ISSN: 2394-7659
IMPACT FACTOR- 2.785

International Journal of Engineering Researches and Management Studies

SECURING DATA FROM COLLUSION ATTACK USING SPLITTING AND COMPRESSION SCHEMES

P.ANUSUYA[*1] and A.SENTHIL KUMAR[2]

[*1]Research Scholar, Dept.of.Computer Science, Tamil University, Thanjavur-613010

[2]Asst.professor, Dept.of.Computer science, Tamil University, Thanjavur-613010

ABSTRACT

In case of sharing data in the cloud we need to secure data because of thread attacks. In this research paper we the authors propose the novel techniques such as splitting and compression for sharing data. The objective of image compression is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form. The main scope of this research is to encrypt image with AES algorithm and compress the image with deflate or Gzip stream algorithm and send to the receiver side and they decompressed image efficiently reconstructed with auxiliary information. At the time of reconstruction process the original content should not be modified. We can also balance the bandwidth range in sender channel provider and receiver side messages.

Keywods:-*Encryption, compression, Splitting, Secret key, Decryption.*

I. INTRODUCTION

Benefited from cloud computing, users can achieve an effective and economical approach for data sharing among group members in the cloud with the characters of low maintenance and little management cost. Meanwhile, we must provide security guarantees for the sharing data files since they are outsourced[3] . Unfortunately, because of the frequent change of the membership, sharing data while providing privacy-preserving is still a challenging issue, especially for an untrusted cloud due to the collusion attack.

Cloud computing, with the characteristics of intrinsic data sharing and low maintenance, provides a better utilization of resources. In cloud computing, cloud service providers offer an abstraction of infinite storage space for clients to host data. It can help clients reduce their financial overhead of data managements by mi2grating the local managements system into cloud servers [1].To preserve data privacy, a common approach is to encrypt data files before the clients upload the encrypted data into the cloud. A cryptographic storage system that enables secure
Data sharing on untrustworthy servers based on the techniques that dividing files into file groups and encrypting each file group with a file-block key.

Drawbacks of Existing System

- Security concerns become the main constraint as we now outsource the storage of data, which is possibly sensitive, to cloud providers
- It is difficult to design a secure and efficient data sharing scheme, especially for dynamic groups in the cloud [2].
- The file-block keys need to be updated and distributed for a user revocation; therefore, the system had a heavy key distribution overhead.

Proposed System

In this paper, we propose a secure data sharing scheme, which can achieve secure key distribution and data sharing for dynamic group. The main contributions of our
Scheme includes:

- We provide a secure way for key distribution without any secure communication channels. The users can securely obtain their private keys from group manager without any Certificate Authorities due to the verification for the public key of the user.

ISSN: 2394-7659
IMPACT FACTOR- 2.785

International Journal of Engineering Researches and Management Studies

- Our scheme can achieve fine-grained access control, with the help of the group user list, any user in the group can use the source in the cloud and revoked users cannot access the cloud again after they are revoked [5].
- We propose a secure data sharing scheme which can be protected from collusion attack. The revoked users can not be able to get the original data files once they are revoked even if they conspire with the untrusted cloud. Our scheme can achieve secure user revocation with the help of polynomial function [4].
- Our scheme is able to support dynamic groups efficiently, when a new user joins in the group or a user is revoked from the group, the private keys of the other users do not need to be recomputed and updated.

We provide security analysis to prove the security of our scheme. In addition, we also perform simulations to demonstrate the efficiency of our scheme.

II. SYSTEM DESIGN

The users or nodes involved in this paper are Sender, Intermediate and Receiver. In order to send file, the sender has to find out the list of nodes which are connected with the sender. From that available list she/he can choose receiver. Then the sender has to analyze the performance of each and every node which is connected with the sender. The performance analysis list will return the priority based result so that sender can choose the intermediate to send the file. The Intermediate will receive the file from sender then it will analyze the performance so that it can send data to another intermediate or receiver. In the receiver side, the receiver has to select the file path to receive the file from sender or intermediate. Then the receiver can view the file, from the sender where the images are viewed securely.

III. PROCEDURE

Step 1
Sender
Authentication:
Input: Provide username and password to get permission for access.
Output: Became authenticated person to request and process the request.
Select files:
Input: Sender will select the image file to the Receiver.
Output: Receiver will receive that file.
Encrypt file:
Input: Sender will send the image file to the Receiver.
Output: The selected file has been encrypted automatically.
Compress a file:
Input: Sender will send the image file to the Receiver.
Output: The encrypted file is compressed.
Step 2
RECEIVER
Authentication:
Input: Provide username and password to get permission for access.
Output: Became authenticated person to request and process the request.
Retrieve file:
Input: Receiver receives the transmitted file.
Output: Receiver will get a file in the encrypted format.
Decompress a file:
Input: Receiver receives the Sender transmitted file.
Output: The encrypted file is decompressed.
Decrypt file:

ISSN: 2394-7659
IMPACT FACTOR- 2.785

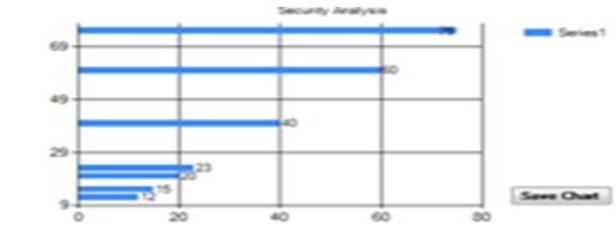
IJERMS

International Journal of Engineering Researches and Management Studies

Input: Receiver receives the Sender transmitted file.
Output: The selected file has been decrypted automatically.

IV. PERFORMANCE ANALYSIS

In this analysis, x- axis depicts the transmission capability of the data (image) size and y-axis depicts the data transmission security while the data transmission done we can analysis the performance of data transmitted from sender to receiver .we achieved the 75% of accuracy in data security through this analysis .



V. CONCLUSION

Thus we propose a novel technique to detect collusion attack and provide solution through implementing image splitting and compression techniques. It suggests a unique solution by analysing the image and it supports information security concepts and enriching by Advanced Encryption Standards (AES) through image compression and cryptographic processes. It uses two popular compression algorithm Deflate and Gzip stream used for compression. The methodology used discrete cosine transform technique for image analysis through that analysis we conclude 75% of accuracy, confidentiality of data can be achieved to mitigate from collusion attacks.

VI. FUTURE ENHANCEMENT

In the future, the encryption methods and algorithm for image compression can be enhanced by analyzing better image ratio-distortion performance with updated tools that hits markets, recent days.

REFERENCE

[1] G.Mercy vimala, R.Vara Prasad, P.Rama Rao ,"A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud.

[2] Amrita Sengupta, Sanjeev Ghosh "Compression of Encrypted Images using Chaos Theory and SPIHT" (2011) http://research.ijcaonline.org/icgct/number1/icgct1303.pdf.

[3] "Computer Networks",Fourth Edition , Andrew S.Tanenbaum.

[4] DhirenR.Patel, "information security", http://www.phindia.com

[5] Akash Raj ". High Resolution Image Encryption & Reconstruction Using Scalable Codes" (2010) http://www.ijera.com/papers/Vol3_issue2/BT32444450.pdf