Enhanced Security for Data Sharing in Multi Cloud Storage (SDSMC)

Dr. K. Subramanian
Assistant Professor
P.G and Research Department of Computer Science
H.H The Rajah's College
Pudukkottai

F.Leo John
Research Scholar
P.G and Research Department of Computer Science
J.J College of Arts and Science (Autonomous)
Pudukkottai

***Abstract*—Multiple Cloud storage has become one of the essential services of cloud computing. This Multi-Cloud storage models allow users to store sliced encrypted data in various cloud drives. Thus, it provides support for various cloud storage services using the single interface rather than using single cloud storage services. Cloud security goal primarily focuses on issues that relate to information privacy and security aspects of cloud computing. This latest data storage service and data moderation prototype focus on malicious insider's access on stored data, protection from malicious files, removal of centralized distribution of data storage and removal of outdated files or downloaded files frequently. Data owner does not necessarily need to worry about the future of the data stored in the Multi-Cloud server may be extracted or depraved. The other is ingress control of data. The proposed method ensures the file or data cannot get access without the knowledge or permission of the owner. Thus, this research aims at offering an architecture which reduces malicious insiders and file threats with an algorithm that improves data sharing security in Multi- Cloud storage services. This technique will offer a secure environment whereby the data owner can store and retrieve data from Multi-Cloud Environment without file merging conflicts and prevents insider attacks to obtain meaningful information. The experimental results indicate that the suggested model is suitable for decision making process for the data owners in the better adoption of multi-cloud storage service for sharing their information securely.**

Keywords*—Malicious Insiders; privacy; Index based Data slicing; Malicious Files; Multi-Cloud Storage; Data Sharing

I. INTRODUCTION

Multi-Cloud is the utilization of various computing services in a single heterogeneous architecture. Multi- Cloud Storage means the utilization of various cloud storage services using a single web interface rather than the defaults provided by the cloud storage vendors in a single heterogeneous architecture. Multi-Cloud data systems have the capacity to enhance data sharing and this aspect will be significantly of great help to data users. It enables data owners to share their data in the cloud. In any cloud computing model, security is regarded as the most crucial aspect due to the sensitivity and delicacy of the user's information or data stored in a cloud. Presently, every Organization is pushing its IT department to scale up their data sharing systems. Most cloud services are not free and possess different sizes. For instance, Single Cloud Storage falls among the services with storage limitation which makes it disadvantageous in comparison to multi-cloud storage. The main advantage of using multi cloud storage is performance and higher security for data sharing. In the single cloud storage data remains on the centralized storage which can be easily accessed by the malicious insiders. Companies should start considering working with more than one cloud provider at a time - for cost savings, performance, disaster recovery and other reasons. Most business organizations share most of their data with either their clients or suppliers and consider data sharing as a priority [1]. Through data sharing, higher productivity levels are reached. With several users from various organizations contributing to the cloud data, cost and time spent would be less compared to the traditional ways of manually sending and sharing data, which often led to the creation of out-of-date and redundant documents [1].

Although many cryptographic data slicing methods [2], [3], [4] have been proposed as the main problem arises in the insider's access to stored data. Insiders are the trusted secondary admin or managers who maintains the third party server with the same authorization as the admin. Since the third party servers or infrastructure has been used to store any sensitive information. Administrators and third parties manage the infrastructure as they have remote access to the servers; if administrators or third party managers are malicious then they gain access to the user's data. The other threat is unlike the single cloud storage, retrieval of the sliced files from the multi-cloud server is not an easy procedure. In addition, malicious files can be easily uploaded in all the existing approaches in single cloud storage and multi-cloud storage. The lesser focus has been applied in designing the multi cloud architecture when malicious files are uploaded. The only existed solution is the integration antivirus tool from the third party or cloud provider which creates customer to wait for a longer time while uploading the files.

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

www.multcloud.com
a sample interface for multicloud storage
Need for Multi-Cloud http://www.huffingtonpost.com/young-entrepreneur-council/the-cloud-and-your-busine_b_13751184.html.
Bank Data set File size take n fromhttps://www.chicagofed.org/banking/financial-institution-reports/commercial-bank-data-complete-2001-2010).
Top Threats Group, "The Treacherous 12 Cloud Computing Top Threats in 2016",http://www.cloudsecurityalliance.org

activity with algorithms. Section 5 explains the experimental solutions, and Section 6 Concludes the report and future work.

II. Related Works

Privacy and security for cloud storage are generally a wide area of research. Numerous academic interrogations have been conducted to identify the potential security issues about this subject. It is important to note that sharing files over cloud platform possess numerous vulnerabilities that can lead to unauthorized access. The attackers of cloud have varied intensions or goals which leads to the poor image of the cloud providers once the goal is achieved. In the view of [2] an architecture has been proposed for sharing health care records in multi-cloud storage using Attribute Based Encryption (ABE) and cryptographic secret sharing. Multi-Cloud proxy splits the encrypted record and stores it in the Multi-Cloud. The main drawback in this approaches are group sharing requires huge computation and long waiting time, since file indexing is not used ambiguous information results in file retrieval process. Since the CP-ABE is provided by third party malicious insider may have easy access to the data. File size more than 50 MBs increase the customer's waiting time. The experiments are performed using a highly configured machine hence it is cost consuming in real time. Malicious files are also easily uploaded by the third party authority or role based managers to corrupt the entire scheme. All the tasks are not automated i.e to upload a file client must create a signed medical record using CP-ABE Scheme. Cloud provider's splits the data and transfers data from multi-cloud proxy to cloud data sources.

In order to enhance the secure data sharing in the multi-cloud storage [3] proposed architecture with an Advanced Encryption Standard Algorithm (AES) which seeks to provide better cloud storage decision making for the customers. But insider attacks, colluding attacks, data integrity, data intruder and malicious files have not been focused.

To protect the data from malicious insiders [4] introduced a Secure Data Sharing in Clouds methodology which uses third party server to store a part of the encryption key and other part is maintained by the user. If the revoked user and third party server colludes data can be retrieved from the cloud. Similarly if the malicious cloud admin and third party server colludes data can be retrieved. This method uses single cloud storage and hence centralized distribution of sensitive data is not recommended for the customers. Larger files of 100 MB reduce the performance of this method and makes customer to wait for a longer time since uploading and encryption process are done consecutively.

[5] Introduced a proxy re-encryption scheme for secure data sharing in cloud but private key gets fully exposed when revoked user and proxy colludes. In addition the entire file is stored in single cloud storage which has low security and efficiency.

The reconstruction of data from multi-cloud requires an effective procedure to merge all the files without changing the meaningful information. In [6] very much similar approach has been proposed but does not guarantee the security for Meta table and failed to encrypt the video and other large files. Once the Meta table information is lost, retrieval process will be a tedious work.

In [7] Secure Scalable and Efficient Multi-owner data sharing scheme has been proposed. This scheme integrates Identity Based Encryption and asymmetric group agreement to enable group-oriented access control for data owners in a many-to-many sharing pattern. However the key generation process is carried out by the third party as a separate process and encryption and decryption process is carried out as another process which is burden to the data owner to wait for the completion of the whole process. Malicious files protection has not been guaranteed. Centralized distribution of data storage has not been much promising to the customers to share their data. Identity based encryption supports only small data of 50MB. Key escrow problem arises in Identity based scheme.

The work of [8] introduced a secure file sharing in multi-cloud using Shamir's secret sharing scheme and base 64 encoding in their algorithm. Malicious insider's attacks have been prevented by this scheme. However, indexing of files has not been used so that in the retrieval process receiver has to select all the shares to encode and reconstruct the file which is burden to the receiver. In addition malicious files are not prevented and automation of all the tasks in this scheme has not been focused which reduces the overall efficiency of this scheme.

Many similar approaches has been proposed but failed to implement an effective architecture and working procedure for the secure data sharing using the Multi Cloud storage providers. The existing above approaches does not guarantee the automation of file slicing, encryption, decryption and retrieval process. Existing research also does not focus on the merging file conflicts in the retrieval process, malicious files, colluding provider attacks, insider attacks, removal of centralized distribution of data and key management while sharing the data in Multi-Cloud Storage. Similarly all the existing architectures of single cloud storage and Multi-Cloud Storage follows the same pattern that is file uploading, encryption and slicing without index. If an encryption process is done before slicing very large files or video files cannot be uploaded securely and in addition it may also result to wait the customer for a longer time. Malicious files can also be easily uploaded which causes damages to the multi cloud server in the existing approaches. Further Malicious files [9] are detected in providers environment or by using third parties only after damage is caused. The proposed model is designed in such a way when the malicious files gets uploaded it first affects the owner's machine.

In order to address the above challenges this paper presents an effective architecture framework with a standard algorithm which would enable to enhance the secure data sharing through index based cryptographic data slicing and retrieval of file without file merging conflicts from the Multi Cloud storage. It also ensures the protection of data from malicious insiders and malicious files while uploading the file.

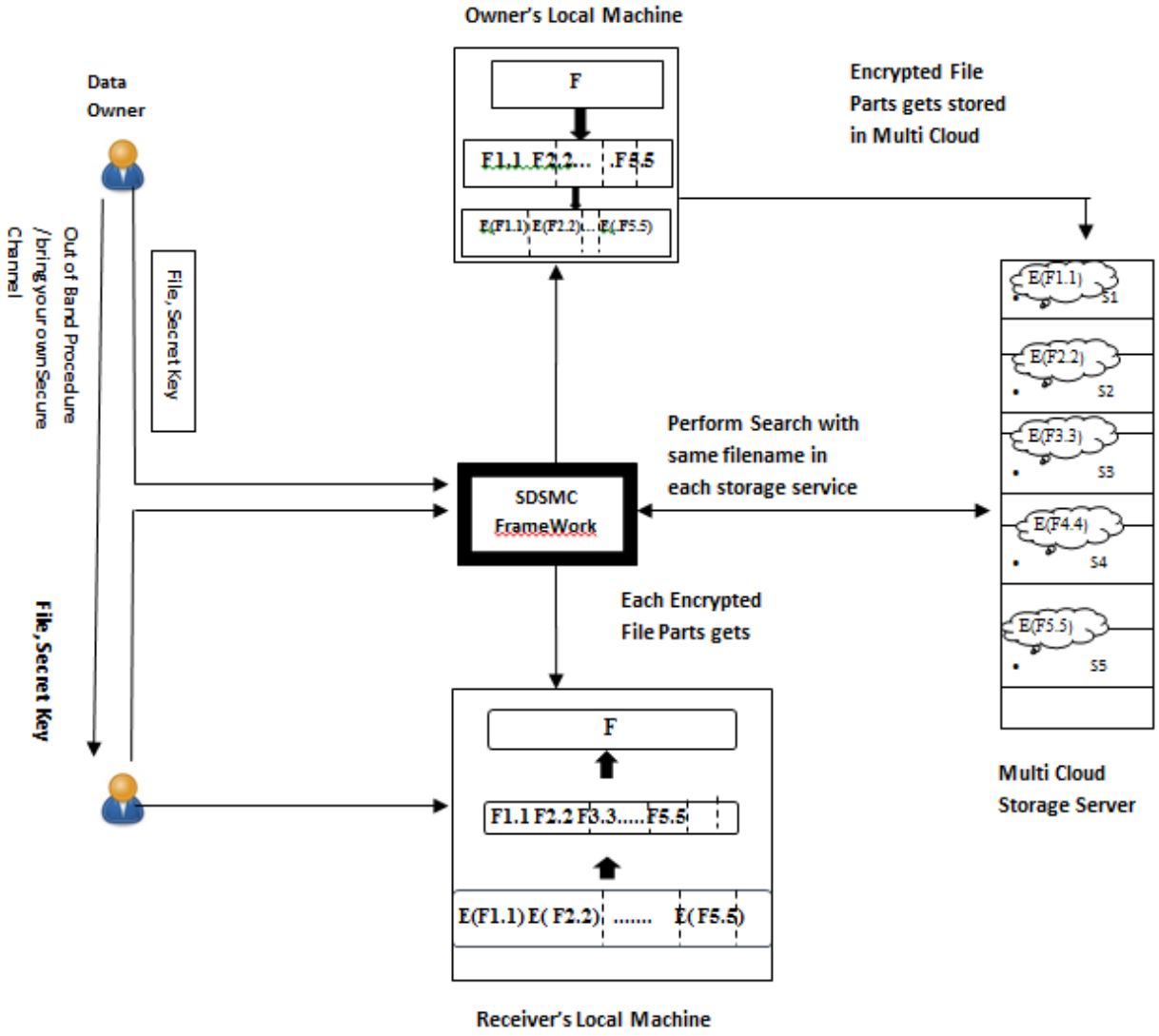

Fig. 1. SDSMC Architecture

III. PROPOSED SYSTEM MODEL

The Overview of Secure Data Sharing Multi Cloud (SDSMC) is shown in Figure-1 and the details are provided in Section 4.

The proposed methodology guarantees the file slicing with index based parts gets encrypted and stored on the Multi-Cloud. This method ensures the file cannot get access without the knowledge or permission of the owner. Data owner uploads the file through the proposed framework interface. The framework uploads the file in the local machine. The framework splits the file with its indexes assigned and encrypts each part of the file using the secret or private key provided by the owner. Each part of the encrypted file gets stored in the owner's machine and then transferred to the multi-cloud server. The receiver sends the decryption request to the owner or the owner can share the required credentials through Bring Your Own Secure Channel (BYOC) or out of band procedure. The receiver enters the credentials through the framework interface. The framework retrieve the file parts and each parts get decrypted, merged and stored the receiver's machine. The major contributions, as described in this report are as follows. The unique feature of this system is to protect the data access from malicious insiders and to protect the datacenters information from malicious files .In addition it also has provision the index based cryptographic data slicing in Multi-Cloud storage services to reduce the file merging conflicts and on demand cost for the customers. It make clients better and fair opportunities for decision making process to choose multi-cloud storage services for secure sharing of data based on trust. The proposed work guarantees that file slicing is based on the number of storage services. More than four cloud storage services are used for confidentiality and none of the Cloud Storage Service Providers can retrieve meaningful information from the pieces of information stored on its servers, without getting some more bits of data from other storage service providers.

In our approach, it is to be presumed that all participating storage cloud service providers, such as Drop Box Google Drive or other CPs, have a common interest securing the infrastructure and data against external, third party

adversaries. Hence the establishment of common and cooperative security mechanisms will be viable, even though many practical and procedural challenges could arise when putting through them in concrete usage scenarios .This work acknowledges those challenges, but consider them out of the scope of our current work.

A. SDSMC Framework

The Secure Data Sharing in Multi Cloud (SDSMC) framework is a web application and it has been described with the overall system flow and various procedures. File uploading, index based file slicing, file encryption, file distribution, file decryption, file retrieval and merging of files, file deletion and Unicode conversion are the automated process performed by the SDSMC framework when using the interface while uploading or downloading a file.

a) ***File Uploading****:* Data owner browse the file from local machine and uploads the file using SDSMC framework interface. This framework uses client resources to upload the file. It means file gets uploaded in the local machine.

b) ***Indexed Based File Slicing:*** This is the process of dividing the uploaded file into two or more parts with respective indices. In this process file slicing is based on the number of storage providers available in the multi-cloud server. At least five storage providers must take part in data sharing and data retrieval process in the proposed approach. This process happens in the owner's local machine.

c) ***File Encryption:*** This is the process of converting a readable file in to unreadable format. This framework encrypts all the index based sliced files using Advanced Encryption Standard (AES) algorithm. Although many existing approaches uses AES it has two draw backs. First it is a weak cipher and the second 128 and 256 bits key make the turnaround time higher which affects the turnaround time process and makes client to wait for a longer time. To overcome the above said limitations slicing is used to make it strong cipher and user defined secret key is used to reduce the turnaround time.

d) ***File Distribution:*** The process of sending the encrypted files along with their indices to different cloud storage providers available in the multi cloud server.

e) ***File Retrieval:*** It is the reversal process of file distribution and file slicing. It is also known as file reconstruction. In this framework the retrieval process starts with submitting the filename without extension. This framework searches the specified filename in each and every cloud storage in multi-cloud server.

f) ***File Decryption:*** Every filename from the multi-cloud server which is associated with specific filename submitted gets decrypted sequentially and stored in the local receiver's machine.

g) ***File Merging****:* This is the process of joining the files with respective indices and gets stored in the receiver's local machine.

h) ***File Deletion****:* This framework performs the automatic removal of files from multi-cloud server and file merging parts in the receiver's machine after the completion of retrieval process.

The idea is about using multiple private clouds simultaneously to deter the risk of disclosure, process tampering and above all, data manipulation in a malicious manner.

IV. Architecture Overview

Figure-1 describes a high level, a standard architecture for a multi-cloud storage service. In the Figure-1 F1.1, F2.2,.. F5.5 denotes the slice file parts name with its index. Similarly E(F1.1),E(F2.2)…E(F5.5) denotes the encrypted sliced parts with its indexing. S1, S2, S3, S4 and S5 are various storage service providers At its core the architecture consists of the following components:

Data Owner: The owner uploads the file with private or secret key. Data Owner acknowledges the request sent by the receiver and sent the details required for the decryption process through the out of band procedure or Bring your own secure channel (BYOSC). In addition the data owner maintains the authorized user's list and keys. Data owner performs the third party duties.

Key Management: There are three options to manage the keys in cloud storage. They are provider's data center, third party server and customer premises. To enhance flexibility and enable sharing of a file to another spacer, it is beneficial to induce the private key at the owner's premise in this approach, as in amazon S3 storage has an enabling option to manage the owner keys.

Multi Cloud Server: It consists of various trusted storage service providers like Cloud A, Cloud B, Cloud C. It stores the encrypted parts of the sliced file from the SDSMC framework to the specific storage service. In this approach minimum five trusted storage service providers are used.

Data Receiver: The receiver will act as a secondary user or sub user. Once the required details are obtained from the owner file can be downloaded.

Owner's Local Machine: All the operations file uploading, indexed slicing and encryption process uses owner's storage device and then encrypted parts are moved towards multi-cloud storage server. This process ensures or guarantee the data owner, uploaded data is highly secured and in addition if malicious files or virus files are uploaded owner's machine will be the priority of those attacks. This is the biggest advantage of our proposed framework and architecture since no additional local server or third parties infrastructure or services are used

Receiver's Local machine: After the successful search operation of the proposed framework, encrypted file parts are downloaded, decrypted and merged in the receiver's device.

TABLE I. NOTATION AND DESCRIPTION

Acronym	Description
F/FN	User's File Name to be uploaded/protected
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

Algorithm-1 SDSMC File Splitting and Encryption

Input: Any file(.xpt, .dicm, video etc.), secret key
Output: Encrypted FilesE (F.1), E (F.2), E (F.3), E (F.4), and E (F.5)

Step 1:
Uploads a file (F) and give 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 service providers integrated with Multi Cloud.
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:
Pass the user defined secret key (SK) to the Unicode Encoding Object to initialize a key(K) and Vector (IV) which can be used to protect repetition pattern in encrypted files.
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 store in the Multi Cloud server.
Step 7:
End

Algorithm -1 explains the application data or file is sliced and transmitted to distinct clouds based on the number of storage services. Files are the most used forms of data storage. The file is uploaded by the user to the Multi Cloud server. The uploaded file gets sliced into five parts with respective indices had been assigned and each part is encrypted using AES encryption algorithm. Five encrypted files are stored in the Multi Cloud Server with respective storage services.

Algorithm-2SDSMC File Decryption and Merging

Input:
File Name without Extension(.xpt, .dicm, video etc.), Secret key (SK)
Output: Decrypted File parts and Merged To get File(F)
Perform:
Step 1:
Get the File Name (FN) and Secret Key (SK) from the data owner or File owner by making request to the processor
Step 2:
Enter or Pass that File Name (FN) and secret Key (SK)
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:
Pass the user defined secret key (SK) to the Unicode Encoding Object to initialize a key (K) and a vector(V) which can be used to create symmetric Decryptor object.
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

Algorithm -2 describes the reverse process of encryption in which authorized receiver using the framework interface passes the file name and secret key obtained from the data owner. The framework start searching the filename associated in the multi-cloud server and then decrypts the file slices sequentially based on the indices and store the decrypted parts in the receiver's locations and finally merges the file based on indices. The merged file is downloaded at the receivers end. After the retrieval process decrypted and encrypted parts of the files are removed from the multi-cloud server and receiver's machine.

V. IMPLEMENTATION

The Secure Data Sharing in Multi Cloud (SDSMC) methodology is proposed to provide following benefits to the outsourced data:

- Confidentiality and secure distributed data sharing in clouds
- Provide protection from colluding service provider attacks

- Removal of centralized distribution of file storage.
- Automation of all the process such as file uploading, file slicing and indexing, encryption, decryption and merging.
- The file is stored on minimum of five storage service providers
- Self-protection of malicious files
- Insider attackers are not able to retrieve meaningful information.
- Removing of file merging conflicts in the retrieval process

A. Experimental Setup

The proposed methodology involves the creation of five private cloud storage services. There is no federated system is available to evaluate performance of the technique. The proposed Secure Data Sharing in Multi Cloud (SDSMC) methodology has been implemented in Visual Studio 2010 Asp.Net with C#. It consists of two entities Multi Cloud Storage Server and Users. The functionality or procedure required by the user is implemented as a client application that connects with Multi Cloud Server to receive the services. The SDSMC web application splits the uploaded file into n pieces based on number of storage services. Each file part has been assigned with indices and encrypted using Advanced Encryption Standard (AES) algorithm to be stored in the respective storage services. All the cryptographic operations are implemented using .net libraries. File name and secret key management gets rectified when it is maintained at the Data Owner premises. As discussed in section IV when malicious files are uploaded it automatically affects the owner's machine. Once the owner receives the request from the receiver or sub user, owner will send the details through the trusted secure channel or Bring Your Own Secure Channel (BYOSC) or out of band procedure for the decryption process. The receiver decrypts all the parts of the file using the details given by the owner and merges in to a File with meaningful information.

Files or Records can be varied in size and format depending on the data contained, which can be plain text or photographic images or even video files. The file sizes used in the first set of experiments are 52MB, 214MB, 345MB, 437 and 552 MB. The experiments are carried out using the following datasets to evaluate our proposed methodology. They are YouTube datasets for video files, Statistical Analysis System (SAS) Commercial Bank Data files with .xpt format containing the variables currently reported on the Report of Condition and Income plus structure and geographical variables (https://www.chicagofed.org/banking/financial-institution-reports/commercial-bank-data-complete-2001-2010) and .DCIM healthcare image datasets (http://www.osirix-viewer.com/datasets). In our methodology five private cloud storages are used for performance evaluation. Both Data Owner and Private Clouds were operated on a Windows 7 Professional 64 bit machine. The machine uses an Intel® Core (TM) 2 Duo CPU T6500 that runs at 2.10 GHz with 4 GB of DDR3 RAM. Retrieval of meaningful information is not possible for malicious insiders. It ensures the data confidentiality for the Data Owners.

B. Numerical Security Analysis

The high level assessment of this multi-cloud approach is performed on the security features such as privacy, insider attacks, confidentiality, secret keys, and data integrity. Table-2 shows the percentage of security obtained in the proposed SDSMC approach. Three models Cipher Text policy Attribute Based Encryption (CP-ABE), Secure Data Sharing in Clouds (SedaSC) and proposed Secure Data Sharing in Multi-Cloud (SDSMC) are allowed in the private clouds for the specific period of time.

TABLE II. COMPARISON OF SECURITY IN VARIOUS APPROACH

S.No	Security Features	SDSMC	CP-ABE [3]	SeDaSC [5]
1	Privacy	80%	60%	40%
2.	Insider Attacks	100%	80%	80%
3.	Confidentiality	90%	30%	30%
4	Secret Keys	60%	60%	60%
5.	Data Integrity	80%	20%	20%

100% means High secure Data sharing in Multi-Cloud Storage.

1) Security Discussions

a) Privacy

The three models were allowed in the multi- cloud for a specific period. The mode of testing was based on the ability, of at least 5 unauthorized persons to go beyond the first step of accessibility. Single cloud was accessible up to the second step by 3 people. The privacy percentage was obtained as follows:

5persons = 100% lack of privacy

3 =? Therefore; 3/5 x 100 = 60%

100% – 60% = 40%

Hence Single cloud was obtained to be 40% privacy.

Multi Cloud was accessible up to the second step by 2 people.

5=100%

2=? Therefore; 2/5 x 100=40%

100%-40%=60%

Hence Multi-cloud was obtained to be 40% privacy.

SDSMC on the other hand was accessed up to the first step by only one person. Mathematically;

5 = 100%

1 =? Therefore; 1/5 x 100 = 20%

100% - 20% = 80%

SDSMC had privacy percentage of 80%.

b) Insider Attacks

This was tested by intentionally allowing the insiders to be aware of the existence of the model. It was checking on the

discipline of the insider and their intensions. Maximum of ten attacks were considered for a period. SeDaSC and CP-ABE approach was attacked successfully twice, but attacks on SDSMC were not successful. Mathematically Single cloud percentage in this case was as shown below;

10 attacks = 100% insecurity

2 =? Therefore; 2/10 x 100% = 20%

100% – 20% = 80%.

SDSMC had zero attacks hence it had 100%, which is the highest quality.

c) Confidentiality

The quality of this feature depended on the number of persons with the secret keys at the first point of access of each model. SDSMC were only known by one person (owner) while cloud to cloud secret keys were known by three users. The higher number of persons with keys for single and multi-cloud lowered its confidentiality as computed as follows. The model is 100% confidential if no one knows the key. 0.9 represents the value of confidentiality if one person knows the key, therefore;

1 = 100%

0.9 =? Hence 0.9 x 100

= 90% confidentiality for SDSMC

If 0.9 = 1 person, then 3 persons = 1/3 x 0.9 = 0.3

1 = 100%

0.3 =? Therefore; 0.3 x 100 = 30% confidentiality for cloud to cloud

d) Secret Keys

Five people were selected randomly who were to guess the first three consecutive keys. 2 people successfully guessed the first two consecutive digits of SDSMC secret keys of first logging. Single and Multi-cloud also had 2 people. Mathematically this was expressed as shown:

5 = 100%

2 =? Therefore; 2/5 x 100 = 40%

100% – 40% = 60%

e) Data Integrity

Five data were allowed into both models. These were managed for a specific period by technicians of both models. Their integrity was later confirmed in case of any corruption. One of the SDSMC data was slightly altered and single and multi-cloud had 4 of its data altered. Mathematically this was expressed as shown:

5 = 100%

1 =? Therefore; 1/5 x 100 = 20%

100% - 20% = 80% for SDSMC

5 = 100%

4 =? Therefore; 4/5 x 100 = 80%

100% - 80% = 20% for cloud to cloud

From the table-II the proposed SDSMC approach has obtained the highest percentage of security in data sharing when compared with other approaches.

C. Performance Analysis

The results obtained from our technique indicate that all processing steps of our architecture can be accomplished with good performance. However, it's more important data owner's waiting time should be minimal for larger file size (500 MB).Since the current implementation performs all operations in memory CPU processing power and memory resources are also concern in performing this technique. It is therefore favorable to operate the proposed technique in firm Multi Cloud Server Environment.

The first set of experiment is carried out using you tube dataset. Table III shows the time taken to complete entire index based file slicing and merging process for the YouTube dataset. Table-IV shows the turnaround times for encryption and decryption process based on the file size of same You Tube Dataset. It is to be noted that file gets uploaded in the local server before the file slicing process started. File slicing or splitting is the process of dividing the files and creating indices for the files based on the number of storage providers.

TABLE III. TIME TAKEN TO COMPLETE SLICING AND MERGING PROCESS FOR YOUTUBE DATA

S.No	File Type	File Size (MB)	Time for Slicing (SECS)	Time for Merging (SECS)
1	.mp4 video	52	0.281	1
2	.mkv video	214	10	2
3	.mkv video	345	22	5
4	.mkv video	437	28	5
5	.mkv video	550	32	7

File slicing computation time is to be observed because it is done before the encryption process and file merging computation time is done after the decryption process. The slicing and merging time increases gradually with respect to the file size. It is to be noted that merging time is very less when compared to slicing time. This is due to the file uploading time is merged with file slicing time.The slicing involves the operation to evaluate the total file size divided by the number of storage services. It will give the constant file size for each storage services. Based on the constant file size, each part of the file has to be created with indices in the respective storage service.

TABLE IV. TIME TAKEN TO COMPLETE ENCRYPTION AND DECRYPTION PROCESS USING AES FOR YOU TUBE DATASET

S.No	File Type	File Size(MB)	Time for Encryption Process(secs)	Time for Decryption Process(secs)
1	.mp4 video	52	3	4
2	.mkv video	214	10	13
3	.mkv video	345	17	18
4	.mkv video	437	21	28
5	.mkv video	552	29	32

It is observed that proposed algorithm shows that encryption and decryption turnaround time has almost taken the same time to complete their process. The above table -IV also proves that the proposed scheme is well suited for non-organizational outsourced data.

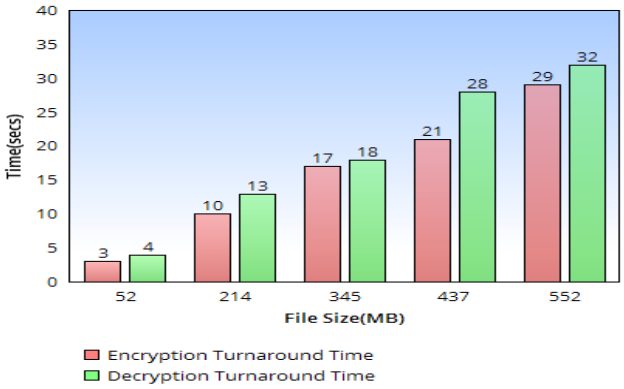

Fig. 2. Encryption and Decryption Turnaround Performance for YouTube Dataset

Above Figure-2 shows the results of YouTube Dataset encryption and decryption turnaround time. Some files have decryption turnaround time more than 7 seconds difference because other process might use the memory resources.The second set of experiments is carried out using commercial bank datasets. The file sizes used are 141,189,234,267 and 337 MB. The same process has been used as in the first set of experiment for the file slicing and merging process. Table-V shows the slice and merging time for bank data. Table -VI shows the encryption and decryption turnaround time for the Bank Data set.Figure-3 shows the results of Encryption Process Time and Decryption Process Time obtained for the Commercial Bank datasets.

TABLE V. TIME TAKEN TO COMPLETE SLICE AND MERGE PROCESS FOR COMMERCIAL BANK DATA

S.No	File Type	File Size (MB)	Slice Time (SECS)	Merge Time (SECS)
1	Call0407.xpt	141	06	02
2	Call0406.xpt	189	10	02
3	Call0209.xpt	234	13	2
4	Call0106.xpt	267	14	3
5	Call0206.xpt	337	24	4

TABLE VI. TIME TAKEN TO COMPLETE ENCRYPTION AND DECRYPTION PROCESS USING AES FOR COMMERCIAL BANK DATASET

S.No	File Type	File Size(MB)	Encryption Time(SECS)	Decryption Time (SECS)
1	Call0407.xpt	141	06	09
2	Call0406.xpt	189	09	12
3	Call0209.xpt	234	11	14
4	Call0106.xpt	267	13	16
5	Call0206.xpt	337	16	20

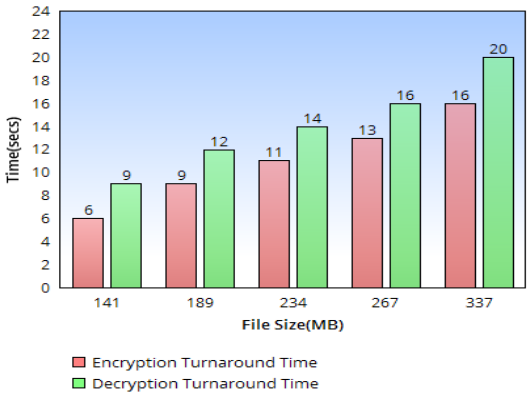

Fig. 3. Encryption and Decryption Turnaround Performance for Bank Dataset

The third set of experiments is carried out using health care data sets. The file here consists of medical records which can be plain text, photographic images or video files. Similar to the first and second experiments the same procedure has been followed in Table-VII and Table-VIII. Table-VII provides the slice and merge time for health care data set. Similarly Table-VIII shows the encryption and decryption turnaround time for healthcare dataset.

TABLE VII. TIME TAKEN TO COMPLETE SLICE AND MERGE PROCESS FOR HEALTH CARE DATASET

S.No	File Type	File Size (MB)	Slice Time (SECS)	Merge Time (SECS)
1	Corstd1.avi	26.3	0.311	0.355
2	Corstd2.avi	36.4	01	0.502
3	Corstd3.avi	79.3	01	01
4	Corstd4.avi	91.3	02	01
5	Corstd5.avi	108	02	01

Above dataset is obtained as DICOM image samples from osirix-viewer.com website. These image samples are converted to .avi files since they are very small in size and used for this research work. The above table-VII shows the various file sizes with slice time and merge time. Whenever file gets sliced indexed is already assigned or in other words file slicing means indexed based file slicing must be assumed throughout this work.

TABLE VIII. TIME TAKEN TO COMPLETE ENCRYPTION AND DECRYPTION PROCESS USING AES FOR HEALTH CARE DATASET

S.No	File Type	File Size (MB)	Encryption Time (SECS)	Decryption Time (SECS)
1	Corstd1.avi	26.3	01	01
2	Corstd2.avi	36.4	02	02
3	Corstd3.avi	79.3	04	05
4	Corstd4.avi	91.3	05	06
5	Corstd5.avi	108	06	07

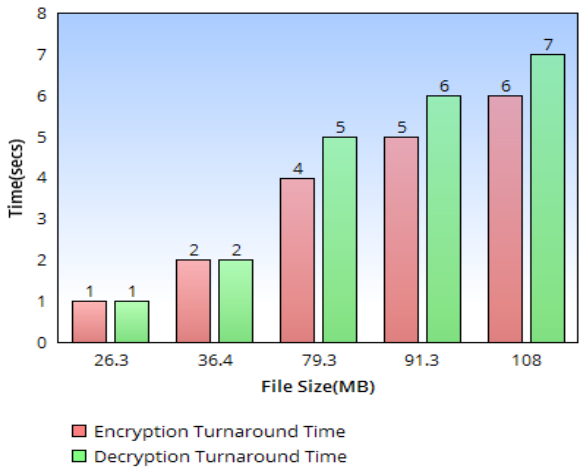

Fig. 4. Encryption and Decryption Turnaround Performance for Health Care Dataset

TABLE IX. COMPARISON OF TURNAROUND TIME WITH DIFFERENT SCHEMES

S.No	File size (MB)	Existing Single and Multi-cloud Storage Schemes						Proposed Scheme	
		[13] CL-PRE		[2]CP-ABE		[5]SeDaSC		SDSMC	
		EPT	DPT	EPT	DPT	EPT	DPT	EPT	DPT
1	1	1	2	0.9	0.9	1	1	0.2	0.2
2	10	13	9	2	2	6	6	1.4	1.6
3	50	53	33	3.4	3.9	9	10	2.4	2.8
4	100	99	57	5.6	5.8	17	20	4	4.8
5	500	369	215	39	40	33	39	26	28.6
6	552	-	-	-	-	-	-	29.2	34.2

EPT-Encryption Process TimeDPT-Decryption Process Time From Table-IX Schemes [13],[2],[5] results are based on single cloud while SDSMC is based on Multi Cloud. The graph has been constructed from the above table for the comparison of Encryption Process Time (EPT) and Decryption Process Time (DPT).

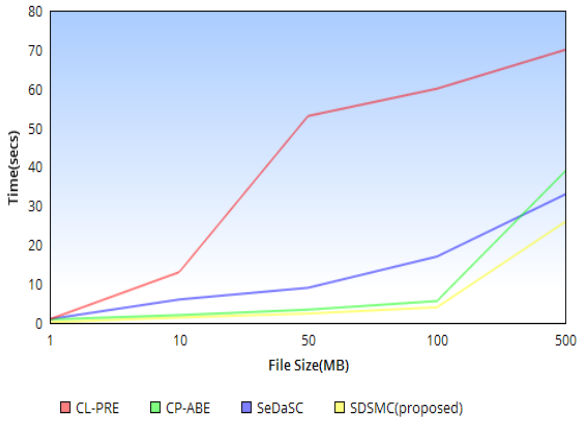

Fig. 5. Comparison of Encryption Process Time

Above figure-5 shows the turnaround performance time of various approaches. It is to be noted that proposed scheme has obtained lesser time seconds for the various file sizes. The consumers waiting time to complete the encryption process has been greatly reduced in the proposed scheme especially for the large file sizes (Mb).

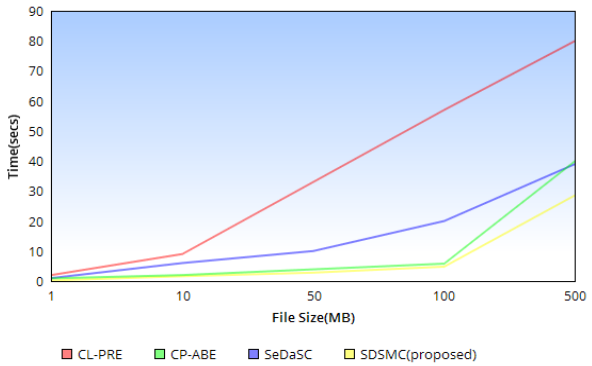

Fig. 6. Comparison of Decryption Process Time

Similarly Figure-6 shows the proposed SDSMC method has far better decryption turnaround time with other existing approaches. In the above table-IX SDSMC column values are obtained from YouTube Dataset, Commercial Bank Dataset and Health care Dataset. The comparison table shows the turnaround times presented in other schemes such as Certificate less Proxy Re-Encryption Scheme (CL-PRE), Cipher TextAttribute Based Encryption Scheme (CP-ABE), Secure Data Sharing in Clouds (SeDaSC) .Although CP-ABE values are very closely related to the proposed approach the share creation and share recovery turnaround times are very high and in addition this scheme uses various software for all the process so automation has not applied as in SDSMC. This scheme (CP-ABE) does not guarantee the malicious insider and file threats and uses high processing machine to obtain the results. Since the files are varied in size and format our methodology supports all types of files which can be used in an organization as well as non-organization for social aspects. Table IX shows the experimental evaluation of existing and the proposed (SDSMC). The experimental results indicate that all processing steps of our proposed architecture can be accomplished with good performance. From the table one can understood that the proposed approach is doing well in terms of time.

In general when the size of file increases time also gets increased but the other security limitations such as privacy, data confidentiality, data integrity and availability of data are far better than single cloud. Similarly when the size of the file, parts of the file and the number of providers increases then the overall performance time decreases because of the parallel execution of all the task at the same time in the proposed SDSMC Multi-Cloud Storage. In the proposed work threshold size of the file is 552 Mb and the minimum threshold number of the storage providers is five. Since the Multi-Cloud Storage is a subscription service the higher the size of the file the higher will be the cost to be paid by the user.

VI. FUTURE WORK

Although the proposed model ensures the protection of data sharing from malicious insiders and files there is a possibility of leakage of key without the owner's knowledge when the framework interface gets accessed from the public networks. When the data owner tries to upload the more files key management becomes cumbersome. To rectify above problems system a public key hybrid crypto system is needed. To enhance the trust of the customers file slicing parts can be defined by the owner itself is the other future directions of our proposed model.

VII. CONCLUSION

The proposed methodology is a Multi Cloud Storage security scheme for organizational as well as non-organizational aspects. Since the various data sets have been used to operate on the SDSMC model and reaches the higher security when compared with other models. The proposed architecture reduces the malicious insider threats and the proposed procedure ensures the providers resource protection from the malicious files. The SDSMC supports all type of files including video files can be encrypted based on the index based cryptographic technique. In the retrieval of the files a standard procedure is used which reduces on demand cost and the conflicts in the merging process. The experimental results justifies the efficiency of the proposed algorithm. The numerical results justifies the data sharing security of the proposed model.

REFERENCES

[1] DananThilakanathan, ShipingChen,Surya Nepal and Rafael A.Calvo "Secure Data Sharing in the Cloud". In Security, Privacy and Trust in Cloud Systems,Springer Berlin Heidelberg,2015,(pp. 45-72).

[2] Benjamin Fabian, Tatiana Ermakova,PhilippJunghanns "Collaborative and secure sharing of healthcare data in multi-clouds". Information Systems, Volume 48 Issue C, 2015,pp 132-150

[3] Balasaraswathi, V. R., &Manikandan, S. (2014)." Enhanced security for multi-cloud storage using cryptographic data splitting with dynamic approach". In Advanced Communication, International Conference onControl and Computing Technologies (ICACCCT), 2014 on (pp. 1190-1194). IEEE.

[4] Mazhar Ali, RevathiDhamotharan, ErajKhan,SameeU.Khan,AthanasiosV.Vasilakos,KeqinLi,Albert.Y.Zomaya "SeDaSC: Secure Data Sharing in Clouds", Systems Journal, IEEE, volume :PP, Issue:99,2015,pp 1-10.

[5] Wang Liang-liang,ChenKe-fei,Mao Xian-ping,Wang Yong-tao "Efficient and Provably-Secure Certificateless Proxy Re-encryption Scheme for Secure Cloud Data Sharing" Journal of Shanghai Jiaotong University Volume 19, issue 4,2014 pp 398-405.

[6] PengXu, XiaqiLiu,ZhenguoSheng,XuanShan,KaiShuang "SSDS-MC: Slice-based Secure Data Storage in MultiCloud Environment" 11th EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE) , 2015,pp 304-309.

[7] ShunganZhou.RuiyingDu,JingChen,HuaDeng,JianShen,Huanguo Zhang "SSEM: Secure, Scalable and Efficient multi-owner data sharing in clouds", China Communications IEEE ,Volume 13,issue 8, 2016,pp 231-243.

[8] Ibrahim Abdullah Althamary, TalalMousaAlkharobi "Secure File Sharing in Multi-Cloud using Shamir's Secret Sharing Scheme",Transactions on Network and communications Vol 4 issue 6, 2016,pp53-67.

[9] Safaa Salam Hatem, Maged H.Wafy,Mahmoud M.EI-Khouly "Malware Detection in cloud Computing",International Journal of Advanced Science and Computer Science Applications,Vol 5 No 2014.

[10] MahaTebaa, Said El Hajji "From Single to Multi-Clouds Computing Privacy and Fault Tolerance", Science Direct (ELSEVIER), InternationalConference on Future Information Engineering,(2014),pp112-118.

[11] YuukiKajiura,Shohei Ueno,Atsushi Kanai, ShigeakiTanimoto, Hiroyuki Sato "An Approach to Selecting Cloud Services for Data Storage in Heterogeneous-Multicloud Environment with High Availability and Confidentiality Autonomous Decentralized Systems" (ISADS) IEEE Twelfth International Symposium,2015,(pp 205 – 210).

[12] Tatiana Ermakova, Benjamin Fabian "Secret Sharing for Health Data in Multi-provider Clouds Business Informatics" (CBI), 2013 IEEE 15th Conference,2013,pp 93-100.

[13] Xu, L., Wu, X., & Zhang, X. "CL-PRE: A certificate less proxy re-encryption scheme for secure data sharing with public cloud". In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security,2012 pp. 87-88 .

[14] Seo, S. H., Nabeel, M., Ding, X., &Bertino, E." An Efficient Certificate less Encryption for Secure Data Sharing in Public Clouds. Knowledge and Data Engineering", IEEE Transactions on, 26 (9), 2013,pp2107-2119.

[15] Abdul Nasir Khan , M. L. Mat Kiaha, Sajjad A. Madanib, MazharAlic, Atta urRehmanKhana , ShahaboddinShamshirbanda "Incremental proxy re-encryption scheme for mobile cloud computing environment". The Journal of Supercomputing, 68 (2), 2014 Pp624-651.

[16] Yashaswisingh, Farah Kandah, WeiyiZhang "A Secured Cost-effective Multi-Cloud Storage in Cloud Computing" IEEE INFOCOM Workshop on Cloud Computing 2011,pp 619-624.

[17] Dr. K.Subramanian, F.Leo john "Data Security in Single and Multi-Cloud Storage- an Overview" *International Journal of innovative Research in Communication Engineering* 2016 pp 19046-19052.

[18] BorkoFurht, Armando Escalante "The Handbook of Cloud Computing",Springer Publications 2012.

[19] Jens-Matthias Bohli,NilsGruschka,MeikoJensen,Luigi Lo Lacono andNinja Marnau,"Security and Privacy-Enhancing Multicloud Architectures" IEEE Transactions On Dependable and Secure Computing 2013 pp-212-224.

[20] VenkataJosyula, Malcom Orr, Greg Page,"CloudComputing:Automating the Virtualized Data Center" Cisco Press 2012.

[21] Alycia Sebastin,Dr.L.Arockiam "A Study on Data Security Issues in Public Cloud", International Journal of Scientific and TechnologyResearchVolume 3 Issue 5 May *2014 pp 144-146.*

[22] B.Rex Cyril, Dr.S.Britto Ramesh Kumar "Cloud Computing Data Security Issues, Challenges, Architectures and Methods-A Survey" InternationalJournal of Engineering and Technology(2015).

AUTHORS PROFILE

Dr. Subramanian Krishnasamy is currently working as an Assistant Professor in H.H The Rajah's College. His area of interest includes Data Mining, Networking, Cloud Computing, Network Security, Big Data, Multi-Cloud and so on.

Mr. Leo John is a part-time research scholar in Computer Science Department, JJ.College of Arts and Science pudukkottai. His area of interest includes Cloud Computing, Unstructured Data Security in multi-cloud, cryptography and so on.