leading to lower electricity consumption, faster recovery times, and a reduction in overall storage costs.

CONCLUSION AND FUTURE ENHANCEMENT

In this paper, we introduced the concept of authorized data deduplication aimed at enhancing data security by incorporating varying user privileges during the duplicate verification process. Additionally, we presented multiple innovative deduplication frameworks that facilitate authorized duplicate checks within a hybrid cloud environment, where the private cloud server generates duplicate-check tokens for files using private keys. Our security analysis confirms that our proposed schemes are robust against both insider and outsider threats as outlined in the established security model. To validate our approach, we developed a prototype of the authorized duplicate check scheme and conducted experimental tests on it. The results indicated that our scheme introduces minimal overhead when compared to convergent encryption and network transfer.

Ultimately, we assert that the security of cloud data storage continues to face numerous challenges and remains critically important, with many research issues yet to be explored. In our proposed work, deduplication has been applied to text and images, and there is potential for further expansion to include audio and video.

REFERENCES

[1]. M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296–312, 2013.

[2]. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.

[3]. M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.

[4]. S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

[5]. P.AndersonandL.Zhang.Fastandsecurelaptopbackupswithencrypted de-duplication. In Proc. of USENIX LISA, 2010.

[6]. M. Bellare, S. Keelveedhi, and T.Ristenpart. Dupless: Server aided encryption for deduplicated storage. In USENIX Security Symposium, 2013.

[7]. M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296– 312, 2013.

[8]. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.

[9]. M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. InCRYPTO, pages 162–177, 2002.

[10]. S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

[11]. J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. The imer. Reclaiming space from duplicate files in a serverless distributed file system .InICDCS,pages 617–624, 2002.

[12]. D. Ferraiolo and R. Kuhn. Role-based access controls. In 15th NIST-NCSC National Computer Security Conf., 1992. [13] Rahman, M. A., Rahim, M. A., Rahman, M. M., Moustafa, N., Razzak, I., Ahmad, & Patwary, M. N. (2022). A secure and intelligent framework for vehicle health monitoring exploiting big-data analytics. IEEE Transactions on Intelligent Transportation Systems, *23*(10), 19727-19742.