An Efficient Reconfigurable Cryptographic Model for Dynamic and Secure Unstructured Data Sharing in Multi-Cloud Storage Server

Parashiva Murthy Basavanapura Muddumadappa[1*], Sumithra Devi Kengeri Anjanappa[2], Mallikarjunaswamy Srikantaswamy[3]

[1] Department of Computer Science and Engineering, JSS Science and Technology University, 570006 Mysuru, India
[2] Department of Computer Science and Engineering, Dayanand Sagar Academy of Technology and Management, 560082 Bengaluru, India
[3] Department of Electronics and Communication Engineering, JSS Academy of Technical Education, 560060 Bengaluru, India

[*] Correspondence: Parashiva Murthy Basavanapura Muddumadappa (mallikarjunaswamys@jssateb.ac.in)

Received: 06-11-2022 **Revised:** 07-18-2022 **Accepted:** 08-02-2022

Citation: P. M. B. Muddumadappa, S. D. K. Anjanappa, and M. Srikantaswamy, "An efficient reconfigurable cryptographic model for dynamic and secure unstructured data sharing in multi-cloud storage server," *J. Intell Syst. Control*, vol. 1, no. 1, pp. 68-78, 2022. https://doi.org/10.56578/jisc010107.

Abstract: This study designs a reconfigurable multi-cloud storage server architecture for dynamic and secure data sharing has been designed, improves the security of unstructured data using cryptographic index-based data slicing (CIBDS), and reduces the malicious insider through data encryption using a third data encryption algorithm (3DEA). Focusing on multi-cloud storage server (MCSS) and data life cycle which includes three stages (i.e., data input, transition and utilization), the authors determined the efficiency of reconfigurable data file slicing, standard format, privacy and trustworthiness of the customers, in contrast to existing methods. Every part of a data file was encrypted using 3DEA, and Rivest Shamir Adleman (RSA) was employed to produce the private key to secure the unstructured data. The results show that the proposed framework effectively searches the data files in MCSS based on tags, such as input file names and private keys. The performance of the framework was measured by the security level, uploading/downloading latency time between our method and conventional methods, under different data sizes in (MB). Overall, our method reduces the malicious insider to 0.23% using 3DEA and RSA, during data encryption in the existing USDS-MC, shortens the uploading/downloading latency time (s) by 10% and 12%, compared to USDS-MC, and enhances the unstructured data security by 12% in comparison with that method. In this way, the authors managed to improve the self-protection of reconfigurable and secure unstructured data files in huge cloud infrastructure. This research optimizes the data security and privacy of encryption, decryption and cryptography technologies, and helps with the online process and its security maintenance during cloud storage.

Keywords: Cryptographic index; Rivest Shamir Adleman (RSA); Cloud computing; Multi cloud storage server; Cloud key management server

1. Introduction

The numerous wireless sensor networks (WSNs) and sensors installed in smart cities have generated a significant amount of data that has been saved in multiple cloud storage servers. For the purpose of security, it is particularly challenging to identify and categorize the various data formats on cloud servers. Thus, this paper proposes a framework that efficiently searches the MCSS data files using tags like the name of the input file that was received and private keys. The proposed and conventional approaches were compared for performance evaluation, in terms of security level, uploading/downloading latency time, and varied data sizes (MB).

Numerous research has been conducted at various stages of time to determine the best way to achieve information security and secrecy in cloud computing. As a way of enforcing information security in the cloud storage scheme, the encryption enables explicit access authorization and cryptography. Sharing information demonstrates inclusivity in data security, particularly for cloud storage. This method cannot guarantee secure key distribution and management. Internal hacks that use a private cloud database are not tracked [1-3]. This significantly lowers the procedure's competence. Some researchers presupposed that a data distribution process was at work in the cloud computing system, enabling a variety of data in addition to various clouds. However, the strategy fails to adequately take into account the distribution of keys in encrypted data channels. As a result, it could potentially compromise data integrity, something that typically occurs during the recovery process [4].

Figure 1 shows the system management of structured and unstructured data communication through integration with various cloud storage servers.

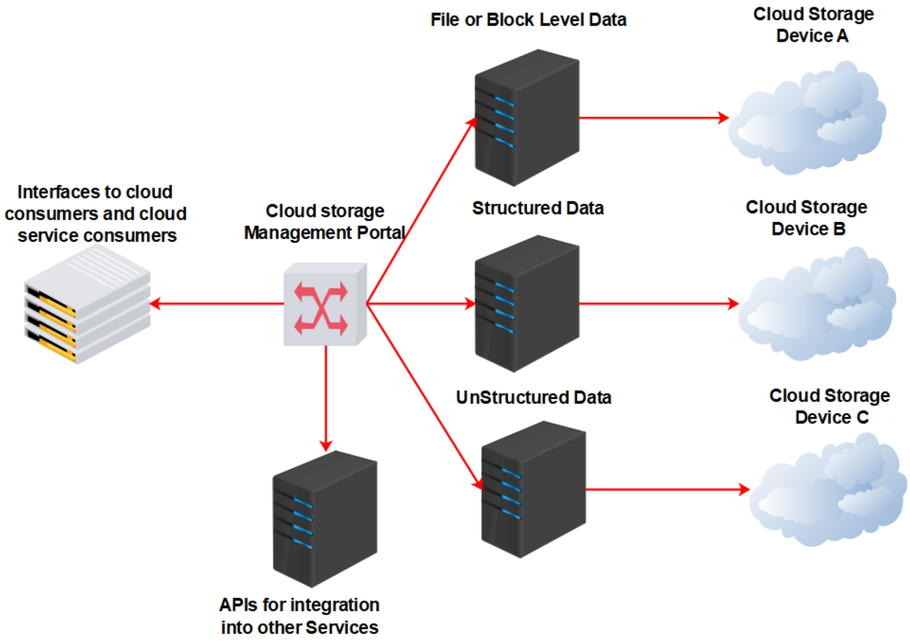
File or Block Level Data
Cloud Storage Device A
Interfaces to cloud consumers and cloud service consumers
Cloud storage Management Portal
Structured Data
Cloud Storage Device B
UnStructured Data
Cloud Storage Device C
APIs for integration into other Services

Figure 1. Fundamental could management system of structured and unstructured data storage

When we observe that the cloud is inaccessible, one of the regular risks is that the techniques for data transfer are adversely affected. The use of cloud computing systems is a common method for data storage. As a result, the breach that develops over time is a welcome response to information security intimidations [5, 6].

Yau et al. [7] devised a dynamic approach and four different types of algorithms to discuss the data security model for cloud computing in terms of cryptography and steganography. Additional switching operations were conducted on the transferred data, with differences in the original information that has been received. These operations significantly lowered the procedure's competence. These researchers have assumed that the data distribution process is effective in enabling the variety of data in addition to the various clouds in the cloud computing system. This could potentially compromise data integrity, something that typically occurs during the recovery process [8]. To increase the security of the cloud model in the medical industry, Manjunath et al. [9] created an algorithm that offers materials readily accessed and effectively used in the e-health system. However, the algorithm causes overlapping and slows down transfer rates from the source to the destination, when utilized for a large volume of data stored in the cloud.

The 3DES and the RSA encryption are two technologies for data encryption. However, one challenge in using the 3DES encryption approach is that it is vulnerable to a brute force attack. Shivaji et al. [10] adopted RSA encryption to increase the security and efficiency of 3DES dynamic file slicing, and proved that this approach helps to consistently encrypt consumer data. The structure of the approach allows for the use of shared encrypted keys in symmetric key cryptography.

In order to maintain a highly secure data storage service and integrate the various forms of encrypted data, Shekhawat et al. [11] developed a model that decreases the capital expense and conversation procedure. It is exceedingly difficult to classify different data packets processed by different algorithms. Bhadlawala and Chachapara [12] implemented data sharing amongst cloud services to maximize cloud service consumption. However, any cloud retains a vast amount of data during the communication on a path, making it very hard to classify the original data.

Therefore, the goal of the current paper is to advance a structure that outshines the aforementioned experiments. Five or more cloud storage facilities at the very least would be submitted as part of the projected design. The proposed configuration makes use of dynamic file slicing to increase privacy. The record coding technique was adopted to enable the structure to maintain a higher level of data integrity. To prevent malicious insiders and additional risks to information, the security of the key sharing processes would take precedence. The proposed approach was proved as the most effective way to maintain data security in a multi-cloud environment [13].

The remainder of this paper is organized as follows: Section 2 introduces the proposed approach; Section 3 defines the architecture outline, constituents and its functioning with algorithms; Section 4 elucidates the test results; Section 5 concludes the work and looks forward to future works.

2. Methodology

2.1 Proposed Design

Figure 2 shows the structural design of our approach. The design shows that the interface structure is how the information holder transmits the data file, an image, and the private key. In SRUD-MD, the recorded database is uploaded. The slicing index and structure are used to locate the uploaded material. The encryption procedure is created using databases that were previously and currently stored on multiple cloud storage servers. Additionally, the private key is encrypted using RSA keys, and the owner is given access to part of the RSA public key and the cloud DB server, respectively. The decryption stage also uses a variety of techniques. Key details are supplied for decryption and combining procedures after choosing the exact image. The secret key is used to decrypt shared files on the recipient's computer after looking at the file title. Particularly, dynamic file sharing is the best active approach, for it includes all safe information transfer methods [14, 15]. The outline provides a clear direction for how the user should conduct the file: file slicing happens before data encryption.

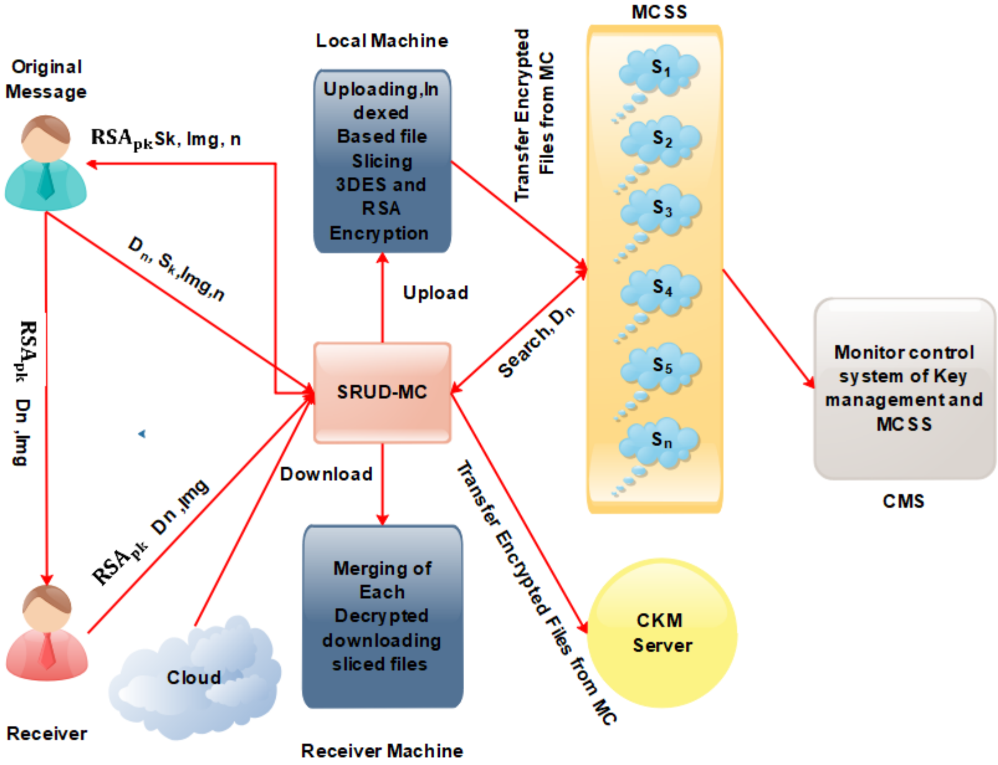
MCSS
Local Machine
Original Message
RSA$_{pk}$Sk, Img, n
Uploading,Indexed Based file Slicing 3DES and RSA Encryption
Transfer Encrypted Files from MC
S$_1$
S$_2$
S$_3$
S$_4$
S$_5$
S$_n$
D$_n$, S$_k$,Img,n
Upload
Search, D$_n$
RSA$_{pk}$ Dn ,Img
SRUD-MC
Monitor control system of Key management and MCSS
CMS
Download
RSA$_{pk}$ Dn ,Img
Merging of Each Decrypted downloading sliced files
Transfer Encrypted Files from MC
CKM Server
Cloud
Receiver
Receiver Machine

Figure 2. SRUD-MC architecture

2.2 SRUD-MC Framework

The SRUD-MC structure functions as a middleware to connect to the MCSS server [16]. The following encryption techniques would be used with the structure outline:

2.2.1 File uploading

At the outset, the establishments stated are the owners of data, which are responsible for attaching the data into the structure over the framework interface [17]. The security of data is emphasized during the switchover process. As a result, the client supplies the slice identity for the file containing the secret key and image [18].

2.2.2 File separation

The process divides up file sharing into several parts for relaxed encryption decisions. The shared file directs users to local server storage [19]. However, the method is also used in the 3DES for encrypting data by smearing the cipher process. The private key is produced using the RSA algorithm based on the most recent and older data that has been kept on the server [20-25].

2.2.3 File encryption and distribution

It is a known fact that the outline transfers the shared file to the multi-cloud server's storage. The file is then only accessible to official parties and is unrestricted from any danger [26]. The installation or bundling of viruses, worms, spyware, Trojan Horses, and other harmful code into files is made possible by file sharing. The likelihood of data being infected by a dangerous virus increases when these files are moved to a multi-cloud storage server. Peer-to-peer (P2P) networks tend to have a higher prevalence, because it is harder to determine whether a file's source is reliable, but organizations are still at danger. Thus, the secret key and file are uploaded via the framework interface utilizing the proposed SRUD-MC for data storage.

Multi-cloud storage server: This is a gathering of various storage amenities which gets linked in a USB interface of application.

Reception of data: The information holder provides such particulars to the receiver, allowing them the admittance to the data confined on the server.

2.2.4 File reconstruction and decryption

This stage consists the following phases:

- The outline displays a popup whenever the receiver arrives following the successful verification, allowing the consumer to choose the particular image for additional dispensation.
- The first step is choosing an image.
- After choosing the image, the customer may have the choice to provide the system with important information. Finally, the file merging technique takes place [27].

2.2.5 File merging

The whole data that the owner sent is received by the receiver. The cryptographic index-based standard is a useful framework to ensure the security of information exchanged among various positions. When the file title and key are applied as input, the scheme runs an automatic process to merge the disparate phases of the file appropriately [28-32].

3. Architecture Overview

The architecture of the proposed approach is explained as follows:

Data owner: This refers to the person in charge of choosing an image to add extra security to the 3DES secret key and to the numerous slices of the shared file. After sharing the file, the outline breaks it up and adopts the RSA key pair creation to encrypt the 3DES secret key before giving the key to the information holder [33]. Figure 2 illustrates the suggested SURD-MC architecture, and Table 1 lists the related abbreviations and descriptions.

- Local machine: This is the component in charge of handling temporary data storage for the shared encrypted files.
- Receiver machine: This device receives decrypted files from the multi-cloud server.
- Cloud monitoring server: This device keeps an eye on the provider's and customer's high-privilege function activities. The super-admin of the cloud platform will be the server manager.
- Cloud key management server: This is the person in charge of keeping track of the encrypted and decrypted keys.
- Data receiver: This device reliably receives data sent by the owner. Nevertheless, they must upload the file and a key.

Table 1. Abbreviation and description

Abbreviation	Abbreviation
$\mathbf{RSA_{pvk}}$	RSA private key
$\mathbf{RSA_{pk}}$	RSA public key
$\mathbf{S_K}$	Secret key
$\mathbf{D_N}$	Customer date name
n	Number of slices
$\mathbf{S_1, S_2, S_3, S_4, ..., S_n}$	File slices without encryption data
$\mathbf{E(S_1), E(S_2), E(S_3), E(S_4), ..., E(S_n),}$	File slices with encryption data
Img	Image

Algorithm 1 examines the method whereby records are shared based on the customer-defined number, but are only allowed to be uploaded to different clouds and dynamic cloud storage services [34-38]. This approach also uses the owner's machine storage for file uploading, indexed dependent sharing, and encryption, aiming to protect against malicious content supplied by malicious users. RSA encryption is implemented to protect the private key and also resolve the key escrow issue. In the end, the global public key is acquired by the owner, a second factor is sent to the cloud database server, and all encrypted share files are then stored on the multi-cloud server.

Algorithm 1: Data file slicing and encryption using SRUD-MC

Step-1:
- **Input:** Data file upload format → (.xpt,jpg,dicm,pdf,video….etc.) and secret key →n,img.

Step-2:
- Upload data file (D) and user identified secret key (S_k)

Step-3
- Determine the data file size (Ds).

Step-4
- Tag n^{th} index value for each slice of data file defined by user.

Step-5
- Create index-based-data file ($\mathbf{S_1, S_2, S_3, S_4, ..., S_n}$) by the original name extension and store it in user local machine.

Step-6
- Initialize encryption of sliced data using RSA and 3DES techniques [utilizing $\mathbf{RSA_{pk}}$ and $\mathbf{RSA_{pvk}}$].

Step-7
- Encrypt each part of the sliced data files [$\mathbf{E(S_1), E(S_2), E(S_3), E(S_4), ..., E(S_n)}$] from local server to be stored in MCSS.

Step-8
- Encrypt data file [$E(S_1)$, $E(S_2)$, $E(S_3)$, $E(S_4)$, ..., $E(S_n)$] and RSA_{pk}, RSA_{pvk}

Step-9
- End

The procedure for file decryption is described in Algorithm 2. The file name, image, and owner's public key are sent under this section via the legitimate document recipient. After recording every detail, the secret key is obtained from the cloud server through RSA decryption. The file names are looked up on the multi-cloud server and then sequentially decrypted on the source of indices before dispensing. The decrypted records are placed in the receiver's location and then concatenated at the index source [39-43].

Algorithm 2: Data file decryption and merging using SDUD-MC

Step-1:
- **Input:** Consider the data file format (Img) with file name without extension (.xpt,jpg,dicm,pdf,video….etc.) and $\mathbf{RSA_{pk}}$**.**

Step-2:
- Verify the obtained correct image.

Step-3:
- Enter data file name (D_n) and public key (P_k).

Step-4:
- Associate the search for data file name with each MCSS tagged directory S_1, S_2, S_3, S_4, ..., S_n and obtain the path of the encrypted files ($E(S_1)$, $E(S_2)$, $E(S_3)$, $E(S_4)$, ..., $E(S_n)$).

Step-5:
- Obtain user defined S_k using P_k and PV_k from cloud servers.

Step-6:
- Decrypt all the encrypted data file using S_k obtained from RSA decryption.

Step-7:
- Merge each sliced part of the decrypted data files [S_1, S_2, S_3, S_4, ..., S_n] from MCSS provider to generate the original data D. Merge the decrypted data file part into a data file *Dn*).

Step-8:
- Remove all encrypted and decrypted tags of each file stored in the respective service.

Step- 9:
- End

4. Results and Discussion

The scheme's advantages include the distribution of keys through untrusted channels, the elimination of file storage in central delivery, clarification of key escrow-related concerns, supervision of the keys in chief checking services, and self-protection from malicious files during upload [44]. Additionally, the system ensures that details about the data cannot be accessed by insiders. The approach therefore aims to eliminate integrity issues from the data recovery process.

4.1 Experimental Setup and Parameters

The effectiveness of our approach was verified using VS2020 C#, which requires the use of the net structure and the safety protocol. The verification was carried out on a 64-bit machine with Windows 10.

In SRUD-MD, the recorded database is uploaded. The slicing index and structure are adopted to locate the uploaded data. The encryption procedure is created using databases that were previously and currently stored on multiple cloud storage servers. The RSA algorithm is utilized to produce the private key based on the most recent and older data that has been kept on the server.

Data has traditionally been categorized based on size and the format in which it is kept on the server. However, the proposed approach divides the data into at least five different private clouds and uses them for the experiments. Our approach was compared with the standard method regarding communication overheads, mobile devices, and energy usage during data transfer into multilevel servers. The comparison shows that our approach improves the turnaround time and decreases power dissipation. Besides, our method uploads data effectively without interruption or data loss [45-50].

4.2 Performance Comparison

Table 2 compares our approach with conventional methods in terms of the uploading/downloading latency time. It can be seen that our approach simplifies the encryption process, and saves the turnaround time for dynamic file slicing. This suits the recent demand of fast computer operations for parallel operations of multi-level cloud data storage processes.

Table 2. Performance comparison

SI.no	FT	D_S(Mb)	Existing methods						Proposed SRUD-MC	
			DSUDS-SC (sec)		SSDS-MC (sec)		USDS-MC (sec)		SRUD-MC (sec)	
			ULT	**DLT**	**ULT**	**DLT**	**ULT**	**DLT**	**ULT**	**DLT**
1.	.pdf	10	9	12	10	10	10	11	9	10
2.	.ppt	15	18	17	15	16	15	17	14	15
3.	.exe	55	16	15	15	16	18	19	14	15
4.	.exl	75	25	28	26	28	24	25	23	24
5.	.jpg	85	30	32	32	34	30	35	30	29
6.	.avi	110	39	40	36	34	36	38	35	35
7.	.flv	220	40	42	40	48	48	45	40	39
8.	.docx	320	38	35	35	39	39	36	34	35

Note: ULT, DLT, and DSUD-SC are short for uploading latency time, downloading latency time, and data security using data slicing over storage clouds, respectively.

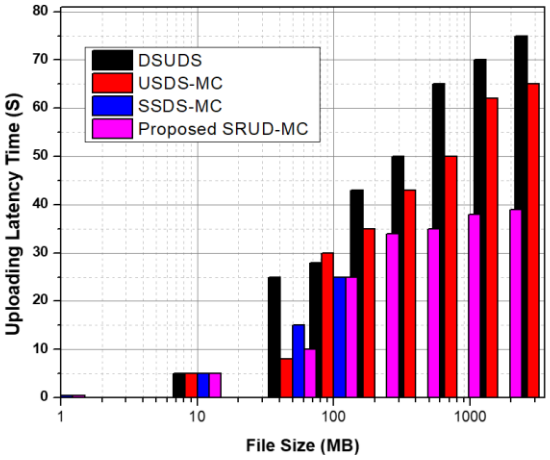

Figure 3. Comparison of uploading latency time (s)

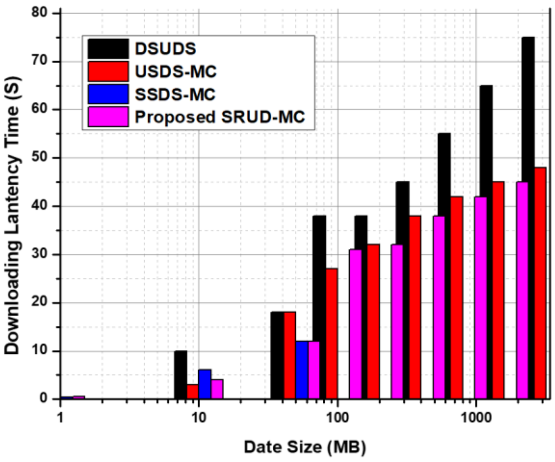

Figure 4. Comparison of downloading latency time (s)

Figure 3 and Figure 4 compare the performance of our method and that of conventional methods in terms of the uploading/downloading latency time for different sizes and formats. It can be seen that our approach SRUD-MC achieves better delay time than the conventional methods. The excellence might be attributed to the dispensation steps of our approach. The task threshold size of the record is 201 Mb and the least threshold of the supplier carriers is 6.

4.3 Security Analysis

4.3.1 Data security

The proposed approach needs to be evaluated in many dimensions of security, such as confidentiality, integrity against insider attacks, and secrecy. This part intends to discuss the data security of our approach, and the combined performance of the said indices. Each dimension was evaluated against a scale of 0-10. In many cases, the cloud-based likelihood of incidence was taken into account. The data security of our approach and existing methods is compared in Table 3.

Table 3. Comparison of data security

SI. No	Security features	SSDS-MC	CP-ABE	DS-MC	USSDS-MC	SRUD-MC
1.	Data Integrity	20%	20%	20%	100%	100%
2.	Confidentiality	75%	75%	80%	100%	100%
3.	Insider Attack	30%	35%	60%	100%	100%
4.	Secret Keys	0%	0%	0%	100%	100%
5.	Privacy	65%	40%	65%	85%	100%

Note: SSDS-MC, CP-ABE, DS-MC, USSDS-MC, and SRUD-MC are short for slice-based secure data storage in multi-cloud environment, ciphertext-policy attribute-based encryption, direct simulation Monte Carlo, unstructured supplementary service data multi-cloud environment, and storage read update and delete multi-cloud environment, respectively.

4.3.2 Privacy

Our approach provides an authenticated procedure for slicing data. An authorized person only has the power to control the slicing of different data formats. Because several third-party servers are used, three unofficial persons are aware of CP-ABE file sharing, and two users are aware of SSDS-MC and DS-MC. If five unauthorized users are able to access the system, then privacy breaches total 100%. DSUDS-MC shows a 100% confidentiality level because only the information owner can notice any file sharing. It has been planned to outline information from end to end, making it simple to obtain while avoiding availability.

4.3.3 Insider attacks

This factor was still assessed using the likelihood of incidence using third-party servers, denial-of-service attacks, conspiring attacks from unhappy customers, the provider environment, information tampering, and repudiation. Further, in SSDS-MC, three out of ten attacks were successful, while in CP-ABE, five out of ten attacks were successful. Attacks on SRUD-MC and USDSMC were unsuccessful, though. Our approach also allowed for the tracking of the insider monitoring service.

4.3.4 Confidentiality

The confidentiality was evaluated based on how many authorized users disclose the secret key and how many file slices are used for each technique. In contrast to other models, where many people were involved, only one person under SRUD-MC knew the key, the image that was chosen, and the number of file sharing. The number of file sharing is fixed when the secret key was discovered. If no one makes a distinction between key and file sharing, privacy is guaranteed.

4.3.5 Malicious file attack

This factor was measured by the zeal with which a data owner seeks to upload a malicious folder to compromise the entire cloud setup. The SRUD-MC and USDS-MC eliminated all harms.

4.3.6 Data integrity

The ability to display data is not maliciously changed while the data is being stored. In other methods, data merging causes so many clatters that it is difficult to tell which information happens first and how the rest is put together. Note that all processes in our approach are automated, such as file sharing, merging, encryption, and decryption. However, these processes are semi-automated in every other method. No information was modified out of the six data entered into the SRUD-MC, demonstrating 100% data integrity. In other methods, five out of six pieces of information are altered in various ways. Figure 5 analyzes the security of the multiple methods. As shown in Figure 4, the proposed approach enhances the security of the multi-cloud platform.

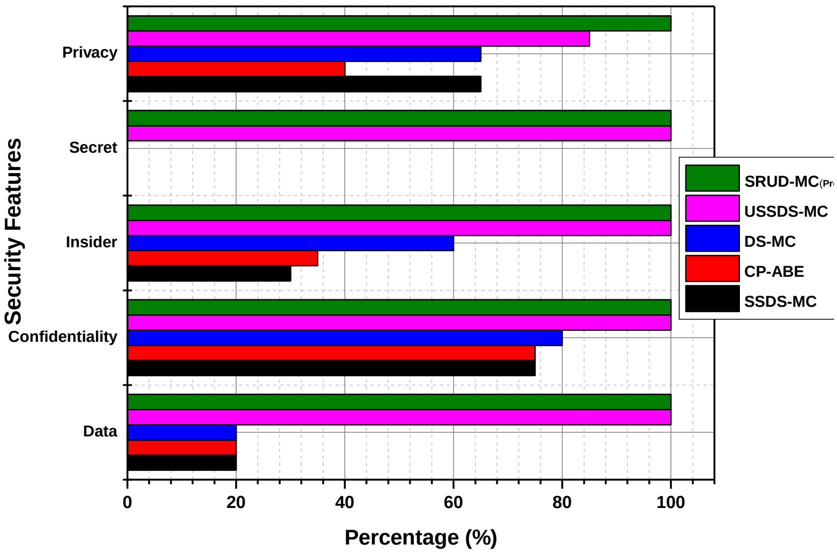

Figure 5. Comparison of security features

Secrecy is assessed by dynamic file sharing. In every contrastive method, all authorized and unauthorized customers have information regarding the file sharing parts. The secure parts depend on the number of providers of storage. By contrast, in the proposed method, Only the owner is able to discern between file parts that considerably increase consumer trust and confidentiality. To ensure anonymity, just one person was able to access the key. When malicious files are posted, no methods can help. If uploaded successfully, the owner's computer becomes ostentatious rather than follow the cloud configuration. When more authorized users are aware of the specifics of their systems' private keys, there is a greater impact on secrecy. Many techniques store the keys that notably upset the concealment using AES encryption and servers run by other parties. The multi-cloud technique measures data integrity as it is being retrieved.

5. Conclusions

This work develops a reconfigurable multi-cloud storage server architecture for unstructured data sharing that is dynamic and secure. In contrast to the current USDS-MC approach, our approach improves the security of unstructured data by 12%. Using 3DEA and RSA throughout data encryption, the USDS-MC reduces the malicious insider to 0.23%. In comparison to USDS-MC, our approach cuts the uploading/downloading latency time (s) down by 10% and 12%, respectively. The proposed framework indexes files using slicing technologies after operating on a variety of file formats. It improves the data sharing process, thanks to its better unstructured data security. It is effective to encrypt and decrypt the sizable unstructured storage in multi-cloud storage servers without losing any information.

If the file can be dynamically sliced, the customers will be more confident. This work improves the data privacy and the migration of the data to multiple clouds. This work can be improved by focusing on the security features of user defined data. More customized features could help to improve secured data sharing in cloud. Every sliced file needs to have a randomly generated key, and data to be retrieved by the key aggregate cryptosystem, making viewing information tedious by attackers. With the dawn of the age of 6G communication, data sharing will reach the speed of 10000 Gb/s, calling for faster encryption and decryption. To adapt to the new age, the proposed approach needs to be further improved to avoid signal attenuation and information loss.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgements

The authors would like to thank JSS Science and Technology University, Mysuru, Dayanand Sagar Academy of Technology and Management, Bengaluru, JSS Academy of Technical Education, Bengaluru, Visvesvaraya technological university (VTU), Belagavi, Vision Group on Science and Technology (VGST), and Karnataka Fund for Infrastructure Strengthening in Science & Technology Level –2 (JSSATEB) for all the support to the research and publication of this paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Wang and Y. Ma, "Unstructured finite-volume lattice Boltzmann method for the multi-group SP3 simulation," *Ann. Nucl. Energy,* vol. 170, Article ID: 109012, 2022. https://doi.org/10.1016/j.anucene.2022.109012.

[2] A. C. R. de Souza, D. K. E. de Carvalho, J. C. A. dos Santos, R. B. Willmersdorf, P. R. M. Lyra, and M. G. Edwards, "An algebraic multiscale solver for the simulation of two-phase flow in heterogeneous and anisotropic porous media using general unstructured grids (AMS-U)," *Appl. Math. Model.,* vol. 103, pp. 792-823, 2022. https://doi.org/10.1016/j.apm.2021.11.017.

[3] H. Ahmed, I. Traore, S. Saad, and M. Mamun, "Automated detection of unstructured context-dependent sensitive information using deep learning," *Internet Things-Neth.,* vol. 16, Article ID: 100444, 2021. https://doi.org/10.1016/j.iot.2021.100444.

[4] S. Mallikarjunaswamy, N. Sharmila, G. K. Siddesh, K. R. Nataraj, and M. Komala, "A novel architecture for cluster based false data injection attack detection and location identification in smart grid," In *Advances in Thermofluids and Renewable Energy,* Singapore, October 22, 2021, P. Mahanta, P. Kalita, A. Paul, and A. Banerjee (Eds.), Springer, pp. 599-611. https://doi.org/10.1007/978-981-16-3497-0_4.

[5] R. Shivaji, K. R. Nataraj, S. Mallikarjunaswamy, and K. R. Rekha, "Implementation of an effective hybrid partial transmit sequence model for peak to average power ratio in MIMO OFDM system," In *ICDSMLA 2020,* Singapore, November 9, 2021, A. Kumar, S. Senatore, and V. K. Gunjan (Eds.), Springer, pp. 1343-1353. https://doi.org/10.1007/978-981-16-3690-5_129.

[6] C. Jain and A. Khunteta, "Novel approach search-associative based searching and secure unstructured data transfer," In *2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering, (ICRAIE),* Jaipur, India, November 22-25, 2018, IEEE, pp. 1-4. https://doi.org/10.1109/ICRAIE.2018.8710385.

[7] Y. C. Yau, P. Khethavath, and J. A. Figueroa, "Secure pattern-based data sensitivity framework for big data in healthcare," In *2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering, (BCD),* Honolulu, HI, USA, May 29-31, 2019, IEEE, pp. 65-70. https://doi.org/10.1109/BCD.2019.8885114.

[8] Z. Wu, E. Zhu, H. Zou, and J. Lu, "Design of unstructured data transmission protocol for electrical information acquisition system," In *2021 13th International Conference on Measuring Technology and Mechatronics Automation, (ICMTMA),* Beihai, China, January 16-17, 2021, IEEE, pp. 335-340. https://doi.org/10.1109/ICMTMA52658.2021.00078.

[9] T. N. Manjunath, S. Mallikarjunaswamy, M. Komala, N. Sharmila, and K. S. Manu, "An efficient hybrid reconfigurable wind gas turbine power management system using MPPT algorithm," *Int J. Power Electron. Drive Syst.,* vol. 12, no. 4, pp. 2501-2510, 2021. https://dx.doi.org/10.11591/ijpeds.v12.i4.pp2501-2510.

[10] R. Shivaji, K. R. Nataraj, S. Mallikarjunaswamy, and K. R. Rekha, "Design and implementation of reconfigurable DCT based adaptive PST techniques in OFDM communication system using interleaver encoder," *Indian J. Sci. Technol.,* vol. 13, no. 29, pp. 2108-2120, 2020. https://doi.org/10.17485/IJST/v13i29.976.

[11] H. Shekhawat, S. Sharma, and R. Koli, "Privacy-preserving techniques for big data analysis in cloud," In *2019 Second International Conference on Advanced Computational and Communication Paradigms, (ICACCP),* Gangtok, India, 2019, IEEE, pp. 1-6. https://doi.org/10.1109/ICACCP.2019.8882922.

[12] S. Bhadlawala and K. Chachapara, "Proposed two layer cryptography key generation for off-premise cloud computing," In *2015 5th Nirma University International Conference on Engineering, (NUiCONE),* Ahmedabad, India, 2015, IEEE, pp. 1-6. https://doi.org/10.1109/NUICONE.2015.7449609.

[13] M. Sasitharagai, A. Renuga, A. Padmashree, and T. Rajendran, "Trust based communication in unstructured P2P networks using reputation management and self certification mechanism," In *2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends, (AICERA),* Kottayam, India, 2012, IEEE, pp. 1-9. https://doi.org/10.1109/AICERA.2012.6306737.

[14] M. R. Sumalatha, K. Praveenraj, and C. Selvakumar, "SK-IR: Secured keyword based retrieval of sensor data in cloud," In *2013 International Conference on Recent Trends in Information Technology, (ICRTIT),* Chennai, India, 2013, IEEE, pp. 341-346. https://doi.org/10.1109/ICRTIT.2013.6844227.

[15] M. L. Umashankar, S. Mallikarjunaswamy, and M. V. Ramakrishna, "Design of high speed reconfigurable distributed life time efficient routing algorithm in wireless sensor network," *J. Comput. Theor. Nanos.,* vol. 17, no. 9-10, pp. 3860-3866, 2020. https://dx.doi.org/10.1166/jctn.2020.8975.

[16] M. L. Umashankar, M. V. Ramakrishna, and S. Mallikarjunaswamy, "Design of high speed reconfigurable deployment intelligent genetic algorithm in maximum coverage wireless sensor network," In *2019 International Conference on Data Science and Communication, (IconDSC)*, Bangalore, India, 2019, IEEE, pp. 1-6. https://dx.doi.org/10.1166/jctn.2020.8975.

[17] P. Satish, M. Srikantaswamy, and N. K. Ramaswamy, "A comprehensive review of blind deconvolution techniques for image deblurring," *Trait. Signal,* vol. 37, no. 3, pp. 527-539, 2020. https://dx.doi.org/10.18280/ts.370321.

[18] T. Wang and J. Yao, "An improved embedded discrete fracture model and domain connectivity algorithms on 3D unstructured grids," *J. Comput. Phys.,* vol. 459, Article ID: 111142, 2022. https://doi.org/10.1016/j.jcp.2022.111142.

[19] B. Xie, Y. Huang, and F. Xiao, "A high-fidelity solver based on hybrid numerical methods on unstructured grids for incompressible multiphase flows," *J. Comput. Phys.*, vol. 463, Article ID: 111299, 2022. https://doi.org/10.1016/j.jcp.2022.111299.

[20] X. Su, M. Zhang, D. Zou, Y. Zhao, J. Zhang, and H. Su, "Numerical scheme for solving the Richard's equation based on finite volume model with unstructured mesh and implicit dual-time stepping," *Comput. Geotech.*, vol. 147, Article ID: 104768, 2022. https://doi.org/10.1016/j.compgeo.2022.104768.

[21] S. Shebin and S. Mallikarjunaswamy, "A software tool that provides relevant information for diabetic patients to help prevent diabetic foot," *IOSR J. Comput. Eng.,* vol. 16, no. 2, pp. 69-73, 2014. https://dx.doi.org/10.9790/0661-16296973.

[22] S. Mallikarjunaswamy, K. R. Nataraj, P. Balachandra, and N. Sharmila, "Design of high speed reconfigurable coprocessor for interleaver and de-interleaver operations," *J. Impact Factor,* vol. 6, no. 1, pp. 30-38, 2015.

[23] S. Shebin and S. Mallikarjunaswamy, "A review on clinical decision support system and its scope in medical field," *Int J. Eng. Res. & Technol.,* vol. 2, no. 13, pp. 417-420, 2018. https://doi.org/10.17577/IJERTCONV2IS13085.

[24] N. Sharmila, K. R. Nataraj, and K. R. Rekha, "An efficient dynamic power management model for a stand-alone DC Microgrid using CPIHC technique," *Int J. Power Electron. Drive Syst.,* vol. 12, no. 3, pp. 1439-1449, 2021. https://dx.doi.org/10.11591/ijpeds.v12.i3.pp1439-1449.

[25] Y. Chen, Y. Xiong, B. Zhang, J. Zhou, and Q. Zhang, "3D point cloud semantic segmentation toward large-scale unstructured agricultural scene classification," *Comput. Electron. Agr.,* vol. 190, Article ID: 106445, 2021. https://doi.org/10.1016/j.compag.2021.106445.

[26] L. Zhou and Y. Zhao, "Improvement of unresolved CFD-DEM by velocity field reconstruction on unstructured grids," *Powder Technol.,* vol. 399, Article ID: 117104, 2022. https://doi.org/10.1016/j.powtec.2021.117104.

[27] M. Hully, T. Lo Barco, A. Kaminska, G. Barcia, C. Cances, and C. Mignot, "Deep phenotyping unstructured data mining in an extensive pediatric database to unravel a common KCNA2 variant in neurodevelopmental syndromes," *Genet. Med.*, vol. 23, no. 5, pp. 968-971, 2021. https://doi.org/10.1038/s41436-020-01039-z.

[28] S. Mallikarjunaswamy, K. R. Nataraj, and K. R. Rekha, "Design of high-speed reconfigurable coprocessor for next-generation communication platform," In *Emerging Research in Electronics, Computer Science and Technology, (ERECST)*, New Delhi, 2014, Springer, pp. 57-67. https://doi.org/10.1007/978-81-322-1157-0_7.

[29] H. N. Mahendra, S. Mallikarjunaswamy, V. Rekha, V. Puspalatha, and N. Sharmila, "Performance analysis of different classifier for remote sensing application," *Int J. Eng. Adv Technol.,* vol. 9, no. 1, pp. 7153-7158, 2019. http://dx.doi.org/10.35940/ijeat.A1879.109119.

[30] S. Thazeen, S. Mallikarjunaswamy, G. K. Siddesh, and N. Sharmila, "Conventional and subspace algorithms for mobile source detection and radiation formation," *Trait. Signal,* vol. 38, no. 1, pp. 135-145, 2021. https://dx.doi.org/10.18280/ts.380114.

[31] E. U. Gallardo-Romero and D. Ruiz-Aguilar, "High order edge-based finite elements for 3D magnetotelluric modeling with unstructured meshes," *Comput. Geosci.,* vol. 158, Article ID: 104971, 2022. https://doi.org/10.1016/j.cageo.2021.104971.

[32] S. Gul, S. Räbiger, and Y. Saygın, "Context-based extraction of concepts from unstructured textual documents," *Inform. Sciences,* vol. 588, pp. 248-264, 2022. https://doi.org/10.1016/j.ins.2021.12.056.

[33] L. Cheng, X. Deng, B. Xie, Y. Jiang, and F. Xiao, "A new 3D OpenFoam solver with improved resolution for hyperbolic systems on hybrid unstructured grids," *Appl. Math. Model.,* vol. 108, pp. 142-166, 2022. https://doi.org/10.1016/j.apm.2022.03.022.

[34] S. Chaitra, V. Rekha, A. M. Harisha, T. A. Madhu, S. Mallikarjunaswamy, N. Sharmila, and H. N. Mahendra, "A comprehensive review of parallel concatenation of LDPC code techniques," *Indian J. Sci. Technol.,* vol. 14, no. 5, pp. 432-444, 2021. https://doi.org/10.17485/IJST/v13i20.459.

[35] M. L. Umashankar, T. N. Anitha, and S. Mallikarjunaswamy, "An efficient hybrid model for cluster head selection to optimize wireless sensor network using simulated annealing algorithm," *Indian J. Sci. Technol.,* vol. 14, no. 3, pp. 270-288, 2021. https://doi.org/10.17485/IJST/v14i3.2318.

[36] S. AC and M. N. Jayaram, "Development of energy efficient and secure routing protocol for M2M communication," *Int J. Performability Eng.*, vol. 18, no. 6, pp. 426-433, 2022. https://doi.org/10.23940/ijpe.22.06.p5.426-433.

[37] K. Shashi Raj, G. K. Siddesh, S. Mallikarjunaswamy, and K. Vivek Raj, "Interference resilient stochastic prediction based dynamic resource allocation model for cognitive MANETs," *Indian J. Sci. Technol.*, vol. 13, no. 41, pp. 4332-4350, 2020. https://doi.org/10.17485/IJST/v13i41.687.

[38] V. B. Chenam and S. T. Ali, "A designated cloud server-based multi-user certificateless public key authenticated encryption with conjunctive keyword search against IKGA," *Comput. Stand. Inter.*, vol. 81, Article ID: 103603, 2022. https://doi.org/10.1016/j.csi.2021.103603.

[39] Q. Wu, T. Lai, L. Zhang, Y. Mu, and F. Rezaeibagha, "Blockchain-enabled multi-authorization and multi-cloud attribute-based keyword search over encrypted data in the cloud," *J. Syst. Architect.*, vol. 129, Article ID: 102569, 2022. https://doi.org/10.1016/j.sysarc.2022.102569.

[40] M. Li, G. Wang, S. Liu, and J. Yu, "Multi-keyword fuzzy search over encrypted cloud storage data," *Procedia Comput. Sci.*, vol. 187, pp. 365-370, 2021. https://doi.org/10.1016/j.procs.2021.04.075.

[41] D. Y. Venkatesh, K. Mallikarjunaiah, and M. Srikantaswamy, "A comprehensive review of low density parity check encoder techniques," *Ing. Sys. Info.*, vol. 27, no. 1, pp. 11-20, 2022. https://doi.org/10.18280/isi.270102.

[42] R. Mishra, D. Ramesh, D. R. Edla, and L. Qi, "DS-Chain: A secure and auditable multi-cloud assisted EHR storage model on efficient deletable blockchain," *J. Ind. Inf. Integr.*, vol. 26, pp. 1-13, 2022. https://doi.org/10.1016/j.jii.2021.100315.

[43] O. Olakanmi and K. Odeyemi, "Faster and efficient cloud-server-aided data de-duplication scheme with an authenticated key agreement for Industrial Internet-of-Things," *Internet Things-Neth.*, vol. 14, pp. 1-12, 2021. https://doi.org/10.1016/j.iot.2021.100376.

[44] S. Thazeen, S. Mallikarjunaswamy, M. N. Saqhib, and N. Sharmila, "DOA method with reduced bias and side lobe suppression," In *2022 International Conference on Communication, Computing and Internet of Things, (IC3IoT)*, Chennai, India, March 10-11, 2022, IEEE, pp. 1-6. https://doi.org/10.1109/IC3IOT53935.2022.9767996.

[45] P. S. Challagidad and M. N. Birje, "Efficient multi-authority access control using attribute-based encryption in cloud storage," *Procedia Comput Sci.*, vol. 167, pp. 840-849, 2020. https://doi.org/10.1016/j.procs.2020.03.423.

[46] Y. Gupta, "Novel distributed load balancing algorithms in cloud storage," *Expert Syst. Appl.*, vol. 186, pp. 1-23, 2021. https://doi.org/10.1016/j.eswa.2021.115713.

[47] H. N. Mahendra and S. Mallikarjunaswamy, "An efficient classification of hyperspectral remotely sensed data using support vector machine," *Int J. Electron. Telec.*, vol. 68, no. 3, pp. 609-617, 2022. https://doi.org/10.24425/ijet.2022.141280.

[48] H. N. Mahendra, S. Mallikarjunaswamy, C. B. Nooli, M. Hrishikesh, N. Kruthik, and H. M. Vakkalanka, "Cloud based centralized smart cart and contactless billing system," In *2022 7th International Conference on Communication and Electronics Systems, (ICCES)*, Coimbatore, India, June 22-24, 2022, IEEE, pp. 820-826. https://doi.org/10.1109/ICCES54183.2022.9835856.

[49] P. Dayananda and M. Srikantaswamy, "Efficient detection of faults and false data injection attacks in smart grid using a reconfigurable Kalman filter," *Int J. Power Electron. Drive Syst.*, vol. 13, no. 4, pp. 2086-2097, 2022. http://doi.org/10.11591/ijpeds.v13.i4.pp2086-2097.

[50] S. Rathod and N. K. Ramaswamy, "An efficient reconfigurable peak cancellation model for peak to average power ratio reduction in orthogonal frequency division multiplexing communication system," *Int J. Electr Comput Eng.*, vol. 12, no. 6, pp. 6239-6247, 2022. http://doi.org/10.11591/ijece.v12i6.pp6239-6247.