International Journal for Rapid Research in Engineering Technology & Applied Science
Vol V Issue IV April 2019
ISSN Online: (2455-4723)

A Secure Group Sharing Scheme with Access Control for Cloud Data

Ayushi Dashore[1] Mohit Jain[2]

1 M.Tech Scholar 2 Head of Department Computer Science

1,2 Department of Computer Science & Engineering

1,2 BM Group of College of Engineering and Technology, Indore, Madhya Pradesh, India
ayushidashore1992@gmail.com,bmctmohitcs@gmail.com***

Abstract –**Cloud computing is a new technology with respect to other technologies. Therefore every day we can see new advancements in cloud system. In this presented work the cloud computing is the main area of investigation. The basic idea of working with the cloud environment is to develop and design an enhance approach of secure data sharing and access control. In this context the different security models are explored and we find that the key management is the complex issue in cloud data sharing. Due to this the communication overhead and the computation overhead is increases. Therefore the proposed work is influenced to overcome the additional overhead of key management. The proposed approach provides a methodology that help to manage the access control and the sharing of files with the low computational and communication overheads. In order to design the proposed technique the ABE encryption technique is used. In this context the proposed technique is implemented in two major parts in first the authentication and authorization is managed. In addition of user's attribute is developed. The computed user attribute is used in next module for encryption and file sharing. Thus this module includes the encryption process using SHA1, MD5 and 3DES. In next for communication of secure keys between all the parties the ECC algorithm is used that encrypt the secure key for decryption. The implementation of the proposed approach is performed using JAVA technology. After implementation the performance of the system is measured in terms of server response time, time complexity and space complexity. The results demonstrate the efficient and secure outcomes of the proposed system.**

Keywords: ***ABE encryption, access control, files sharing, group management, secure file hosting, cryptographic cloud storage;***

I. INTRODUCTION

Cloud computing is the long dreamed vision of computing as a utility, where data owners can remotely store their data. The basic service provided by the Cloud is Data Storage. However, it is a difficult task for sharing data in multi-owner manner where group admin and all group members can store and modify data while preserving data and identity privacy from an un-trusted cloud server, due to the frequent change of the membership. Many of the public cloud computing services have appeared for data storage in group applications. Two important problems that arise when sharing group data in public cloud are the privacy and security of group member's data.

Cloud service providers are separate administrative entities and users don't have access to the cloud internal operational details. Because of the semi trust nature of cloud service provider, the traditional security technologies cannot be directly applied to the public cloud based group data sharing applications. So that Data sharing is increasingly important for many users and sometimes an essential requirement especially for industries. Sharing group resource among cloud users is a major problem, still the data privacy leak. Most of the traditional systems are use Group Key Management method for sharing Key Generation and distribution in the group member or users. Sometimes change to user one group to another group, the group key to enable authenticated users to access the files securely and efficiently is still a challenging problem. Here we are proposing a security framework for dynamic group data sharing that make accessible to data file in secure manner in public cloud environment.

II. PROPOSED WORK

The cloud is one of technology which is frequently accessed now in these days. Now only for computational ability its now in these days also be used for storing data. In this presented work the user access policy and Secure Sharing is primary concern for investigation. This chapter includes the system using which the proposed security system is demonstrated.

A. System Overview

The uses of computer based applications are increases in recent years. In distributed systems users need to share sensitive information with others based on the recipients' ability to satisfy a policy. Within the computing environments, the cloud servers can validate numerous data services, such as access control data, storage and outsourced secure computation. For data storage, the servers kept a huge amount of shared data, which has to be accessed by, authorize centralized users. A number of real world applications become online to serve their consumers continuously. In those applications a number of user with the different roles are interacted and performs their functions. But the individual authorities have their own access rights and their own privacy therefore the secure sharing and access control mechanism is required for regulating the functions appropriately.

In this age of computer system, most of the private information and data have to be secured in public cloud environment. Data encryption is a basic solution to maintain security of data and the encrypted data is uploaded into the cloud. Depending on the possibility to identify privacy and security users cannot join the cloud computing systems. In this concern, in a public cloud environment, sharing sensitive information is a challenging task. This can be done using cryptographic security for sharing data among multiple users.

Therefore, in a simple term, User Attribute based Data Encryption is a wider vision for public key encryption that gives users to encrypt and decrypt messages based on user attributes to protect their confidential sensitive information from unauthorized parties. Therefore in this thesis work, we have proposed secured data model named ***Attribute based Access Control for Dynamic Group Secure Sharing Approach*** proposed which ensures both the security and privacy of data sharing and trust for end user. Therefore the proposed mechanism of data access control for secure sharing is designed and implemented. This approach offers the cryptographic security of data using the user based attributes.

This section provides the formal overview of the proposed system in next section, the problem formulation and proposed methodology of the system design is explained in details.

B. Problem Identification

Basically in an access control based data sharing system the access policies and user credentials are included. The access policy defines the role of users in the groups and decides which user can perform which task. Therefore the system can be demonstrated using the following tuple.

[User list, access policy]

In this context the user list is fluctuating factor which is depends on the users exist in a group, additionally their revocation of group and joining new members increases the overhead of key managers. If the user attributes are directly used then for each joining of new member and revocation impact on the key generation. And need to be update keys frequently. Therefore in order to deal with the existing problem without including the additional authority a new kind of solution is required. That helps to improve the existing system. The proposed work is intended to reduce the overhead of key manager, that also reduce communication overhead and frequent key generation and updates.

C. Methodology

"Methodology" implies more than simply the methods we intend to use to collect data. It is often necessary to include a consideration of the concepts and theories which underlie the methods. Unlike an algorithm, a methodology is not a formula but a set of preparations.

The proposed model for securing the data over the group sharing technique is demonstrated in this section. The proposed system is described in two major modules as:

1. **Authentication and authorization:** basically in this module the primary user credentials are used for profiling of the target user. By profile extraction of the user system decides the user role for access control activity and the user attributes which are used for performing the cryptographic operations.
2. **Data encryption and sharing:** this module provides the information about how securely the system processes the data files and how we can use the system to securely share the files among different group users.

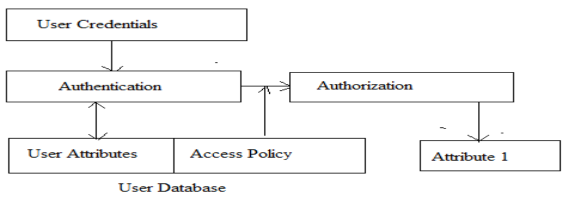

Figure 2.1 authentication and authorization

The figure 2.1 demonstrates how the authentication and authorization on the system is taken place for the proposed secure cloud data sharing environment. The details about the given system in figure 2.1 according to their involved components are given as:

1. **User credentials:** in order to access the proposed secure data sharing system user registration process required. During the registration process user provides some private attributes and also generate the login credentials i.e. user id and password. Therefore to access the system user requires the user id and password for authentication.
2. **User database:** the user database is a secure data storage unit which contains two basic part of user information. First the user attributes which are provided during the registration process with user id and password. Secondly part of information is user's access policy. Therefore each user instance is defined through the access flag for identifying which user can access which kind of data. In other term this attribute define the user role for access control management.
3. **Authentication:** the authentication is process for verifying the user and their identity for secure data access management. In this context the user provides its login credentials (i.e. user id and password). If the given information is available on the user database then system provides access to system otherwise prompt error for valid inputs. When the user get access to the system the user profiling is initiated this is described in next section.
4. **Authorization:** as user get access to the system. The system extracts the additional information from the user database. Therefore a random user information (user attribute) and the corresponding access policy is extracted and combined in a string. The access policy decides the user authority for the system.
5. **Attribute:** the combine information i.e. user randomly selected attribute and the access policy flag is organized in a string which is works are user attribute information for cryptographic scenarios.

As the user attribute is generated the user can use the system for uploading, downloading and sharing of data among different users who are available on the system. The cryptographic scenario for processing the data files are demonstrated in figure 2.2. Additionally their different components are listed as:

Input file: This file or data which is required to upload to the cryptographic server. User can select a text file for local computer and use as the input for the proposed system. In this presented work the text file is used for experimentation. After cryptographic process this file either hosted on secure server or shared among multiple users.

Attribute 1: the attribute which is generated is previous module is used as the secondary input to the system. Basically it works as signature of actual file owner with their role and the private attribute (information). Similarly each user has their own set of signature. Additionally as the user included sharing the file the target users credentials (attribute 1) is also included for security and data access purpose.

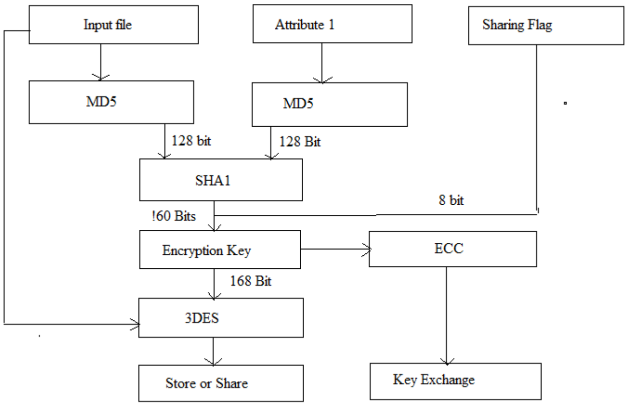

Figure 2.2 data sharing and cryptography

Sharing flag: that is the third and essential input to the system which is prepared for sharing purposed and the data movement tracking purpose. Therefore a fixed length of bits (8 bits) are prepared for identifying the sharing, for example if the user want to just store the data for their own then the sharing flag becomes 00000001, or for one to one sharing it becomes 00000010 and so on. Therefore these bits are depends on the system designer but the only restriction is that it have only 8 bits.

MD5: the MD5 is also known as message-digest version 5. This algorithm is a hash generation algorithm which accepts any length of string and generates only 128 bits as hash key. In this work the MD5 algorithm is used two times first for generating the hash code for input file and second for user attribute 1. Both the 128 bits is used in further processes for generating the cryptographic key.

SHA1: the SHA1 is also a hash generation algorithm that is stronger than MD5 algorithm. This algorithm generates the 160 bit of hash key. The two different hashes generated by MD5 in previous phase is produced in this algorithm therefore an 256 bit string is passed on the SHA1 algorithm to generate 160 bit hash code.

Encryption key: in this phase 160 bits of SHA1 hash code is used for preparing the final encryption key. Therefore 160 bits of SHA1 hash and 8 bit of sharing flag are combined to generate the complete 168 bits of encryption key.

ECC: input message is processed using the ECC (elliptic curve cryptographic technique). The ECC algorithm is asymmetric key encryption techniques which generate two keys namely one private key and second the public key. The encryption algorithm encrypts the data using public key and only the private key can decrypt the data. The ECC algorithm works on the following manner.

Suppose Q= public key

P= a point in curve

d= private key

M= original message

K= random number

Then, using above parameters we get the two cipher text blocks which are denoted using $C_1$and C_2.

$$C_1 = K.P$$

$$C_2 = M + KQ$$

The encryption process input file convert into byte array and then set x_1, y_1, x_2, y_2, for curve generation. Now get points on the curve and generate a random no. d from 1 to N. Calculate public key with the help of d and p and generate cipher text C_1 and sequence of C_2. Using the above equations the message can be defined as:

$$M = C_2 - d * C_1$$

$$M = C_2 - KQ$$

At the network scenarios the cipher C_1 and C_2 is sanded on network and the recovery of the original message can be found using the below given expression.

$$M = C_2 - d * C_1$$

$$C_2 - d * C_1 = (M + KQ) - d * (K * p)$$

In next step

$$C_2 - d = M + KQ C_2 = M + KQ$$

$$M = M$$

The 168 bit cryptographic key is encrypted using ECC algorithm for secure key exchange. Additionally that is kept preserve for the future data decryption.

Key exchange: the 168 bit hash code which is generated using the SHA1 hash code and the file sharing flag bits are encrypted using the ECC algorithm. That is secure and light weight thus the encryption and decryption keys are exchanged securely.

3DES: in this phase the initial input text file which is required to secure and share is provided as input. In addition of that the 168 bit generated key is also included as input. The triple DES encryption algorithm is applied over the selected input text file using the 168 bit encryption key.

Store or share: the outcome of the 3DES algorithm is used for sharing or hosting in the cloud server where different user can get the files for their use.

D. Proposed Algorithm

This section provides the process steps of the proposed system in terms of algorithm steps, the algorithm followed for the system is designed in two phases both the algorithm steps are described as:

```
Input: user ID U, Password P
Output: generate user attribute A

Process:
    1. $AU = verifyUser(U, P)$
    2. $if (AU == True)$
        a. $UserA = ExtractUserAttribute\ (U)$
        b. $AP = ExtractAccessPolicy(U)$
        c. $A = UserA + AP$
    3. Else
        a. Create Error Message
    4. End if
    5. Return A
```

Table 2.1 authentication & authorization

```
Input: file to secure F, Attribute A, Sharing flag F
Output: Encrypted Key E, File to share FS

Process:
    1. $R = readFile(F)$
    2. $Key_{128}^{F} = MD5.GenrateHash(R)$
    3. $Key_{128}^{A} = MD5.GenrateHash(A)$
```

```
4. $Key = Key^{A}_{128} + Key^{F}_{128}$
5. $Key^{SHA}_{160} = SHA1.GenrateHash(Key)$
6. $KeyF = Key^{SHA}_{160} + F^{Share}_{8}$
7. $FS = 3DES.Encrypt(F, keyF)$
8. $E = ECC.Encrypt(KeyF)$
9. Return FS, E
```

Table 2.2 data sharing and cryptography.

III. Results analysis

The experimental evaluation and the system performance is computed and demonstrated in this chapter. Therefore some essential performance parameters are obtained and listed with their obtained observations.

A. Encryption Memory

The amount of main memory required to execute the algorithm with the input amount of data is known as the encryption memory. The total memory consumption of the algorithm is computed using the following formula.

$$\text{Consumed Memory} = \text{Total Memory} - \text{Free Memory}$$

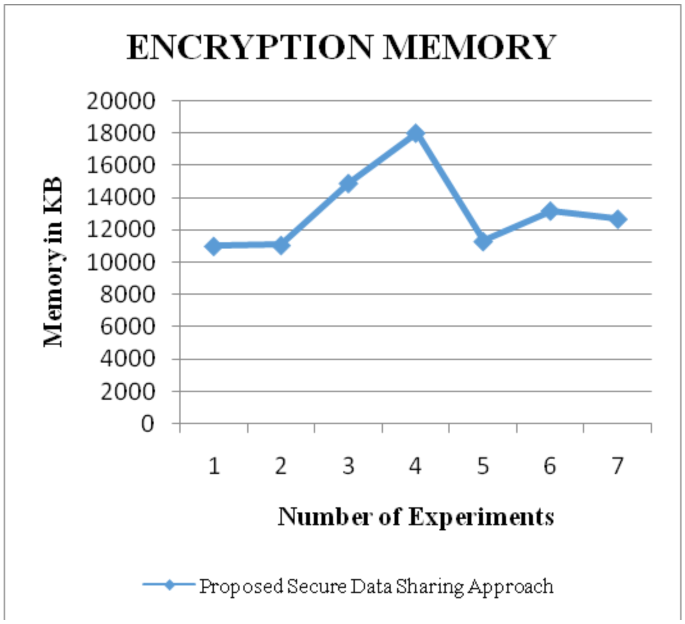

Figure 3.1 Encryption Memory

The figure 3.1 and the table 3.1 show the encryption memory consumption of the proposed approach. In this diagram the amount of main memory consumed is given in Y axis and the number of experiments are reported in X axis. According to the obtained performance the proposed algorithm consumes fewer resources as we seen during the execution of algorithm.

Table 3.1 Memory Consumption

Number of Experiments	Proposed Secure Data Sharing Approach
1	11005 KB
2	11061 KB
3	14904 KB
4	18014 KB
5	11301 KB
6	13194 KB
7	12689 KB

B. Decryption Memory

The amount of main memory required to recover the original file from the cipher text is known as the decryption memory consumption. The figure 3.2 and table 3.2 shows the amount of main memory consumed during the data recovery process.

In this diagram the X-axis depicts the different experiments of different file size used for decryption and the Y axis shows the amount of main memory consumed during the decrypting data file. According to the obtained results the amount of main memory used is less than of encryption memory and consume less space of proposed algorithm.

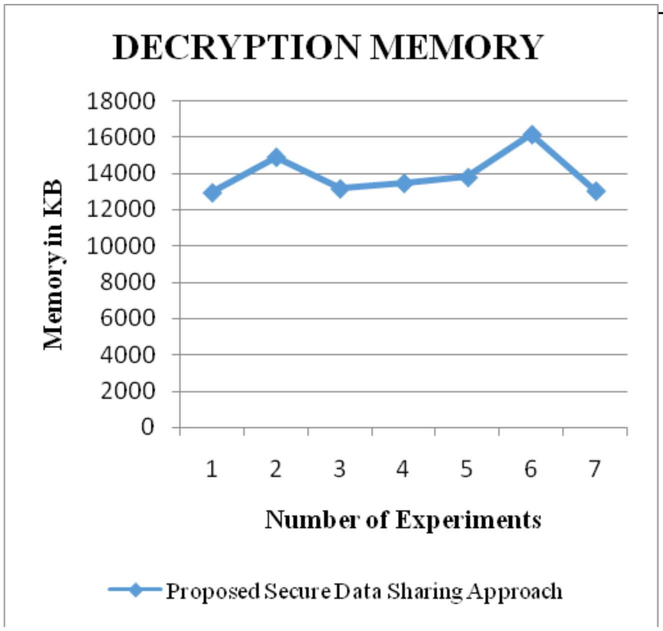

Figure 3.2 Decryption Memory

Table 3.2 Decryption Memory

Number of Experiments	Proposed Secure Data Sharing Approach
1	12951 KB
2	14891 KB
3	13162 KB
4	13465 KB
5	13784 KB
6	16132 KB
7	13041 KB

C. Encryption Time

The amount of time required to perform encryption using the selected algorithm is termed as the encryption time of the system. The encryption time of the proposed system is demonstrated using figure 3.3 and the table 3.3.

$$\text{Time consumption} = \text{Algorithm End Time} - \text{Algorithm Start Time}$$

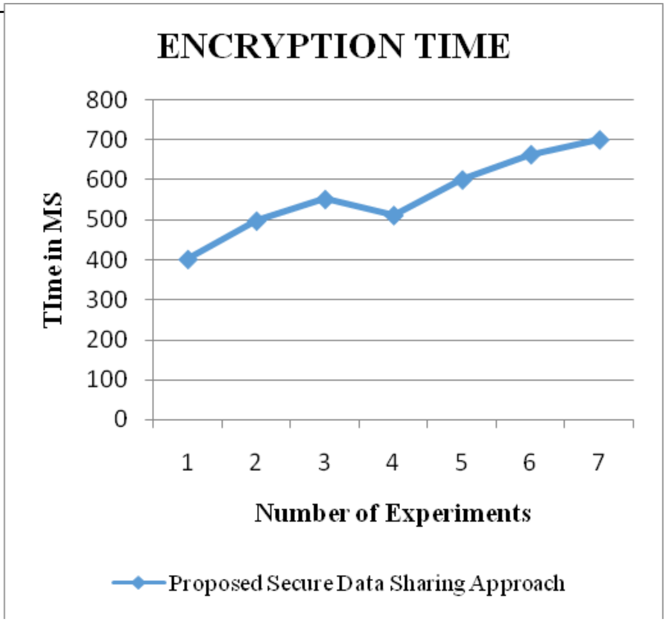

Figure 3.3 Encryption Time

In order to show the performance of implemented data sharing scheme, encryption execution time is reported in figure 3.3 and table 3.3. In this diagram the X axis shows the different experiments on which we run different files as an input and the Y axis shows the amount of time consumed for encrypting the input text file. Additionally the performance of proposed system is given using blue line. According to the given results the proposed system consumes less time for file uploading. Additionally the results shows the amount of time consumed is depends on the amount of data provided for execution. Moreover, while using proposed data security, enhance the security respect to the sharing of file among different parties.

Table 3.3 Encryption Time

Number of Experiments	Proposed Secure Data Sharing Approach
1	402 MS
2	498 MS
3	552 MS
4	512 MS
5	601 MS
6	664 MS

7	701 MS

D. Decryption Time

The amount of time required to recover (Decrypt) the original data from the cipher text is known as the decryption time of the algorithms. The figure 3.4 and table 3.4 shows the obtained performance of the system in terms of millisecond. To show the performance of secure sharing scheme the blue line shows the performance of proposed algorithm.

In given figure 3.4, X-axis shows the different numbers of experiments are performed and the Y-axis shows the amount of time consumed for decryption process. According to the generated results the encryption time is higher than the decryption time in the system, but the decryption time of the proposed algorithm is much adaptable and after secure sharing user can be downloaded in their system.

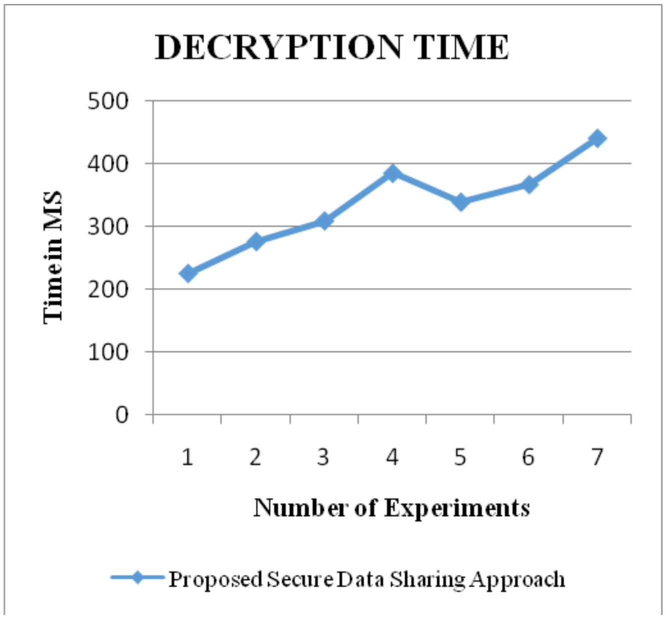

Figure 3.4 Decryption Time

Table 3.4 Decryption Time

Number of Experiments	Proposed Secure Data Sharing Approach
1	225 MS
2	276 MS
3	309 MS
4	385 MS
5	339 MS
6	367 MS
7	441 MS

E. Server Response Time

The amount of time required to produce the outcome after making the request from the server is termed as the server response time. The response time not included the encryption or decryption activity during these measurements. The computed response time for proposed cryptographic technique is shown in figure 3.5 and table 3.5.

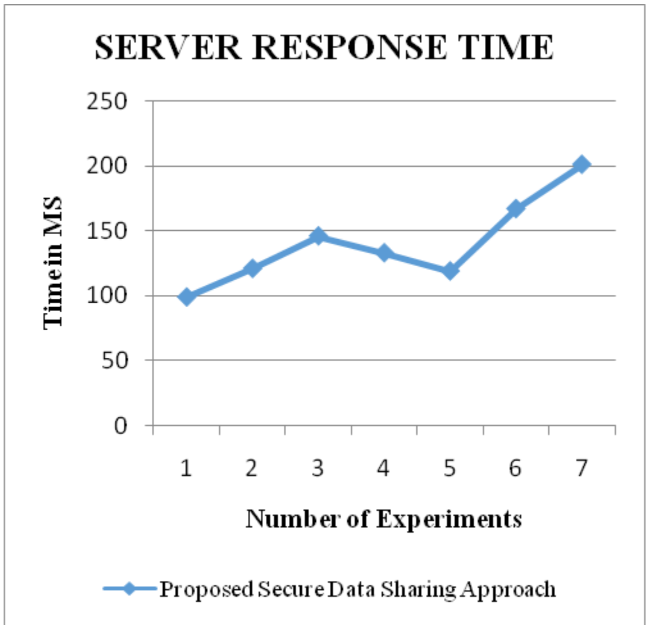

Figure 3.5 Response Time

X axis of this diagram contains the amount of experiments performed and the Y axis shows the amount of time required for generating the response through the server. This can also term as the communication overhead for the system. According to the computed results the response time is not depends on the amount of file size or other parameters. That is directly depends on the amount of work load on the target server where the data is stored or the application is hosted.

Table 3.5 Response Time

Number of Experiments	Proposed Secure Data Sharing Approach
1	99
2	121
3	146

4	133
5	119
6	167
7	201

IV. CONCLUSION & FUTURE WORK

The proposed work is intended to provide a dynamically secure group data sharing and access services in a decentralized manner. This chapter provides the summary of the performed for cloud oriented in security concerns and the future extension of the work is also suggested.

A. Conclusion

Access policy is a mechanism that provides security facilitates the data to user in a controlled manner. The traditional mechanism is that the data is encrypted with the user's public keys. The data owner encrypts the data using this user's public key and then uploads the file to the cloud. The user whenever wanted to download the file should decrypt the file with his generated secret key. By doing this there are a few problems like the owner has to get the public key of the user and the same data is encrypted with different public keys this results in storage overhead.

Public clouds are popular nowadays, where they are generally used in the storage and retrieval of the user's information. It is given as a secure way of data sharing with multiple members. It has very impact in the user's way of data storage. The study of secure data sharing is an increasingly research problem. The main aim of the proposed work is to provide a secure and efficient data group sharing and storage services using the public cloud. Reducing data storage and processing cost is a mandatory requirement of any organization, while analysis of data and information is always the most important tasks in all the organizations for decision making.

In this proposed access control based secure sharing approach, file sharing is initiated and uploaded in the cloud dispersed environment. The file is shared in the user group which created dynamically. The proposed model is based on the user attribute which is generated by user authentication and authorization process and implement access control by means of user credential. Finally, for data encryption and decryption process for group data sharing, we take user input file and attribute simultaneously. Thereafter, 128 bit of MD5 generate of these two inputs. This MD5 pass to SHA1 algorithm, this hash accept 256 hash value of MD5. Therefore, SHA1 produces 160 bit hash value and additionally, sharing flag generate 8 bit flag which input to the SHA1, Hence, SHA1 finally produce 168 bit hash value. By this hash value we have generated key and encrypt it. The encrypted key size is 168 bit. The encryption key will also pass to ECC algorithm which is performed key exchange operation. For file encryption, we input file and apply triple DES algorithm and generate encrypted key, on this input file. After encryption, process we will share this file among the group or store to cloud server. For getting original file, we have follows this process as in reverse manner and user can download the original file.

The implementation of the proposed approach is provided using JAVA environment. After implementation of the system the performance of the system over different parameters are computed. Based on the experimentation the following performance outcomes are obtained as listed using table 4.1

Table 4.1 Performance Summary

S. no	Parameters	Remark
1	Encryption time	The encryption time is acceptable for secure data storage service and increase with the amount of data of to be encrypt
2	Decryption time	The decryption time is less than the decryption time
3	Encryption memory	The memory consumption is acceptable for encryption process
4	Decryption memory	The memory consumption of decryption algorithm is less than the encryption process
5	Response time	Efficient response time even the server implement the cryptographic scenarios

The proposed approach is secure and efficient sharing of data file in public cloud environment, secure and availability of data purpose thus the proposed system is acceptable for data hosting.

B. Future Work

The proposed work for user attribute for access control based dynamic secure group sharing approach and their secure access from the dispersed manner is implemented successfully. Additionally the system performance with the cryptographic implementation of the system is also obtained which is adoptable.

In near future the proposed technique can be extendable for the following application areas.

- ✓ Providing security for the open access clouds such as the social media applications
- ✓ Improving security and access of data in banking applications
- ✓ In future we will focus to improve the efficiency of our proposed model, such as shortening the size of user key, reducing the amount of public information and
- ✓ developing faster encryption/decryption algorithms as traditional in behave now days.

REFERENCES

[1] Zhu, Zhongma, and Rui Jiang. "A secure anti-collusion data sharing scheme for dynamic groups in the cloud." IEEE Transactions on parallel and distributed systems 27, no. 1 (2016): 40-50.

[2] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in Grid Computing Environments Workshop, 2008. GCE '08, 2008, pp. 1-10

[3] Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee et al. Above the clouds: A berkeley view of cloud computing. Vol. 4. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.

[4] Sookhak, Mehdi, et al. "Remote data auditing in cloud computing environments: a survey, taxonomy, and open issues." ACM Computing Surveys (CSUR) 47.4 (2015): 65.

[5] Mell, Peter, and Tim Grance, "The NIST definition of cloud computing," Communications of the ACM 53, no. 6 (2010): 50

[6] "Introduction to Cloud Computing", Dialiogic, available online at: https://www.dialogic.com/~/media/products/docs/whitepapers/12023-cloud-computing-wp.pdf

[7] J. F. Yang and Z. B. Chen, "Cloud Computing Research and Security Issues," 2010 IEEE International Conference on

[8] Computational Intelligence and Software Engineering (CiSE), Wuhan pp. 1-3.

[9] Carroll, Mariana, Alta Van Der Merwe, and Paula Kotze. "Secure cloud computing: Benefits, risks and controls." In Information Security South Africa (ISSA), 2011, pp. 1-9. IEEE, 2011.

[10] Herdman, R. "Information security and privacy in network environments." The Office of Technology Assessment (OTA) (1994).

[11] Sattarova Feruza Y. and Tao-hoon Kim, "IT Security Review: Privacy, Protection, Access Control, Assurance and System Security", International Journal of Multimedia and Ubiquitous Engineering Vol. 2, No. 2, April, 2007.

[12] Oscarson, Per. "Information security fundamentals, graphical conceptualisations for understanding: research group VITS, Department of Business Administration, Economics." Statistics and Informatics, Örebro University, Sweden (2003).

[13] Chaeikar, Saman Shojae, Mohammadreza Jafari, and Hamed Taherdoost. "Definitions and criteria of CIA security triangle in electronic voting system." Information Technology (IJACSIT) 1, no. 1 (2012).

[14] "Introduction of Computer and Network Security", Lecture Notes (Syracuse University), available online at: http://www.cis.syr.edu/~wedu/Teaching/IntrCompSec/LectureNotes_New/Introduction.pdf.

[15] Hussain, Syed Asad, Mehwish Fatima, Atif Saeed, Imran Raza, and Raja Khurram Shahzad. "Multilevel classification of security concerns in cloud computing." Applied Computing and Informatics 13, no. 1 (2017): 57-65.