Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Dynamic and secure unstructured data sharing in multi-cloud storage using the hybrid crypto-system

K. Subramanian [1], F. Leo John [2, *]

[1]P. G. and Research Department of Computer Science, H. H. The Rajah's College, Pudukkottai, India
[2]P. G. and Research Department of Computer Science, J. J College of Arts and Science (Autonomous), Pudukkottai, India

ARTICLE INFO

Article history:
Received 25 May 2017
Received in revised form
18 October 2017
Accepted 11 November 2017

Keywords:
Multi-cloud storage
Malicious insider
Malicious files
Dynamic index based data slicing
Cryptography
Data sharing
3DES

ABSTRACT

The aim of the paper is to offer an architectural framework to help in securing data sharing processes in the multi-cloud storage. For example, the paper highlights the ways of achieving a secure data sharing through the application of the cryptographic index-based data slicing techniques. It seeks to prevent malicious insider through data encryption using 3DES in every part of the file and RSA for the encryption of private key. In addition this work also enhances the privacy of secure data sharing using dynamic file slicing in which customer can define the number of file parts to be sliced using the framework interface. This architecture solves the key management and key distribution problem. The proposed algorithm uses self-protection or counter attack mechanism in order to safe guard cloud infrastructure from malicious files. The cloud-storage of information is a security-intensive process because a multi-cloud process entails a collection of storage servers. The framework only searches for the file in the multi-cloud server when the receiver inputs the file name and the private key. Moreover, it is the only way of accessing the file. This study focused on multi-cloud storage. The life cycle of the data involves three stages that include data input, transition, and usage. The report is more about cloud storage and transition. The experimental and numerical results shows the efficiency of the various file formats and privacy gets increased through dynamic file slicing which in turn increases the trustworthy of the customers. The data security increase during cloud-storage due to cryptographic and encryption techniques. Cloud storage is important for the organization those who are doing online process. Therefore, it is benefit to them to access secure data storage and transfer processes.

1. Introduction

The security of data storage is important because it offers an opportunity for one to access secure data later. The paper seeks to address the process of data security for multi-cloud storage operations in its architectural framework. A secure data sharing process is necessary for every information system because it protects the information from any danger (Subramanian and John, 2017a). Data threats are common in the modern society due to the increase in knowledge of information security systems in public.

In particular, the threat emanates from the existence of external hackers and malicious insiders. Noteworthy is that they tend to manipulate the customer information in the system and interfere with the arrangement and the integrity of said data (Han et al., 2017). The most common danger posed by hackers is accessing the information illegally and using it to intimidate the respective organization.

Among other risks relating to multi-cloud storage that makes it necessary to adopt better storage mechanisms for the information that include the interference of confidentiality, lack of information integrity, intrusion, and loss of the information (Satapathy et al., 2016). Ideally, the sources of such threats take advantage of the fact that some of the organizations cannot afford the high-cost single cloud platforms. Another common threat is that when the cloud is inaccessible, there is a negative impact on the data delivery processes.

The adoption of cloud systems is a standard mechanism for data storage, and it attracted most of the developing organizations (Borazjani, 2017). Therefore, the gap that exists in the process is an amicable solution to data security threats. The hope

* Corresponding Author.
Email Address: stleojohn@gmail.com (F. Leo John)
https://doi.org/10.21833/ijaas.2018.01.003
2313-626X/© 2017 The Authors. Published by IASE.

of finding a lasting settlement in multi-cloud storage, cryptographic index-based slicing techniques takes priority as one of the measures to ensure the security of data transferred between systems and different storage media in an organization.

The remainder of the paper is formed as follows. Section 2 describes the overview of the related work in the field. Section 3 discusses the proposed System model. Section 4 describes the overview of architecture, components and its operating activity with algorithms. Section 5 explains the experimental solutions, and Section 6 Concludes the report and future work.

2. Literature survey

Many research studies have been conducted in different periods to determine the best way to achieve the goals of information security and privacy in cloud computing. However, most of the researchers provide advice towards the use of an architecture integrated with encryption. The encryption would allow specific admittance authorization and cryptography as a process of enforcing the security of information in the cloud storage system. Data or file slicing is a process that targets inclusivity in data security especially in cloud-storage (Vaidya and Nehe, 2015). This model does not guarantee safe key management and key distribution. In addition file slicing is not focused in which cloud admin or super user from the providers environment support is required during the file merging process. Private cloud database remains unmonitored which leads to insider attacks. This greatly reduces the efficiency of the algorithm. Improvement in security standards is the major factor in the system of cloud computing (Razaque et al., 2016). The authors considered data sharing algorithm as effective in enabling the spread of information along the different cloud in the cloud computing systems. However, the approach fails to pay adequate attention to the distribution of keys in secure information channels. As a result, it is possible to corrupt the integrity of information, a process that commonly occurs in the retrieval process.

The work of Balasaraswathi and Manikandan (2014) introduced two clouds are used for file storage, and another cloud is employed for storage of metadata of files such as file access paths, passwords, and secret keys. Since data stays in the cloud storage for quite long, the two cloud providers can collude to breach the security of the underlying data. There is a constant update of file access paths under this model. The decryption process during the update of file access paths requires intensive computation.

Ali et al. (2017) and Levitin et al. (2017) proposed a secure data transfer process in the cloud model. In particular, the authors considered the prevention of the invasion to be malicious insiders. However, the authors proposed a clouding model that does not prevent the colliding attacks that are common because the third-party server contributes in the process of maintaining part of the key. On the same note, the approach only uses a single cloud storage process and centralized data distribution.

The work of Fabian et al. (2015) proposed architecture to share health care records using Attribute Based Encryption and cryptographic secret sharing. This approach does not guarantee the malicious attacks, data integrity and efficiency of the overall process such as uploading, file slicing, and group sharing and so on requires huge computation.

To ensure secure data sharing in a Multi-Cloud, Xu et al. (2015) proposed similar to the above model slice based secure data sharing. This model does not support the video files and Meta table remains unsecured, which leads to malicious insider attacks. In the work of Zibouh et al. (2016) Multi-Cloud architecture has been proposed with fully homomorphic encryption to enhance the performance and the time of data processing. However if the size of the file increases computation overhead arises. Another potential drawback is homomorphic cryptosystems are vulnerable to malware.

The work of Subramanian and John (2017b) which uses index based cryptography data slicing to overcome the data integrity and file merging conflicts in the retrieval process. This model uses AES encryption to encrypt sliced parts of the file. However, this model possesses additional burden to the owners when the number of files uploaded gets increased since the key management is done by the owner. The proposed model is very similar to the work of Subramanian and John (2017b) which uses RSA encryption to enhance the security of the private key used by AES encryption. It also solves the key management and key distribution problem arises in the previous approaches. However this approach does not guarantee the privacy and trustworthy to the customers.

Two data encryption mechanisms include the Triple Data Encryption Standards (3DES) and the Rivest-Shamir-Adleman (RSA) encryption. However, one common challenge in the utilization of the 3DES encryption process is the fact that it is vulnerable to brute force attack or meets in the middle attack. To enhance the security and efficiency of 3DES dynamic file slicing along with RSA encryption is used. Therefore, it commonly serves as a reliable encryption to user data. The system tends to pass the common encrypted common keys for the symmetric key cryptography. The latter can perform bulk encryption and encryption procedures faster.

Therefore, the current study seeks to develop a framework that solves the challenges mentioned above. The proposed architecture will involve the application of at least the five-cloud storage services. To enhance the privacy the proposed scheme uses dynamic file slicing. None of the providers or even receivers does not have any knowledge about the file parts since dynamic file slicing is used. It also increases the trustworthy of the customers. In cases in which file indexing is inapplicable, the file coding

enables an arrangement of the different parts of the sliced file accordingly. Therefore, due to the different codes allocated to the different parts of the file during slicing, it is possible to arrange them appropriately during file merging. The file coding process also enables the framework to maintain a greater level of information integrity. Additionally, in the current framework, the system supports all file formats. Since other works may not support the video file formats, information owners resort to using cryptographic systems, which guarantee adequate information security. The third-party server will not be one of the key maintenance approaches. However, the process will restrict the task to the owner such that the third party also exists as a foreign body to prevent colluding attacks. Additionally, the security of the major distribution processes will take precedence to prevent malicious insiders and other threats to information. The integration of architecture with the encryption and cryptography processes makes it easy to conserve data in a secure transfer, storage, and usage process. It is the most efficient technique for maintaining data security in the multi-cloud surrounding.

3. Proposed system design

The Fig. 1 shows the architecture of the proposed system. According to the architecture, data owner transmits the file, an image and the secret key via the framework interface. The file is uploaded to the Dynamic and Secure Unstructured Data Sharing in Multi-Cloud (DSUDS-MC) by the framework and indexed based slicing and encryption consequently performed on the files before being transferred to the multi-cloud storage server. Furthermore, the secret key is also encrypted using the RSA keys and a portion of the RSA public key transmitted to the owner and RSA private key to the cloud database server. The decryption phase also involves a number of processes. For instance, upon receiving the necessary credentials from the owner, the filename, an image and the public key are transmitted using the untrusted or semi-trusted channel. Only after choosing the correct image key details are entered for decryption and merging process. The file name is searched and the private key is used to decrypt the sliced files to the receiver's computer. In particular, dynamic file slicing is the most effective process because the secure data transfer process will be inclusive. The framework offers a clear path through which the user sends the file. File slicing takes place before data encryption occurs.

3.1. DSUDS-MC Framework

The DSUDS-MC framework act as a middleware or web API to connect with Multi-cloud server. The architectural framework will make use of the encryption techniques highlighted below:

a) **File uploading**: In the framework, the institutions mentioned are the data owners. The data owner is responsible for uploading the information into the system through the framework interface. The focus in the transition process is data security. Therefore, the user enters the number of slices for the file with the private key and image.

b) File separation: The process involves the file slicing into unrelated parts for easy encryption purposes. The sliced file goes to storage on the local server. However, the process is common in the 3DES algorithm, which applies the cipher algorithm to encrypt. For the secret key, the encryption process makes use of the RSA algorithm before the file goes to storage.

c) Encryption and distribution of files: It is the point at which the framework transfers the sliced file into the multi-cloud server for storage. At that point, the file is accessible only to the authorized parties and free from any threat. In the same aspect, the framework encrypts each section of the sliced file such that there is no possibility of interference with data integrity and confidentiality.

d) Multi-cloud storage server: It is a collection of several storage services that gets connected through a single application interface.

e) Reception of data: The owner of the data gives such details to the receiver to enable them access to the information contained on the server. In particular, the owner takes control of the process by retaining the power to allow other users to access the information selectively.

f) Reconstruction of the file and decryption: The reconstruction and decryption process is a systematic process that involves sets of steps:

a) For instance, once the receiver enters after the successful verification,
b) The framework will display a window through which the user can choose the correct image for additional processing.
c) The choice of an image comes first.
d) After choosing the image, the user can access a chance to enter key details into the system and finally file merging process materializes.

g) Merging of Files: The receiver gets full information as sent by the owner. As a result, the cryptographic index-based protocol is a useful framework for ensuring the security of data transferred between different points. The system runs an automatic procedure that enables it to merge the different sections of the file appropriately once the name of the file and the key are input. .

4. Architecture overview

The following is a description of the architecture components of the proposed system.

Data owner: It involves the party responsible for choosing an image as an additional security to the 3DES private key and the several slices of the slice file. After slicing the file, the framework disintegrates the file and encrypts the 3DES private key using the

RSA key couple generation process and sends the slice of the key to the data owner.

Local machine: It is the entity responsible for temporary data storage for the encrypted sliced files.

Receiver machine: An entity receives decrypted files from the multi-cloud server.

Cloud monitoring server: An entity monitors the activities of the consumer and provider's high privilege operator actions. The super-admin of the cloud-platform is the manager of the server.

Cloud key management server: The party responsible for the management of the encryption and decryption keys.

Data receiver: It is the entity that securely receives data conveyed by the owner. However, they must upload a key and the file name.

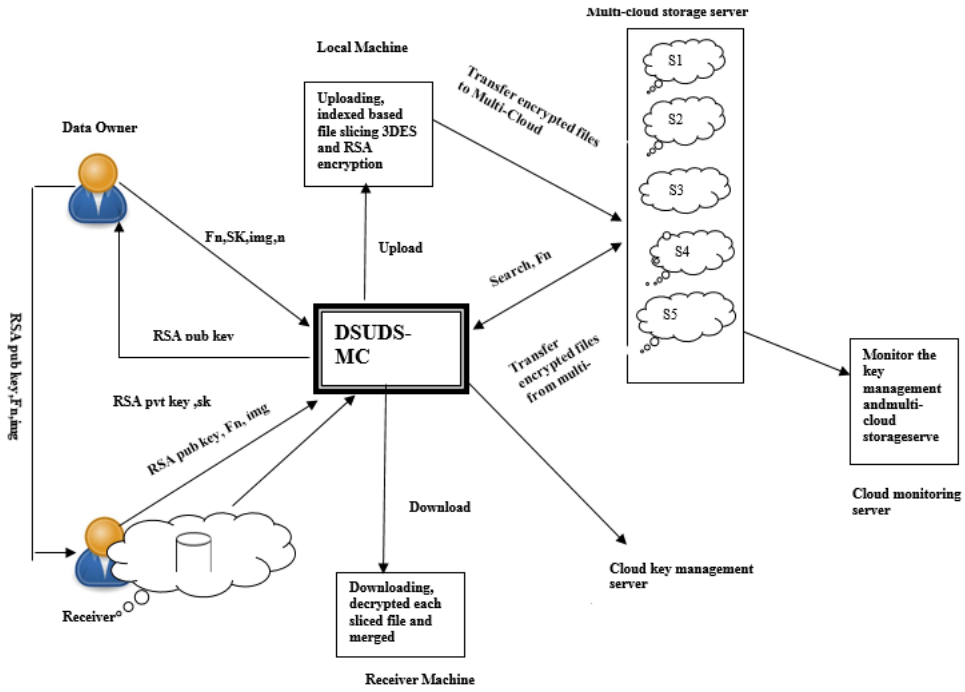

Fig. 1: DSUDS-MC architecture

Table 1 Shows the Notation and description used in the Fig. 1.

Table 1: Notation and description

Acronym	Description
F/FN	User's File Name to be uploaded/protected
n	Number of Slices
img	Image
F.1, F.2..Fn	Sliced parts of the file without encryption
E(F.1),E(F.2).....E(Fn)	Sliced parts of the Encrypted File
SK	Secret Key
RSA pub key	RSA public key
RSA pvt key	RSA private Key

Algorithm-1 Examines the method by means of which records are sliced based on the user defined number, but limited to active cloud storage services and uploaded to diverse clouds. This procedure also uses owner's machine storage for file upload process, indexed based slicing and encryption procedure so that it will guard from the malicious document uploaded by the malicious user, RSA encryption is used to protect the secret key and also solves the key escrow problem. As a final result, the owner gets the general public key and any other component is sent to the cloud database server, finally, the entire encrypted slice documents get stored at the multi-cloud server.

Algorithm 1- DSUDS-MC file slicing and encryption

Input*: Any file (.xpt, .dicm, video etc.), secret key,n,img*

Output*: Encrypted Files E (F.1), E (F.2), E (F.3), E (F.4), and E (F.5), RSA pub key and RSA pvt key*

Step 1:

Uploads a file (F) and assign user defined secret key (SK)

Step 2:

Find the size of a file (SF)

Step 3:

Slice or Divide the size of a file (SF) by the user defined nth value.

Step 4:

Index based files (F.0, F.1, F.2, F.3 and F.4) are created with the same file name and get stored in the owner's local machine.

Step 5:

Use RSA encryption to asymmetrically encrypt the user defined secret key assigned in 3DES encryption process. Publish the RSA pub key to the owner and the other part RSA pvt key to the cloud database server.

Step 6:

Encrypt each part of the sliced file E (F.1), E (F.2), E (F.3), E (F.4), and E (F.5) from local server and stored in the Multi Cloud server.

Step 7:

End

Algorithm-2 describes the technique of file decryption. Beneath this section, the document name, image along with the public key obtained from the owner is handed via the valid document recipient. In one case the correct icon is taken in the framework interface the receiver can go into the file and key details. After recording all the above details private key is found out from the cloud server using RSA decryption. The filenames are searched in the

multi-cloud server and then sequentially decrypted on the basis of indices earlier assigned. The decrypted documents are stored in the vicinity of the receiver and finally merged on the basis of the indices.

Algorithm-2 DSUDS-MC File Decryption and Merging
Input:
Img, File Name without Extension (.xpt, .dicm, video etc.), RSA Pub key (PK)
Output*: Decrypted File parts and Merged To get File (F)*
Perform:
Step 1:
Choose the correct image for verification process.
Step 2:
Enter or Pass that File Name (FN) and public Key (PK)
Step 3:
Perform a search with the filename associated in each Multi Cloud storage service provider directory (F.0, F.1, F.2, F.3 and F.4) and obtain the path of the encrypted files E (F.1), E (F.2), E (F.3), E (F.4) and E(F.5).
Step 4:
Obtain the user defined secret key (SK) using owner's public key (PK) and private key from the cloud server. Decrypt all the encrypted file parts using secret key obtained from RSA decryption.
Step 5:
Merge each part of the decrypted files F1, F2, F3, F4, and F5 from Multi Cloud storage service provider to obtain the original file F.
Step 6:
Auto removal of all decrypted and encrypted parts of the files stored in the respective services.
Step 7:
End

5. Implementation

The benefits derived from the system include the distribution of keys through the untrusted canals, removal of the central delivery of file storage; solve the key escrow problems that emanate in the process, the management of the keys in central monitoring services, and self-protection from the malicious files during upload. Additionally, the system ensures that insiders cannot access details about the data. Therefore, the system targets the removal of integrity conflicts in data retrieval process.

5.1. Experimental setup

The evaluation of the effectiveness of the structure involved the implementation of the Visual Studio 2010 C#. The procedure entailed the use of the net framework and the security analysis protocol. The process was achievable using a 64-bit machine running on Windows 7 Professional. The machine operates using an Intel Core (TM) 2 Duo CPU T6500 with a speed of 2.10GHz. The DDR3 RAM is 4GB.

The procedure involved a classification of the clouds, and the system could use at least five multiple private clouds for each file involved in the experimental setup. Data Security using data slicing over storage clouds (Vaidya and Nehe, 2015), Secure and reliable Unstructured Data Sharing in Multi-Cloud Storage using the Hybrid Crypto System(USDSMC)(Subramanian and John, 2017b), Slice-based Secure Data Storage in MultiCloud Environment(SSDS-MC) (Junghanns et al. 2017) Compared to communication overhead, energy consumption of the mobile devices used for data transfer to multi-cloud servers, turnaround time improves significantly when the file upload and encryption processes occur smoothly and free from interference. In particular, the framework offers the latency time for overall uploading process and downloading process.

5.2. Performance analysis

The performance of the proposed algorithm is provided on Table 2. The graph has been constructed from the above table for the comparison of latency Time for uploading process and Downloading Process. Though 3DES is used through dynamic file slicing the turn-around time to complete the encryption process has been greatly reduced in the proposed scheme especially for the large file sizes (Mb). However today, fast computers have significantly reducing the execution times related to the multi-cloud storage. This storage service follows the parallel processing to complete the task.

Table 2: Latency time comparison

S. No	FT	FS (MB)	Existing Multi-cloud Storage Approaches						Proposed Scheme	
			Data security using data slicing over storage clouds (secs)		USDSMC (secs)		SSDS-MC (secs)		DSUDS-MC (secs)	
			Uploading Latency Time	Downloading Latency Time	Uploading Latency Time	Downloading Latency Time	Uploading Latency Time	Downloading Latency Time	Uploading Latency Time	Downloading Latency Time
1	.docx	1	1	2	0.9	2	1	1	0.9	2
2	.pdf	10	13	9	2	2	6	6		
3	.exe	50	15	17	16	17	14.5	15	4	4.5
4	.avi	100	30	35	28	29	-	-	14.5	15
5	.flv	200	36	35	29	30	-	-	28 29	30 30

Data Security using data slicing over storage clouds (Vaidya and Nehe, 2015), USDSMC (Subramanian and John, 2017b). SSDS-MC-DSUDS-MC (proposed approach)

Fig. 2 shows the uploading latency time of various approaches. The latency time is the time taken to switch the complete information to the multi-cloud storage from the uploading system. It is to be noted that proposed scheme has had lesser time in phrases of seconds for the numerous file sizes and record formats. The existing method (Xu et al., 2015) do not support video file format. Within the current techniques, encryption is managed for the complete data at the beginning, after which gets sliced however inside the proposed method record is sliced first and then encrypted all the sliced documents on the same time or parallel. This greatly enhances the performance of the turnaround time. When records theft or loss has taken place it gets rectified right away considering index based slicing is used lacking components may be without difficulty retrieved.

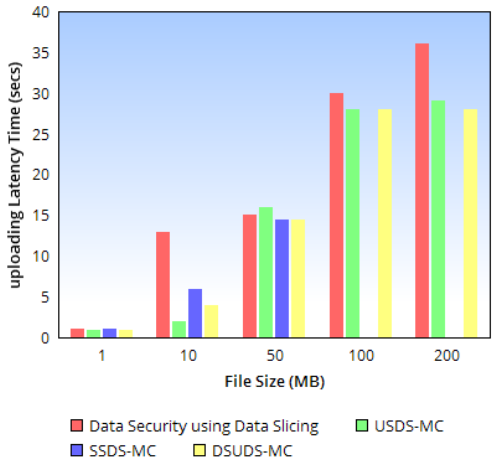

Fig. 2: Comparison of uploading latency time

In addition, Fig. 3 indicates the proposed DSUDS-MC technique has far better downloading latency time with other existing techniques. The downloading latency time is the time taken to the complete downloading technique (i.e., searching, decryption and merging) .The assessment table also indicates the uploading latency time and downloading latency time offered in other schemes. The experimental consequences indicate that processing steps of our proposed technique may be achieved with good performance. From the table, one can be understood that the proposed approach is doing properly in terms of time. The work threshold size of the file is 200 Mb and the minimum threshold quantity of the provider carriers is five because the Multi-Cloud storage is a subscription service the higher the scale of the record the higher will be the value to be paid by the user.

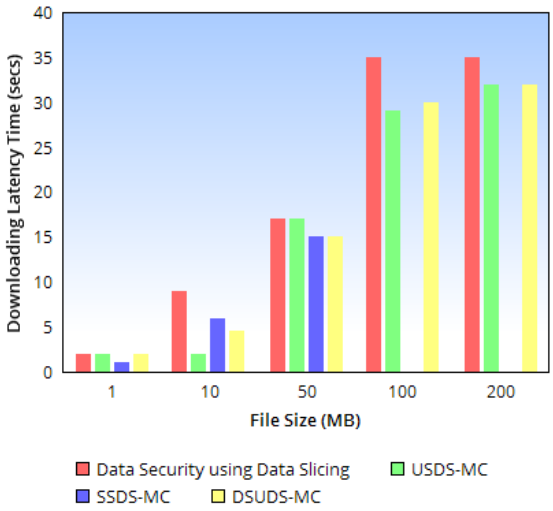

Fig. 3: Comparison of downloading latency time

5.3. Security discussions

5.3.1. Numerical security analysis

Various dimensions of security such as integrity, confidentiality, insider attacks and privacy will be used to conduct the numerical case study for the proposed system. The numerical study of each security feature was determined in consultation with the expert group contributing to this report, drawing on their collective experience. The resulting assessment is measured on a scale of 0 to 10 that can be evaluated against risk acceptance criteria. In many cases the estimate of likelihood depends heavily on the cloud model or architecture under consideration. Table 3 shows the comparison features of various approaches.

Privacy: privacy was tested on the ability of authorized users who knows about the file slicing other than the owner. It was discovered that CP-ABE file slicing is known by 3 unauthorized individuals since so many third party servers are used, SSDS-MC and DS-MC were known to 2 users. One is the receiver and other is the third party user or admin since both architecture uses third party and private cloud database server. USDSMC was file slicing is known to the receiver. If 5 unauthorized users could access the system, then there was 100% lack of privacy. In DSUDS-MC only data owner knows about the file slicing thus representing a privacy level of 100%. Tracing data end to end has been implemented in the proposed technique so bypassing accessibility is easily caught.

Table 3: Comparison security features in various approaches

S. No	Security Features	DSUDS-MC	USSDSMC	DS-MC	CP-ABE	SSDS-MC
1	Privacy	100%	80%	60%	40%	60%
2	Insider Attack	100%	100%	60%	40%	30%
3	Confidentiality	100%	100%	80%	70%	70%
4	Secret Keys	100%	100%	0%	0%	0%
5	Data Integrity	100%	100%	20%	20%	20%

For other approaches like DS-MC, and SSDS-MC was accessed by 2 people

5=100%
*Therefore; 2/5*100=40% lack of privacy*

100%-40%=60% privacy level
For USDSMC accessed by one person
*1/5*100=20% lack of privacy*
100%-20%=80% privacy level
For CP-ABE accessed by 3 person
5=100%
*Therefore; 3/5*100=60% lack of privacy*
100%-60%=40% privacy level

Insider attacks: This feature was measured by examine the possibility of the occurrences in every approach when the third party servers are used, denial of service attacks, colluding attacks from the revoked user as well as from the providers environment, tampering with data and repudiation (lack of ability to trace user when an user performs illegal operation). It is one of the important security features which affect all kinds of cloud services. It was discovered that out of 5 attacks 2, were successful for DS-MC respectively. Similarly 3 out of 10 attacks were successful in SSDS-MC and 5 out of 10 attacks were successful in CP-ABE. However, attacks directed at DSUDS-MC and USDSMC were unsuccessful and to track of insiders monitoring service has also been implemented in this approach.

5 attacks = 100% insecurity
For DS-MC architecture does not focus on tampering of data and repudiation attacks
2 =? Therefore; 2/5 x 100% = 40% insecure
100% – 40% = 60%. secure
For SSDS-MC architecture uses third party server, does not focus on tampering of data and repudiation attacks
*3/5*100=60% insecure*
100%-60%=40% secure
For CP-ABE architecture uses third party server, does not focus on tampering of data, colluding providers attacks and repudiation attacks
*4/5*100=80% insecure*
100%-80%=30% secure
100% means zero successful insider attacks

Confidentiality: The assessment of confidentiality relied on the number of authorized users who knows about the secret key or private key and number of file slices for all approaches. One person knew the key, image chosen and number of file slices under DSUDS-MC as opposed to other models in which many people knew the secret key, number of file slices is fixed. If nobody knows the key and file slices, confidentiality becomes 100%.In the proposed and USDSMC approach the owner knows only RSA public key. Since know no one knows about the private key the confidentiality is 100%. But in all other approaches data owner, receiver and third party privileged user has an option to know about the secret keys. For three people, in SSDS-MC and CP-ABE confidentiality becomes 3/10*100=30% lack of confidentiality. Since both approaches uses third party servers without monitoring.

Therefore its confidentiality level becomes 100%-30%=70%.

If two people know the key regarding DS-MC, the ask of confidentiality becomes 2/10*100=20%.

Therefore its confidentiality level becomes 100%-20%=80%. Two authorized users know about the secret key.

Malicious file attacks: This was tested by evaluating the ease with which a Data owner or user tries to upload a malware file to corrupt the entire cloud infrastructure. Five others have put in the malicious list their ability to upload the malicious file is evaluated. For the DSUDS-MC and USDS-MC, no damage was possible. For each of the other approaches, five people on the malicious list were able to upload the file representing a security level of 0% unless they utilize third party antivirus software with their approach. As discussed in section 4 file is stored on local machine, not on the cloud storage directly so it damages only local machine first. For USDS-MC and DSUDS-MC 100% is the level of security since none gets affected

For the other models, 5/5*100=100% lack of security all malicious files are successfully uploaded.

Data integrity: The property that data has not been maliciously or accidentally altered during storage or transmission. The integrity of these data was later evaluated at the time of the retrieval process. In other models data or file merging causes so many conflicts since it was hard to find which part of the data occurs first and consecutive parts forms the rest. It is to be noted that all the operations such as file slicing, file merging, encryption and decryption are automated. But in all other models they are semi-automated. Out of 5 data into the DSUDS-MC, none data was altered representing 100% data integrity levels. In other approaches, 4 out of 5 data gets altered. Additional support from provider's environment is required to reconstruct the file without merging conflicts.

If 5 = 100% lacks integrity (all 5 data gets altered) If 4 data gets altered. Therefore; 4/5 x 100 = 80% lacks integrity.

100% - 80% = 20% supports integrity for DS-MC, SSDS-MC, CP- ABE and SDS-MC.

Fig. 4 is a security analysis for the various models.

As depicted in Fig. 4, the security of the multi-cloud platform will be enhanced by the proposed system.

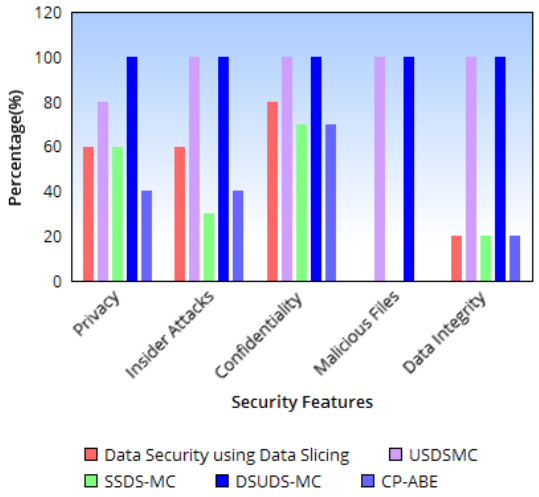

Fig. 4: Security comparison

6. Results and discussions

Privacy is measured by using the dynamic file slicing. Since in all other approaches all the authorized and unauthorized users have knowledge about the file slicing parts because of their fixed parts based on the number of storage providers. But in the proposed approach owner only knows about the file parts which greatly increase the privacy and trustworthy of the clients. Insider attacks are measured by the most possible occurrence in designing the architectures. If monitoring service is not implemented any type of insider attacks are possible as discussed earlier. The accessibility to the key was restricted to single individual to ensure confidentiality. None of the approaches support the malicious file attacks when uploaded. Since they rely on third party antivirus software, but the proposed approach doesn't involve any third party software for file uploading operation first occurs on the owner's machine. If uploaded successfully only the owner's machine gets affected not the cloud infrastructure. There was a greater impact on confidentiality when the number of authorized users knows about the secret key details in their approaches. Most of the approaches use AES encryption and third party servers to store the keys which greatly affect the confidentiality. Data Integrity in the Multi-Cloud approach is measured during the retrieval process. Since none of the existing techniques used Index based slicing merging file conflict occurs which part of the file forms the first and other consecutive parts. It requires additional support from the provider's side environment. The merging and decryption process are not automated it becomes very tedious when there is lack of admin in the providers environment.

7. Future enhancement

The ability to dynamically slice the file greatly increases the trustworthiness of the customers and provides better resilience. This paper guarantees the higher protection for the privacy of data on Multi-Cloud storage and data transition, future research should seek to determine the need for data security which can be based on user defined. The customer can create or add their own features for secure data sharing in cloud storages. The dimension is to generate random keys for each sliced encrypted file and retrieval process is done using the key aggregate cryptosystem. It makes the attackers or any insiders to steal or view the information more tedious.

8. Conclusion

It is evident that cloud storage is the most delicate process in data security. The proposed methodology is the Multi-Cloud Storage security scheme for those who are doing online transactions. This paper ensures the privacy has been greatly increased by using the dynamic file slicing technique which in turn enhances the trust of the customers. The proposed architecture guarantee the insider attacks and malicious file attacks are not possible. The proposed framework supports various file formats and index based slicing technique supports integrity at the retrieval process. The experimental results justify the efficiency of the proposed algorithm. The numerical results justify the data sharing privacy of the proposed model.

References

Ali M, Dhamotharan R, Khan E, Khan SU, Vasilakos AV, Li K, and Zomaya AY (2017). SeDaSC: Secure data sharing in clouds. IEEE Systems Journal, 11(2): 395-404.

Balasaraswathi VR and Manikandan S (2014). Enhanced security for multi-cloud storage using cryptographic data splitting with dynamic approach. In the International Conference on Advanced Communication Control and Computing Technologies, IEEE, Ramanathapuram, India: 1190-1194. https://doi.org/10.1109/ICACCCT.2014.7019286

Borazjani PN (2017). Security Issues in Cloud Computing. In the International Conference on Green, Pervasive, and Cloud Computing, Cham, Springer, Cetara, Italy: 800-811. https://doi.org/10.1007/978-3-319-57186-7_58

Fabian B, Ermakova T, and Junghanns P (2015). Collaborative and secure sharing of healthcare data in multi-clouds. Information Systems, 48: 132-150.

Han Y, Chan J, Alpcan T, and Leckie C (2017). Using virtual machine allocation policies to defend against co-resident attacks in cloud computing. IEEE Transactions on Dependable and Secure Computing, 14(1): 95-108.

Junghanns P, Fabian B, and Ermakova T (2016). Engineering of secure multi-cloud storage. Computers in Industry, 83: 108-120.

Levitin G, Xing L, and Dai Y (2017). Optimal data partitioning in cloud computing system with random server assignment. Future Generation Computer Systems, 70: 17-25.

Razaque A, Nadimpalli SSV, Vommina S, Atukuri DK, Reddy DN, Anne P, and Malllapu VS (2016). Secure data sharing in multi-clouds. In the International Conference on Electrical, Electronics, and Optimization Techniques, IEEE, Chennai, India: 1909-1913. https://doi.org/10.1109/ICEEOT.2016.7755020

Satapathy SC, Bhateja V, Raju KS, andJanakiramaiah B (2016). Computer communication, networking and internet security (Volume 5). Springer Singapore, Singapore.

Subramanian K and John FL (2017a). Enhanced security for data sharing in multi cloud storage (SDSMC). International Journal of Advanced Computer Science and Applications, 8(3): 176-185.

Subramanian K and John FL (2017b). Secure and reliable unstructured data sharing in multi-cloud storage using the hybrid crypto system. International Journal of Computer Science and Network Security, 17(6): 196-206

Vaidya MB and Nehe S (2015). Data security using data slicing over storage clouds. In the International Conference on Information Processing, IEEE, Pune, India: 322-325. https://doi.org/10.1109/INFOP.2015.7489401

Xu P, Liu X, Sheng Z, Shan X, and Shuang K (2015). SSDS-MC: slice-based secure data storage in multi-cloud environment. In the 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, IEEE, Taipei, Taiwan: 304-309.

Zibouh O, Dalli A, and Drissi H (2016). Cloud computing security through parallelizing fully homomorphic encryption applied

to multi-cloud approach. Journal of Theoretical and Applied Information Technology, 87(2): 300-307.