International Journal of Engineering & Technology, 7 (4.19) (2018) 485-492

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

SDBE-MOG: Secure Dynamic Bits Encryption for Multi Owner Group Data Sharing in Cloud

Sathishkumar Easwaramoorthy[1*], Dhivya Subburaman[2], Deepanjali.S[3]

[1]Independent Researcher, [2, 3]SRM University Chennai 603203, India
**Corresponding Author Email: [1]srisathishkumarve@gmail.com*

Abstract

Cloud computing is considered as the cost-efficient and practical resolution for group data sharing within the cloud group members. Unluckily, distributing data in a multi- owner way and defending data with unique privacy in cloud is a challenging issue because of the repeated change or deletion of the members. The proposed multi-owner group management system uses novel scheme GCAURDL-FBS(Group Creation, Add Member, Revocation, Deletion, Login with Forward & Backward Secrecy) to control overall group management. GCAURDL-FBS using SDBAES (Secure Dynamic Bits Advance Encryption Standard) secures the group data sharing. The major objectives of this research is 1) data security, 2) data integrity, 3) data sharing without using secret key re-computation / re-encryption, 4) forward and backward secrecy, 5) proper group owner / member revocation and 6) avoiding collusion attack, brute force attack and Structured Query Language Injection attack. Any member in group can confidently impart their information through entrusted server. Owner / Member revocation is achieved by a novel scheme; it is not necessary to compute/alter the secret keys of other members. The proposed scheme supports data integrity and efficient group member/owner revocation. Ultimately, it is found that, proposed scheme is also secure, scalable and efficient in relative to the relevant schemes. This is revealed in the security and tentative analysis

Keywords: *SDBAES, Group Signature, Data Integrity, Data Sharing, Forward & Backward Secrecy, Group Owner / Member Revocation.*

1. Introduction

Cloud computing is a latest and rapid growing technology which distributes various sources to its members dynamically through the internet [1].The cloud is a model for convenient demand network which succeeds a group of computing resources. The cloud storage provides a logical pool to store the digital data which is considered as the most eminent service. It is reliable, cost effective and easy to manage the data. Its associates can store and share the data with each other. A member can not only access and modify the data but can also share with others, so that the data can be accessed and modified by multiple members. In some cases cloud servers may return with hardware and software failure. So, the human attack maintenance and malicious, new forms of affirmative data honesty and openness are required for the security and protection of cloud member's information. Various techniques are proposed to check the correctness of the shared data which is independent of the integrity of the data. The existing work suggests techniques that can be used to crosscheck the single owner's shared data besides multi-owner data.

For giving veracity and availability to the storage in the cloud, some methods were recommended. In these schemes, it is called as dynamic scheme, when it supports data modification or else static one. If the given scheme is publicly verifiable then the uniqueness in the data can be checked also by the third parties, apart from the data owners. This paper proposes a methodology named Secure Dynamic Bits Advance Encryption Standard (SDBAES) that deals with the data protection within the shared group. The system adopts the following parameters: Members, a cryptographic server and the cloud [2].

To generate an Access Control List (ACL) to the Cloud System (CS), the data owner should submit the data. The CS is a reliable third party and it is noted for key management, encryption, decryption, and access control. The symmetric key will be generated by CS and the data is encrypted by the generated key. Since the single part of the key is not adequate for reproducing the key, the key is divided into two parts by CS for every member in the respective groups. The original key produced was eliminated by overwriting it in the protected manner. One part of the key is fixed within the ACL which is maintained by the CS of the data file and the other part is transmitted to the respective member in the group. The input parameters given by the user is responsible for generating the ACL. The produced secret data is stored in the cloud and the member sends a request to the CS for getting the information from the cloud,. The CS recognizes the request from the member by receiving the part of the key from the member and in this way downloads the data from the cloud. Working on the member segment of the key results in the production of the key and the particular CS is kept up for that specific member. The member will get the decoded information again. The two segments of the key are generated for recently joining member and it is added to the ACL [3]., The record is erased from the ACL for a member who is withdrawing. The withdrawing individual can't decrypt the information all alone as he/she only have a fragment of the key. So, no continuous decryption and encryption are required if there arises any changes in the grouping membership.

This paper discusses about the existing problems and constructing more integrity and security with both owner revocation and member revocation. This work is explained as follows,

1) This paper suggests an eminent method of defense and well-versed shared data integrity examination of multi-owner operation for cipher text database.
2) The paper analyzed the security and efficiency of this scheme. The results resolved that the scheme is best suitable for above mentioned parameters.
3) By implementing the primitives of GCAURDL-FBS; Group creation, Add Member, Revocation, Deletion, Login with Forward & Backward Secrecy are enabled which give rise to few new aspects such as traceability and countability [4].
4)

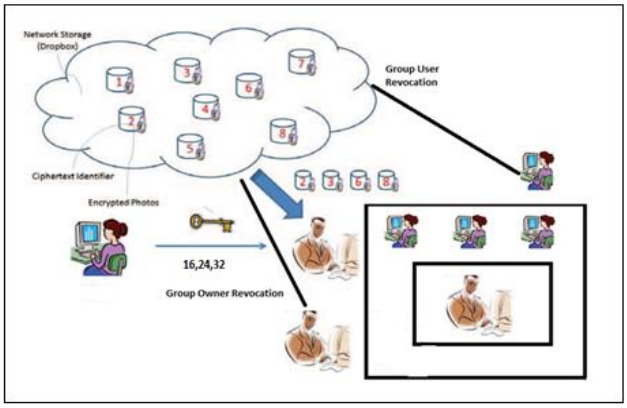

Fig. 1: Model for threat

The following gives the model of the research paper in an organized way. In part 2, it declares the properties of the existing problem formulation. Part 3 tells about the preliminaries used. Further, it presents the details of theproposed scheme in part 4. The security and efficiency analysis is explained in part 5 and 6. Results are concluded in section 7.

2. Problem Formulation

The model of cloud storage is given in figure 1.It has three entities, namely the cloud storage, server, group members. Each group member comprises the owner for the data. The owner is the one who gives authorization to the members and number of members who are sanctioned to use or delete the data. The data given by individual member of the group is shared in the cloud which was not a trusted storage. The integrity of the data shared in the cloud server is examined by an unit in the cloud called the auditor. In the proposed theory, the secure data integrity from both owner revocation and member revocation are implemented. The authorized member can encode the data and upload it to cloud storage and further no modification to be done by the owner. The alteration of the data and deletion was prevented in the proposed scheme. So, the unauthorized person will not modify the uploaded data [4].
This model considers the following types of attacks,

- Collusion Attack
- Brute Force Attack
- SQL Injection Attack

In cloud storage, cloud storage server is assumed as the semi-trusted platform, because a revoked member will collude with the cloud server. It will create the scheme insecure against t- types of attacks which pave way for getting the secret key of revoked members during the member revocation phase. The cloud will generate the data D that needed to be revoked, into an attack D'. In the member revocation process, the cloud can make D' become valid. To reduce the above problems, the following security methods were implemented [11].
1. Security- The new scheme is highly secure, because any anonymous person cannot modify or delete any data in the cloud.
2. Correctness- A scheme is correct if, the output verification by loud storage server is always the value D for any data D updated by a valid group member
3. Efficiency- This scheme is more efficient. The computation and storage overhead invested by any client for any data must be independent of the size of the shared data [11].

3. System Preliminaries

Bilinear groups are made use in this scheme. It depends on strong Diffie-Hellman and bilinear mapping. The system preliminaries contain Bilinear Mapping, Computational Diffie-Hellman (CDH) Assumption and Discrete Logarithmic (DL) Algorithm.

3.1. Bilinear Mapping

Consider G2 as an additive group and a multiplicative Prime Orders' (PO) cyclic group respectively. Let, $G_1 \to G_2$ mentions the bilinear map which is formed with theories stated below,
1) Bilinear- $a,b \in Z_q^* and M,N \in G_1, e(aM,bN) = e(M,N)^{ab}$ for all.
2) Non degenerate-It exists at a point P such that $e(g_1,g_2) \neq 1$
Computable-To compute $e(P,Q)\ for any P,Q \in G_1$

3.2. Computational Diffie-Hellman Assumption

1) The main security problem is Diffie-Hellman. For example, Let G1 and G2 be the cyclic group of prime order po, where G1 =G2. Here producer of G1 and G2 are g1 and g2 respectively.
2) The second security problem is the conclusion to the linear problem. Let, G1 and G2 are an array of prime order p. Given that $v^a to v^c \in G_1$ is the input, the output is yes, if and only if a+b=c.
3) The third main problem is Square Computational Diffie-Hellman. Let, $g \in G_1\ (g, g^y)$ for $y \in_R Z_p$ as input then the output is g^{y^2}.

3.3. GCAURDL-FBS

GCAURDL-FBS is a primary factor in cryptography which plays a prominent role in security standards like voting, identification, zero-knowledge proof etc. It should disclose the information about the committed message, and it needs the commitment mechanism which restricts the sender to modify the committed message. This is the main property of the vector commitment.
This paper proposes a more secure and integrity method namely GCAURDL-FBS that was implemented for Group Creation, Add Member, Revocation, Deletion, Login with Forward & Backward Secrecy. This method is collection of following parameters,
1. $GCAURDL - FBS(GC, AU, GMR, GOR, GD)$
2. $GCAURDL - FBS \to GC(1^k, p)$, where, GC is the Group Creation with parameter k and size p. The key generation outputs have some signature parameters pp.
3. $GCAURDL - FBS \to AU(d_1, ..., d_p)$ On input sequence of p with data's $d_1, d_2, ..d_m \in D$.
4. $GCAURDL - FBS \to open_{pp}(d_1, ..., d_p, j, aux)$
This is to produce a proof j with jth data D. Here, the auxiliary can include the information when some updates have occurred. After

the update, it will verify the given algorithm by running this command.

5. $GCAURDL - FBS \rightarrow Ver_{pp}(C, d_1, ..., d_p, j, \Delta_j)$

This algorithm will accepts only a valid data with the created sequence $d_1, d_2, ..d_m \in D$. It will again update the procedure once it is valid. Once the coding given below gets executed, it updates the data.

6. $GCAURDL - FBS \rightarrow Update_{pp}(C, d_1, d', j)$ .This will be considered as input for the old data d, the new data d'.

3.4. Group Signature and Member Revocation

There are two types of revocation that are implemented in this scheme.

- Group Owner Revocation
- Group Member Revocation

1) $GCAURDL - FBS \rightarrow KeyGen(n)$ This algorithm is taken as input n, number of group members. It takes the outputs as group login signature gls, and n-element of signature keys $gls[1], gls[2], ...gls[n]$ with n-element of group owner revocation (i.e.).

$$GCAURDL - FBS \rightarrow GOR(gls[1], gls[2], ...gls[n])$$

$$GCAURDL - FBS \rightarrow GMR(gls[1], gls[2], ...gls[n])$$

2) Similarly for group signature, it takes input as the group login signature GLS with data D. It returns the following command.

$$GCAURDL - FBS \rightarrow GroupSign(gls[1], gls[2], ...gls[n], D)$$

$$GCAURDL - FBS \rightarrow Verify(gls[1], gls[2], ...gls[n], RL, D)$$

3) The group login signature (GLS) is taken as input by the authentication process, a set of revocation (group Owner Revocation and group Member Revocation) RL with input data D.

4. Scheme Construction

In this section, two important things are explained. 1) Definitions of the proposed scheme 2) Procedure for the proposed scheme.

4.1. New Framework

Consider the database DB as a set of value (x, D_x), where x is an index and D_x is its respective value. Normally, a secure integrity auditing scheme with updates allows the resource-restricted client to outsource the storage of a very large database to a remote server.

The predicted framework is converged on the sharing of dynamic data with secure group member revocation, which is given as follows,

4.1.1. Setup $(1^k, DB)$

Let $DB = (J, D_j)$ be the database for $1 \le j \le p$ and this database is shared by a group which has n members with a single data owner.

1) The proprietor can produce a group by executing the command $GC \rightarrow pp \leftarrow GC(1^k, p)$ where, pp represents public parameters.

2) The parameters used to create the group are $GC \rightarrow (uID, OwnerName, eID, GroupName)$. It will use this command to create the group for login.

3) Here, login signature is used to login the group which is called as group login signature.

4) The group login signature contains the command $GLS \rightarrow (gID, ON, GN, uID, eID, GS, tS)$

Whereas, ON- Owner Name, GN-Group Name, GS-Group Sign, ts-tsamp

5) The owner will start to add the member into the group. The parameters used to add a member into the group is $AU(eID)$. It will automatically generate the group signature for the login.

6) The members were added by executing the commands $AUGLS \rightarrow (gID, ON, GN, uID, eID, GS, tS)$

Where, AUGLS is the Add Member Login Signature

7) Finally set the Group login signature parameter $AUGLS(pp, gls, C(t-1), C(t))$

4.1.2. Query ($GCAURDL - FBS \rightarrow (PP, DB, gls, j)$

To find the proof, a team member will run the algorithm $\Delta_j = GCAURDL - FBS.Open_{pp}(C_j, gls, j, tstamp)$,

Where, Δ_j is the j-th committed data and return proof γ. $\gamma = (c_o, \Delta_j, \sum(t))$

4.1.3. Verify ($GCAURDL - FBS \rightarrow (gls, RL, j, \gamma)$)

Take $\gamma = (c_j, \Delta_j, \sum(t))$, the given signature is considered as valid after running the algorithm if it verifies the below.

$$\{0,1\} \leftarrow GCAURDL - FBS.Verify(gls, RL, \sum(t)$$

Then, it will run the verification algorithm of,

$\{0,1\} \leftarrow GCAURDL - FBS.Ver_{pp}(C(t), \sigma^t, c_j, j, \Delta_j)$.

The algorithm gives the Δ_j when its output is 1.where $C_1, ..., C_q$ would produce C^t such that $C = C_j$

4.1.4. Update (j, γ)

1) A group verifies the created database to make it valid. It will obtain $\gamma \leftarrow Query(gls, RL, j, \gamma)$ and will check that $Verify(gls, j, \gamma, RL) = d_j$

2) It will update the information over the new data and updated output, which is as follows,

$$(C', U) \leftarrow GCAURDL - FBSUpdate(C, m, m', j)$$

4.1.5. Proof Update $(C, \Delta_j, c_j', j, U)$

1) Run the algorithm which is formulated and for the data at position J calculate the proof of update as follows $GCAURDL - FBSUpdate_{pp}(C, \Delta_j, m'_j, j, U)$, such that Δ_j contains the value of m'. The update information is $U = (m, m', j)$.

2) Verify the above C' with proof Δ_t .

4.1.6. Member Revocation (gls, j, γ)

1) Here the verification algorithm can be run by the auditor for both member revocation and member revocation and return either valid or invalid. The command to run the revocation is,

$\{0,1\} \leftarrow GCAURDL - FBS.Verify(gls, RL, \eta, D)$

2) Parameters used for the revocation of group owner and Group Member Revocation is

$GCAURDL - FBS \rightarrow GOR(gls[1], gls[2], ...gls[n])$

$GCAURDL - FBS \rightarrow GMR(gls[1], gls[2], ...gls[n])$

4.1.7. Group Deletion

1) The owner can delete the group in cloud storage. The parameters used to delete the group are GD. The command is given by,

$GD(Del(GC \Rightarrow gID, GroupName)$ - To delete the group

$GD(Del(GLS \Rightarrow gID, GroupName)$ -To delete the Group login Signature

$GD(Del(SKA \Rightarrow gID, GroupName)$ -To access the Secret Key

4.1.8. Group Login

The login procedure takes the below command to login.

$GL(GroupName, eID, groupSign, Grev)$

4.2. Data Encryption and Decryption

This paper proposes a **S**ecure **D**ynamic **B**its **A**dvance **E**ncryption **S**tandard (SDBAES) which was implemented to encrypt and decrypt the data. A schematic diagram of the proposed method is given below,

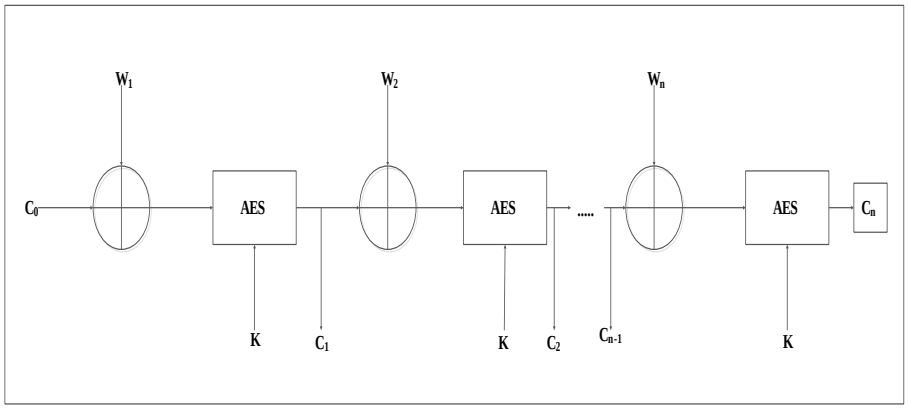

Fig. 2: Schematic diagram for SDBAES operation

The following crypto blocks in CBC mode occur in case of any change in a message block. This mode of CBC can be made use in MAC (Message Authentication Code) or authentication purpose which can be done in the following way. The first block can (e.g.) be formed simply with 0-bits [16]. The transmitter has a message that is formed of message blocks W1….Wn. applying a secret key and by using CBC mode K, it computes the corresponding crypto blocks $C_1 \ldots C_n$. The sender will transmit the message block and C to the receiver, who can validate the key c_n.

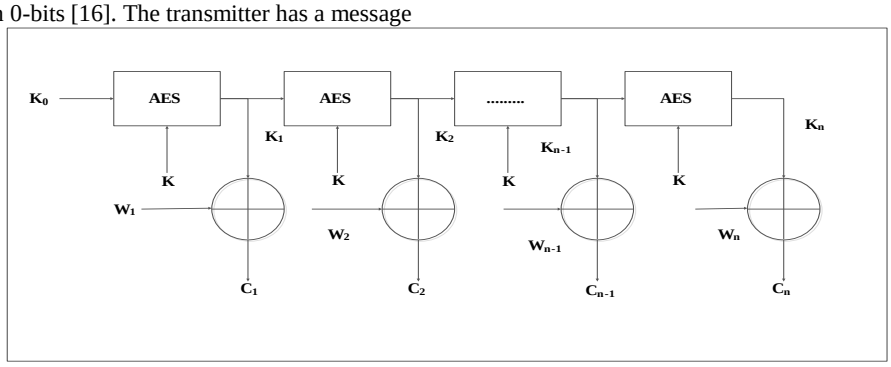

Fig. 3: Block diagram for AES operation on OFB mode

CFB mode (Cipher Feedback) is a variant, which is given rise by the OFB mode, where the key Ki of the key stream is made by encoding the successive crypto block. By encoding the initial block C0 [16] k1 is got again.

In AES the plain text block length I_B and the independent block length I_K may be either 128 or 192 or 256 bits. By Dividing with 32, the result will be,

$$N_B = \frac{I_B}{32} \quad \& \quad N_K = \frac{I_K}{32}$$

Bits are handled as bytes of bits. Finite field F_{2^8} is comprised of 8-bit components which has the residue representation

$$b_0 + b_1 y + b_2 y^2 + b_3 y^3 + b_4 y^4 + b_5 y^5 + b_6 y^6 + b_7 y^7 .$$

The key in bytes are usually given as $4 \times N_k$ matrix elements. When the key is in byte by bytes,

$$k = k_{00} k_{10} k_{20} k_{30} k_{01} k_{11} k_{21} \ldots k_{3,N_K - 1}$$

Hence the corresponding matrix is

$$K = \begin{pmatrix} k_{00} & k_{01} & k_{02} & \dots & k_{0,N_{K-1}} \\ k_{10} & k_{11} & k_{12} & \dots & k_{1,N_{K-1}} \\ k_{20} & k_{21} & k_{22} & \dots & k_{2,N_{K-1}} \\ k_{30} & k_{31} & k_{32} & \dots & k_{3,N_{K-1}} \end{pmatrix}$$

Similarly if the input block plaintext block is, byte by byte,

$$p = p_{00}\, p_{10}\, p_{20}\, p_{30}\, p_{01}\, p_{11}\, p_{21} \dots p_3, N_{B-1}$$

$$(bs_{00} \oplus bs_{10} z \oplus bs_{20} z^2 \oplus bs_{30} z^3, bs_{01} \oplus bs_{11} z \oplus bs_{21} z^2 \oplus bs_{31} z^3, \dots, bs_{0,N_{B-1}} \oplus bs_{0,N_{B-1}} z \oplus bs_{0,N_{B-1}} z^2 \oplus bs_{0,N_{B-1}} z^3)$$

For a unique representation, F_{2^8} must be constructed in such a way that it contains a given fixed complex polynomial of degree 8 from $Z_2[x]$ [16].

4.3. Overview of SDBAES

Hence the respective matrix is,

$$P = \begin{pmatrix} p_{00} & p_{01} & p_{02} & \cdots & p_{0,N_{B-1}} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1,N_{B-1}} \\ p_{20} & p_{21} & p_{22} & \cdots & p_{2,N_{B-1}} \\ p_{30} & p_{31} & p_{32} & \cdots & p_{3,N_{B-1}} \end{pmatrix}$$

During encryption, a bit sequence of length l_B will be dealt, the so-called state, which is also expressed as byte by byte like the block in the form of a $4 \times N_k$ matrix:

$$BS = \begin{pmatrix} bs_{00} & bs_{01} & bs_{02} & \dots & bs_{0,N_{B-1}} \\ bs_{10} & bs_{11} & bs_{12} & \dots & bs_{1,N_{B-1}} \\ bs_{20} & bs_{21} & bs_{22} & \dots & bs_{2,N_{B-1}} \\ bs_{30} & bs_{31} & bs_{32} & \dots & bs_{3,N_{B-1}} \end{pmatrix}$$

The elements of the field F_{2^8} can be defined as the constituents of the matrices K, P and BS, which are bytes of 8 bits. Consider the columns of the matrix as sequences of elements of the field of length, which is the other method to interpret the martrix. 4. It can be further interpreted by considering confidents polynomials with maximum degree 3 from the polynomial ring F_{2^8} [z] as the vertical elements. Thus the above mentioned state S corresponds to the polynomial sequence. In order to randomize the encrypted data the proposed scheme is constructed partially on the basis of randomized convergent encryption scheme. This makes the proposed scheme resistant to the chosen-plaintext attack .But Reduplication of the data is unavoidable [16]. Figure 4 depicts the outline of the suggested methodology and its security objectives.

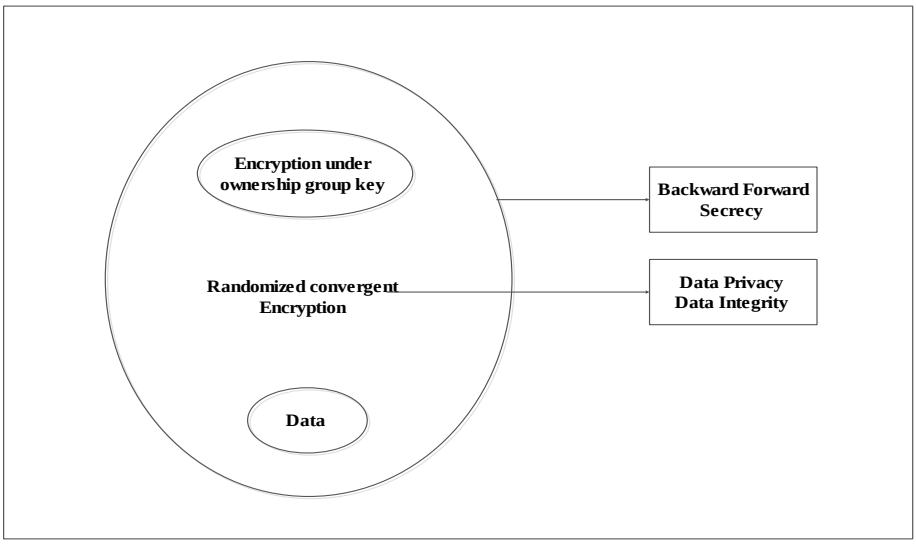

Fig. 4: Overview of proposed scheme

4.4. Algorithm steps for SDBAES

- AES have 3 bits level encryption & decryption such as 128 bits, 192 bits and 256 bits.
- File owner upload a file to cloud in background secure architecture, which will pick one level of bit level (128 or 192 or 256) then converted into byte.
- Based on byte value random generator it will generate, the public key of AES input Key. Then use the public key to encrypt the file and upload them to cloud.
- Same as generating the secret key of uploaded file they synchronize the secret key to every group members uniquely. The Secret key is effective among the active group members.
- File=>PubKey=>Encrypt=>PrivateKey=>PubKey=>Decrypt=>File

The above scheme contains the following algorithms,

1) $KEY \leftarrow KEYGen(U):$ A set of members U is taken as input by the KEY generation algorithm and renders output KEYs for every member in U. This enables secure group key distribution.

2) $C \leftarrow Encrypt(D,1^{\gamma})$: Data D and a security parameter γ is taken as input by the encrypted algorithm and it gives output which is the cipher text C of the data. C comprises encrypted data and its details.

3) $D \leftarrow Decrypt(C',K,PK)$: The elements of the decryption algorithm (a fixed algorithm) are input C', message encryption key K and a set of KEYs PK which encrypts an ownership group key GK. The output is a data D, if K calculated from D and GK is not retracted for the ownership, G inserts JNt into Gi, creates, and stores Ci in the storage, if ut is the first up loader for Di.

4.5. Data Decryption

It can decrypt the data by running $Decrypt(C_i^{'},K_i,PK_t)$, if $u_t \in G_i$.This is done in case of receiving a cipher C_i' by member u_t from cloud server. It contains group key ownership and data decryption.

Where,
CD –data or file size
Cc - encrypted data size
CK –key size
CT – Tag size
CJN – size of a member's identity
CT – node value
n- Number of members in the system
m- Number of owner.

5. Attacks Prevention

This secure architecture blocks the data modification from unauthorized person in the cloud storage and improves the data integrity with more security. It prevents the collusion attack, Brute Force Attack and SQL Injection Attack.

5.1. Collusion Attack

The changes often occurs in the membership of file groups therefore, repudiation in these schemes and the collusion attacks is directly incremented with the number of data proprietors and denied clients. To overcome this issue the proposed methodology introduced a GCAURDL-FBS algorithm for group management [12]. It will support for both group owner and group member revocation. It prevents the collusion attacks through group management process.

5.2. Brute Force Attack

In brute force attack, automated programs are utilized to create an extensive number of successive speculations with regards to the estimation of the information. It might be utilized by offenders to break the encoded information or by security experts to test the security [10]. Here, SDBAES calculation is utilized to encode and decode the information. The key for both encoding and decoding the information is having n-number of blends. So it will consume long time to break the data. Brute Force Attack failed to address this issue [9].

5.3. SQL Injection Attack

SQL Injection attacks occur when a invalidated data supposed by a foreign user are directly included in SQL query. So, this approach prevents these attacks through group management [13]. In that, the key used for information will be validated with session validation. So, the information will not be hacked by the hacker easily.

6. Result Analysis

The result for the proposed scheme is represented below. The graphs given below shows (fig: 5, 6, 7,8) the time required for uploading, downloading, encrypting and decrypting the file. The file size was taken in kb and time was considered in milliseconds. The time required for uploading time is higher than the downloading time.

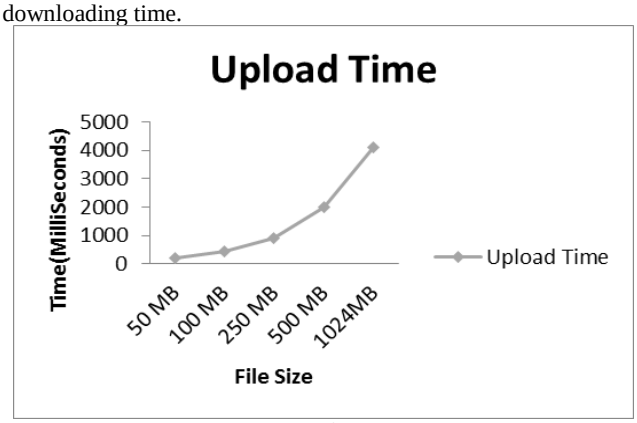

Fig. 5: Upload Time for Proposed scheme

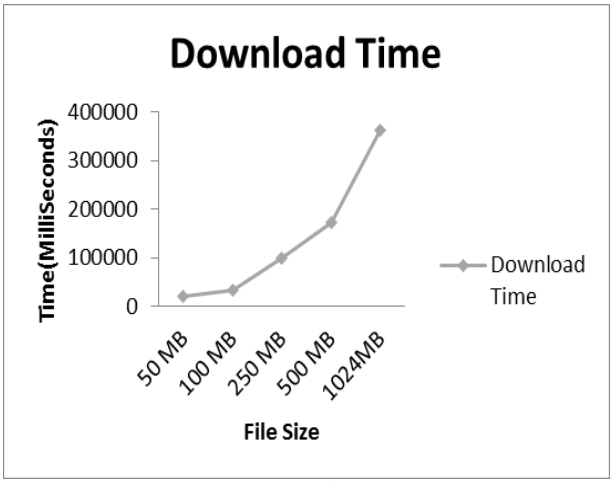

Fig. 6: Download Time for Proposed scheme

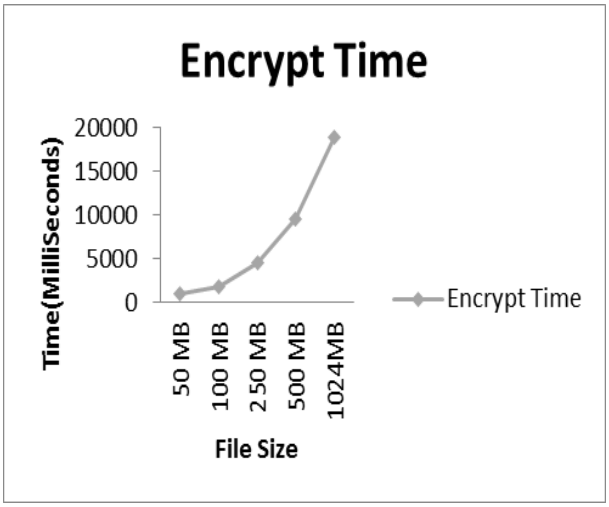

Fig. 7: Encrypt Time for Proposed scheme

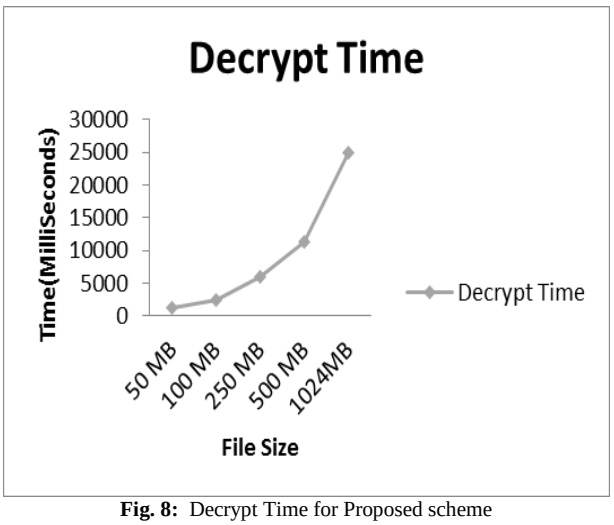

Fig. 8: Decrypt Time for Proposed scheme

7. Performance Evaluation

Here, the mathematical evaluation of this scheme is matched with the previous work and it is resistant to collision attack of the cloud storage server and the revoked members in the potent scheme. It is more effective than the existing scheme, this is because the computational dependence enclosed by the client does not depend on the size of the given data. The detailed performance evaluation is stated in table 1.

Table 1: Work Evaluation of the proposed and existing theory

theory	theory 1 [10]	Theory 2 [11]	Proposed Scheme
Query	-	$(q-1)(Mul+Exp)$	$s(Pair+Mul+EXP)$
Verify	$Exp+2Pair$	$7Pair+Mul+9Exp+5Hash$	$Mul+2EXP+16SRand$
Update	$(s+2)Exp+(s+1)Mul$	$2s(Mul+Exp)\ ors(Mul+Exp)^*$	$s(Mul/Pair+EXP)$
Member Revocation	$c(Pair+Exp)$	$z(Mul+2Pair)$	$U(MUL+5Pair/EXP)$
Group Owner Revocation	-	-	$O(MUL+5Pair/CAP)$

The comparison for upload and download time for the scheme is graphically represented in Fig 9, 10.

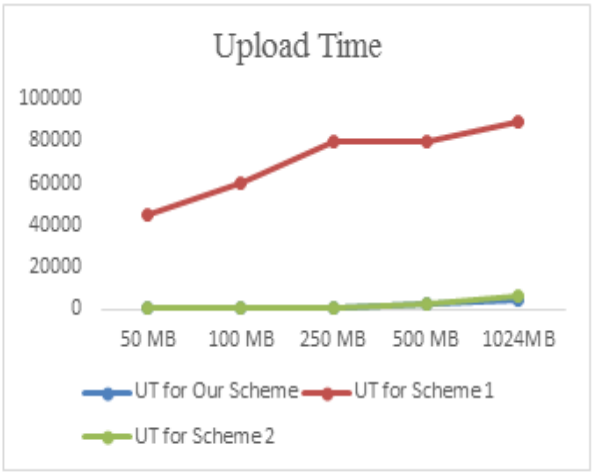

Fig. 9: Comparison results for Upload Time

Figure 9 shows that the time required for both existing and proposed schemes. It is observed that the size of 50Mb file was uploaded in 218ms and 245ms for proposed and existing schemes respectively which shows that the proposed method is more efficient than existing one.

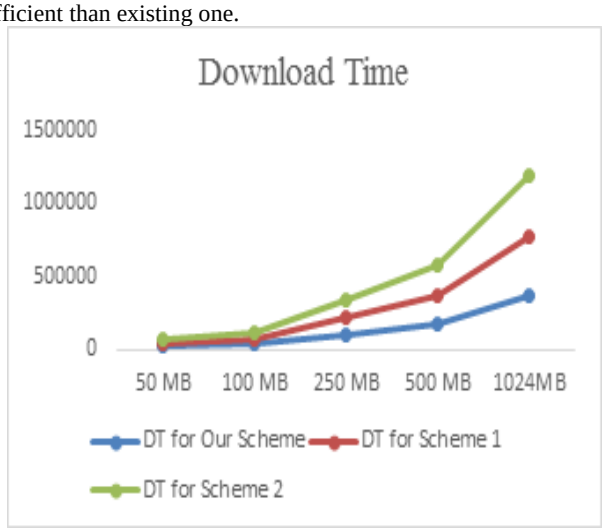

Fig. 10: Comparison results for Download Time

From above figure, it is observed that the time required to download the 50Mb file for proposed scheme is 20156ms, which shows that the proposed scheme is less than the existing scheme.

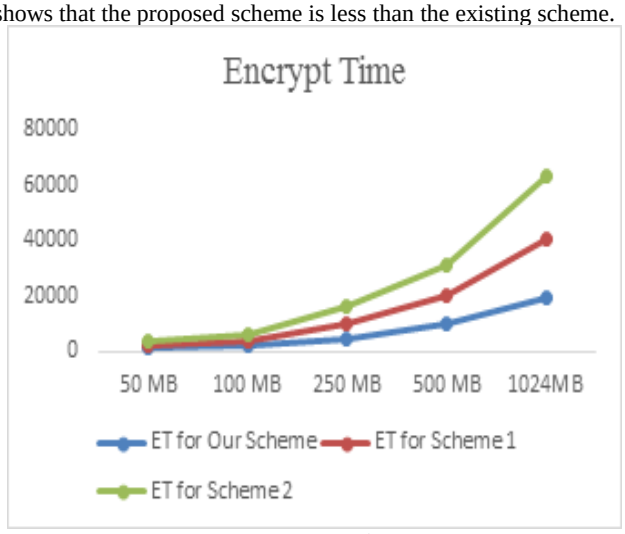

Fig. 11: Comparison results for Encrypt Time

The figure 11 denotes the encrypt time for both existing and proposed schemes. In that, the 50Mb file was encrypted in 998ms in the proposed scheme and 1120ms in existing scheme. It is reasoned that, the proposed scheme is safer than the current methods.

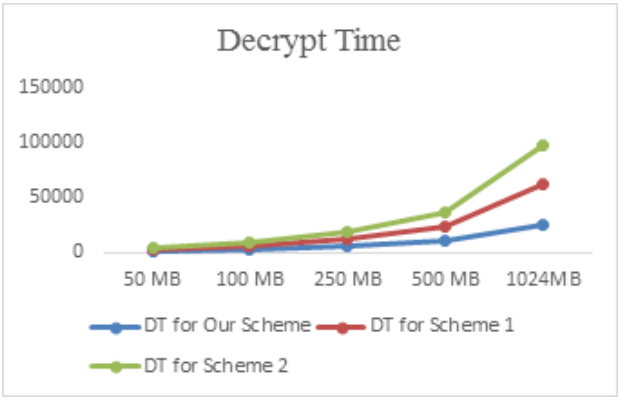

Fig. 12: Comparison results for Decrypt Time

Figure 12 shows the time required for both existing and proposed schemes. It is observed that the size of 50Mb file is decrypted in 1300ms and 1451ms for proposed and existing schemes respectively.

8. Conclusion

The GCAURDL-FBS & SDBAES methodology is a cloud storage security scheme for multi-owner group management and this is suggested in this paper. The suggested methodology offers data security & integrity, blocking access control for resentful insiders, keeps up secrecy in the entangling/disentangling. Data can be shared protectively without re-computation/ re-encryption. Moreover, the proposed methodology provides assured group owner / member revocation .The group deletion can be done by deleting the parameters demanded for grouping the owner credentials. The encryption and decryption functionalities are worked out in the cloud storage, a trusted SDBAES methodology. The working of GCAURDL-FBS & SDBAES formally control multi-owner group management using Group Creation, Group Owner Adding Members, Group Revocation - Owner / Member, Group Deletion ,Group Login, File Upload, File Encryption, File Download, File Decryption and so on. The execution of the SDBAES methodology is evaluated on the basis of time consumption during the key generation, file upload, file encryption, file download and file decryption operations. The conclusion unveils that the GCAURDL-FBS & SDBAES methodology can be practically applied in the cloud for sharing the data securely among the multi-owner group management.
The suggested methodology can be further extended by limiting the faith level in the cloud storage, which helps the system to survive with insider menace and avoiding data de-duplication in group data sharing. Moreover, evaluation of the response of the methodology with altering dynamic key sizes can be done.

References

[1] S. Akanksha, and Patil "A Secure Multiowner Dynamic Groups Data Sharing In Cloud", International Journal of Advances in Engineering & Technology, 9(1), 32. 2016.
[2] Ali. M. Dhamotharan, R. Khan, E. Khan, S. U. Vasilakos, A. V. Li, K, and A. Y. Zomaya, SeDaSC: secure data sharing clouds. IEEE Systems Journal, 2015.
[3] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," in Proc. of IEEE CLOUD 2012, Hawaii, USA, pp. 295–302, Jun. 2012.
[4] B. Wang, L. Baochun, and L. Hui, "Public auditing for shared data with efficient member revocation in the cloud," in Proc. Of IEEE INFOCOM 2013, Turin, Italy, pp. 2904–2912, Apr. 2013.
[5] K. S. Babu, and J. Mahalakshmi, "Group Member Revocation And Integrity Auditing Of Shared Data In Cloud Environment," IJITR, 4(4), 3237-3240, 2016.
[6] M. A. Chitra, , P. Prince, and V. V. Varthan, "Access Control based Dynamic Groups to overcome Collusion Attacks in Cloud",
[7] B. Cui, Z. Liu, and L. Wang, "Key-aggregate searchable encryption (KASE) for group data sharing via cloud storage", IEEE Transactions on computers, 65(8), 2374-2385, 2016.
[8] Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in Proc. of Asiacrypt 2001, Gold Coast, Australia, Dec. 2001, pp. 514–532.
[9] J. Hur, D. Koo, Y. Shin, and K. Kang," Secure data deduplication with dynamic ownership management in cloud storage", IEEE Transactions on Knowledge and Data Engineering, 28(11), 3113-3125, 2016.
[10] J. Yuan and S. Yu, "Efficient public integrity checking for cloud data sharing with multi-member modification," in Proc. of IEEE INFOCOM 2014, Toronto, Canada, Apr. 2014, pp. 2121–2129.
[11] Jiang, Tao, Xiaofeng Chen, and Jianfeng Ma. "Public integrity auditing for shared dynamic cloud data with group member revocation." IEEE Transactions on Computers 65.8 (2016): 2363-2373.
[12] D. S Kasunde, and A. A. Manjrekar, " Verification of multi-owner shared data with collusion resistant member revocation in cloud", In Computational Techniques in Information and Communication Technologies (ICCTICT), pp. 182-185, 2016.
[13] M. Kavya, and M. J Reddy, " Privacy Preserving Data Sharing in Multi Groups".
[14] K. Kowsalya, and V. Ramesh, "Multi Owner Data Sharing & Outsourced Revocation Using Identity Based Encryption on Cloud", International Journal of Advanced Networking and Applications, 7(5), 2899, 2016.
[15] N. Singh, A. Jangra,U. Lakhina, and R. Sharma, "SQL Injection Attack Detection & Prevention over Cloud Services" International Journal of Computer Science and Information Security, 14(4), 256, 2016
[16] W. Patterson, "Mathematical cryptology for computer scientists and mathematicians,", Rowman & Littlefield, 1987.