Enhancing Security of Cloud based File Sharing Systems using AES and Proxy-Transformation

Abhishek Iche
Computer Engineering, Modern Education Society's College of Engineering, Pune, India

Abhishek Mhamane
Computer Engineering, Modern Education Society's College of Engineering, Pune, India

Ayan Shaikh
Computer Engineering, Modern Education Society's College of Engineering, Pune, India

Mahesh Kadam
Computer Engineering,
Modern Education Society's College of Engineering,
Pune, India

ABSTRACT

Cloud computing is one of the newest technologies in the era Cloud computing has become an extensively exploited research area in academic and industry. The security challenges related to cloud computing have been extensively studied. This review is aimed to highlight the existing research studies on cloud computing security, threats, and challenges and also proposed the security essentials for web-based applications. Cloud services are facing problems in providing security regarding data and file storage as well as sharing the files through cloud, securing the cloud from the threat actors causing external threats we are trying to address these issues to make reliable and secure file sharing and storage on cloud. We have developed file sharing system, security powered with AES encryption algorithm along with the Proxy Re-encryption (Proxy Transformation) to enhance the security on the cloud platform as well as while transferring the data.

Keywords

Cryptography, Cloud Security, File Sharing, AES, Proxy-Transformation

1. INTRODUCTION

File sharing has been a means of distributing or providing access to digitally saved data, this data can be computer programs, multimedia system files like audio, pictures, video, documents, or digital books. Cloud Computing" is outlined as delivery of computing services over the Internet (The Cloud) to supply quicker innovation and increase productivity.

The essential challenges round-faced with the standard method of file sharing and storage techniques such as manual filing or sharing of images, documents etc. is that users have very little management over or even knowledge of the kind of file being shared, if it contains a virus. Also, users don't or generally don't bear in mind the names of those files, the files contents, and their storage locations, this will increase the chance of loss. And also, the main concern regarding the protection of the files held on the cloud.

File Sharing is a sharing of computer data or files publicly or privately with various levels of access privilege. File sharing allows the use of a file shared by the user. File sharing can also mean having an allocated amount of personal file storage in a common file system.

Cloud computing is the delivery of different services through the internet. Rather than keeping files on hard drive or local storage device, cloud storage makes it possible to save them on a remote database or on a cloud. Cloud computing is the popular option for people and businesses to save their time, to save their cost of storing files on local computers.

Encryption is the process of encoding useful data using various encryption algorithms called as encryption algorithms. Whereas decryption takes encrypted data, decode that data and outputs original content on which encryption algorithm applied.

This research focuses on a cloud primarily based file sharing system, by inculcating the potential edges and removing Security problems related to sharing of files simply for straightforward retrieval.

The main aim of this research, is to highlight concept of Encryption and Proxy Transformation

1.1 Motivation

Information sharing is changing into more and more vital for finished users, enterprises and even health industries. There is currently a push for IT organizations to increase their information-sharing efforts. per a survey by data Week, nearly all organizations shared their data in some way, with 74% sharing their data with customers and 64% sharing it with suppliers due to this cloud computing is developing terribly quickly for it offers high computing power and capacity. The facility of cloud computing permits dynamic ability of utilizations and effort in a completely different methodology of file transfer in our advanced technology age. However, challenges arise when considering the massive variety of files to be shared. This project focuses on coming up with and implementing a Cloud Based Secured File Sharing System, the potential edges and technical problems associated with sharing of files simply for straightforward retrieval are all taken into consideration.

2. RELATED WORK

Scheme that provides the facility of secure storing and sharing the data for dynamic groups in the cloud and also a user is able to share data with others in the group without revealing their identity to the cloud. Reference [1] proposed scheme consists of design goals that includes access control, data confidentiality, anonymity, traceability and efficiency. They have also proposed functionalities for group owners for smooth and efficient management.

SeGShare[2], a new architecture for end-to-end encrypted, group-based file sharing using trusted execution environments (TEE), e.g., Intel SGX. Intel Software Guard Extensions (SGX): is a set of security-related instruction codes that are incorporated into some modern Intel (CPUs). SeGShare protects the confidentiality and integrity of content files, the file system structure, permissions, existing groups, group memberships.

Attribute based encryption is one of the most attractive ways to manage and control file sharing in the cloud with its special attribute computing properties. Attribute-based[3] encryption is a type of public-key encryption in which the secret key of a user and the cipher text are dependent upon attributes. In such a system, the decryption of a cipher text is achievable only if the set of attributes of the user key matches the exact attributes of the cipher text.

A proxy re-encryption scheme is introduced and combined with a distributed erasure-code. Such a secure and powerful data storage and retrieval, but also lets a user to share his data on the cloud with another user in the encrypted format itself. Reference [4] focuses on CLOUD STORAGE USING AES AND PROXY RE- ENCRYPTION and uses the techniques of both encryption and sharing the data.

Proposed approach in [5] can evaluate the risk associated with specified threats only using the STRIDE thread model, which provides a more fine-grained evaluation of the threats in the cloud. Furthermore, this approach can also be applied in other networked-based computing environments.

Data tampering and thefts were amongst the highly discussed topics in this. Other associated security flaws were attached with the data intrusion and data storage in the cloud computing system. In [6] authors have demonstrated security threats and risk mitigation strategies, commercial cloud computing providers and security challenges, role of blockchain technology in the security of cloud data.

Revocable-storage identity-based encryption (RS-IBE) [7], which can provide ciphertext forward/reverse security by introducing user revocation and ciphertext update functions at the same time. In Reference [7] presented a concrete construction of RS-IBE, and proved its security in the defined security model.

From the perspective of protecting cloud data confidentiality, Reference [8] proposed a Cloud Secure Storage Mechanism named CSSM. To avoid data breach at the storage layer, CSSM [8] integrated data dispersion and distributed storage to realize encrypted, chucked and distributed storage.

Reference [9] proposed the concept of cloud data storage security strategy capable to overcome the shortcomings of traditional data protection algorithms and improving security using steganography, AES encryption decryption techniques, compression and splitting technique adoptable to better security for the data stored on the cloud

Reference [16] enhanced cloud data security using AES algorithm, In [16] current available systems for data security which focuses on providing security to the stored data in cloud storage have revived. However, emphasizing less on securing the data whereas it's being transferred.

Introduction of a special type of CPRE [17], sender specified PRE (SSPRE), which permits the delegator to delegate the right to decrypt the ciphertexts of a given sender to its delegate. Formal definition of SSPRE and its security model is given. And also provided the concrete constructions of a secure IND-CPA SS-PRE schema and a secure IND-CCA SS-PRE schema with one-way and single-use properties and demonstrate the security of both schemas in the standard model.

SeDaSC [19]. In this paper, a very new and unique approach towards the Secure Data Sharing in Clouds (SeDaSC) methodology is proposed The SeDaSC methodology encrypts a file with a single encryption key. For each of the users two different key shares are generated, with the user only getting one share. The single share possession of a key allows the SeDaSC methodology to counter the insider threats. The other key share is stored by a trusted third party, which is known as the cryptographic server. To conventional and mobile cloud computing environments, the SeDaSC methodology is applicable.

Group Key Management Protocol, for file sharing on cloud storage (GKMP) is proposed in [20]. Faced with network attacks from public channel, a group key generation scheme based on mixed encryption technology is proposed. And a verification scheme is used to prevent shared files from being attacked by the collusion attack of cloud providers' and group members'.

3. AES ENCRYPTION AND PROXY-TRANSFORMATION

The present research work aims to secure cloud-based file sharing of users using AES (Advanced Encryption Standard) Encryption and Proxy Re-Encryption technique. It aims in encrypting the data both on server side as well as client (user) side to ensure maximum security, below is the proposed model along with related diagrams.

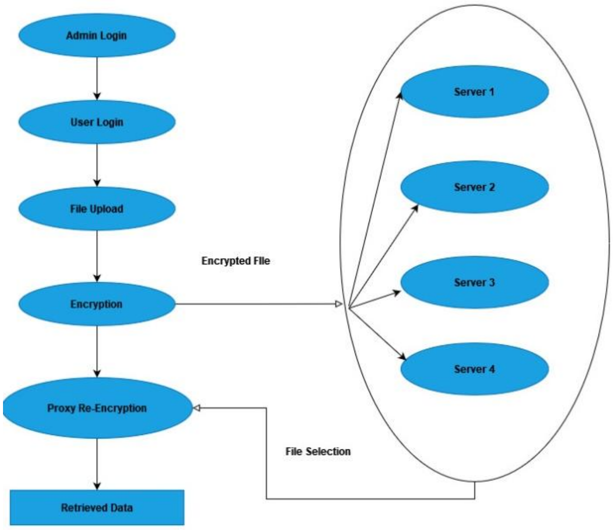


Fig 1: System Architecture

It is necessary to provide administrator login and user login to file sharing systems for standard authentication and authorization. Normal system users can upload file on file server and can share those files with other users on system. Files stored on server will be AES encrypted and can be split on multiple file servers. While downloading files, they will undergo proxy re-encryption process.

4. AES ENCRYPTION AND PROXY RE-TRANSFORMATION MODEL

AES encryption [16] may be very famous due to the fact the encryption technique of AES is rather clean to understand. This enables easier implementation and makes processing of encryption and decryption faster. In addition, AES is known for using much less memory comparing with many different sorts of encryption (like DES), which makes it a real winner when it comes to deciding on your selected encryption approach. AES is a symmetric sort of encryption, because it makes use of the identical key to each encrypt and decrypt data. It additionally makes use of the SPN (substitution permutation network) algorithm, making use of a couple of rounds to encrypt data. These encryption rounds are the purpose at the back of the impenetrability of AES, as there are too many rounds to break through.

There are 3 lengths of AES encryption keys. Each key length has a specific number of feasible key combinations:

- 128-bit key length: 3.4 x 1038
- 192-bit key length: 6.2 x 1057
- 256-bit key length: 1.1 x 1077

Even though the key length of this encryption technique varies, its block size - 128-bits (or 16 bytes) - remains constant.

It converts these individual blocks with the use of keys of 128, 192, and 256 bits. Once it encrypts these blocks, it joins them collectively to form the ciphertext.

Because of numerous advantages of the AES Encryption algorithm, in our proposed methodology we've got used a 128 bit key for a 128 bit block which is constant, to encrypt files at rest.

Following are the steps for a 128-bit block of AES encryption:

- Derive the set of round keys from the cipher key.
- Initialize the state array with the block data (plaintext).
- Add the preliminary round key to the beginning of state array.
- Perform 9 rounds of state manipulation.
- Perform the 10th and very last round of state manipulation.
- Copy the end state array out as the encrypted data (ciphertext).

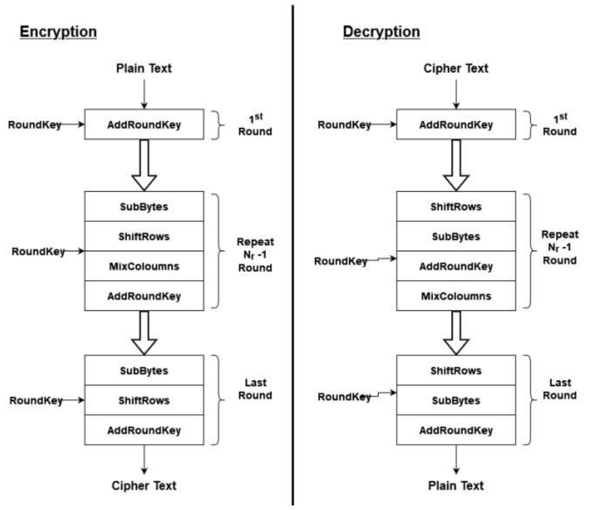


Fig 2: AES Encryption and Decryption Algorithm

Proxy re-encryption [4] is the process where the data that is already encrypted on server by a certain encrypting algorithm is again encoded using a hashing algorithm which provides highly secured information stored in the cloud. This is done to improve security of the stored files.

Every user will have a public key and private key. Public key of each user is well-known to everybody but the private key is known only by the particular user. Following figure shows the concept of proxy re-encryption and how and where keys are used in the process.

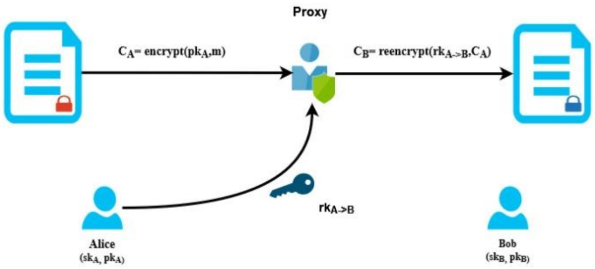


Fig 3: Proxy Transformation

Input – Input to this model will be any type of file which will be uploaded by any registered and authenticated user on our system.

Output – Output for this model will be an re-encrypted file which will be the output of a proxy re-encryption server. This output can be decrypted by the user.

Algorithm–

Step 1) The file is stored in the cloud by encrypting the file using AES encryption technique.

Step 2) While sharing files, AES encrypted file stored, will be directly sent to receiver.

Step 3) Using senders private key and receivers public key, transformation key will be generated and send to proxy server.

Step 4) With the transformation key sent on proxy, the secret key for the AES encrypted file will be transformed (re-encrypted) on proxy and sent to receiver.

Step 5) Transformation received by receiver will be decrypted by its private key, through which receiver will get AES decryption key for file.

Step 6) AES decryption key then will be used by receiver to decrypt encrypted file.

Following Figure 4 shows Processing Data using Encryption and Proxy Re-Encryption using users Alice and Bob.

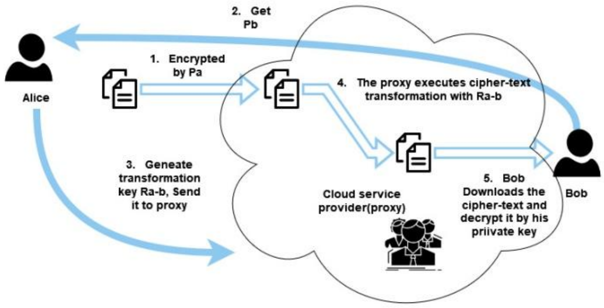

2. Get Pb
1. Encrypted by Pa
Alice
4. The proxy executes cipher-text transformation with Ra-b
3. Geneate transformation key Ra-b, Send it to proxy
Cloud service provider(proxy)
5. Bob Downloads the cipher-text and decrypt it by his priivate key
Bob

Fig 4: Processing Data using Encryption and Proxy Transformation

5. TECHNICAL AND EXPERIMENTAL DETAILS

Security is the need for modern applications. But it's not only about security we should focus on while development, speed or performance of an application is also one of the important factors while developing any type of application. This system needs more hardware to speed up operations, as any user operations needs to go through encryption, decryption and file processing.

We have experimented with some key management techniques which helped us manage secret keys. For key generation, we appended randomized strings starting and ending at the primary attribute and generating a secure key using MD5.

And we have observed splitting a secret key into multiple parts and storing it on different servers is an added advantage on its security as it is very difficult for threat actors to penetrate into systems and get secured keys by bypassing firewalls and other security protocols.

Observations – We have observed that standard cloud-based file sharing system without security implementation is vulnerable to threats. Security of the system can be improved by encrypting data at rest on cloud and providing end to end encryption using proxy re-encryption technique. Figure 5 is the visualization pf effect of AES encryption and Proxy Re-Encryption on Security of System.

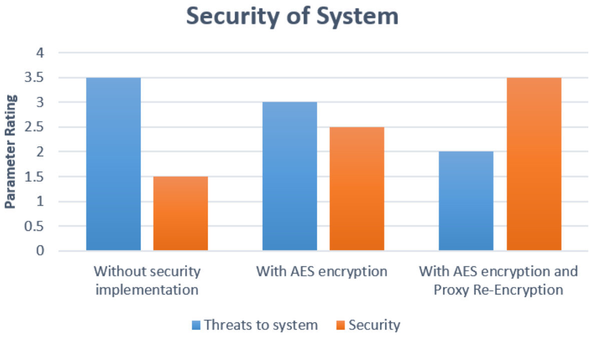


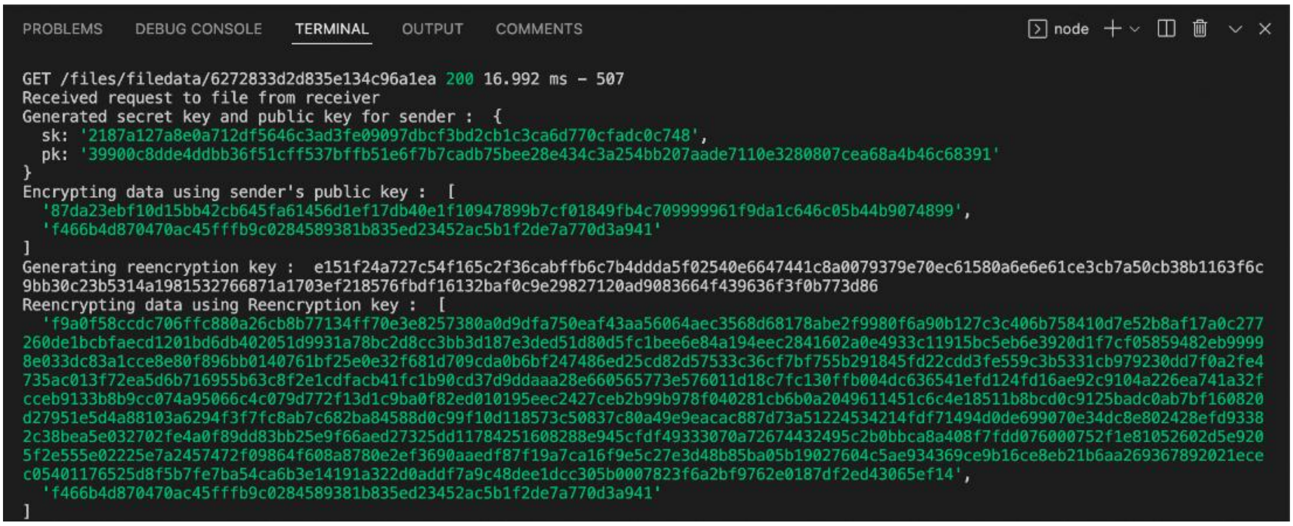
Fig 5: Effect of AES encryption and Proxy Re-Encryption on Security of System

6. RESULT AND DISCUSSION

We have reviewed and implemented a user-friendly user interface with sharing files securely with others. In our analysis we have provided security to data in transit (motion) and data at rest on cloud. For data at rest, using AES encryption algorithm which is better than any other because of its simplicity as a symmetric encryption algorithm and it is secure because it is less susceptible to cryptanalysis.

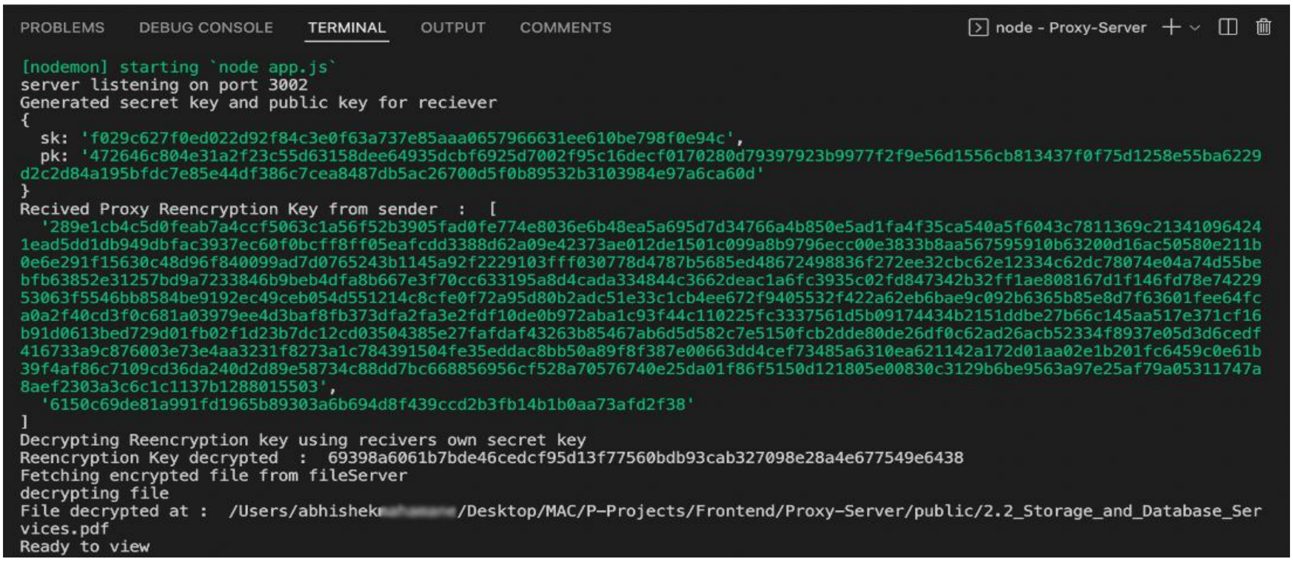
For the data in transit from server to client and client to server we have implemented Proxy re-encryption with proxy in between server and client. This provides security to the server by not allowing any user to directly communicate with the server and passing the user through proxy. Proxy Re-encryption results in a double security advantage. Using proxy is one. And further we are re-encrypting data in transit, which will protect data in motion from eavesdroppers.

For implementation of AES and Proxy Re-Encryption in Cloud Based File Sharing System we used NodeJS, Express and React which have a base in JavaScript, in application development, benefits in performance and rapid development. Following are some of the backend view of proxy re-encryption process.



```
GET /files/filedata/6272833d2d835e134c96a1ea 200 16.992 ms - 507
Received request to file from receiver
Generated secret key and public key for sender :  {
  sk: '2187a127a8e0a712df5646c3ad3fe09097dbcf3bd2cb1c3ca6d770cfadc0c748',
  pk: '39900c8dde4ddbb36f51cff537bffb51e6f7b7cadb75bee28e434c3a254bb207aade7110e3280807cea68a4b46c68391'
}
Encrypting data using sender's public key :  [
  '87da23ebf10d15bb42cb645fa61456d1ef17db40e1f10947899b7cf01849fb4c709999961f9da1c646c05b44b9074899',
  'f466b4d870470ac45fffb9c0284589381b835ed23452ac5b1f2de7a770d3a941'
]
Generating reencryption key :  e151f24a727c54f165c2f36cabffb6c7b4ddda5f02540e6647441c8a0079379e70ec61580a6e6e61ce3cb7a50cb38b1163f6c
9bb30c23b5314a1981532766871a1703ef218576fbdf16132baf0c9e29827120ad9083664f439636f3f0b773d86
Reencrypting data using Reencryption key :  [
  'f9a0f58ccdc706ffc880a26cb8b77134ff70e3e8257380a0d9dfa750eaf43aa56064aec3568d68178abe2f9980f6a90b127c3c406b758410d7e52b8af17a0c277
260de1bcbfaecd1201bd6db402051d9931a78bc2d8cc3bb3d187e3ded51d80d5fc1bee6e84a194eec2841602a0e4933c11915bc5eb6e3920d1f7cf05859482eb9999
8e033dc83a1cce8e80f896bb0140761bf25e0e32f681d709cda0b6bf247486ed25cd82d57533c36cf7bf755b291845fd22cdd3fe559c3b5331cb979230dd7f0a2fe4
735ac013f72ea5d6b716955b63c8f2e1cdfacb41fc1b90cd37d9ddaaa28e660565773e576011d18c7fc130ffb004dc636541efd124fd16ae92c9104a226ea741a32f
cceb9133b8b9cc074a95066c4c079d772f13d1c9ba0f82ed010195eec2427ceb2b99b978f040281cb6b0a2049611451c6c4e18511b8bcd0c9125badc0ab7bf160820
d27951e5d4a88103a6294f3f7fc8ab7c682ba84588d0c99f10d118573c50837c80a49e9eacac887d73a51224534214fdf71494d0de699070e34dc8e802428efd9338
2c38bea5e032702fe4a0f89dd83bb25e9f66aed27325dd11784251608288e945cfdf49333070a72674432495c2b0bbca8a408f7fdd076000752f1e81052602d5e920
5f2e555e02225e7a2457472f09864f608a8780e2ef3690aaedf87f19a7ca16f9e5c27e3d48b85ba05b19027604c5ae934369ce9b16ce8eb21b6aa269367892021ece
c05401176525d8f5b7fe7ba54ca6b3e14191a322d0addf7a9c48dee1dcc305b0007823f6a2bf9762e0187df2ed43065ef14',
  'f466b4d870470ac45fffb9c0284589381b835ed23452ac5b1f2de7a770d3a941'
]
```

Figure 6. Generating Keys for Proxy Re-Encryption



```
[nodemon] starting `node app.js`
server listening on port 3002
Generated secret key and public key for reciever
{
  sk: 'f029c627f0ed022d92f84c3e0f63a737e85aaa0657966631ee610be798f0e94c',
  pk: '472646c804e31a2f23c55d63158dee64935dcbf6925d7002f95c16decf0170280d79397923b9977f2f9e56d1556cb813437f0f75d1258e55ba6229
d2c2d84a195bfdc7e85e44df386c7cea8487db5ac26700d5f0b89532b3103984e97a6ca60d'
}
Recived Proxy Reencryption Key from sender  :  [
  '289e1cb4c5d0feab7a4ccf5063c1a56f52b3905fad0fe774e8036e6b48ea5a695d7d34766a4b850e5ad1fa4f35ca540a5f6043c7811369c21341096424
1ead5dd1db949dbfac3937ec60f0bcff8ff05eafcdd3388d62a09e42373ae012de1501c099a8b9796ecc00e3833b8aa567595910b63200d16ac50580e211b
0e6e291f15630c48d96f840099ad7d0765243b1145a92f2229103fff030778d4787b5685ed48672498836f272ee32cbc62e12334c62dc78074e04a74d55be
bfb63852e31257bd9a7233846b9beb4dfa8b667e3f70cc633195a8d4cada334844c3662deac1a6fc3935c02fd847342b32ff1ae808167d1f146fd78e74229
53063f5546bb8584be9192ec49ceb054d551214c8cfe0f72a95d80b2adc51e33c1cb4ee672f9405532f422a62eb6bae9c092b6365b85e8d7f63601fee64fc
a0a2f40cd3f0c681a03979ee4d3baf8fb373dfa2fa3e2fdf10de0b972aba1c93f44c110225fc3337561d5b09174434b2151ddbe27b66c145aa517e371cf16
b91d0613bed729d01fb02f1d23b7dc12cd03504385e27fafdaf43263b85467ab6d5d582c7e5150fcb2dde80de26df0c62ad26acb52334f8937e05d3d6cedf
416733a9c876003e73e4aa3231f8273a1c784391504fe35eddac8bb50a89f8f387e00663dd4cef73485a6310ea621142a172d01aa02e1b201fc6459c0e61b
39f4af86c7109cd36da240d2d89e58734c88dd7bc668856956cf528a70576740e25da01f86f5150d121805e00830c3129b6be9563a97e25af79a05311747a
8aef2303a3c6c1c1137b1288015503',
  '6150c69de81a991fd1965b89303a6b694d8f439ccd2b3fb14b1b0aa73afd2f38'
]
Decrypting Reencryption key using recivers own secret key
Reencryption Key decrypted  :  69398a6061b7bde46cedcf95d13f77560bdb93cab327098e28a4e677549e6438
Fetching encrypted file from fileServer
decrypting file
File decrypted at :  /Users/abhishek[redacted]/Desktop/MAC/P-Projects/Frontend/Proxy-Server/public/2.2_Storage_and_Database_Ser
vices.pdf
Ready to view
```

Figure 7. Console view of proxy re-encryption process

7. CONCLUSION

We have presented a complete overview of the cloud computing paradigm, as well as distinct techniques used in file sharing systems to improve security. We discussed cloud security issues and we have seen techniques to address these issues such as encryption, proxy re encryption and access control.

Security to the file in the cloud at rest is provided by AES encryption and for files in transit proxy re-encryption is providing secure shield protecting files and maintaining confidentiality and integrity.

To make file sharing system more secure we need to use combinations of techniques such as AES, Proxy re-encryption. Also, we can use a combination of access control techniques to identify user and its role correctly.

8. REFERENCES

[1] M. Malarvizhi, J. A. J. Sujana and T. Revathi, "Secure file sharing using cryptographic techniques in cloud," 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), 2014, pp. 1-6, doi: 10.1109/ICGCCEE.2014.6921421.

[2] B. Fuhry, L. Hirschoff, S. Koesnadi and F. Kerschbaum, "SeGShare: Secure Group File Sharing in the Cloud using Enclaves," 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2020, pp. 476-488, doi: 10.1109/DSN48063.2020.00061.

[3] S. Zhu, X. Yang and X. Wu, "Secure Cloud File System with Attribute Based Encryption," 2013 5th International Conference on Intelligent Networking and Collaborative Systems, 2013, pp. 99-102, doi: 10.1109/INCoS.2013.22.

[4] R. Nivedhaa and J. J. Justus, "A Secure Erasure Cloud Storage System Using Advanced Encryption Standard Algorithm and Proxy Re-Encryption," 2018 International Conference on Communication and Signal Processing (ICCSP),2018, pp. 0755-0759, doi: 10.1109/ICCSP.2018.8524257.

[5] Armstrong Nhlabatsi, Jin B. Hong, Dong Seong Kim, Rachael Fernandez, Alaa Hussein, Noora Fetais, and Khaled M. Khan, "Threat-specific Security Risk Evaluation in the Cloud", Oct 2018 IEEE Transactions on Cloud Computing, DOI 10.1109/TCC.2018.2883063

[6] BADER ALOUFFI, MUHAMMAD HASNAIN, ABDULLAH ALHARBI, WAEL ALOSAIMI, HASHEM ALYAMI AND MUHAMMAD AYAZ, "A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies", IEEE Access Volume 9 2021 DOI:10.1109/ACCESS.2021.3073203

[7] Jianghong Wei, Wenfen Liu, Xuexian Hu, "Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption", IEEE JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015

[8] HEQING SONG, JIFEI LI, AND HAOTENG LI," A Cloud Secure Storage Mechanism Based on Data Dispersion and Encryption", IEEE Access Volume 9 2021 DOI:10.1109/ACCESS.2021.3075340

[9] Kajal Rani, Raj Kumar Sagar, "Enhanced Data Storage Security in Cloud Environment using Encryption, Compression and Splitting technique", IEEE 2017 2nd International Conference on Telecommunication and Networks (TEL-NET 2017)

[10] Khashan, "Secure Outsourcing and Sharing of Cloud Data Using a User-Side Encrypted File System," in IEEE Access, vol. 8, pp. 210855-210867, 2020.

[11] E. Chen, Y. Zhu, K. Liang and H. Yin, "Secure Remote Cloud File Sharing with Attribute-based Access Control and Performance Optimization," in IEEE Transactions on Cloud Computing.

[12] Bhagat and N. Rathee, "Addressing Techniques for Secure Data Sharing in Cloud," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018.

[13] Jacob and V. R. Rekha, "Secured and reliable file sharing system with de-duplication using erasure correction code," 2017 International Conference on Networks & Advances in Computational Technologies (NetACT),

2017.

[14] X. Zhang, W. Guo, Z. Li, X. Zhao and X. Qin, "MLFS: A multiple layers shared file system for cloud computing," 2014 IEEE Globecom Workshops (GC Wkshps), 2014.

[15] L. Selvam and R. J. Arokia, "Secure Data Sharing of Personal Health Records in Cloud Using Fine-Grained and Enhanced Attribute-Based Encryption," 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018.

[16] K. M. Akhil, M. P. Kumar and B. R. Pushpa, "Enhanced cloud data security using AES algorithm," 2017 International Conference on Intelligent Computing and Control (I2C2), 2017.

[17] P. Zeng and K. R. Choo, "A New Kind of Conditional Proxy Re-Encryption for Secure Cloud Storage," in IEEE Access, vol. 6.

[18] J. Wei, W. Liu and X. Hu, "Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption," in IEEE Transactions on Cloud Computing, vol. 6, no. 4, pp. 1136-1148, 1 Oct.-Dec. 2018

[19] M. Ali et al., "SeDaSC: Secure Data Sharing in Clouds," in IEEE Systems Journal, vol. 11, no. 2, pp. 395-404, June 2017.

[20] S. Zhang, S. Han, B. Zheng, K. Han and E. Pang, "Group Key Management Protocol for File Sharing on Cloud Storage," in IEEE Access, vol. 8.

[21] Research Clue. (2020). design and implementation of a cloud based file sharing system.[Accessed: 2022-3-30].