International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: SJIF: 5.995
Available Online at www.journalijcar.org
Volume 7; Issue 2(F); February 2018; Page No. 10036-10043
DOI: http://dx.doi.org/10.24327/ijcar.2018.10043.1683

Research Article

SDSPG: SECURED DATA SHARING WITH PRIVACY AS A GROUP IN CLOUD COMPUTING

Muthi Reddy P., Manjula S H and Venugopal K R

Department of Computer Science and Engineering, Bangalore University, Bengaluru-560001, India

ARTICLE INFO

Article History:
Received 14th November, 2017
Received in revised form 6th December, 2017
Accepted 18th January, 2018
Published online 28th February, 2018

Key words:

Authentication, Collaborative, Efficiency Group sharing, Individual, Privacy

ABSTRACT

Cloud computing is a prominent emerging data interactive pattern, which helps in online storing the data in the cloud server. Cloud services make available for the users to get beneficiated in the form of accessing, storing data in the cloud. In large organizations, the data sharing may be in a collaborative manner and thus sharing of the data becomes more important to achieve the benefits. Proving of data security over the cloud becomes more challenge when data may be accessed by all group members. Therefore need to focus on the authentication which must be provided to have a legal access to private data. The existing different issues are *privacy*, *authentication*, and data *access* permissions. To overcome these issues, we proposed *Secured Data Sharing with Privacy as a Group in Cloud Computing* (SDSPG). In this paper, more importance is given to the data access permissions on the basis of whether the data is *public* or *private*. SDSPG mechanism is proposed to save timings by assigning sharing and authentication to the individual users. It provides more *security* and *privacy* thus to increase *efficiency* and *effectiveness* of CPU utilization, access *time* and *energy* of the file.

INTRODUCTION

Cloud computing is the main paradigm which is helpful for the individuals as well as for organizations. It provides the data storage facility in an efficient and effective manner. It is an advantage service of cloud computing to provide on-demand self-service, omnipresent network access, and location independent resource pooling [1], [2]. It works outstandingly irrespective of any infrastructures, platform, software, and others are applied for present interconnections. Recent studies have been made to promote the cloud computing and developed gradually towards the internet of services [3], [4]. Afterward, the issues such as *security* and *privacy* are becoming main concerns with which increasing popularity of cloud services.

The main service approaches of using cloud computing are that the user can access the data from anywhere. Along with that, users may want to access and share the data with each other's authorized data to achieve fruitful benefits. This can bring a new security and privacy challenges for the cloud storage [5]. To identify the main motivation, there exist various interest groups for such as supplier, carrier, and retailer etc. in the system.

In the cloud, the data stored may be accessed by any of the users, therefore privacy and security issues must be handled carefully. Instead of allowing all the users to access the particular file, distinguish the private and public file. The private file is the one which can only be accessed by the authorized user. Public file is the one which can be accessed by all users. This distinguishing of the file is made because of the organization, as there will be various types of users in which file may be necessary. The private file can be accessed if and only if a person is an authorized user. The valid user is the one, who had taken the permission from the admin. We proposed the concepts like group management is provided to save the time of admin who gives authentication to all users upon getting information from the data owner. The data owner is the person who uploads the file to the cloud and who want others to access user private data if and only if it is a private file. The feature comparison of different schemes by considering different parameters are shown in Table 1.

Table 1

Feature Comparison				
Sl. No.	**Name of the scheme**	**Type of authentication**	**Accessing method**	**Time consumption**
1	SAPA [1]	Individual	Private/Public	More
2	MONA [7]	Individual	Private/Public	More
3	SDSPG	Group	Private/Public	Less

Motivation

To provide *Security* and *Privacy*, which are the main concerns of the data, which needs to be stored in the cloud. To provide such security over the cloud, Shared Authority Based Privacy Preserving Authentication Protocol in Cloud Computing (SAPA) [1] approached in an efficient way. This paper plays an important role in putting some effective part in terms of concept analysis and implementation. In the SAPA, the

Corresponding author:* **Muthi Reddy P
Department of Computer Science and Engineering, Bangalore University, Bengaluru-560001, India

concept is only taken care of the *individual* users, therefore it consumes more *time* and *energy* to access the file. To overcome these drawbacks, we proposed *Secured Data Sharing with Privacy as a Group in Cloud Computing* (SDSPG) mechanism.

Contribution

As described in motivation, SAPA [1] which is based on privacy and security provides authentication for the individual users. Administrator (*Admin*) is the person who assigns authentication for the users to get access to the *private* data. Whether to access private data will be intimated by the data owner who uploads the private data on to the cloud. We proposed *group sharing* scheme i.e. for providing authentication in the group manner which helps in saving the *time* and *energy* of admin is being focused. Instead of providing an individual assignment, our scheme SDSPG helps in assigning authentication to the collaborative users. Secured Data Sharing with Privacy as a Group in Cloud Computing (SDSPG) reduces the access time by assigning multiple users at a single instance of time and it is a less cost-effective thus increases the throughput of the CPU utilization.

Organization: The structure of the paper is listed out as follows. We presented related work in section II. We displayed problem definition in section III. In section IV, the background of the paper is discussed. Section V gives the algorithms of SDSPG scheme. Performance evaluation focused in section VI and final in section VII conclusion of SDSPG is represented.

Related Work

The scheme which supports operations of data dynamically is mentioned in the dynamic cloud data [6] indicates the scheme is able to stand against malicious data modification attack, Byzantine failure, and server colluding attacks.

The users' access to the data [7] will give authentication to generate a dynamic as well as a mobile object such that several resources by fetching the Java Archives (JAR) programmable capability encapsulation. The approach used in [8], [9], [10] the scheme contains distributed auditing mechanisms which are provided to users' for data control in most effectively and efficiently. In the previously discussed works, the main issue being focused on security.

In cloud area, the issue which takes more attention is privacy as many of the users will be handling the data with several operations of the cloud. Here, such issues in [11], [12], [13] are identified and taken as a new challenge for accessing the valid data, by also considering the authentication to overcome the privacy preserving as well as access authority sharing. To provide authentication and authorization, the attribute based access control and also proxy re-encryption mechanism is applicable.

Privacy preserving [14], [15] is the one of the main concern which needs to be taken care while storing the data in the cloud as it is a public storage [16]. To secure dependable service for the purpose of safe storage the scheme [17] is been efficiently implemented which is been referred to this work. In a distributed system, the data sharing in a collaborative manner [18], [19] becomes the main issue during sharing of data in the cloud thus to focus more on distributed sharing, the scheme [20] is used. In the cloud, the operations which take place do not only make use of different types of files but media can also be used, the scheme [21] explains about the context in media of placements.

The cloud can be managed with group-wise also as it enables users to have an access to the cloud such as reading a file, writing a file and access permission over the cloud in a collaborative manner. As cloud enables collaborative access control, it provides the group management for data sharing mechanism as a dynamic group data sharing. As there is group management which is been explained in [22], it supports dynamic group sharing of data with respect to different users, but the time complexity of this scheme is less efficient to the SAPA scheme.

To have a reference over the data in the cloud need to have a public key management in group sharing also, to get the access method the scheme [23], [24], is implemented with sharing of keys with respect to the group. But the performance is evaluated in an effective way in SAPA scheme as compared to this scheme.

Table 2 Notation for efficiency computation

Abbreviation	Definition
JAR	Java Archives
GUA	Group User Authentication
IUA	Individual User Authentication
S	Cloud Server
U_x, U_y	Two User's
PHP	Hypertext Preprocessor

Definition

The solution and objective are taken care by analysis of problem definition by considering the problem statement and the objective as follows:

Problem Statement

The current system is intended to provide the security over software and web applications. Nowadays it is necessary to provide the security and privacy of the data which is stored in the cloud. Providing security and privacy over the cloud, mainly in a collaborative relationship, people can access the data in a necessary task.

Objective

To overcome the issues in SAPA [1], we proposed Secured Data Sharing with Privacy as a Group in Cloud Computing (SDSPG). It provides security of data in group sharing as follows; i) Encryption and Decryption process of data file. ii) Automatic file generation for encrypted data and decrypted data. iii) Accessing private data only for authentic users and iv) Group file sharing authentication.

To get an effective encryption and decryption process, including automatic file generation for encrypted data and decrypted data, thus to have an access to private data. For authentic users to provide group file sharing authentication is facilitated.

Background

The background of the paper is presented with two figures such as system model and flowchart.

System Model

In Fig. 1, illustration is made on the System model for the cloud storage interactional aspects which consists of three important entities: *Admin, User* and *Cloud server.*

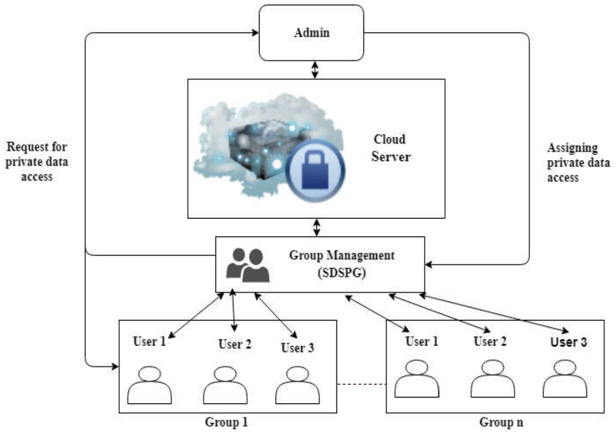

Fig 1 The cloud storage interactional system model.

- ***Admin:*** It is an entity which acts as the main person who manages almost all the operations, which acts as an admin who controls all the actions.
- ***User:*** User may be an individual or group, who will be intended to store the data in the cloud or those who want to download and perform some of the actions (read, write, alter, store etc.) on that downloaded data.
- ***Cloud server:*** Cloud server is the one which manages the data, provide data storage and computing services by a particular cloud service provider or cloud application operator.

The cloud storage interaction system model, SDSPG allows a user in addition to support the group management which contains the set of groups. In each group there exist set of users who and all belong to that particular group.

It provides security of public or private data using a) Single encryption and b) Double encryption techniques.

Single Encryption

The Data owner who want to upload data into the cloud server is given the permission of the admin. The keys automatically generated by server, it is given to authorized requested data owners to upload for encryption. Data owners, need to specify the file access type; whether private or public. Any remote user wants to download file, request sent to admin. Admin will check the registered user or not. If the authorized users in the group admin given the permission to download the files. If the file is declared private, and another user needs to access it, they must request the admin for the decryption key and authenticate the files. If the data owner approves data access to the user then the admin sends key to distributed user in the group. But the user will not receive the key itself. For public files, the admin verifies the authenticity of the user and allows downloads.

Double Encryption

For more security purposes, the files are encrypted twice and uploaded to the server by the data owner. This method of double encryption does not alter the file name. After first encryption, the file without its name being changed, is stored on the server for ciphertext. This file, is once again, encrypted and stored on the server. During downloads, this encrypted ciphertext file is decrypted and again re-decrypted same file, to provide the actual file to the user.

In group sharing GSAPA provides more security for private data by authenticate access permission and sending key of particular file to the remote users in the group. The public file sharing all the members in the group, but not for private file. A user in addition to support the group management which contains the set of groups. In each group there exist set of users who and all belong to that particular group.

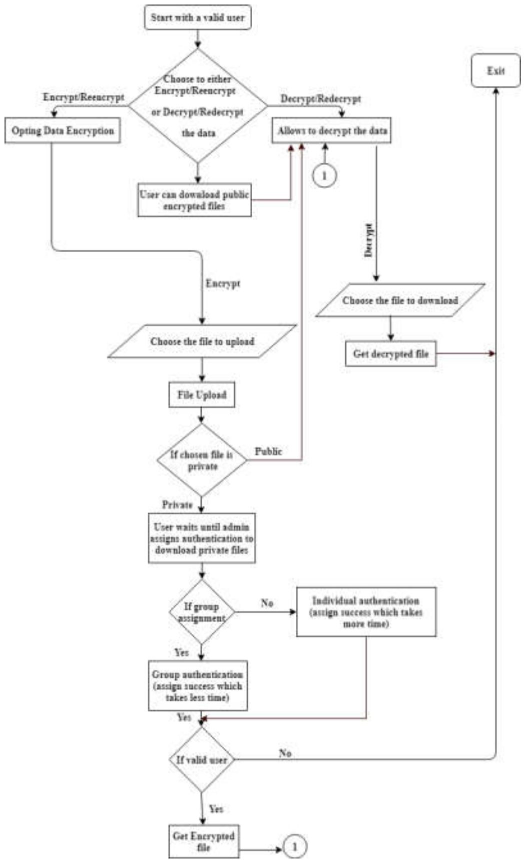
Start with a valid user
Choose to either Encrypt/Reencrypt or Decrypt/Redecrypt the data
Encrypt/Reencrypt
Decrypt/Redecrypt
Opting Data Encryption
Allows to decrypt the data
Exit
User can download public encrypted files
1
Decrypt
Encrypt
Choose the file to download
Get decrypted file
Choose the file to upload
File Upload
If chosen file is private
Public
Private
User waits until admin assigns authentication to download private files
If group assignment
No
Individual authentication (assign success which takes more time)
Yes
Group authentication (assign success which takes less time)
Yes
If valid user
No
Yes
Get Encrypted file
1

Fig 2 Workflow of the SDSPG scheme.

As explained earlier admin plays an important role. Admin is the person who authenticates all valid users. Authentication by admin occurs only when users try to access private files from the cloud management. In the system model, users are the interactional entities who can become a data owner. The data owner is the person or any user, who uploads data in the cloud and informs the admin by listing the other users through any communication resources (example: email and SMS etc.) to whom data owner wants to provide access when uploaded file is a private file.

Flowchart

The actual workflow of our SDSPG scheme is illustrated in Fig. 2. Initially, start the process with a valid user i.e. after a user performs the registration algorithm. Then a user will get the option to choose whether to encrypt or decrypt the data. By opting the data encryption, the user can encrypt the data i.e. convert plain text to cipher text. We choose the data decryption, the user can decrypt the data i.e. converting the data from cipher text to plain text. The security can be provided by the process of encryption file while uploading and decryption of file while downloading. The user can assign whether that particular file is private or public. If the uploaded file is a public one, the user can download it without any restriction. If the uploaded file is private then check whether the user is valid user to download that file or else take the permission from the admin. Only the admin will be able to fix the particular user is valid or not. Once the user gets the permission from the admin, the user can be able to download that particular file by entering the username and password. The SDSPG scheme can reform re-decryption, which helps the user to opt re-decryption method. Re-decryption is the process which performs the decryption double time. This process occurs when an encrypted file is being encrypted one more time. On the performance of double encryption process, the file must be decrypted twice, so re-decryption operation needs to be carried out.

Algorithms

There are 5 main processes which play an important key role in our SDSPG scheme, which occurs serially such as *Registration*, *Encryption*, *Decryption*, *Individual user authentication* and *Group management* as well as stepwise as shown below.

Registration ( )

1. Get registration form in the browser.
2. Enter data on the respective fields on the basis of validation constraint.
3. The user can choose the process and the user can upload a file for encryption or decryption who becomes the data owner of the particular file.

Encryption ()

1. A user uploads the file to convert the data to encrypted format.
2. Perform encryption process on the data uploaded with parameters as follows:

- ***Key***: It is a secret key to convert plaintext to ciphertext.
- ***Data***: amount of information which needs to be converted to ciphertext
- ***Cipher***: name of the algorithm which is built-in in PHP language.
- ***Mode***: There are certain modes to pass as parameters, such as cbc, ecb, cfb, ofb, nofb, Stream.
- ***Iv_size***: This is a parameter which is passed for initialization of modes.

1. Creates a random file using *generateRandomString()* function and which gives the same name for the file.
2. Copies all encrypted data to randomly generated the file as the ciphertext.
3. Store the data by computing the function:

Uploadfile=Upload_file(size, etime, filename, privacy)

The variables defined as

- *Size*: Size of the file.
- *ETime*: Time is taken for encrypting the data.
- *Filename*: Name of the file which is being uploaded.
- *Privacy*: Whether file is private or public,
- Private represents '0'
- Public represents '1'

Decryption ()

1. User uploads the file to convert the data to decrypted format. Any user can go for decryption but same encryption key must be used which is constantly written in the code.
2. Perform decryption process on the data uploaded with parameters are *Key, Data, Cipher, Mode, Iv_size*.
3. User uploads the file to convert the data to decrypted format. Any user can go for decryption but same encryption key must be used which is constantly written in the code.
4. Perform decryption process on the data uploaded with parameters are *Key, Data, Cipher, Mode, Iv_size*.

Where, Key: It is a secret key to convert ciphertext to plaintext. Both the keys which is being used for encryption and decryption must be same.

1. Creates a random file using *generateRandomString()* function and which gives the same name for the file, same process which takes place in the encryption as well.
2. Copies all decrypted data to randomly generated the file as the plaintext.
3. Get the following information and store it in a database
4. Store the data by computing the function:

Decryptfile=Decrypt_file(size, dtime, filename)
The variables defined as

- *Size*: Size of the file
- *DTime*: Time is taken for decrypting the data
- *Filename*: Name of the file which is being uploaded.

Only while uploading a file for encryption, the files are being differentiated whether it is private or public. During decryption process, any file can be decrypted irrespective of the private and public file. Decryption is an open function. There is a provision for re-decryption which takes double decryption process.

Individual User Authentication ()

1. It is a job of admin who would assign authentication for the user to get access to private data for *Individual User Authentication (IUA)*
2. Displays a list of names where admin can select names to which the user wants to give access permission.
3. Admin gets information about the user such as the first name and file from the database on the basis of following equations:

$$S = \sum_{i=0}^{n} \$r[First\_name] \quad (1)$$

$$F = \sum_{i=0}^{n} \$r[File\_name] \quad (2)$$

1. Select names of the user and respective file to give access permission.
2. Once access permission is assigned, user can login and get private data.

Group User Authentication ()

1. *Group User Authentication (GUA)* is the job of the admin who would assign authentication for the group to get access to private data.
2. *Displays a drop-down of groups*: On opting group, it displays all names of the user from that particular group. After choosing a group, admin can select names to which user wants to give access permission.
3. *Admin* gets information about the user such as first name and file from the database on the basis of equations (1) and (2) as shown above.
4. Select names of the user and respective file to give access permission of that particular file. After choosing name of the user as well as file if admin go for "*active*" option from another dropdown then access is successfully assigned.
5. If opted for "Inactive" user permanently don't get access to that particular file to that particular user unless and until it's been removed from the database.
6. Once access permission is assigned, the user can log in and get private data. To save the time of admin, group management helps in such a way that, to whole group access permission can be assigned without selecting the users one by one.

Performance Evaluation

This paper is implemented using PHP as the main language which helps in certain operations on a database such as storing data on to the database, fetching data from the database to the user interface to perform some actions by the users as well as admin. In this section, we are discussing the simulation setup and experimental results.

Simulation Setup

For creating user interfaces, a framework that is well known and popular as "*Bootstrap*" is used for GUI implementation. Using Bootstrap framework which will reduce time in writing the style for stylesheet which needs to be applied to the user interfaces. The bootstrap framework allows the user to download a zip file which contains Cascading Style Sheet (CSS), fonts, JavaScript (JS). These files contain some initializing and configuration files which can be imported to respective HTML files.

Experimental Results

The process of encryption (or reencryption) takes an argument and data considered is the document file which is given by the user. The encrypted function then converts from raw data to encrypted form by creating another randomly named document file. The randomly generated file which contains encrypted data will then be stored in the cloud. Then that particular file can be accessed based upon whether it is a *public* or *private* file. Currently, it is being focused on the path of admin device.

The following functions processes for each user who wants to get access to private data. Group management is a task of admin to whom authentication must be provided for the groups.
Functions:

$$S = \sum_{i=0}^{n} \$r[First\_name] \tag{3}$$

$$F = \sum_{i=0}^{n} \$r[File\_name] \tag{4}$$

Here, *S* is the variable which contains the array of the names of the user, $\$r$ is the representation of array with the name 'r', *First_name* is the name of the user to whom authentication must be provided and *F* is the variable which contains the array of the names of the file
The cloud storage system includes a cloud server *S*, users U_a and U_b which have independent access authorities on their own users. Let $F(R_{U_x}^{U_y}(R_{U_y}^{U_x})^T) = Cont$ is the algebraic relation of $\{R_{U_x}^{U_y}, R_{U_y}^{U_x}\}$ and $Cont$ is a constant. Any randomized polynomial time algorithm *A* there is a negligible function p(*k)* for a sufficiently large value *k*:

$$Prob[\{(x,x'); (y,y')\} \leftarrow A(1^k): (x \neq x', y \neq y') \wedge F(R_{U_y}^{U_x} (R_{U'_x}^{U'_y})^T) = Cont]] \leq p(k).p(k) \tag{5}$$

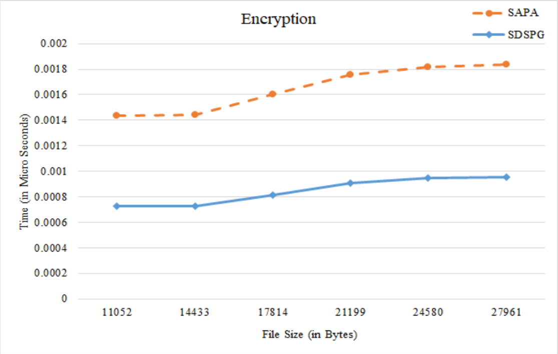

Fig 3 The different file encryption delay using SAPA and SDSPG scheme.

The file size plotted as the x-axis and encryption time as the y-axis is as shown in Fig. 3. The number of group members considered as six. The encryption time and size of the file is in terms of microseconds and bytes respectively. The records from a database are stored by the process of encryption of a file chosen by the user. In this figure SAPA and SDSPG schemes considered gives the slight linear graph. The value in the y-axis, the points differ from 0.0002. The file sizes approximately differ from one point to another in the x-axis is 3k bytes. The file size 11052 bytes takes decryption time 0.00071 microseconds in SDSPG scheme. Therefore the first point lies in between 0.0006 to 0.0008 microseconds. In SAPA scheme, the first point, file is of 11052 bytes and it takes time 0.00142 microseconds (ms).

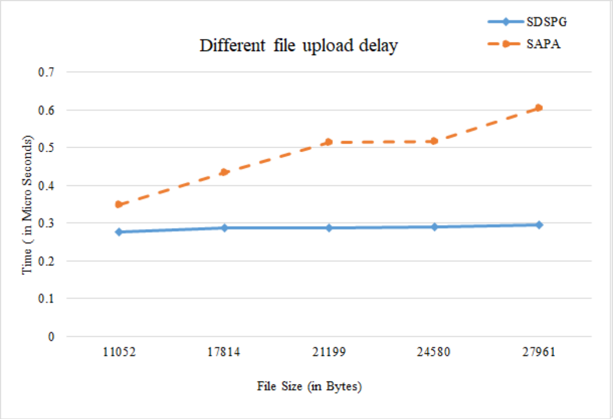

Fig 4 The different file upload delay using SAPA and SDSPG scheme with group size is five.

Therefore the first point lies in between 0.0014 to 0.0016 microseconds. Similarly, as in case of sixth point, the file is of 27961 bytes and it takes time 0.00088 microseconds in SDSPG schem and SAPA scheme it takes time 0.00182 microseconds.

As the time taken varies upon the considered file size, the points on the graph will be drawn slight linearly. Therefore, the SAPA scheme consumes more time to encrypt a file compared to SDSPG scheme.

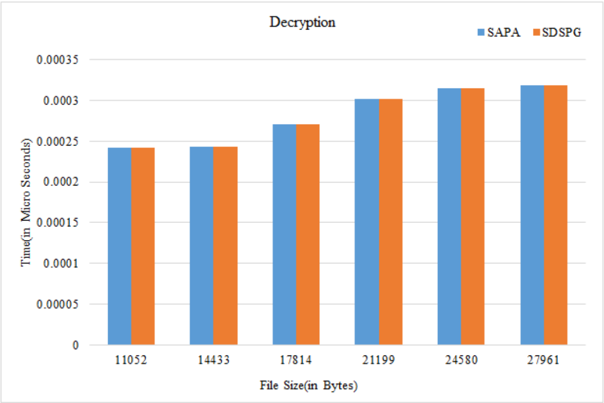

Fig 5 The files decryption delay using SAPA and SDSPG scheme.

The encryption of file to upload in one group of size is five users illustrated in Fig. 4. The upload time is less in SDSPG scheme because of group assignment, whereas in SAPA scheme is used in individual assignment.

In Fig. 5, the number of group members considered as six. The Group member request a file to download, admin giving authentication and sent key to user.

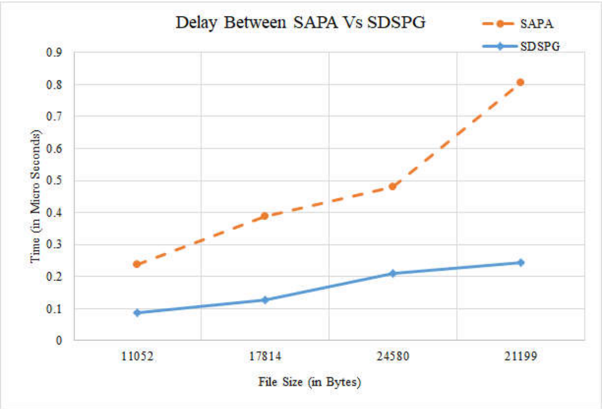

Fig 6 The average time delay to share different files in SAPA and SDSPG scheme of different groups.

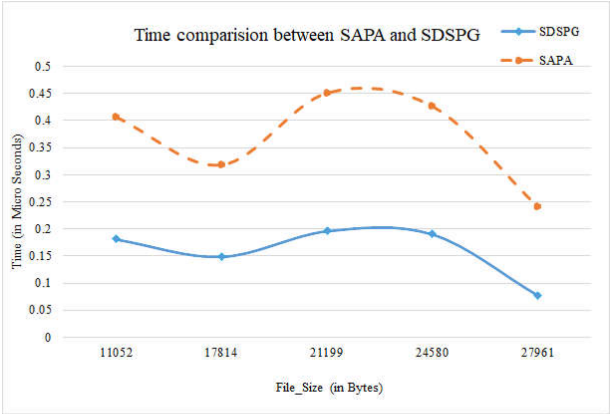

Fig 7 Comparison of SAPA and SDSPG considering the single group of file size five.

Later, user download a file by using key. Therefore, SDSPG scheme is more secure. Whereas in SAPA scheme authentication of admin is not required, so it is less secure compare to SDSPG scheme. The time taken by the decryption function gives the linearly as the size of the file downloaded goes on increasing. To download a file using SAPA or SDSPG scheme of same key when upload secret key only. The time taken to decrypt and download a file in both schemes are same.

The comparison of individual assignment (SAPA) and group assignment (SDSPG) is illustrated in Fig. 6. The number of groups considered as four with different sizes. In SAPA scheme to share encrypted file using individual assignment for group members consumes more time. Because each time file separately upload to each member. GSAPA scheme to upload encrypted file using collaborative manner for group members at same time, therefore, it takes less time. The time consumption is more in SAPA to share file, it leads cost is high. GSAPA takes less time to upload file then the cost is less. The file uploaded goes on changes, as because of the random file is created to store the decrypted data. The value in the y-axis, the points differ from 0.1. The members in the groups are varied, the sharing time also differs. The file size 11052 bytes takes decryption time 0.0867 ms in SDSPG scheme. In SAPA scheme, the first point, file is of 11052 bytes and it takes time 0.2374 ms. Similarly the file size is of 21199 bytes and it takes time 0.244826 microseconds in SDSPG scheme and SAPA scheme it takes time 0.8055 ms. The file size is not in a synchronized way and thus the graph not linear. The time for SDSPG is calculated by considering the different number of members of the group and taking an average of it. Therefore, the SAPA scheme consumes more time to share a file compared to SDSPG scheme.

Table 3 Comparison of time delay for different groups with same file

File Size(in Bytes)	Group Size	SDSPG(in ms)	SAPA(in ms)
27961	2	0.137	0.1482
27961	3	0.1968	0.2257
27961	4	0.212	0.2378
27961	5	0.3025	0.3198

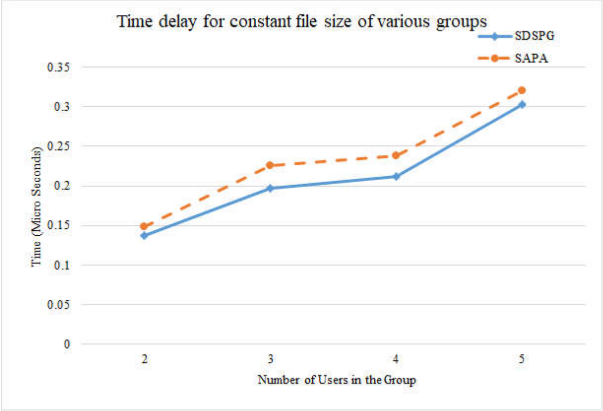

Fig 8 Time delay for shared constant file using SAPA and SDSPG scheme of different group sizes.

After considering the group assignment, the time and size of the file is noted down as an access time to plot a graph from individual user authentication with group user authentication. By observing the graph i.e. Fig. 7. The individual user authentication takes more access time compare to group user authentication. Thus access time for group user authentication is more efficient. Access delay is calculated between individual assignment and group assignment considering single group i.e. members of the group may vary from considered one value to another value. The file size increases or group size increases the time consumption is more in individual assignment various group assignment is less compared to SAPA. The result is shown between individual assignment and group assignment considering but by considering the same file size of 27961 bytes. We can see that the group assignment SDSPG takes less time compared to individual i.e. in SAPA.

The Time delay for shared constant file using SAPA and SDSPG scheme is illustrated in Fig. 8, the time for SDSPG is less when compare to SAPA scheme. The different time delay to share a constant file of different groups as shown in table 3. The access time is less in SDSPG scheme, thus cost of sharing file reduces. The access time is more in SAPA scheme, the cost is high when compare to SDSPG scheme.

CONCLUSION

In SAPA [1], the authentication is provided by considering *individual* users, which increases the time and cost. To overcome these issues in the security aspects, SDSPG scheme provides more authentications which are performed by the admin wherein SAPA, admin entity is considered has optional. In SDSPG scheme *group assignment* is focused more as well as for dynamic groups. The automatic file generation to store encrypted data and decrypted data. *Private* files cannot be accessed by the unauthorized user, where *public* files can be accessed by any of the users as that particular file is kept open to all. The private file can be downloaded only by that user who got the permission from an admin. In this way private and public files can be distinguished successfully. Instead of using only individual allocation i.e. admin selecting each user to get access, group mechanism plays important role in SDSPG scheme.

The SDSPG scheme consumes *less* access time and *cost* compare to SAPA scheme and also CPU utilization is more, hence it is more *efficient* and *effective* to provide high *security* and *privacy* of data. Selecting multiple groups at a time can be considered for the future enhancement.

References

1. Hong Liu, Huansheng Ning, Qingxu Xiong and Laurence T.Yang, 2015. Shared Authority Based Privacy-Preserving Authentication Protocol in Cloud Computing, IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 1, pp. 241-251.
2. Muthi Reddy P, Manjula S. H. and Venugopal K. R., 2017. Secure Data Sharing in Cloud Computing: A Comprehensive Review, *International Journal of Computer (IJC),* ISSN 2307-4523, vol. 25, no. 1, pp. 80-115.
3. P. Mell and T. Grance, 2009. Draft NIST Working Definition of Cloud Computing, Nat'l Inst. of Standards and Technology.
4. A. Mishra, R. Jain, and A. Durresi, 2012. Cloud Computing: Networking and Communication Challenges, IEEE Comm. Magazine, vol. 50, no. 9, pp. 24-25.
5. R. Moreno-Vozmediano, R.S. Montero, and I.M. Llorente, 2013. Key Challenges in Cloud Computing to Enable the Future Internet of Services, IEEE Internet Computing, vol.17, no.4, pp.18-25.
6. L. A. Dunning and R. Kresman, 2013. Privacy Preserving Data Sharing with Anonymous ID Assignment, IEEE Trans. Information Forensics and Security, vol. 8, no. 2, pp. 402-413.
7. X. Liu, Y. Zhang, B. Wang, and J. Yan, 2013. Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud, IEEE Trans. *Parallel and Distributed Systems,* vol.24, no.6, pp.1182-1191.
8. Tao Jiang, Xiaofeng Chen, and Jianfeng Ma, 2016. Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation, IEEE Trans. Computers, vol. 65, no. 8, pp. 2363-2373.
9. Yichao Jin, Yonggang Wen and Kyle Guan, 2016. Toward Cost-Efficient Content Placement in Media Cloud: Modeling and Analysis, IEEE Trans. Multimedia, vol. 18, no. 5, pp. 807-819.
10. Zhijie Wang, Dijiang Huang, Yan Zhu, Member, Bing Li and Chun-Jen Chung, 2015. Efficient Attribute-Based Comparable Data Access Control, IEEE Trans. Computers, vol. 64, no. 12, pp. 3430-3443.
11. Wei Li, Kaiping Xue, Yingjie Xue, and Jianan Hong, 2016. TMACS: A Robust and Verifiable Threshold Multi-Authority Access Control System in Public Cloud Storage, IEEE Trans. *Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1484-1496.
12. S. Grzonkowski and P.M. Corcoran, 2011. Sharing Cloud Services: User Authentication for Social Enhancement of Home Networking, IEEE Trans. Consumer Electronics, vol. 57, no. 3, pp. 1424-1432.
13. Xinfeng Ye, 2016. Privacy Preserving and Delegated Access Control for Cloud Applications, Tsinghua Science and Technology, vol. 21, no. 1, pp. 40-54.
14. Yong Yu, Yannan Li, Jianbing Ni, Guomin Yang, Yi Mu, and Willy Susilo, 2016. Public Integrity Auditing for Dynamic Data Sharing With Multiuser Modification, IEEE Trans. *Information Forensics and Security*, vol. 11, no. 3.
15. Song Han, Shuai Zhao, Qinghua Li, Chun-Hua Ju, and Wanlei Zhou, 2016. PPM-HDA: Privacy-Preserving and Multifunctional Health Data Aggregation With Fault Tolerance, IEEE Trans. Information Forensics and Security, vol. 11, no. 9, pp. 1940-1955.
16. Kai Fan1, Qiong Tian, Junxiong Wang, Hui Li and Yintang Yang, 2017. Privacy Protection Based Access Control Scheme in Cloud-Based Services, *IEEE Journal,* vol. 14, no. 1, pp. 61-71.
17. M. Nabeel, N. Shang, and E. Bertino, 2013. Privacy Preserving Policy based Content Sharing in Public Clouds, IEEE Trans. Knowledge and Data Eng., vol. 25, no. 11, pp. 2602-2614.
18. C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, 2012. Toward Secure and Dependable Storage Services in Cloud Computing, IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232.
19. Nirnay Ghosh, Debangshu Chatterjee, Soumya K Ghosh, and Sajal K Das, 2016. Securing Loosely-Coupled Collaboration in Cloud Environment through Dynamic Detection and Removal of Access Conflicts, IEEE Trans. *Cloud Computing*, vol. 4, no. 3, pp. 349-362.
20. S. Sundareswaran, A.C. Squicciarini, and D. Lin, 2012. Ensuring Distributed Accountability for Data Sharing in the Cloud, IEEE Trans. *Dependable and Secure Computing,* vol. 9, no. 4, pp. 556-568.

21. Jinbo Xiong, Ximeng Liu, Zhiqiang Yao, Jianfeng Ma, Qi Li, Kui Geng, and Patrick S. Chen, 2014. A Secure Data Self-Destructing Scheme in Cloud Computing, IEEE Trans. *on Cloud computing*, vol. 2, no. 4, pp. 448-458.
22. Mastering C++ 2nd Edition, Venugopal K R and Rajkumar Buyya, 2013. ISBN (13): 978-1-25902994-3, ISBN (10) : 1-25-902994-8, pp. 881, McGraw Hill Education.
23. Zhongma Zhu and Rui Jiang , January 2016. A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud, IEEE Trans. Parallel and Distributed Systems, vol. 27, no. 1, pp. 40-50.
24. Wei Song, Hua Zou, Haowen Liu and Jun Chen, 2016. A Practical Group Key Management Algorithm for Cloud Data Sharing with Dynamic Group, *IEEE Journal.* Security scheme and solutions, vol. 13, no.8, pp. 205-216.

How to cite this article:

Muthi Reddy P., Manjula S H and Venugopal K R (2018) ' Sdspg: Secured Data Sharing with Privacy as a Group in Cloud Computing ', *International Journal of Current Advanced Research,* 07(2), pp. 10036-10043.
DOI: http://dx.doi.org/10.24327/ijcar.2018.10043.1683

****