| ISSN: 2395-7852 | www.ijarasem.com | Bimonthly, Peer Reviewed & Referred Journal|

| Volume 9, Issue 1, January 2022 |

Cloud Data Security Using Efficient Cryptosystem Technique

Aditya Bastawad, Deepthi A, Neha R and Spoorthy A N, Mrs. S Vidhya

Dept. of I.S.E., The Oxford college of Engineering, Bangalore Karnataka, India

Dept. of I.S.E., The Oxford college of Engineering, Bangalore Karnataka, India

Dept. of I.S.E., The Oxford college of Engineering, Bangalore Karnataka, India

Dept. of I.S.E., The Oxford college of Engineering, Bangalore Karnataka, India

Dept. of I.S.E., The Oxford college of Engineering, Bangalore Karnataka, India

ABSTRACT: Services are distributed among all servers and between the users and individuals in the cloud environment. Cloud providers have trouble guaranteeing file protection as security is the biggest issue in data handling and transfer as it can be accessed, misused and destroyed the original data form. Cloud security is a big concern in the cloud computing environment. To safeguard the cloud environment, many research works are being proposed. To overcome the security issue and achieve the CIA property (confidentiality, integrity and availability) the cryptography is used. Cryptography is the most useful technique to ensure a high level of data transfer and storage security. To solve this we are going to introduce a symmetric-block cipher for high data security and confidentiality. The symmetric - block cipher used here is Triple DES Algorithm.

KEYWORDS: Cloud Computing, Cryptosystem, Encryption, Decryption, 3DES.

I. Introduction

Cloud infrastructures can be roughly categorized as either private or public. In a private cloud, the infrastructure is managed and owned by the customer and located on-premise (i.e., in the customers region of control). In particular, this means that access to customer data is under its control and is only granted to parties it trusts. In a public cloud the infrastructure is owned and managed by a cloud service provider and is located off-premise (i.e., in the service provider's region of control). This means that customer data is outside its control and could potentially be granted to untrusted parties. When an organization elects to store data or host applications on the public cloud, it loses its ability to have physical access to the servers hosting its information. As a result, potentially sensitive data is at risk from insider attacks.

To address the concerns outlined above and increase the adoption of cloud storage, cloud data security is must. To address users' concerns over potential data leaks in cloud storage, a common approach is for the data owner to encrypt all the data before uploading them to the cloud, such that later the encrypted data may be retrieved and decrypted by those who have the decryption keys. Such cloud storage is often called the cryptographic cloud storage.

Cryptography is a type of a rule or a technique by which private or sensitive information is secured from the public or other members. It focuses on the confidential data, authentication, data integrity etc. The use of Cryptography in passwords is a very famous example.

The idea of cryptography is to convey a private message or piece of information from the sender party to the intended recipient without getting the message intruded on by a malicious or untrusted party. In the world of cryptography, this suspicious third party that is trying to sneak into a private communication to extract something sensitive out of it is called an adversary.

Cryptography protects us from these unwanted adversaries by offering a range of algorithms required to hide or protect our message in the best way possible and transmit it comfortably over a not-so-secure network.

II. Related work

[1] Title: Cloud Data Security using Attribute-based Key-Aggregate Cryptosystem. Authors: Pooja More Year: 2020 Technique: Baojiang Cui et al proposed a key aggregate searchable encryption (KASE) scheme. The proposed KASE scheme applies to any cloud storage that supports the searchable group data sharing functionality Advantages: Users may selectively share a group of selected files with a group of selected users, therefore we can acquire security.

Disadvantages: If request to the file access is greater than the threshold then it would be difficult to handle all the request while simultaneously checking the users that can access the file. [2] Title: Performance Analysis of Homomorphic Cryptosystem on Data Security in Cloud Computing. Authors: Mebiratu Beyene, Dr. K. Raja Shekar. Year: 2020. Technique: We have designed a method for security for the data in the cloud, with FHC algorithm and the complexity of three homomorphic cryptographic algorithm which are SDC FHE, Paillier and RSA with encryption time, decryption time, throughput and memory usage. Advantages: Only the client holds the decryption keys and provided the required steps to read the information which has been in the cloud. Disadvantages: In this method the data in the secure systems will be encrypted by the client and then sent to the cloud for storage or processing. Hence the client will have to manually encrypt data each time when it wants to store the data in the cloud. [3] Title: Providing Security, Integrity and Authentication Using ECC Algorithm in cloud storage. Authors: Akanksha Bansal and Arun Agrawal. Year: 2019 Technique: Providing Security using ECC as compared to RSA. Integrity checking using Metadata. Advantages: In this paper we are secure saving of our data in the cloud storage in an efficient way which need use less CPU Power and Processing time so we are using Electronic Curve Cryptography (ECC) algorithm for security purpose. Disadvantages: The size of cipher text is not constant, so it increases the cost of storing and sharing cipher text. [4] Title: Data Security in Cloud Computing with Elliptic Curve Cryptography. Authors: Veerraju Gampala, Srilakshmi Inuganti and Satish Muppidi. Year: 2019 Technique: Send secure data by applying digital signature and encryption to data with ECC and decrypt with private key and verify the signature with public key. Advantages: Integration of both encryption and decryption approaches in this dimension can improve the performances of security of messages against several cryptographic attacks. Disadvantages: To make a powerful decryption key is not concerned in this scheme. [5] Title: Cryptographic Implementation of Aggregate-Key Encryption for Data Sharing in Cloud Storage Authors: Priyanka P. Kendrekar and M. K. Chavan. Year: 2018 Technique: Construction of a Key aggregate cryptosystem is implemented in Java with Java pairing based cryptography (JPBC) library for underlying elliptic-curve group and pairing operations. Advantages: This paper proposes an aggregate-key cryptosystem in which the decryption key generation mechanism is so powerful that the generated aggregate key is able to decrypt any subset of cipher text classes generated during data upload. Disadvantages: The predefined bound on the number of cipher text classes

III. PROPOSED ALGORITHM

A. *Design Considerations:*

- The entries of each row are permutations of values between 0 and 15.
- If we change a single bit in the input, two or more bits will be changed in the output.
- DES uses sixteen rounds of Feistel ciphers.
- It has been proved that after eight rounds, each ciphertext is a function of every plaintext bit and every key bit.
- The ciphertext is thoroughly a random function of plaintext and ciphertext. Therefore, it looks like eight rounds should be enough.
- However, experiments have found that DES versions with less than sixteen rounds are even more vulnerable to known-plaintext attacks than brute-force attack, which justifies the use of sixteen rounds by the designers of DES.

B. *Description of the Proposed Algorithm:*

Aim of the proposed algorithm is to protect against brute force attacks. The original DES symmetric encryption algorithm specified the use of 56-bit keys -- not enough, by 1999, to protect against practical brute force attacks. Triple DES specifies the use of three distinct DES keys, for a total key length of 168 bits.

The encryption-decryption process is as follows −

- Encrypt the plaintext blocks using single DES with key K1.
- Now decrypt the output of step 1 using single DES with key K2.
- Finally, encrypt the output of step 2 using single DES with key K3.
- The output of step 3 is the ciphertext.
- Decryption of a ciphertext is a reverse process. User first decrypt using K3, then encrypt with K2, and finally decrypt with K1.

IV. PSEUDO CODE

```
Cipher (plainBlock[64], RoundKeys[16, 48], cipherBlock[64])
{
    permute (64, 64, plainBlock, inBlock, InitialPermutationTable)
    split (64, 32, inBlock, leftBlock, rightBlock)
    for (round = 1 to 16)
{
mixer (leftBlock, rightBlock, RoundKeys[round])
if (round!=16) swapper (leftBlock, rightBlock)
}
     combine (32, 64, leftBlock, rightBlock, outBlock)
     permute (64, 64, outBlock, cipherBlock, FinalPermutationTable) }
     mixer (leftBlock[48], rightBlock[48], RoundKey[48])
{
     copy (32, rightBlock, T1) function (T1, RoundKey, T2)
     exclusiveOr (32, leftBlock, T2, T3)
     copy (32, T3, rightBlock)
}
swapper (leftBlock[32], rigthBlock[32])
{
      copy (32, leftBlock, T)
      copy (32, rightBlock, leftBlock)
      copy (32, T, rightBlock)
}
function (inBlock[32], RoundKey[48], outBlock[32])
{
      permute (32, 48, inBlock, T1, ExpansionPermutationTable)
      exclusiveOr (48, T1, RoundKey, T2)
      substitute (T2, T3, SubstituteTables)
      permute (32, 32, T3, outBlock, StraightPermutationTable)
} substitute (inBlock[32], outBlock[48], SubstitutionTables[8, 4, 16])
{
for (i = 1 to 8)
{
row <- 2 x inBlock[i x 6 + 1] + inBlock [i x 6 + 6]
col <- 8 x inBlock[i x 6 + 2] + 4 x inBlock[i ¥ 6 + 3] +
      2 x inBlock[i x 6 + 4] + inBlock[i x 6 + 5]
      value = SubstitutionTables [i][row][col]
      outBlock[[i x 4 + 1] <- value / 8; value <- value mod 8
      outBlock[[i x 4 + 2] <- value / 4; value <- value mod 4
      outBlock[[i x 4 + 3] <- value / 2; value <- value mod 2
      outBlock[[i ¥ 4 + 4] ¨ value
      }
      }
```

V. Simulation Results

The system execution details are shown with the help of following screenshots:

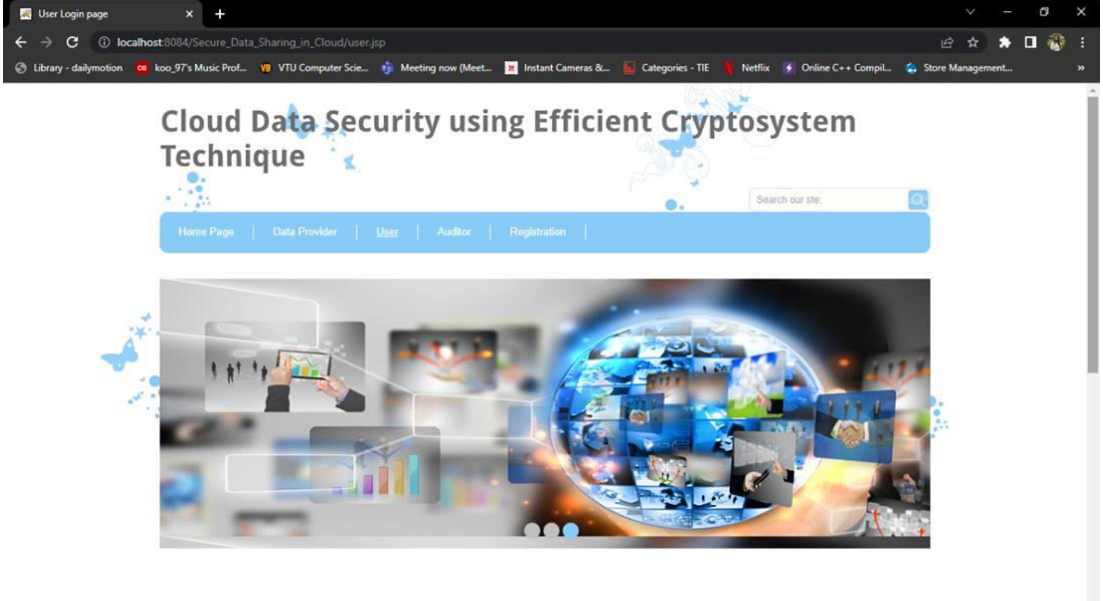

Fig 3.1 Home Page

In the above fig, home page is shown where the Data Provider, User, Auditor and registration applications can be accessed to provide data and retrieve data.

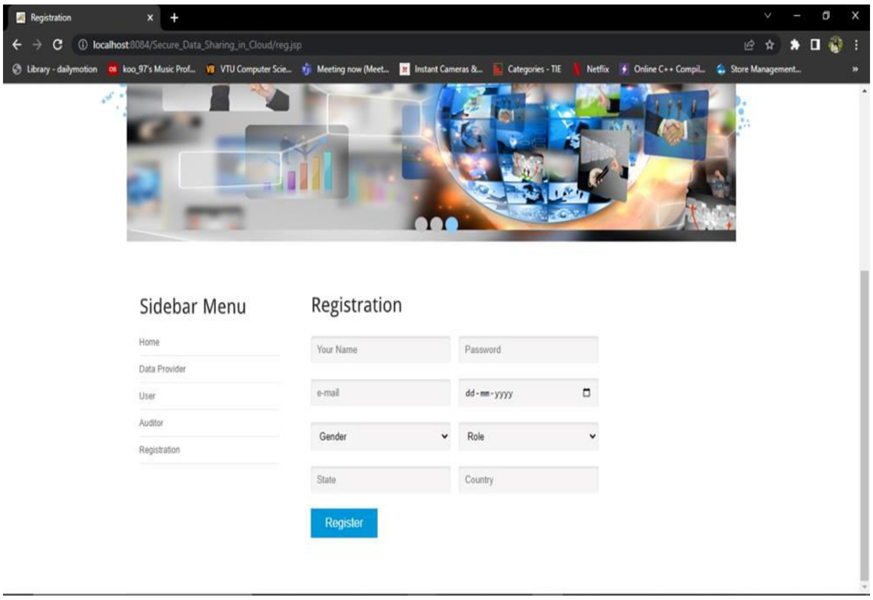

Fig 3.2 Registration

In the above figure, we can see that the registration page is seen. Here we can register new data provider and users in order if they want to add data or retrieve data from our page for more security.

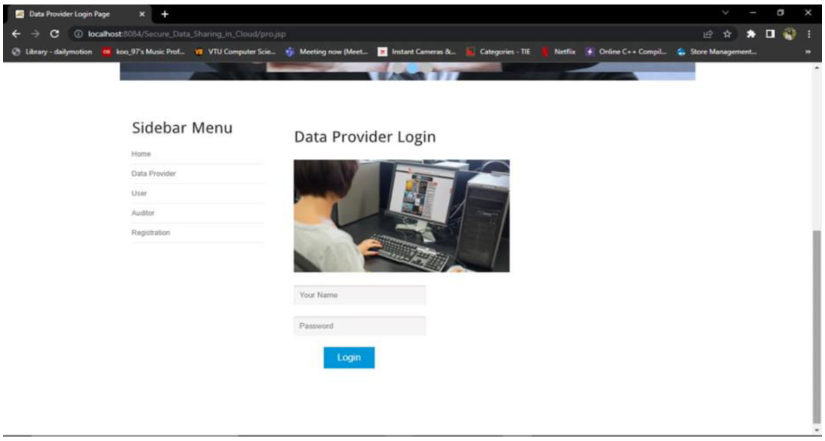

Fig 3.3 Data Provider Login

As shown above, data provider page is now open and here we can give our credentials.

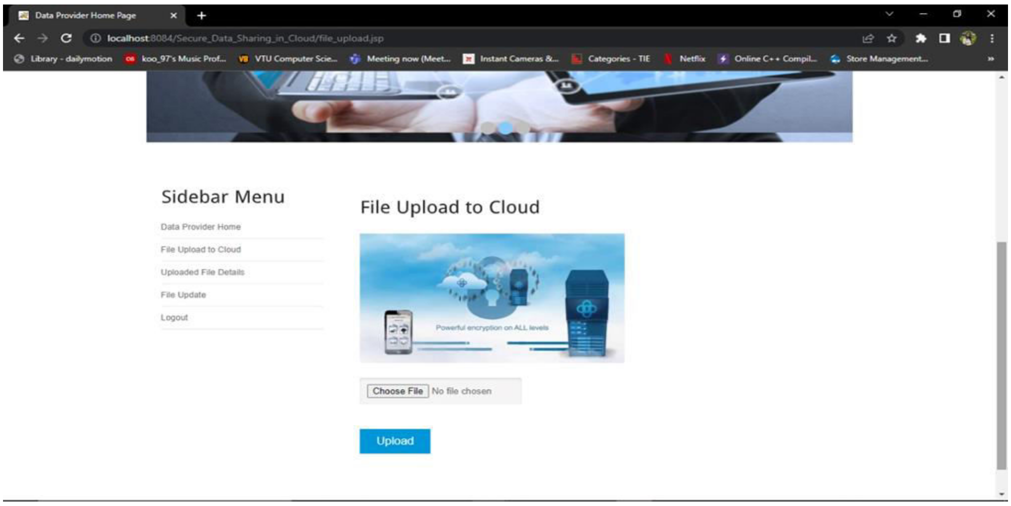

Fig 3.4 File Upload to Cloud

The above figure illustrates that the data provider can now upload any data they want others to access to the cloud. As soon as the data provider uploads the data it will automatically be encrypted and stored in the cloud. Even if the data provider wants to view the doc stored in the cloud, it will be encrypted. He needs to use secret key to access the data.

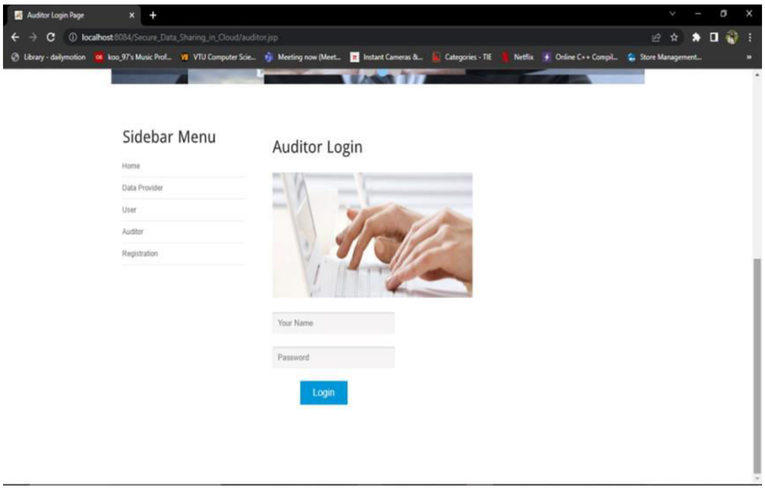

Fig 3.5 Auditor Login

As shown in the above figure, the auditor will login using its credentials. An auditor login is a Key Management System. This handles all the requests from the users to access certain data from the cloud. He will generate the secret keys according to the users request and he will have all details of the provider n the user.

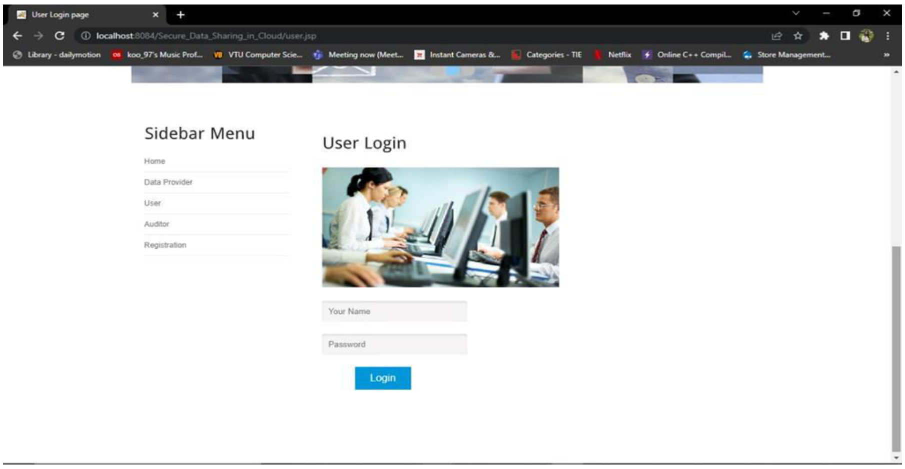

Fig 3.6 User Login

As shown in above figure, The user can finally login into our system and request for the files uploaded by the data provider. First he will request to download a file from the cloud. This request will be sent to the Auditor, he will handle all the requests and send the appropriate key to the user. Later the user will receive this secret key through mail. He will retrieve the file by just typing the secret key. Suppose the key is incorrect the text file will not be downloaded into the system.

VI. Conclusion and Future Work

In this project, we have used the Triple Data Encryption(3DES) algorithm to provide security, privacy and trust to the user's sensitive data. The owner possesses a secret key which can be extracted for different classes. The plaintext will be encrypted using encryption algorithm and secret key hence, proposing ciphertext to the receiver, the receiver will receive the cipher text which is later decrypted using the decryption algorithm and the secret key. Therefore, the plaintext (original text) is received. Thus we will use cryptography for providing security to the Cloud Data.

References

[1] X.F. Chen, J. Li, X.Y. Huang, J.W. Li, Y. Xiang, "Secure Outsourced Attribute-based Signatures", IEEE Trans. on Parallel and Distributed Systems DOI.ieeecomputersociety.org/10.1109/TPDS.2013.180, 2013.

[2] K. Yang, X. Jia, and K. Ren, "DAC-MACS: Effective data access control for multi-authority cloud storage systems", in Proceedings of the 32nd IEEE International Conference on Computer Communications. IEEE, 2013, pp. 2895–2903.

[3] Y. Wu, Z. Wei, and H. Deng, "Attribute-based access to scalable media in cloud-assisted content sharing", IEEE Transactions on Multimedia, vol. 15, no. 4, pp. 778–788, 2013.

[4] Z. Liu, Z. Wang, X. Cheng, et al. "Multi-user Searchable Encryption with Coarser-Grained Access Control in Hybrid Cloud", Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, pp. 249-255, 2013.

[5] R. A. Popa ,N. Zeldovich, "Multi-key searchable encryption", Cryptology ePrint Archive, Report 2013/508, 2013.

[6] X. Liu, Y. Zhang, B. Wang and J. Yan. "Mona: secure multi owner data sharing for dynamic groups in the cloud", IEEE Transactions on Parallel and Distributed Systems, 2013, 24(6): 1182-1191.

[7] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability", IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 8, pp. 2201–2210, 2014.

[8] Sourya Joyee De, Asim K. Pal, "A Policy-based Security Framework for Storage and Computation on Enterprise Data in the Cloud", 2014 47th Hawaii International Conference on System Science.

[9] C. Chu, S. Chow,W. Tzeng, et al. "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, 2014, 25(2): 468-477.

[10] Baojiang Cui, Zheli Liu and Lingyu Wang, "Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage", IEEE Transactions on computers, vol. 6, no. 1, January 2014.

[11] Wei Li, Kaiping Xue, Yingjie Xue, and Jianan Hong, "TMACS: A Robust and Verifiable Threshold Multi-Authority Access Control System in Public Cloud Storage", DOI 10.1109/TPDS.2015.2448095, IEEE Transactions on Parallel and Distributed Systems.

[12] Wei Teng, Geng Yang, Yang Xiang, Ting Zhang and D

[13] H. Deng, Q. Wu, B. Qin, J. Domingo-Ferrerd and L. Zhang, "Ciphertext Policy hierarchical attribute-based encryption with short ciphertexts" , Information Sciences,vol.275,pp:370-384,2014.

[14] J.Li, X.F. Chen, M.Q. Li, J.W. Li, P. Lee, Wenjing Lou. "Secure Deduplication with Efficient and Reliable Convergent Key Management", IEEE Transactions on Parallel and Distributed Systems, 25(6): 1615-1625, 2014.

[15] Smital Erande1 , V. S. Ranmalkar2, "Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage", International Journal of Science and Research (IJSR) , Volume 5 Issue 12, December 2016.