International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 1, 2025

A symmetric Encryption Based Secure Data Sharing in Cloud Environments

Rajkumar V[1], Sivaranjini R[2]
Department of Computer Science and Engineering, Krishnasamy College of Engineering and Technology
(Affiliated to Anna University, Chennai),Cuddalore, Tamilnadu, India, raj_win7@yahoo.com[1], ranjiniraja13@gmail.com[2]

Abstract – **The main advantage of cloud computing is the cloud storage which is helpful in storing data. But it deals with a major concern of security issues. It mainly occurs highly in group sharing of data which faces a two-side problem both cloud specific and conventional insider threats. To overcome this issue a secure data sharing in cloud methodology has been proposed which provides data confidentiality and integrity, access control, data forwarding, insider threat security and also it provides forward and backward access control. This method encrypts a file with a single key. Two different keys are generated one is sent to the user which encounters in finding and other is maintained by the trusted third party which is a Storage Server(SS). A working prototype has been implemented with this methodology and its performance is evaluated based on the time consuming of various operations. We also verify this secure data sharing cloud by using Asymmetric key generation. This method is potentially effective to provide security in group sharing data in cloud.**

Keywords –Asymmetric Key Generation, Cloud Security, Secure Data Sharing, Security, Storage Server.

I. Introduction

Cloud computing is an emerging technology to provide elastic provisioning, flexibility and on demand storage with several new computing to the users [7]. The main security concern is the loss of control over the data and computation raises many security issues. Privacy and confidentiality of the data is the one which are mainly recommended by the customers. The confidentiality of the customer is that customer data should not be stolen out. Cryptography act as a main tool for providing confidentiality and privacy to the data. The steps involved mainly for providing security by means of cryptographic tool is encryption, access control, key management and decryption process [1]. The existing system has a main disadvantage of when a newly added group member can access the data where they may act as an insider threats. To overcome this proposed methodology helps in providing security for the shared group data.

This methodology works with the following modules 1.Cloud 2. Data Provider 3. Storage server access 4. Asymmetric Key generation 5. File uploading & downloading.

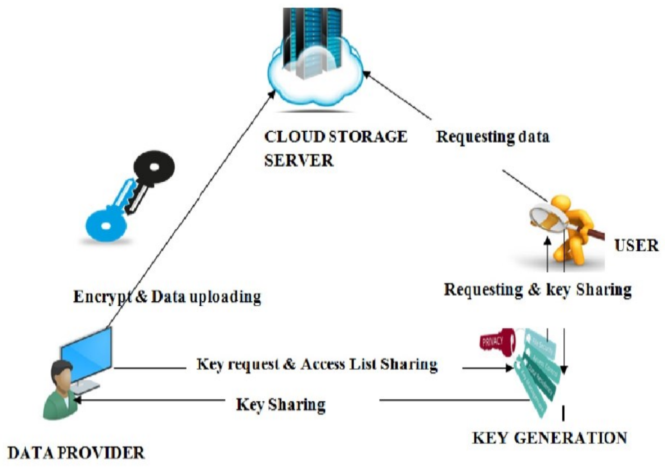

Fig: 1 Architecture of Asymmetric Encryption

ISSN 2347–3657

Volume 13, Issue 1, 2025

The advantage of the proposed system is to improve good quality of service(QOS) and secure data distributing among different providers and taking advantage of secret sharing where every user has their own master key.

II. RELATED WORK

Mazhar Ali et al.[2] considered the SeDaSC methodology, which is a cloud storage warranty scheme for accumulation data. this methodology provides encouraged deletion by deleting the parameters required to decrypt a file. The encryption and decryption functionalities are performed at the CS particularly a trusted third party in the SeDaSC methodology. The proposed methodology can be also employed to mobile cloud computing right to the fact that compute-intensive tasks are performed at the CS. The working of SeDaSC was formally analyzed by HLPNs, the SMT-Lib, and a Z3 solver [13]. The performance of the SeDaSC methodology was evaluated based on the anticipate consumption during the key generation.

Xu et al. [5] considered a certificate less proxy re-encryption (CL-PRE) step by step diagram for securely show and tell the broadcast within a group in the community cloud. In the CL-PRE scheme, the message owner encrypts the message mutually the symmetric key. Subsequently, the symmetric time signature is encrypted by all of the public key of the dataowner. Both the encrypted data and the key are uploaded to the cloud. The encrypted key is re-encrypted by the cloud (that acts as a proxy reencryption agent) that becomes decrypt able by the user's unknown key [3]. The public–private keys generated in the proposed scheme are not based on the certificates. The user's identity is used to prompt the public–private key pair [11]. The proxy re-encryption is based on bilinear pairing and the BDH that makes the CL-PRE scheme computationally intensive. The computational cost of the bilinear pairing is valuable as compared with the standard operations in finite fields.

Chen and Tzeng [4] considered a methodology based on the shared key derivation approach for securing message sharing in a group. The methodology uses a dual tree for the computation of keys. However, the computational charge of the proposed scheme is high as the rekeying equipment is strongly employed in the considered scheme [16]. Moreover, the scheme is not tailored for public cloud systems because actual operations demand centralizedmediations. A similar Rivest–Shamir–Adleman (RSA) based concern was also proposed in [6]. However, the scheme was poor against collusion attacks.

III. METHODOLOGY

In this section our proposed methodology has been explained neatly with the working procedure for the secure data sharing in cloud and system design has been implemented.

A. *Cloud:*

The cloud provides computerized information services to the user. The data on the cloud wish to be secured at variance with privacy breaches. The confidentiality of the data is ensured by storing encrypted data from one end to the other the cloud. The cloud in the proposed system only involves basic cloud operations of file upload and download. Therefore, no changes at the code of behavior or implementation level on the cloud are required.

B. *Data Provider*

Data Provider is having valid rights and complete control from one end to the other a single bit or set of data elements. It defines and provides information practically the rightful owner of data assets and the acquisition, handle and distribution policy implemented by the data owner.

C. *Storage Server*

The SS is a trusted party and is responsible for security operations, a well-known as key management, encryption, decryption, the management of the ACL for providing confidentiality, and secure data forwarding in the group [14]. The users of this proposed systemare required forthcoming registered with the SS to take in the security services. TheSS is assumed to be a secure entity in the proposed methodology. The SS can be maintained by an organization or can be owned by a third-party provider. However, the SSmaintained by an organization will generate more trust in the system.

D. *Users*

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 1, 2025

The clients are the users of the storage cloud. For each data file, such user will be the manager of the file, whereas the others in the group will be the data consumers. The owner of the file decides the secure rights of the other group members. The access rights are given and revoked based on the decision of the owner. The access rights are managed every SS in the form of an ACL file. A separate ACL is maintained all ofthe data files

E. *Key Authority*

Key Authority is a set of roles, policies, and procedures needed to construct, manage, distribute, use, store, and share the key to client.KA schemes construct on cryptographic time signature, which provides asymmetry during the perfect parties and the attacker [10]. This module describes time signature distribution approaches and focuses on the approach based on public-key certificates authority and public-key the common people that can verify large number of parties.

IV. SYSTEM DESIGN

A. *File Upload:*

Whenever a need to share data among the group arises, the owner first requests the key to Key Authority (KA). The request is accompanied by the Date (D) and a list (L) of users that are to be granted access to the file. L also contains the access rights for each of the users. The users may have READ-only and/or READ–WRITE access to the file [12]. Other parameters can be also set to enforce fine-grained access control over the data.L is used to generate the ACL for the data by the KA. L is store in KA only if the data are to be shared with a new proposed group. If the group already exists, the encryption request will not contain L; rather, the group ID of the existing group will be sent The SS, after receiving the encryption request for the file, generates the ACL from the list and creates a group of the users [9]. The ACL is separately maintained for each file. The ACL contains information regarding the file such as its unique ID, size, owner ID, the list of the user Ids with whom the file is being shared, and other metadata. When a group already exists only the ACL for the

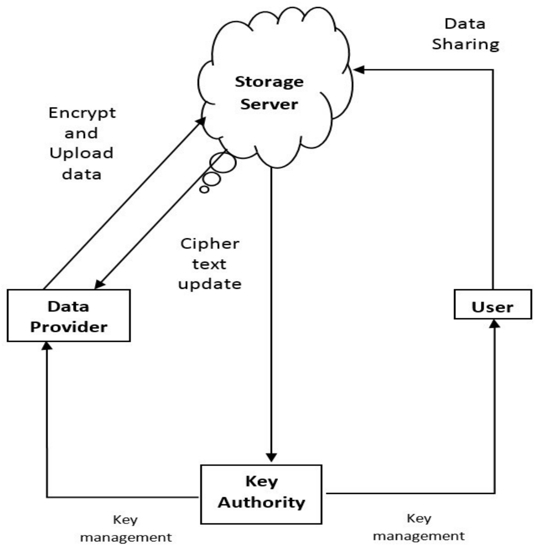

Fig: 1 Basic idea for secure data sharing using A_{ska}

file is created. Next, the KA generates K according to the procedure defined and encrypts the file with an appropriate asymmetric block cipher (we have used the ECC for encryption purposes). The result is an encrypted file (CT). Subsequently, the KA generates Ki and K i for every user and deletes K by secure overwriting. Secure overwriting is a concept in which the bits in the memory are constantly flipped to make sure that a memory cell never grips a charge for enough duration for it to be remembered and recovered. The Ki for each user is inserted into the ACL for later use[8]. The KAalso computes the message authentication code (MAC) signature on every encrypted file. A similar procedure for the MAC key is adopted. However, the MAC key is kept by the KA only. The encrypted data, the group ID (in the case of a newly generated group), and the K i for the owner are sent to the requesting data owner The group ID and the Ki for the rest of the group users are directly store in secure storage.

ISSN 2347–3657

Volume 13, Issue 1, 2025

B. *File Download:*

The authorized user sends a download request to the SS or downloads the encrypted file (CT) from the cloud and sends the decryption request to the KA. The cloud verifies the authorization of the user through a locally maintained ACL. The decryption request is accompanied by the user portion of the key, i.e., Ki, along with other authentication credentials. The KA computes K by applying XOR operation over Ki and the corresponding Ki from the ACL. As each ofthe users correspond to a different pair of Ki and Ki, none of the users can use other users' K i to masquerade identity. Subsequently, the KA proceeds with the decryption processafter verifying the integrity of the file [12]. If the correct Ki is received by the KA, the result will be a successful decryption process;

V. ALGORITHM IMPLEMENTATION

Secret Key Computation: This proposed system maintains a base hit cryptographic key individually of the data files. However, after encryption/decryption, the entire key is not stored and possessed by any of the involved parties. The key is partitioned into two constituent parts and are possessed by disparate entities. The consequently are the keys that are used within system.

Asymmetric Key K: K is a random secret generated by the SS for each of the data files. The length of K inthis proposed system is 256 bits, as it is recommended by most of the standards regarding key length for Asymmetric key algorithms (A_{ska}). However, the length of the key can be altered according to the requirements of the underlying A_{ska}. K is obtained in a two-step process. In the first step, a random number R of length 256 bits is generated such that R = {0, 1}256. In the next step, R is passedthrough a hash function that could be any hash function with a 256-bit output. In our case, we used secure hash algorithm 256 (SHA-256). The second step completely randomizes the initial user-derived random number R. The output of the hash function is termed as K and is used in symmetric key encryption [e.g., the Elliptic curve cryptography (ECC)] for securing the data.

KA Key Share Ki: For each of the users in the group, the KA generates Ki, such that Ki = $\{0, 1\}^{256}$. Ki serves as the KA portion of the key and is used to compute K whenever an encryption/decryption request is received by the KA. Moreover, it is ensured by comparison that the distinct Ki is generated for every file user.

A. *Key Generation and Encryption Algorithm*

```
Input:
Data,
Access Control List
The $A_{SKA}$,
The 256-Bit
Hash Function $H_F$

Compute:
R$num$=$\{0,1\}^{256}$
        key=$H_F$(R$num$)
        CT=$A_{SKA}$(Data, key)
for each user$i$ in the Access Control List, do
key$_i$=$\{0,1\}^{256}$
        key$_i$ = key   XOR  key$_i$
Add $K^1_i$ for user i  in the Access Control List
send $K^1_i$ for user i
end for
delete(Key)
delete($K^1_i$)
return CTto the data owner or upload to the cloud.
```

ISSN 2347–3657

Volume 13, Issue 1, 2025

B. *Decryption Algorithm Algorithm*

```
Input:
CT
Access Control List
The $A_{SKA}$

Compute:
Get $K^1_i$ from the requesting user
Generate secret key
Get CT from the requesting user or download from the cloud

Retrieve $K_i$ from the Access Control List
        if $K_i$ does not exist in the Access Control List, then

return the access denied message to the user

else
        key=key XORkey$_i$
        Data=$A_{ska}$(CT,key)
        Send Data to the user
end if
        delete(K)
  delete($K_i^1$)
```

VI. Performance Evaluation

Performance Evaluation in cloud computing is a challenging task that evaluates the systems. In cloud computing we need a methodology that regards the specific configuration of possibilities and measurements.

Our methodology is implemented in Visual Studio2010 C# using the .Net 4 framework. The proposed methodology consists of modules, i.e,cloud, data provider, storage server access, asymmetric key generation and file uploading and downloading the Amazon s3 is used as a cloud server in our approach. to communicate with the server Amazon software development kit is used with .net application programming. The Storage Server is used as a trusted third party [15]. the communication between the modules were established with the help of .net libraries.

Result:

The proposed methodology has been evaluated with the following three different ways

A. *key generation:*

Asymmetric key is generated for each file. key sharing is done separately as it is computes for each user in the group. Weevaluated for time consumption in key generation.The time is computed for different numbers of users. We set the number of users to be 20, 40, 60,80, and 100.the results are shown in the fig(2).

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 1, 2025

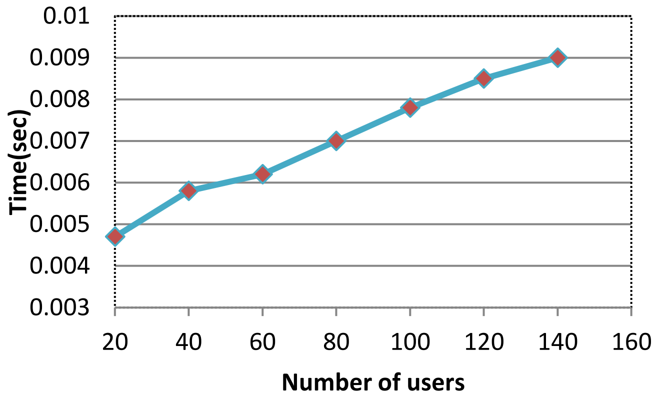

Fig 2: Time consumption for key generation

B. Encryption and Decryption:

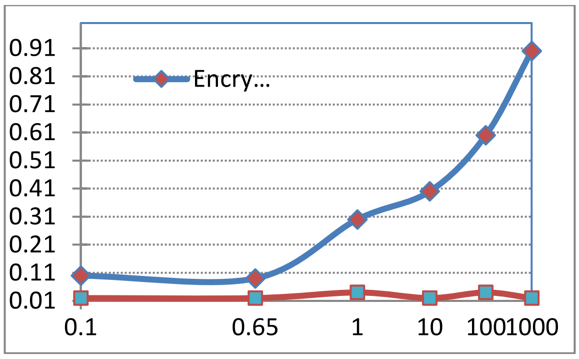

Fig 3: Performance of file encryption

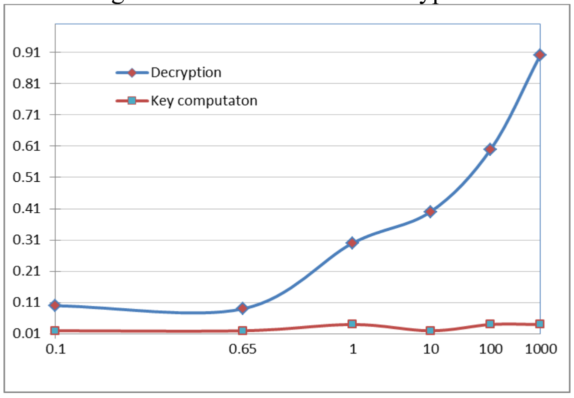

Fig 4:Performance of file decryption

The time consumption is mainly evaluated here in the encryption and decryption of the files with different sizes The file sizes used were 0.1, 0.5, 1, 10, 50, the main purpose is to monitor the time overhead of key generation during each encryption and decryption process.

VII. CONCLUSION

A Secure data sharing in cloud methodology provides security to the group shared data. This method provides high confidentiality, access control, and secure data sharing. It also provides high efficiency in avoiding the insider threats and also malicious attacks.

Moreover this methodology provides assured encryption and decryption. The functionalities of encryption and decryption are done at Storage server which is a trusted third party. this methodology can also be used in mobile cloud computing in which the computed intensive tasks are done by the Storage Server. The proposed performance methodology has been evaluated with the time consumption accrued during key generation, uploading and downloading operations. this methodology can be practically implemented in the cloud for secure data sharing among the group

REFERENCES

[1] V. Rajkumar, M. Prakash, and V. Vennila "Secure Data Sharing with Confidentiality, Integrity and Access Control in Cloud Environment," Comput. Syst. Sci. Eng., vol. 40, no. 2, pp. 779-793. 2022

[2] Mazhar Ali, Revathi Dhamotharan, Eraj Khan, Samee U Khan, Athanasios V Vasilakos, Keqin Li, & Albert Y Zomaya, “SeDaSC: Secure Data Sharing in Clouds”, IEEE systems journal, vol. 11, no. 2, 2017.

[3] A N Khan, M L M Kiah, S U Khan, & S A Madani, “Towardssecure mobile cloud computing: A survey”, Future Generation Computer Systems, vol. 29, no. 5, pp. 1278-1299, 2013.

[4] A N Khan, M M Kiah, S A Madani, M Ali, & S Shamshir-band,“Incremental proxy re-encryption scheme for mobile cloud computingenvironment”, Journal of Supercomputing, vol. 68, no. 2, pp. 624-651,2014.

[5] Y Chen & W Tzeng, “Efficient and provably-secure group key management scheme using key derivation”, in Proceedings of the IEEE 11th International Conference on TrustComputing, pp. 295-302,2012.

[6] L Xu, X Wu, & X Zhang, “CL-PRE: A certificateless proxy reencryptionscheme for secure data sharing with public cloud”, in Proceedings of 7th ACM Symposium. Inf., Comput. Commun. Security, pp. 87–88,2012.

[7] Y Chen, J D Tygar, & W Tzeng, “Secure group key management usinguni-directional proxy re-encryption schemes”, in Proc. IEEE INFOCOM, pp. 1952-1960.

[8] Cloud security Alliance, “Security guidelines for critical areas of focus in cloud computing v3.0”, 2011.

[9] P Gutmann, “Secure deletion of data from magnetic and solid-state memory”, in Proc. 6th USENIX Security Symp. Focusing Appl. Cryptography,p.8,1996.

[10] I Indu, P M. Rubesh Anand & Shaicy P Shaji,“Secure File Sharing Mechanism and Key Management for Mobile Cloud Computing Environment”, Indian Journal of Science and Technology, vol. 9, no. 48,2016.

[11] H Li, Y Dai, L Tian, & H. Yang, “Identity-based authentication for cloud computing”, in Proc. CloudCom, pp. 157-166,2009.

[12] H Sun, Q Wen, H Zhang, & Z Jin, “A novel remote user authentication and key agreement scheme for mobile client–server environment”, Appl.Math. Inf. Sci., vol. 7, no. 4, pp. 1365-1374, 2013.

[13] N Fernando, S W Loke, & W Rahayu, “Mobile cloud computing: A survey”, Future Gen. Comput. Sys., vol. 29, no. 1, pp. 84-106, 2013.

[14] A Mishra, R Jain, &A Durresi,“Cloud computing: Networking and communication challenges”, IEEE Communications Magazine, vol. 50, no. 9, pp.24–29,2012.

[15] M Prakash & G Singaravel, “Haphazard, Enhanced Haphazard and Personalised Anonymisation for Privacy Preserving Data Mining on Sensitive Data Sources”, International Journal of Business Intelligence and Data Mining, vol. 13, no. 4, pp. 456-474, 2018.

[16] J L Tsai , Lo NW. “A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services”, IEEESystem Journal,vol. 9, no. 3, pp. 805–820,2015.