Secure and Reliable Unstructured Data Sharing in Multi-Cloud Storage using the Hybrid Crypto System

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

[1]Assistant Professor, P.G & Research Department of Computer Science, HH The Rajah's College,Pudukkottai
[2*] Research Scholar, P.G & Research Department of Computer Science, J.J College of Arts and Science(Autonomous),Pudukkottai

Abstract
Multi-Cloud Storage is one of the essential service in cloud computing which is used to store and access data remotely. This Multi-Cloud storage models enable users to store sliced encrypted information in varied cloud drives. Thus, it provides support for varied cloud storage services using the only interface instead of using single cloud storage services. Despite the fact that cloud computing streamlines the process of sharing resources among groups and users, there are some security concerns about using the cloud. Sharing data among multiple users while still maintaining data integrity and privacy is still a big challenge as a result of the constant change of the data ownership. The proposed model offers the optimal solution for malicious insider's attack, protection from malicious files when uploaded by malicious user, and decentralized distribution of data storage using an index based cryptographic data slicing and sharing of the key through untrusted or semi-trusted secure channels with high efficiency and flexibility. In this research, a new security model is used with an algorithm to ensure the security of data sharing in the Multi-Cloud. This technique also uses the hybrid cryptosystem to enhance the secure data sharing, secret key confidentiality and also solves the key escrow problem. The algorithm presented in this method reduce the threats caused by malicious insiders and malicious users when sharing data in multi-cloud storage. Various datasets block or file size has been chosen randomly to evaluate the efficiency and its probability with various file formats using the proposed technique. The results of the experiment indicate that the proposed model is the best choice for data owners who are looking for the most secure and efficient way to share their data over multi-cloud storage.

Key-words:
Multi-Cloud Storage, Malicious Insider, Cryptography, Data Sharing, Index Based Data Slicing, Malicious Files, Counter Attack

1. Introduction

There are several security challenges associated with single cloud systems. For insiders, malicious insiders and external hackers may pose a threat to the customer information. After gaining unauthorized access to an organization's data, a hacker can use the data for fraud purposes or use the information to blackmail an organization. There are also additional threats such as lack of confidentiality, data loss, intrusion, and lack data integrity Subramanian, K & Leo, J (2016). Maintenance of single cloud platforms requires high costs which are not sustainable by most organizations. There is also the issue of one single point of failure; if the cloud is unreachable, then service delivery is impacted. Data security has been an area of focus for a long time. However, the nature of the internet today has accelerated the need for information security. The current information systems should have high levels of security and should safeguard personal and sensitive information from being exposed to the public Thilakanathan, et.al. (2015,).

Multi-Cloud employs two or more cloud services and avoids over reliance on one single cloud. 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. Data storage and sharing in the current world means businesses. Compared to over 200 years ago, companies nowadays virtually compete on data. Most organizations generally move their data to the cloud platforms to reduce costs of operations. Organizations, both large and small, are embracing cloud computing platforms to achieve leaner and efficient systems. The data stored in the cloud can be shared among many individuals as well as organizations. Data sharing enables high levels of productivity.

The security of information stored in the multi-cloud architecture is implemented using several techniques Bohli, et.al (2013). Khasim Shaik et.al(2017) analyse the most important security encryption algorithms for data protection in cloud computing.The architecture of multi-cloud alone makes it difficult for attackers to breach the security of information stored on these platforms. The main problem arises in the key management, distribution of keys for re-sharing the file, malicious insiders and users. Malicious users are one those who upload the virus files to damage the entire service. Malicious insiders are the trusted admin or managers who maintain the third party server with admin authorization.

Several approaches Razaque et.al(2016), Ali,et.al (2015), Balasaraswathi, V. R., &Manikandan, S.(2014), Vaidya, M. B., and Sandeep Nehe (2016) had been offered for

Manuscript received June 5, 2017
Manuscript revised June 20, 2017

secure sharing of unstructured information in multi-cloud, but do not undertake the dependable and reliable architecture. Existing procedures in multi-cloud storage does not ensure the protection against the secure distribution of keys, key management and malicious insiders and files. File merging conflicts arise in the retrieval process which reduce the data integrity since the indexed based cryptographic data splitting is not employed in existing approaches. AES128 bit encryption is also practiced in many approaches which reduces the performance of the response time when the size of the file increases encryption process time also increased.

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

Security and privacy of cloud networks are an area of extensive research. Several research studies have been done to discover the security issues associated with using cloud systems. The use of an architecture integrated with attribute-based encryption for specific access authorization and cryptography is one of the common techniques used for enforcing security in the multi-cloud environment. To enhance the leverage of security in multi-cloud Abdul Razaque etal. (2016) proposed a data sharing architecture which aims at spreading data in several clouds. This approach has some limitations in that protection of the encryption keys does not focus on sharing of keys in a semi-trusted, secure channels and data integrity in the retrieval process, since the index based slicing is not used. To protect the data from malicious insiders Ali et.al (2015) proposed the secure data sharing in the cloud model. This model does not prevent colluding attacks since the third party server is used to maintain the part of the key. In addition, this methodology uses single cloud storage and centralized distribution of data which is not recommended. The work of Balasaraswathi, V. R., and Manikandan.S (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.

The use of standard architecture and secure data storage procedure is common among several cloud systems. The standard data storage procedure does not employ multi-cloud in its operation. The work of Vaidya, M. B., and Sandeep Nehe (2016)proposed a unified framework to guarantee the secure sharing of data in a Multi - Cloud. It uses cryptography, data slicing method and encryption techniques to store data in the public clouds. Meta data (file segmentation, file distribution information, etc.) are stored on the private cloud database. 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 a burden to the receiver. In addition, key management and key distribution are not focused, private cloud database remains unmonitored which leads to malicious insider attacks. Automation of all the tasks in this scheme has not been focused which reduces the overall efficiency of this scheme. 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.

Fabian et.al. proposed an 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.

The work of Xu, L,et.al.(2016) also makes use of proxy re-encryption techniques which does not require the use of public key certificates to address the key escrow problem. Proxy-encryption under this scheme requires heavy computation and is expensive thus making it least preferred.

In the work of Ouadia,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 proposed model is very similar to the work of Subramanian,K and Leo, J (2017) 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 retrieval of the lost key is impossible and data also cannot be decrypted. The distribution of key is done through the owner's trusted secure channel. The model remains unmonitored to track of any issues.

To protect the mobile privacy Indu, I.,et.al.(2017) proposed a secure file storing and retrieving mechanism using single cloud storage for very small file sizes. But this approach is fully depends on the trusted third party and file sizes analyzed has not been presented on the approach.

To address the above challenges the proposed model uses RSA encryption to secure the AES private key and stores a part of the key to the server. In all the above approaches key sharing cannot be sent using the semi-trusted or untrusted channel when the files are used. AES Encryption gets improved when owner defined private key is used and RSA to protect the private key. So a part of the key is maintained by the owner and another part in the cloud server. File types and sizes supported should also be considered for all the existing approaches. The proposed architecture ensures the distribution keys can be done using untrusted channel, counter attack mechanism against malicious files, monitoring of malicious insiders and reduces the ambiguous information in the retrieval process since the file slicing used index-based access. Various file types get supported in the proposed model. In addition, key management is maintained in the centralized distribution and get gets monitored frequently by the service provider.

3. Proposed Design

The figure-1 provides the architecture of the proposed system using hybrid cryptosystem. According to the architecture, data owner transmits the file and the secret key via the framework interface. The file is uploaded to the USDSMC 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 key and a portion of the key transmitted to the owner and cloud database server. The decryption phase also involves a number of processes. For instance, upon receiving the necessary credentials from the owner, the filename and the public key are transmitted using the untrusted or semi-trusted channel. The file name is searched and the private keys used to decrypt the sliced files to the receiver's computer.

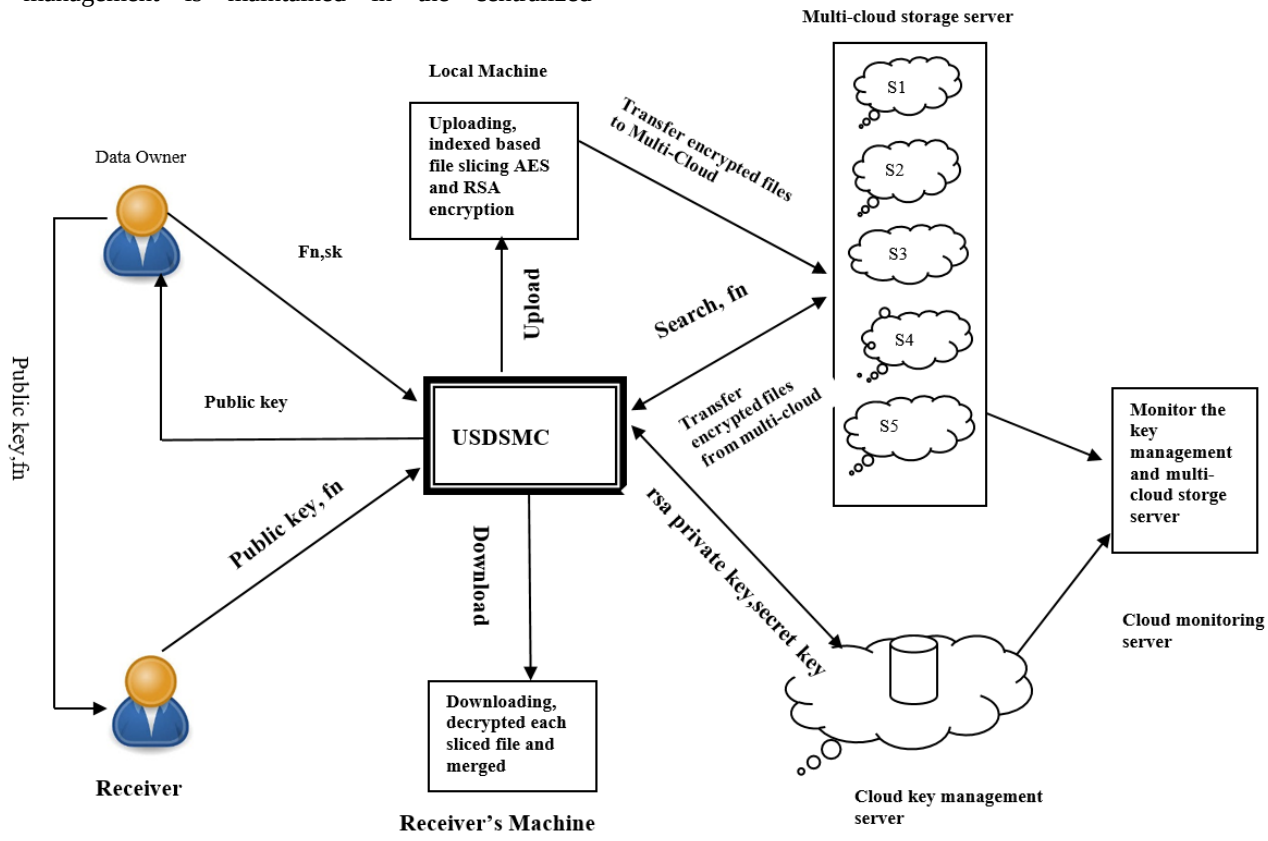


Fig. 1 USDSMC Architecture

3.1 USDSMC Framework

The USDSMC framework act as a middleware or web API to connect with Multi-cloud server. The following are the operations or process performed by the framework when the file gets uploaded or downloaded.

a) File Uploading: File gets uploaded in the owner's local machine.

b) Indexed Based File Slicing: File gets sliced based on the number of storage service providers. All the sliced file sizes are constant.

c) RSA Encryption: Owner's a secret key gets encrypted and as a result, framework sends the public key to the owner and another part to the cloud database server.

d) AES Encryption: Each part of the sliced file is encrypted and sent to the multi-cloud storage server. It uses 8-bit secret key to reducing the round trip process while performing encryption which in turn increase the efficiency of the proposed algorithm.

e). File Downloading: Once the receiver enters the public key and file name via framework interface it searches the file name and downloads all the parts in the receiver's machine.

f) File Merging: Once the file parts are received merging process will result to give a required file. All the processes get monitored at the provider's end.

4. Architecture Overview

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

Data Owner: This entity is charged with the responsibility if uploading the file and secret key. Moreover, this entity also ensures the transfer of public key.

Local Machine: The local machine entity assumes the responsibility of a temporary storage for the encrypted sliced files. Each part of the sliced file is encrypted and sent to the multi-cloud storage server consecutively.

Receiver Machine: This entity is charged with the responsibility of receiving the decrypted files stored in the multi-cloud storage server.

Cloud Monitoring Server: The cloud monitoring server is used to monitor the consumer's activities and provider's high privileged user's activities. This server is managed by the super admin of the cloud platform.

Cloud Key Management Server: This entity is responsible for managing the encryption and decryption keys. Prior to being uploaded to the multi-cloud storage, the file slices are encrypted using a user-defined private key. The private key is encrypted by the RSA and a part of the key is maintained in the cloud key management server along with the secret key. During the retrieval of the file slices, public key from the owner and private key from the cloud server is used to generate the decryption key

Data Receiver: this entity is charged with the responsibility of uploading the public key and the file name onto the cloud storage. It also securely receives the data transmitted by the owner.

Table 1: 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
RSA pub key	RSA public key
RSA pvt key	RSA private Key

Algorithm 1- USDSMC File Slicing and Encryption
Input: Any file (.xpt, .dicm, video etc.), secret key
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 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:
Use RSA encryption to asymmetrically encrypt the user defined secret key assigned in AES 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-1 Examines the process by which data is sliced and uploaded to various clouds. In order to safeguard from the malicious file as a precaution measure, algorithm-1 uses owner's infrastructure for file upload and indexed based slicing and encryption process. RSA encryption is used to protect the user defined key and also solves the key escrow problem As a result, the owner receives the public key and another part is sent to the cloud database server. Finally, all the encrypted slice files get stored on the multi-cloud server.

Algorithm-2 USDSMC File Decryption and Merging
Input:
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:
Get the File Name (FN) and public Key (PK) from the data owner or File owner by making request to the processor
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 parts from the receiver's machine, when all the parts are fully merged.
Step 7:
End

Algorithm-2 describes the process of file decryption. Under this phase, the file name along with the public key received from the owner is passed by the legitimate file recipient. Secret key is revealed 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 files are stored in the location of the receiver and eventually merged on the basis of the indices.

5. Implementation

The Secure and reliable Unstructured Data Sharing in Multi-Cloud (USDSMC) 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.
- Provides a solution for key escrow problem.
- The file is stored on minimum of five storage service providers
- Self-protection from malicious files when uploaded.
- Insiders are not able to retrieve meaningful information.
- Removing of Data Integrity conflicts in the retrieval process.
- Enhance the performance in encryption and decryption process.

5.1 Experimental Setup

The proposed methodology uses the five public cloud storage services such as drop box, box, one drive, Google drive and amazon. All the storage services apps are installed on the owner's local machine. The proposed Secure and reliable unstructured Data Sharing in Multi-Cloud (USDSMC) methodology has been implemented in Visual Studio 2010 Asp.Net with C# with all the required cryptographic libraries. It consists of two entities Multi-Cloud Storage Server and Users. The USDSMC framework act as a middleware to connect Users with Multi-Cloud server via framework interface. 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 USDSMC web application splits the uploaded file into n pieces based on the 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. Key escrow problem gets rectified when RSA encryption is used to protect the user defined secret key. 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, the owner will send the details through the untrusted or semi-trusted channel or for the decryption process. The receiver decrypts all the parts of the file using the details given by the owner and merges into 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 set of experiments are 1MB, 10MB, 50MB, 100MB, 200MB and 300 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 from the work of Subramanian,K and Leo, J (2017). In our methodology, five public cloud storage services are used for performance evaluation. Both Data Owner and Public Cloud services

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.

5.2 Performance Analysis

The performance of the proposed algorithm is provided on table-2. The other Multi-Cloud Data Sharing (MCDS) Schemes values are average turn-around time for encryption and decryption process. The graph has been constructed from the above table for the comparison of Encryption Process Time (EPT) and Decryption Process Time (DPT). The turn-around time to complete the encryption process has been greatly reduced in the proposed scheme especially for the large file sizes (Mb).

Table-2 Turn Around Time Comparison

S.No	FT	FS (MB)	Existing Single and Multi-cloud Storage Schemes							Proposed Scheme
			SeDaSc [1] (secs)		Multi-Cloud Proxy Cp-ABE [27] (secs)		Other MCDS [6],[8],[28] (secs)			USDSMC (secs)
			EPT	DPT	EPT	DPT	EPT	DPT	EPT	DPT
1	.xpt	1	1	2	0.9	0.9	1	1	0.2	0.2
2	.xpt	10	13	9	2	2	6	6	1.4	1.6
3	.exe	50	53	33	3.4	3.9	9	10	2.4	2.8
4	.avi	100	99	57	5.6	5.8	17	20	4	4.8
5	.flv	200	369	215	39	40	33	39	26	28.6
6	.mkv	300	-	-	-	-	-	-	29.2	34.2

EPT- Encryption Process Time DPT-Decryption Process Time FT-File Type FS-File Size

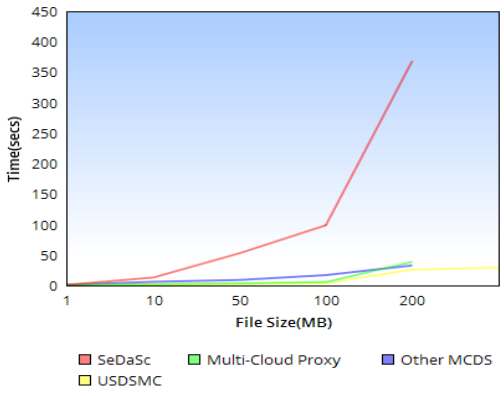


Fig. 2 Comparison of Encryption Process Time

Figure-2 shows the turnaround performance time of various approaches. It is to be noted that proposed scheme has obtained lesser time in terms of seconds for the various file sizes and file formats. The existing approaches do not support video file format. In the existing approaches encryption is managed for the whole file at the beginning and then gets sliced but in the proposed approach file is sliced first and then encrypted all the sliced files at the same time or parallel. This greatly enhances the efficiency of the turnaround time. When data theft or loss has occurred it gets rectified immediately since index based slicing is used missing parts can be easily retrieved.

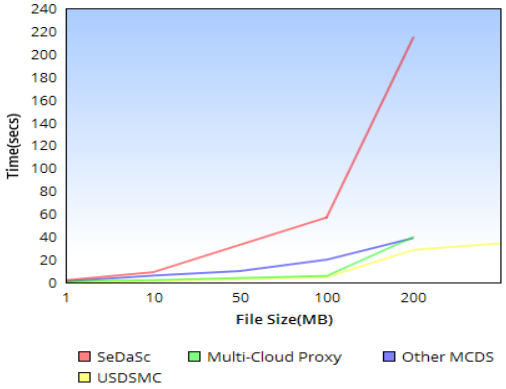


Fig. 3 Comparison of Decryption Process Time

Similarly, Figure-3 shows the proposed SDSMC method has far better decryption turnaround time with other existing approaches. The comparison table also shows the turn-around time performance of encryption and decryption process presented in other schemes such as Ali et.al (2015) Secure Data Sharing in Clouds (SeDaSC), Benjamin Fabian et.al (2015) Multi-cloud proxy Cipher Text Attribute-Based Encryption Scheme (CP-ABE) and other Multi-Cloud Data Sharing Scheme Razaque et.al(2016), Ali,et.al (2015), PengXu,et.al.(2015),Balasaraswathi, V. R., &Manikandan, S.(2014), Vaidya, M. B., and Sandeep Nehe (2016). The experimental results indicate that all processing steps of our proposed architecture can be accomplished with good performance. From the table, one can be understood that the proposed approach is doing well in terms of time. When the proposed USDSMC architectural framework gets implemented in the real time it will yield better turn-around performance since Multi-Cloud is based on parallel processing. There will be a parallel execution of all the similar task simultaneously. The work threshold size of the file is 300 Mb and the minimum threshold number of the service 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.

5.3 Security Discussions

Numerical Security Analysis

Various dimensions of security such as integrity, confidentiality, insider attacks and privacy will be used to conduct the security analysis for the proposed system. The table below depicts the enhanced levels of security under the USDSMC approach. Security Features of the proposed approach(USDSMC) is compared to various approaches like Abdul Razaque et.al.(2016) Secure Data Sharing in Multi-clouds, Vaidya, M. B., and Sandeep Nehe.(2016) DS-MC Data security using data slicing over storage clouds, Fabian et.al(2016) CP-MCP Cipher Text Policy Attribute based Encryption using Multi-cloud Proxy and PengXu,et.al.(2015) SSDS-MC: Slice-based Secure Data Storage in Multi-Cloud Environment Below the table gives the detailed description of the numerical study of the various security parameters and how those table values are obtained.

Table 3: Comparison Security Features in various Multi-Cloud approaches

S.No	Security Features	USDSMC	SDS-MC	DS-MC	CP-ABE	SSDS-MC
1	Privacy	80%	60%	60%	40%	60%
2	Insider Attack	100%	80%	80%	50%	70%
3	Confidentiality	90%	60%	45%	45%	60%
4	Secret Keys	100%	80%	80%	60%	60%
5	Data Integrity	100%	20%	20%	20%	20%

- **Confidentiality**: This feature was measured based on the number of individuals with secret keys for every approach. It was discovered that under USDSMC, the key was known by just one individual while in the other models the key was known by many individuals. The confidentiality of the other systems was impacted by the higher number of individuals who were in possession of keys. If the key is known by no one, the confidentiality is 100%, if the key is known by 1 person, the confidentiality is 90%.
  It is 100% if the key is known by no one,
  0.9 is the confidentiality value if the key is known by one person
  0.9*100=90% confidentiality for USDSMC
  For SDS-MC and SSDS-MC
  3 persons=1/3*0.9=0.3
  0.3*100=30% confidentiality.
  For DS-MC and CP-ABE
  2 persons=1/2*0.9=0.45
  0.45*100=45% confidentiality

- **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 SSDS-MC and CP-ABE secret keys of first logging. Similarly 1 expert successfully guessed the first two consecutive digits of SDS-MC and DS-MC. However, none was unsuccessful in USDSMC because RSA uses 2048 bit key to secure the private key. As a result, USDSMC provides 100% secure for guessing attacks. Mathematically this was expressed as shown:
  5 = 100%
  For SSDS-MC and CP-ABE
  2 =? Therefore; 2/5 x 100 = 40%
  100% – 40% = 60%
  For SDS-MC and DS-MC
  1/5*100=20
  100%-20%=80%

- **Privacy**: privacy was tested on the ability of 5 users or more to bypass the first stage of accessibility. It was discovered that CP-ABE was accessible up to stage two by 3 individuals, SDS-MC, SSDS-MC and DS-MC were accessed by 2 persons. If 5 unauthorized users could access the system, then there was 100% lack of privacy. Three people, therefore, represented 60%. USDSMC was only accessed by 1 individual up to the first stage thus representing a privacy level of 80%.
  USDSMC was accessed by 1 person
  This implies, 1/5*100=20%
  100%-20%=80% privacy for USDSMC.
  For other approaches like DS-MC, SSDS-MC and SDS-MC was accessed by 2 person
  5=100%
  2=? Therefore; 2/5*100=40%
  100%-40%=60%
  For CP-ABE accessed by 3 person
  5=100%
  3=? Therefore; 3/5*100=60%
  100%-60%=40%

- **Data integrity**: Data was permitted in both models and managed over a specific period of time. The integrity of these data was later evaluated at the time of the retrieval process. It was discovered that USDSMC data was least corrupted. 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 USDSMC, only one data was altered representing 80% data

integrity levels. In other approaches, 4 out of 5 data gets altered.

5 = 100%

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

100% - 20% = 80% for USDSMC

5 = 100%

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

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

100%-80%=20%

- **Insider attacks:** This feature was tested by evaluating the success rate of insider attacks among the models over a specific period. It was discovered that out of 10 attacks 2, were successful for DS-MC and SDS-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 USDSMC were unsuccessful and to track of insiders monitoring service has also been implemented in this approach.

10 attacks = 100% insecurity

For DS-MC and SDS-MC

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

100% – 20% = 80%.

For SSDS-MC

3/10*100=30%

100%-30%=70%

For CP-ABE

5/10*100=50%

100%-50%=50%

100% means zero successful insider attacks

The graph below is a security analysis for the various models.

Fig. 4 Security Comparison

As depicted in figure 4 above, the security of the multi-cloud platform will be enhanced by the proposed system.
20

5.4 Reliability Discussions

Numerical Reliability Analysis

Reliability refers to the likelihood that a proposed system will perform its required function under particular conditions over a certain period of time Wen, Zhenyu, et al(2016). Reliability will be determined by examining the time process of the different models in 5 cycles. Various dimensions such as file formats supported, collusion attack, key escrow problem, file size and malicious file threats are applied to evaluate the proposed framework reliability. Table 4 below depicts reliability comparison of various approaches.

Table 4: Comparison of Reliability Features in various approach

S.No	Reliability Features	USDSMC	SDS-MC	DS-MC	CP-ABE	SSDS-MC
1	File Formats Supported	100%	80%	80%	20%	60%
2	Collusion Attack	100%	80%	80%	60%	60%
3	Key Escrow	100%	60%	60%	60%	60%
4	Malicious Files	100%	0%	0%	0%	0%
5	File Size	100%	80%	80%	80%	80%

- **File Formats Supported**: This feature was tested by uploading various file formats. The success rate depends on the most formats or all formats gets uploaded and must perform the slicing and encryption process. Out of 5 file formats, all were successful on USDSMC representing a 100% reliability. Nevertheless, one out of 5 file formats were successful in CP-ABE model, thus representing a reliability rate of 80%. In SSDS-MC, 3 out 5 file formats were successful and 4 out 5 file formats were successful in DS-MC and SDS-MC respectively.
  100%= All formats supported. Therefore USDSMC had 100% to support various file formats
  For CP-ABE , 1/5*100%=20%
  For SSDS-MC 3/5*100=60%
  For SDS-MC and DS-MC 4/5*100=80%
- **Collusion attack**: Collusion attack can be caused by unauthorized or revoked user or third party admin tries to collude with cloud server for data theft or loss on a successful attack reduces the reliability. Out of 5 Collusion attacks on USDSMC, none was successful representing 100% reliability. Other models 2 out of 5 attacks were successful.
  For SSDS-MC and CP-ABE, 2/5*100=40%
  100%-40%=60%.
  For SDS-MC and DS-MC 1 out of 5 attacks were successful
  1/5*100=20
  100%-20%=80%
- **Key Escrow:** This was tested by measuring the ease with which an escrow key can be compromised. An encryption key was entrusted to a third party for all the three models and 5 people asked to access the key in each model. For USDSMC, only 1 person was able to successfully access the key representing reliability of 80%. Two people were able to access the keys to the other models.
  For USDSMC, 1/5*100=20%
  100%-20%=80%
  For the other models
  2/5*100=40%
  100%-40%=60%.
- **Malicious Files**: 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 USDSMC, no damage was possible. For each of the other approaches, five people on the malicious list were able to upload the file representing a reliability 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 USDSMC-100% since no one was able to access
  For the other models, 5/5*100=100%
  100%-100%=0% reliability rate
- **File Size:** Block or file sizes of 10MB, 50MB, 100 MB, 200 MB and 300 MB were fed into five models over a specific period of time. All the models were able to upload up to the file size of 200 MB successfully. Only in the USDSMC maximum file size of 300 MB can be uploaded as shown in Table-2. So 300 MB is the maximum file size or threshold value that can be uploaded in the proposed model. For USDSMC, all the data were retrieved representing 100%.
  For the other models, 4/5*100=80%

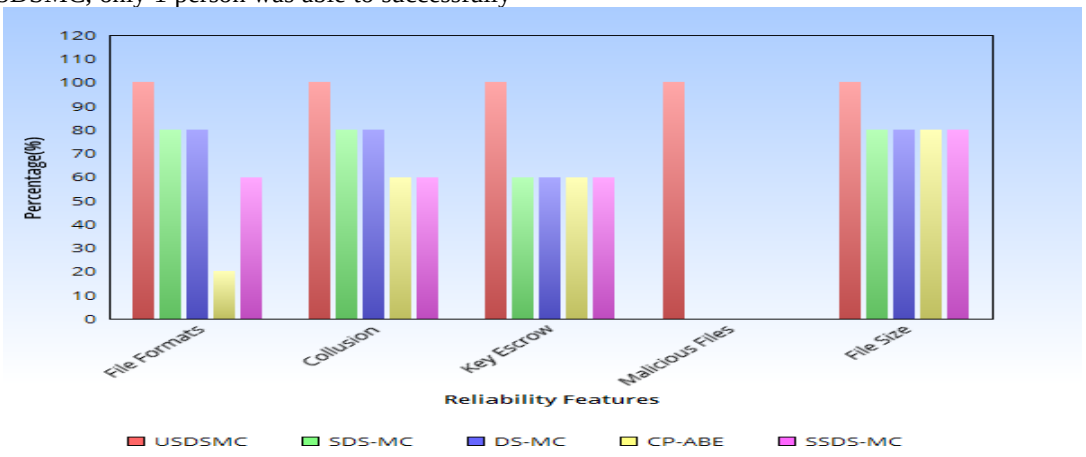


Fig. 5 Reliability Comparison

The figure-5 shows the reliability comparison report of the various approaches. It is to be observed that the proposed method provides 100% reliability features whereas other approaches are not capable of providing those percentages. It is because most of the existing approaches provide a lesser focus on malicious insider, malicious files and monitoring event failures in designing the infrastructure. As discussed earlier reliability main features have been weighed in the analysis and 100% reliability has been accomplished without altering the efficiency and flexibility of the existing approach.

6 Future Enhancement

Although the proposed Multi-Cloud model offers the good experimental results for the providers and consumers there are some limitations to be considered. The outcomes of the monitoring data can be applied to analyze the security in the proposed approach, however, as the number of log sources grows, and then does the volume of the log data being gathered. This growth never follows a linear path. Each system generates more and more data; and with each system, another system comes into the scope. If all systems and devices are sending logs to a centralized system, which is the ultimate goal, the volume of data quickly becomes unmanageable. With the systems now producing more log information than ever before, and diverse data sources required to seek out and locate a threat within the network, a new path to perform data analysis and identify correlated events is required. Big Data arises as a result. In future dynamic data slicing using 3DES can be done.

7. Conclusion

The proposed approach offers a solution for malicious files, insider threats, and key escrow problem while sharing unstructured data in multi-cloud. It supports the various file format to outsource their data into the clouds. This work offers a solution in designing safe, dependable and efficient architecture for the Multi-Cloud service providers. The experimental results demonstrate the public presentation of the proposed framework which is desirable for any customers or organizations to share their data securely in the Multi-Cloud. The security and reliability analysis of the proposed framework also enhance the confidence of the consumers. The observational results and analytical reports prove that the suggested method is safe, reliable and efficient while sharing the unstructured data in Multi-Cloud Storage.

References

[1] DananThilakanathan, et.al., “Secure Data Sharing in the Cloud”. In Security, Privacy and Trust in Cloud Systems,Springer Berlin Heidelberg,2015,(pp. 45-72).

[2] Subramanian,K and Leo, J. “Data Security in single and Multi-Cloud Storage-An overview”. International Journal of Innovative Research in Computer and Communication Engineering, Volume-4 Issue-11, 2016, pp.19046-19052.

[3] Jens-Matthias Bohli, et.al “Security and Privacy –Enhancing Multi-Cloud Architectures, IEEE Tranasactions on Dependable and Secure Computing, Volume: 10,NO:4,2013, pp.212-224.

[4] Abdul Razaque et.al.,“Secure Data Sharing in Multi-clouds.” International Conference on Electrical,Electronics and Optimization Techniques(ICEEOT),March 2016.

[5] Mazhar Ali,et.al.,“SeDaSC: Secure Data Sharing in Clouds”, Systems Journal, IEEE, volume :PP Issue:99,2015,pp.1-10.

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

[7] Vaidya, M. B., and Sandeep Nehe. "Data security using data slicing over storage clouds." Information Processing (ICIP), 2015 International Conference on. IEEE, 2016.

[8] Safaa Salam Hatem,et.al., “Malware Detection in cloud Computing”,*International Journal of Advanced Science and Computer Science Applications*,Vol 5 No-4 2014,pp.187-192.

[9] PengXu,et.al., “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.

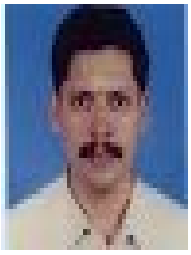
[10] Benjamin Fabian et.al.,”Collaborative and secure sharing of healthcare data in multi-clouds” Information Systems, Volume 48 Issue C, 2015,pp 132-150.

[11] Vaishal, Chauhan and Ani, Singh. “Security Pitfalls In Multi-Cloud Computing Environment.” International Journal of Science, Technology and Management, Volume-5 Issue -01,Jan 2016 pp.150-155.

[12] Xu, L,et.al., “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

[13] Pratheeba, M. S., and K. G. Santhiya. "Security Enabled Group Key Agreement in Multi Cloud." International Journal of Engineering

Science 3157, Volume-6 Issue-11, 2016, pp.3157-3561.
[14] Zibouh, Ouadia,et.al., "Cloud Computing Security Through Parallelizing Fully Homomorphic Encryption Applied To Multi-Cloud Aproach", Journal of Theoretical and Applied Information Technology 87.2,May 2016,Volume-87, pp. 300-307.
[15] Subramanian,K and Leo, J. “Enhanced Security for Data Sharing in Multi-Cloud Storage (SDSMC)”. International Journal of Advanced Computer Science and Applications, Volume 8 Isuue-3 2017, pp.176-185.
[16] Jacob, Dr, and Dr A. Murugan. "Towards the Secure Storage of Images on Multi-Cloud System." arXiv preprint arXiv: 1611.07633 (2016).
[17] Kaur, Satpreet, and Mandeep Singh. "A Novel Cryptographic Key Distribution Scheme for Cloud Platforms." International Journal of Computer Applications Volume-122 Issue-3, 2015 pp.22-25.
[18] Indu, I.,et.al., "Secure File Sharing Mechanism and Key Management for Mobile Cloud Computing Environment." Indian Journal of Science and Technology Volume- 9 Issue-48 , 2017 pp.1-8
[19] Arote, Swapnali Vilas, and R. L. Paikrao. rop." Improved Shamir's Secret Based Key Aggregate Mechanism for Secure Data Sharing in Multi Cloud Storage”International Journal of scientific Research in Science and Technology.Volume-3 Isuue-1, 2017 pp.236-243.
[20] Afolaranmi S.et.al.,"Methodology to obtain the security controls in multi-cloud applications.”In Proceedings of the 6th International Conference on Cloud Computing and Services Science “- Volume 1, 2016, pp.327-332.
[21] Djemame Karim,et.al., "A risk assessment framework for cloud computing." IEEE Transactions on Cloud Computing Volume-4 Issue-3, 2016,pp. 265-278.
[22] Ashalatha,et.al.,. "Data storage security algorithms for multi cloud environment." Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2016 2nd International Conference on. IEEE, 2016.
[23] Hadji, Makhlouf. "Scalable and Cost-Efficient Algorithms for Reliable and Distributed Cloud Storage." International Conference on Cloud Computing and Services Science. Springer International Publishing, 2016.
[24] Wen,et.al.,. "Cost effective, reliable and secure workflow deployment over federated clouds." IEEE Transactions on Services Computing Volume-PP Issue-99, 2016.
[25] Reddy,et.al.,. "Distributed authentication for federated clouds in secure cloud data storage." Indian Journal of Science and Technology Volume-9 Issue-19, 2016,pp.1-7.
[26] Wei, Hang, and Pei-Li Qiao. "Reliability Assessment of Cloud Computing Platform Based on Semiquantitative Information and Evidential Reasoning." Journal of Control Science and Engineering 2016 pp.1-10.
[27] Kalyani, B and Ramchand, Kolosani. “Significance Of Software Reliability Assessment In Multi Cloud Computing Systems A Review.” International Journal of Computer Engineering and Technology (IJCET), Volume 6, Issue 7, July 2015, pp. 35-40 2016.
[28] Khasim Shaik et al., “Implementation of Encryption Algorithm for Data Security in Cloud Computing”,. International Journal of Advanced Research in Computer Science Volume 8, No.3,April 2017.pp.579-583.



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.

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