Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

# Duplicate Detection on Cloud Data with Re-Encryption based Data Sharing

**N. Saranya[1], T. Dheepa[2], V. Dhanalakshmi[3], C. Radhakrishnan[4]**

[1,2,3,4]*Department of Computer Science and Engineering, Kongunadu College of Engineering and Technology, Trichy, India*

***Abstract.*** Cloud data storage has gained immense popularity in recent years due to its scalability and accessibility. However, the efficient management of data in the cloud, especially with the ever-increasing volume of data, remains a significant challenge. Data deduplication is a crucial technique to decrease storing costs and enhance data management efficiency. It also concerns the secure cloud data storage and sharing. Here implements a novel approach to deduplication-based cloud data storage using chunk-based partitioning. In this approach, data is divided into fixed-size chunks, and a deduplication mechanism is applied to identify and eliminate redundant chunks. The process of duplicate detection leading to improved storage efficiency and reduced data redundancy. It not only conserves storage space but also enhances data retrieval performance by dropping the volume of data to be moved over the network. Proposed system also enhances the security of cloud data storage by employing Base64 encoding for data representation and implementing re-encryption techniques based on the Advanced Encryption Standard (AES) for secure data sharing. The Base64 encoding methodology is employed to transform binary data into a text format, rendering it less susceptible to unintended manipulations during data transmission and storage. This encoding mechanism ensures data integrity and facilitates its secure storage in cloud environments. By utilizing AES encryption, data shared among users is re-encrypted with unique keys, ensuring that only authorized recipients can access and decrypt the shared information. The proposed approach significantly enhances data confidentiality and mitigates risks associated with unauthorized access, thereby fortifying the security of cloud-based data sharing.

***Keywords:*** *Cloud system, Deduplication, Chunk based similarity checking, Base64 encoding, Re-encryption, Advanced encryption standard, Secure data sharing.*

## 1 Introduction

In the dynamic landscape of cloud data management, ensuring efficient storage utilization, duplicate detection, and secure data sharing has become increasingly crucial. This project introduces a novel approach known as chunk-based similarity checking, designed to enhance duplicate detection on cloud-stored data and facilitate more efficient sharing practices. The key innovation lies in breaking down data into smaller, manageable chunks, which are then subjected to similarity checks to identify duplicates. By adopting this approach, the system optimizes storage space by eliminating redundant data chunks, promoting a more streamlined and economic data storage environment.

Furthermore, the project integrates a re-encryption mechanism to fortify data security during sharing. Leveraging this approach, data owners can re-encrypt chunks of data with new encryption keys before sharing them with authorized users. This provides an additional layer of protection, ensuring that only designated individuals with the appropriate decryption keys can access and utilize the shared information. The re-encryption process not only contributes to data security but also facilitates controlled and secure data sharing practices, aligning with contemporary standards of privacy and confidentiality in cloud computing.

In essence, the combination of chunk-based similarity checking and re-encryption offers a comprehensive solution to the challenges of duplicate detection, efficient storage management, and secure data sharing in cloud environments. This project aims to advance the capabilities of cloud data systems, fostering a more resilient and streamlined approach to data management in an era where efficient resource utilization and data security are paramount.

### 1.1 Cloud Computing

Cloud computing refers to the use of computing resources, such as servers, storage, and software, delivered over the internet as a service. In order to store and carry out the intended business processes, users of the cloud computing paradigm must grant access to their data. Because there is a vast amount of sensitive and important data stored on clouds, cloud service providers are required to offer trust and security. Concerns exist about fine-grained, adaptable, and scalable access control in cloud computing.

Major cloud computing providers like Amazon, Google, Microsoft, Yahoo, and others offer services such as Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Storage-as-a-Service, and Infrastructure-as-a-Service (IaaS). Cloud computing is steadily expanding. Furthermore, cloud computing is a fantastic technology since it may significantly reduce expenses through optimization and maximize both operating and economic effectiveness. Additionally, cloud computing may greatly increase its collaboration, speed, and range, enabling an internet infrastructure that supports a global computing model. Furthermore, cloud computing offers benefits in providing more scalable and resilient services.

### 1.2 Cloud Security

There are several definitions of security. Combining confidentiality—the avoidance of unauthorized information disclosure—integrity—the avoidance of unauthorized information modification or deletion—and availability—the avoidance of unauthorized information withholding—makes up security.

Resource security, resource management, and resource monitoring are the main problems with cloud computing. As of right now, cloud application deployment is not governed by any standard rules or regulations, and cloud standardization control is lacking. Many innovative methods have been developed and deployed in the cloud; nonetheless, because of the characteristics of the cloud environment, these methods are unable to guarantee complete security.

There is a discussion of the inherent problems with data security, governance, and control management in cloud computing. The main concerns of security, privacy, and trust in the current cloud computing environment are discussed, along with how users might identify both concrete and abstract risks associated with using it. The authors list security, privacy, and trust as the three main areas where cloud computing could be threatened. In the current period of long-dreamed computing as a utility vision, security is crucial. It falls into four subcategories: safeguards against illegal insider operations and service hijacking, monitoring or tracing of cloud servers, data confidentiality, and safety systems. For networks using cloud computing, a data security strategy is suggested. The writers' primary topic of discussion was cloud data storage security. Additionally, there are a few patents pertaining to data storage security methods. Provide a survey on critical infrastructure security using cloud computing. For RFID technology integrated into cloud computing, which will merge cloud computing and the Internet of Things, a security & privacy architecture has been proposed as shown in Fig 1.

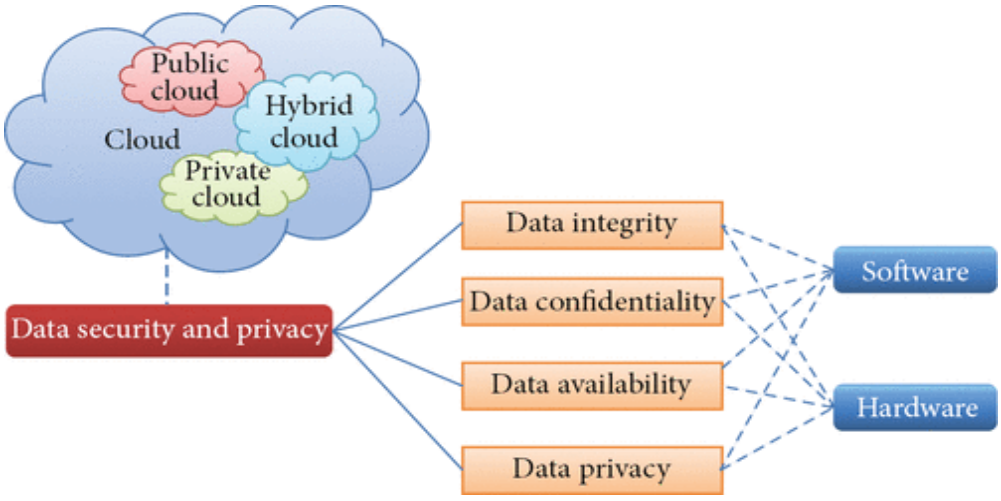


**Fig. 1. Cloud security model**

## 2 Related work

Yuan, et.al,…[1] developed a the proposed data deduplication plan utilizes random sampling and the convergent all-or-nothing transform (CAONT) to ensure secure re-encryption. The approach, resilient against stub-reserved attacks, aims to safeguard sensitive data privacy when outsourced to a remote cloud service provider. Targeted at user groups or enterprises, the scheme involves three entities: cloud user, key server, and cloud service provider (CSP). File-level deduplication, focusing on fixed-size chunks, is the primary focus. The cloud user uploads data, deletes M to save storage, and can share the file key using CP-ABE key server assistance. The CSP, assumed sincere but curious, handles storage, removing unnecessary data while retaining one duplicate of each item.

Liu, et.al,…[2] deployed sequential multi-signature and proxy re-encryption to introduce a novel blockchain-assisted EMR in a cloud context. First off, the system performs highly secure even in the absence of a reliable centre thanks to blockchain. Second, to safeguard private medical information while facilitating doctors' access to patients' past medical records, employ proxy re-encryption. The doctors have also employed a consecutive multisignature, which is useful and can improve safety performance. A sequential multisignature and a group key are used in BCEMR to improve information safety (a analysis may be made by one or more doctors). Doctors can access patient medical records from the past while maintaining data security because to proxy re-encryption. Blockchain technology in particular has made it possible for BC-EMR to significantly increase security and overcome numerous shortcomings in traditional cloud-assisted EMR. The hospital server allots a medical team to the patient based on their initial condition if the identity is accepted. The patient's diagnosis results are protected by a group key that is established by the team, the server, and en. When a physician receives a message from a previous physician regarding a diagnosis, the physician first confirms the signatures of all previous physicians. In the event that it passes, the physician will diagnose the patient and publish their signature on the blockchain. If not, he or she asks the previous physician to send the message again. After the final physician signs, the patient's public key is used to encrypt the result. The encrypted data, along with signatures, is stored on the blockchain and cloud, pending successful verification.

Uyyala, et.al,…[3] implemented an useful public key cryptography tool called accessible public key encryption (SPKE). The proposed SPKE (Symmetric Password Key Exchange) system, called endorsement-based accessible encryption, enhances security for initial messages and keyword searches on encrypted communications. Unlike other SPKE systems, it addresses flaws, including vulnerability to keyword guessing attacks. Inspired by sign-cryption and certificate-based cryptography, it provides defense against such attacks, along with features like verifiable verification, no key escrow, and no secure channel requirement. The designed encryption scheme demonstrates security and practicality in the face of various attacks, making it a robust solution.

Obour Agyekum, et.al,…[4] presented a safe and effective Proxy Re-Encryption (PRE) scheme that incorporates an Inner-Product Encryption (IPE) scheme. Under this scheme, Data can be decrypted using the associated ciphertext and the inner product of the private key, connected to a set of attributes specified by the data owner, equal zero (0). In this case, a blockchain network is used, and the processing node re-encrypts the data while serving as the proxy server. In brief, leverage blockchain's cryptographic foundations and consensus mechanism in this study. Employ proxy re-encryption using Attribute-Based Encryption (ABE) to ensure robust data confidentiality. The blockchain's processing node re-encrypts the data, splitting it between the blockchain and a cloud server. This dual storage makes it practically impossible for a revoked user, even with access to the cloud server, to retrieve the complete data. Moreover, the blockchain's processing nodes handle data computations, relieving the load on the cloud server. The proposed proxy re-encryption complies with fine-grained access control since the ABE method allows users to access various data sets. In essence, the resistance to collusion lies in the inability of the cloud server, proxy, and the revoked user to cooperate in retrieving the data. This is made possible by the decentralised nature of the blockchain network, where all activities (transactions) are recorded and saved into blocks in addition to being watched over by each network member. Additionally, because blockchain creates a trustworthy environment among participants, there is a notable level of confidence between the users and the data owner.

Wang, et.al,…[5] introduced a productive proxy re-encryption (PRE) system into the ICN architecture to support minimise user-side overhead and ensure adaptable data exchange amongst subscribers and their co-operator. Ad-

ditional advantages of this concept are its resilience to collusion and lack of interaction. Additionally, we demonstrate that our technique is protected in contradiction of chosen ciphertext attacks (CCA) in complete ICN encoding and adaptive repayable adaptive chosen ciphertext attacks (RCCA) in re-encryption [6]. For efficient content transmission and reduced user-side computational load, this Proxy Re-Encryption (PRE) scheme integrates with Shamir's Secret Sharing mechanism and the Publish/Subscribe Networking content distribution service [7]. The content provider establishes an access control policy, assigns authorized subscribers, and encrypts content using a router's share and Shamir's Secret polynomial. The resulting encrypted enabling block is sent to the storage router, and proxy re-encryption is employed for this functionality [8]. Assume that Subscriber A generates a re-encryption key, sends it to a proxy, which then uses the key to re-encrypt information in a router. The re-encrypted data is transmitted to Subscriber B, who can decrypt it using their private key [9].

## 3 Existing Methodologies

Proxy Re-encryption (PRE) is employed for information retrieval, and Public Key Encryption with Keyword Search (PEKS) matches keywords with trapdoors to detect file duplication. The two key components of the scheme are data recovery and data de-duplication[10]. The re-encryption key is stored in the ciphertext chain table, and the file tag is connected to the file ciphertext. To enable data de-duplication, the data owner uploads the file ciphertext, file tag, and re-encryption key to the cloud server. If the test is successful, the file has already been saved on the server[11]. Data recovery is achieved through the use of Proxy Re-encryption (PRE), and file duplication is detected by Public Key Encryption with Keyword Search (PEKS), which matches keywords with trapdoors. Data de-duplication and data recovery are the two main parts of the plan. The file tag corresponds to the file ciphertext, and the re-encryption key is stored in the respective ciphertext chain table[12]. For data de-duplication, the data owner uploads the file ciphertext, file tag, and re-encryption key to the cloud server. If the test is successful, the file is saved on the server. To retrieve the file, the user requests it from the server, which then re-encrypts and creates modified ciphertext using the user's re-encryption key from the ciphertext chain table. The user can decrypt the changed ciphertext with their private key[13][14].

**Public Key Encryption**

- Public key encryption, also known as asymmetric encryption, employs two mathematically linked keys: a public key and a private key[15][16].
- This method is frequently applied to digital signatures, authentication, and secure data transfer[16].

**Re-encryption**

- The user's re-encryption key is stored in the ciphertext chain table corresponding to the cloud server's ciphertext[17][18].
- This key, formed from the file encryption key and the user's public key, establishes a one-to-one relationship between the file and the user's identity[19].
- Consequently, the user can decrypt the corresponding modified ciphertext using their private key[20][21].

## 4 Proposed Methodologies

Cloud services would rather concentrate on their main business than handle the massive volumes of data that are added every day. This study looked at a number of data deduplication-related topics. A study was conducted on the different requirements of cloud services, including encryption, data deduplication, and management. The experiment put into practice a scenario in which the uploaded data may be deduplicated by the cloud service. The project integrates a number of cutting-edge methods to improve security, controlled data sharing, and the effectiveness of data storage. Data is first stored on the server after being encrypted in the Base64 format. Base64 encoding ensures that the data is stored and retrieved with integrity, allowing for smooth interchange and readability across multiple platforms. The project uses chunk-based deduplication to maximize storage space. In order to do this, the data must first be divided into smaller pieces and any duplicates must be found before being saved on the server. Storage space is preserved, improving efficiency and lowering the total storage footprint, by removing superfluous data portions. An AES-based re-encryption method combined with key sharing is used for safe data exchange. The data owner creates a new encryption key using the Advanced Encryption Standard (AES)

whenever a user requests access to encrypted data. A new key is generated and employed to re-encrypt the data, enhancing its security. Data owner gives the asking user the decryption key in a secure manner at the same time. Only authorized users who possess the necessary keys can decrypt and access the shared data thanks to this safe exchange. The procedure offers a regulated and secure way to share data, protecting sensitive data and respecting user privacy.

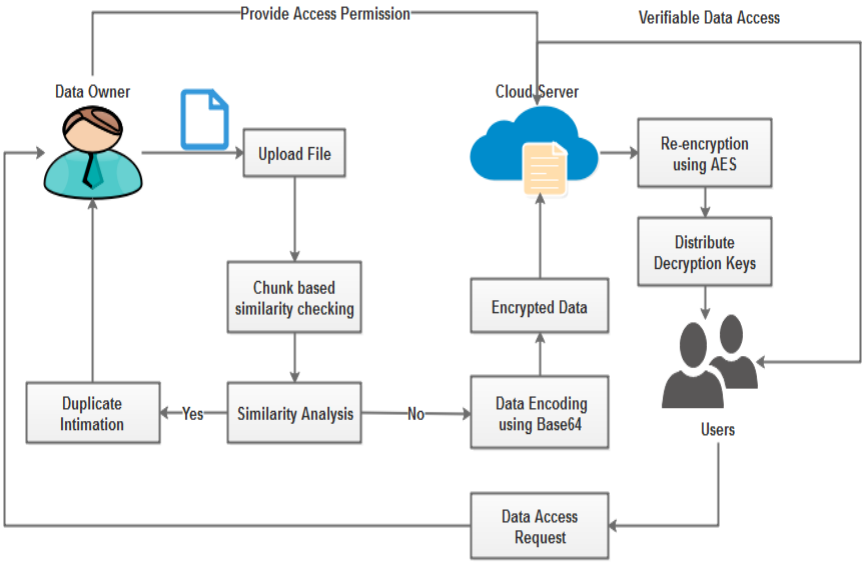


**Fig. 2. Proposed Framework**

### Cloud Storage Framework

Cloud computing and storage options provide users and businesses the flexibility to process and store data in either privately owned or third-party data centres, which can be situated anywhere from local to global locations. Coherence in cloud computing is achieved by resource sharing. Two different user categories, such as data owners and data providers, are possible under this system. A cloud service owner, whether a user or a provider, is the legitimate owner of a cloud service depending on ownership of the hosting cloud shown in Fig 2. Users obtain storage space from cloud service providers, allowing multiple data owners to share the same storage. Data owners can upload their files to the storage system for future use.

### De-duplication Checking

Data compression is a specific data compression method used in computing to get rid of duplicate copies of material that repeats. Single-instance (data) storage and intelligent (data) compression are words that are related and somewhat synonymous. This method can be used to decrease the number of bytes that need to be delivered during network data transfers as well as to increase storage utilisation. Analysed during the compression process, distinct data segments, or byte patterns, are found and saved. As the analysis continues, additional chunks are compared to a stored copy. If a match is found, the redundant chunk is replaced with a reference to the saved one. Users can verify files using the file name and contents in this module. Files that are encrypted are divided into sections. The chunks are checked by the service provider when files are uploaded. In order to conserve cloud storage space, data owners should only upload original files. Here can verify the text, document, and image file deduplication.

### File Encryption

The best approach to ensure data security is via encryption. A secret key that allows for file decryption must be available to the user in order to read an encrypted file. Base64 encoding transforms binary data into a text-based format, consisting of a set of ASCII characters, making it more suitable for storage and transfer across different systems. By converting binary information into a human-readable text representation, Base64 encoding facilitates compatibility with a wide range of devices and applications. This encoding process is particularly relevant in the context of cloud storage, where data is often transmitted over networks and stored in various formats. Base64-

encoded data is not only platform-independent but also aids in the prevention of data corruption during transmission. Moreover, the utilization of Base64 encoding contributes to an added layer of security. While it is not a method of encryption, Base64 encoding obscures the original content, making it less susceptible to casual observation. This can be advantageous when sensitive data is being transferred or stored on cloud servers, providing a degree of obfuscation that adds a level of security, albeit not a replacement for robust encryption mechanisms.

**Data Access Request**

An efficient and secure approach is essential when a person logs into a cloud service and needs to seek access to specific data that belongs to another user. The request contains information about the particular files or data types required as well as cloud verification for the access. The data owner receives a notification alert from the cloud application alerting them to the access request simultaneously. After login into their account, the data owner can review the request and determine whether or not access should be granted. Both sides receive real-time updates on the request's status to guarantee transparency.

**Re-encryption and Data Sharing**

In this module AES-based data re-encryption is employed, the process involves encrypting information with a unique key and subsequently re-encrypting it with a different key to enable secure sharing. When a user needs access to encrypted data, an AES-based data re-encryption mechanism comes into play. The data owner, holding the original encryption key, starts the process by creating a new encryption key. Using this new key, the data is re-encrypted, creating a secure envelope around it. Simultaneously, the data owner securely shares the decryption key associated with the newly encrypted data with the requesting user. This key exchange occurs through a secure channel, ensuring confidentiality. Once the requesting user receives the decryption key, they can use it to decrypt the re-encrypted data and access the information securely. This approach provides an additional layer of security and control, allowing data owners to share information selectively while preserving the integrity of the initial encryption.

## METHODOLOGY

### AES ALGORITHM

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm employing a block cipher for both data encryption and decryption. The standard defines three key sizes: AES-128, AES-192, and AES-256. The following steps make up the algorithm:

**Key Expansion:** The 128-bit, encryption key is expanded into a key schedule of 10, 12, or 14 round keys, respectively. Round keys are generated from the original encryption key through a key schedule algorithm.

**Initial Round:** The plaintext is segmented into 128-bit blocks and then XORed with the first round key.

**Rounds:** The encryption process contains of a set of rounds (10, 12, or 14) that operate on the state of the cipher. The four transformations that make up a round are SubBytes, ShiftRows, MixColumns, and AddRoundKey.

**SubBytes:** Every byte in the state is substituted with a corresponding byte from a substitution box (S-box), introducing confusion and enhancing resistance against linear cryptanalysis.

**ShiftRows:** The state is cycled through a set number of steps shifting each row. The second row shifts one step left, the third row shifts two steps left, and the fourth row shifts three steps left.

**MixColumns:** Each column of the state undergoes multiplication by a fixed polynomial, contributing to diffusion and aiding in the prevention of differential cryptanalysis.

**AddRoundKey:** The state is XORed with the round key for the current round.

**Final Round:** The final round is like previous rounds, excluding the MixColumns transformation.

**Output:** The final ciphertext is the output state of the cipher.

The opposite of the encryption process is the decryption procedure. The cipher text is divided into 128-bit blocks and XORed with the last round key. Then, the rounds are performed in reverse order, with the inverse of each transformation used. The final result is the original plain text.

## 5 EXPERIMENTAL RESULTS

The methodology is implemented with PHP as the front end and MySQL as the back end, offering a customizable development environment. The experiment involves various file sizes (KB) using the AES algorithm. Files of varying sizes are input into the system, and the evaluation is performed based on parameters such as encryption time and decryption time, listed in Table 1 and refer Fig 3.

**Table 1. Results**

| File Size in kb | 10 | 50 | 100 | 200 | 500 |
|---|---|---|---|---|---|
| Encryption Time in ms | 13 | 16 | 78 | 140 | 218 |
| Decryption Time in ms | 6 | 20 | 22 | 59 | 96 |

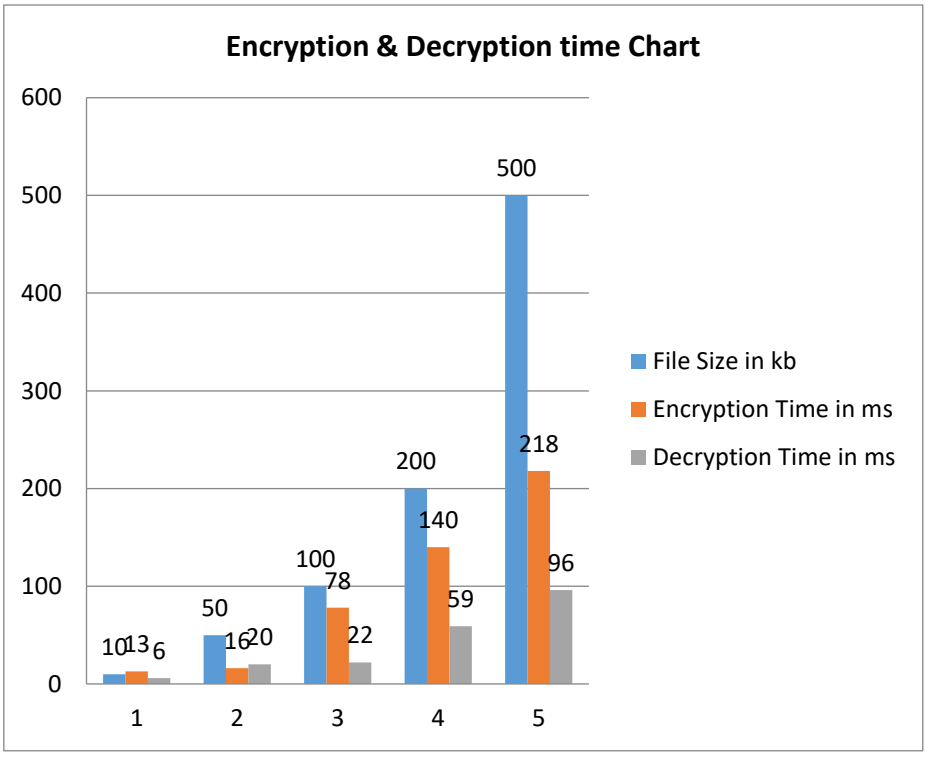


**Fig. 3. Encryption & Decryption time Chart**

## 6 Conclusion

In Conclusion, for safe and effective data management, the proposed project offers cloud data storage with AES encryption, chunk-based similarity checking method for duplication detection, and re-encryption for data sharing. AES encryption ensures the privacy and security of data in cloud storage, preventing unauthorized access.Modern data storage is incomplete without AES encryption because of its strong security features, which offer a strong defence against data breaches and privacy violations. The system's capacity to recognise and remove redundant data pieces lowers storage expenses and improves the cloud-based storage system's overall performance. Re-encryption is a crucial advance in data sharing since it allows for safe data interchange without sacrificing data integrity.

**References**

1. Yuan, Haoran, Xiaofeng Chen, Jin Li, Tao Jiang, Jianfeng Wang, and Robert H. Deng. "Secure cloud data deduplication with efficient re-encryption." IEEE Transactions on Services Computing 15, no. 1 (2019): 442-456.
2. Liu, Xiaoguang, Jun Yan, Shuqiang Shan, and Rongjun Wu. "A blockchain-assisted electronic medical records by using proxy reencryption and multisignature." Security and Communication Networks 2022 (2022).
3. Uyyala, Prabhakara. "Secure Channel Free Certificate-Based Searchable Encryption Withstanding Outside and Inside Keyword Guessing Attacks." The International journal of analytical and experimental modal analysis 13, no. 2 (2021): 2467-2474.
4. Obour Agyekum, Kwame Opuni-Boachie, Qi Xia, Emmanuel Boateng Sifah, Jianbin Gao, Hu Xia, Xiaojiang Du, and Moshen Guizani. "A secured proxy-based data sharing module in IoT environments using blockchain." Sensors 19, no. 5 (2019): 1235.
5. Wang, Qiang, Wenchao Li, and Zhiguang Qin. "Proxy re-encryption in access control framework of information-centric networks." IEEE Access 7 (2019): 48417-48429.
6. Ni, Fan, and Song Jiang. "RapidCDC: Leveraging duplicate locality to accelerate chunking in CDC-based deduplication systems." In Proceedings of the ACM Symposium on Cloud Computing, pp. 220-232. 2019.
7. Ni, Fan, Xing Lin, and Song Jiang. "SS-CDC: A two-stage parallel content-defined chunking for deduplicating backup storage." In Proceedings of the 12th ACM International Conference on Systems and Storage, pp. 86-96. 2019.
8. Wu, Huijun, Chen Wang, Yinjin Fu, Sherif Sakr, Liming Zhu, and Kai Lu. "Hpdedup: A hybrid prioritized data deduplication mechanism for primary storage in the cloud." arXiv preprint arXiv:1702.08153 (2017).
9. Sekaran, R., Munnangi, A. K., Ramachandran, M., & Gandomi, A. H. (2022). 3D brain slice classification and feature extraction using Deformable Hierarchical Heuristic Model. Computers in Biology and Medicine, 149, 105990-105990.
10. Ramesh, S. (2017). An efficient secure routing for intermittently connected mobile networks. Wireless Personal Communications, 94, 2705-2718.
11. Sekaran, R., Al-Turjman, F., Patan, R., & Ramasamy, V. (2023). Tripartite transmitting methodology for intermittently connected mobile network (ICMN). ACM Transactions on Internet Technology, 22(4), 1-18.
12. Douglis, Fred, Abhinav Duggal, Philip Shilane, Tony Wong, Shiqin Yan, and Fabiano Botelho. "The logic of physical garbage collection in deduplicating storage." In 15th USENIX Conference on File and Storage Technologies (FAST 17), pp. 29-44. 2017.
13. Guo, Wei, Hua Zhang, Sujuan Qin, Fei Gao, Zhengping Jin, Wenmin Li, and Qiaoyan Wen. "Outsourced dynamic provable data possession with batch update for secure cloud storage." Future Generation Computer Systems 95 (2019): 309-322.
14. 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. 22nd Int. Conf. Distrib. Comput. Syst., Vienna, Austria, 2002, pp. 617–624.
15. A. Agarwala, P. Singh, and P. K. Atrey, ‘‘DICE: A dual integrity convergent encryption protocol for clientside secure data deduplication,’’ in Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC), Banff, AB, Canada, Oct. 2017, pp. 2176–2181.
16. P. Anderson and L. Zhang, ‘‘Fast and secure laptop backups with encrypted deduplication,’’ in Proc. 24th Large Installation Syst. Admin. Conf., San jose, CA, USA, 2010, pp. 1–12.
17. D. Harnik, B. Pinkas, and A. Shulman-Peleg, ‘‘Side channels in cloud services: Deduplication in cloud storage,’’ IEEE Security Privacy, vol. 8, no. 6, pp. 40–47, Nov./Dec. 2010.

18. J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, ‘‘Secure deduplication with efficient and reliable convergent key management,’’ IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 6, pp. 1615–1625, Jun. 2014.
19. Saravanakumar, S., & Thangaraj, P. (2019). A computer aided diagnosis system for identifying Alzheimer's from MRI scan using improved Adaboost. Journal of medical systems, 43(3), 76.
20. Kumaresan, T., Saravanakumar, S., & Balamurugan, R. (2019). Visual and textual features based email spam classification using S-Cuckoo search and hybrid kernel support vector machine. Cluster Computing, 22(Suppl 1), 33-46.
21. Saravanakumar, S., & Saravanan, T. (2023). Secure personal authentication in fog devices via multimodal rank-level fusion. Concurrency and Computation: Practice and Experience, 35(10), e7673.
22. Thangavel, S., & Selvaraj, S. (2023). Machine Learning Model and Cuckoo Search in a modular system to identify Alzheimer's disease from MRI scan images. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 11(5), 1753-1761.
23. Saravanakumar, S. (2020). Certain analysis of authentic user behavioral and opinion pattern mining using classification techniques. Solid State Technology, 63(6), 9220-9234.
24. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, ‘‘Public key encryption with keyword search,’’ in Proc. Int. Conf. Theory Appl. Cryptograph. Techn., C. Cachin and J. Camenisch, Eds. Interlaken, Switzerland, 2004, pp. 506–522.
25. M. H. Au, J. Chen, J. K. Liu, Y. Mu, D. S. Wong, and G. Yang, ‘‘Malicious KGC attacks in certificateless cryptography,’’ in Proc. 2nd ACM Symp. Inf., Comput. Commun. Secur., New York, NY, USA, 2007, pp. 302–311.
26. M. Bellare, S. Keelveedhi, and T. Ristenpart, ‘‘Message-locked encryption and secure deduplication,’’ in Proc. 32nd Annu. Int. Conf. Theory Appl. Cryptograph. Techn., T. Johansson and P. Q. Nguyen, Eds. Athens, Greece, 2013, pp. 296–312.
27. S. Keelveedhi, M. Bellare, and T. Ristenpart, ‘‘DupLESS: Server-aided encryption for deduplicated storage,’’ in Proc. 22th USENIX Secur. Symp., S. T. King, Ed. Washington, DC, USA, 2013, pp. 179–194.
28. M. Abadi, D. Boneh, I. R. A. Mironov, and G. Segev, ‘‘Message-locked encryption for lock-dependent messages,’’ in Proc. 33rd Annu. Cryptol. Conf., Santa barbara, CA, USA, 2013, pp. 374–391.
29. Rathinam, G., Balamurugan, M., Arulkumar, V., Kumaresan, M., Annamalai, S., Bhuvana, J., Enhanced Security for Large-Scale 6G Cloud Computing: A Novel Approach to Identity based Encryption Key Generation, (2023), Journal of Machine and Computingthis link is disabled, 3(2), pp. 80–91