# IMPLEMENTATION OF DIFFIE-HELLMAN KEY EXCHANGE BASED ON CLOUD

[1]Sowmya G V, [2]Nideep G P, [3]Adithya Narayan, [4]Adithya K R

[1]Assistant Professor, [2,3,4]Students, 8th sem,
[1,2,3,4]Department of IS&E,
[1,2,3,4]J N N College of Engineering, Shivamogga, India

***Abstract:*** Most businesses adopt the cloud computing architecture. It offers on-demand service, extensive network connectivity, flexibility, and other benefits. However, the use of these characteristics is hampered by several security issues. When a cloud service user uploads sensitive data to the cloud platform, it must be sent securely. It should be sent through a secure communication channel for this purpose. The authentication procedure preserves the idea of a verifier who ensures the identity of the second user and is responsible for communicating with the first user. Data confidentiality implies that the data sent through the channel is private and can only be seen by authorised users. As a result, sensitive data is only safe when it is not tampered with or targeted by an unauthorised person.

***Index Terms* - Diffie-Hellman key exchange, AES, Cloud.**

## I. INTRODUCTION

Many security concerns impede the usage of cloud characteristics such as on-demand service, broad network access, flexibility, and so on. When a cloud service user uploads sensitive data to the cloud platform, it must be sent securely. It should be sent through a secure communication channel for this purpose. A Diffie-Hellman key exchange technique is a feasible solution to this problem. It is used to safely exchange cryptographic keys over a public channel. However, the present technique is vulnerable to MiM and plain-text attacks. To address this, an enhanced version of the Diffie-Hellman algorithm was developed.

In the proposed work, different encryption models are designed. Chaotic sequences are used for key sequences. The rest of the sections are as follows: In section II, a literature survey is carried out. The proposed work is explained in section III. The result obtained is presented in section IV. The analysis of the results is carried out in section V. The conclusion of the proposed work is given in section VI.

## II. LITERATURE SURVEY

To improve security in cloud computing by thwarting Mim and plain-text assaults, an expanded version of the Diffie-Hellman key exchange method was proposed. It also offers a cloud-based architecture that uses the algorithm to guarantee the privacy, accuracy, and personalization of data [1]. It was suggested to use a hybrid strategy that combines the symmetric key Blowfish method for secrecy, the SHA3 algorithm for data integrity, and the Diffie-Hellman key exchange algorithm for secure data transfer. Scalable processing and cloud storage are provided by using Amazon's EC2 and S3 services [2]. Elliptic Curve Cryptography and Diffie-Hellman key exchange are used in the efficient authentication mechanism (EAAP) for safe cloud computing to improve security in the cloud environment [3]. It was discussed the necessity of a secure gateway to stop data theft and leaks during data transfer. To improve security and stop unauthorised access to or alteration of data, it suggests a more complicated version of the Diffie-Hellman algorithm for key exchange [4]. It [5] uses the Geffe generator to generate a secure binary sequence, performs tests to ensure its quality, and stores keys as hashes on the server, preventing unauthorised transmission. Information technology and security are greatly aided by the Diffie-Hellman key exchange protocol, a trustworthy server, and cryptographically secure CRC. The implementation is simple since the server's duties are reduced to providing a secure random divisor polynomial. By employing random, sizable, and distinct private key values in each session, this method improves security by making man-in-the-middle assaults simple to spot [6].

By adopting the Diffie-Hellman method for key exchange and proving effective performance at a lower cost than previous techniques, this research increases the effectiveness of cloud auditing. The suggested encryption technique improves cloud data integrity and has the possibility for commercial cloud audits [7]. In order to provide resistance against security assaults, this study presents a strong authentication system that overcomes flaws in existing methods. Its efficiency in terms of computation and communication overhead, as shown by the performance analysis, makes it suited for combining massive service servers with a centralised authorization centre [8]. The Diffie-Hellman algorithm's suggested enhanced version offers simplified computing with fewer steps and does away with exponentiation operations, which shortens computation time. The technique is readily adapted for any number of participants and is illustrated with eight participants [9]. The difficulties associated with data storage, retrieval, and security in organisations are discussed in the study. It [10] suggests a brand-new encryption system, one that uses the Diffusion-based cryptography method and the Diffie-Hellman key exchange algorithm to safeguard data from unauthorised users. The main objective is to verify identities and restrict access to authorised users. A secondary objective is to encrypt data while it is being stored and retrieved to improve security and guard against abuse by both internal and external intruders. The studies mentioned above help us comprehend information security better.

## III. PROPOSED WORK

The proposed work as shown in Fig.1 entails the safe exchange of private keys between two users. The private keys are retrieved from a form submission and saved for later use. The modified Diffie Hellman key exchange technique is used to produce shared keys, which are subsequently utilised for encryption and decryption. The AES method in Cypher Feedback (CFB) mode is used to encrypt submitted data, and the resultant ciphertext and initialization vector are saved. Boto3, a Python module, is used to upload the encrypted file to an S3 bucket in a chosen AWS region. To download and decrypt the file, the AWS access key and secret access key are needed, and the decrypted file is stored to disc. In the event of an exception, such as an erroneous key.

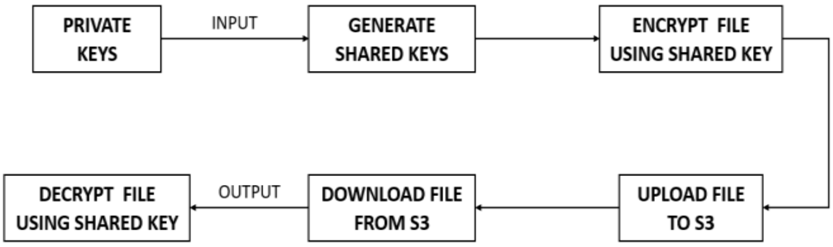


**Fig.1 Proposed Design**

## IV. ALGORITHM

The existing Diffie-Hellman key exchange algorithm was modified to secure algorithm from MiM and plain-text attack, and implemented on cloud platform as follows:

1. Two publicly known numbers are used ‘q’ and ‘n’. (n<q)

2. User 1 generates private key ‘x’ and calculates public key as ‘A’.

$$A = n^x \bmod q$$

3. Similarly, user 2 generates private key ‘y’ and calculates public key as ‘B’.

$$B = n^{y} \bmod q$$

4. Both shares their public keys A and B with each other.
5. User 1 and user2 calculates the secret keys as ‘K1’ and ‘K2’ respectively.

$$K1 = B^{x} \bmod q$$

$$K2 = A^{y} \bmod q$$

6. Further, they select the random arbitrary numbers ’t’ and ‘s’.
7. Using logarithmic function user 1 and user 2 calculates ‘C’ and ‘D’ respectively.

$$C = \log_{m} (t, K1)$$
$$D = \log_{m} (s, K2)$$

8. Another public keys are generated by both the users.

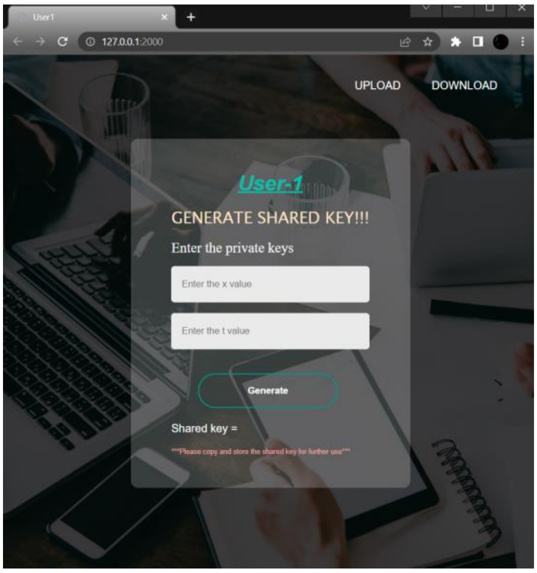
$$G = C * t$$
$$H = D * s$$

9. Finally, both exchanges the public keys to generate final shared symmetric key.

$$FK_{A} = G * H$$
$$FK_{B} = H * G$$

10. User 1 encrypt the file using AES encryption algorithm with the shared key generated. Later the encrypted file is uploaded to Amazon S3.
11. User 2 downloads the uploaded file from the Amazon S3 and decrypts it using the shared key generated.
12. If the shared key generated is not same as the key used for encryption, then error message is displayed.

## V. RESULTS



User-1

GENERATE SHARED KEY!!!

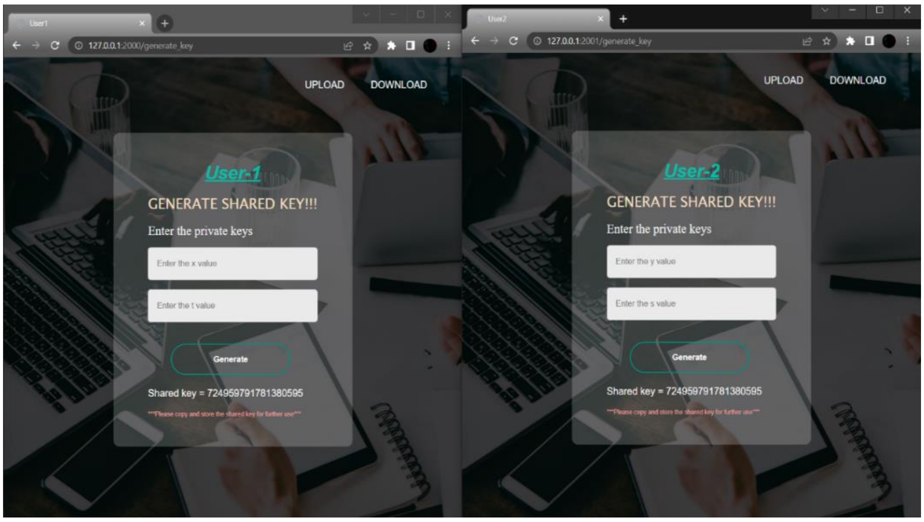
Enter the private keys

Shared key =

***Please copy and store the shared key for further use***

**Fig.2 User interface for generating shared key**

The above Fig.2 represents the web interface for users to generate shared key. Users needs to provide the private keys value as an input and by using modified DHKE final shared key is generated.



User-1

GENERATE SHARED KEY!!!

Enter the private keys

Enter the x value

Enter the t value

Shared key = 724959791781380595

**Please copy and store the shared key for further use**

User-2

GENERATE SHARED KEY!!!

Enter the private keys

Enter the y value

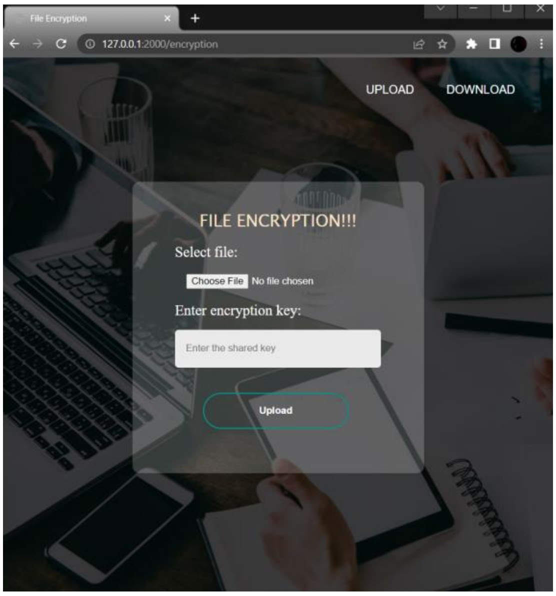
Enter the s value

Shared key = 724959791781380595

**Please copy and store the shared key for further use**

**Fig.3 Shared key generated**

Fig.3 represents the shared key generated and displayed on the both the users web page. By using modified Diffie Hellman key exchange final shared key is generated. Both the users needs to copy and save the shared key for the further use.



FILE ENCRYPTION!!!

Select file:

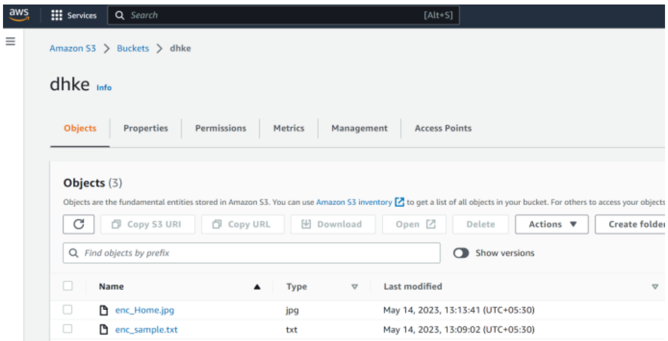
No file chosen

Enter encryption key:

Enter the shared key

**Fig.4 File upload user interface**

Fig.4 represents the user interface for uploading files to the cloud. The interface consists of a form where user needs to select the file that need to be uploaded and user needs to enter the encryption key which is shared key generated in previous step and that key is used in AES encryption technique for encrypting file.



dhke

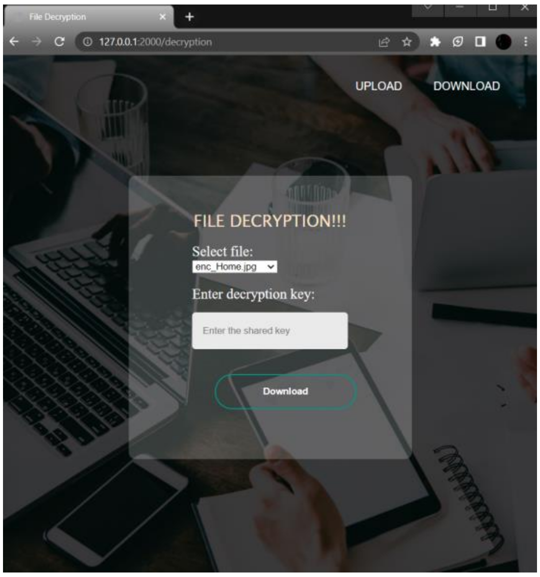
Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects

| | Name | Type | Last modified |
|---|---|---|---|
| □ | enc_Home.jpg | jpg | May 14, 2023, 13:13:41 (UTC+05:30) |
| □ | enc_sample.txt | txt | May 14, 2023, 13:09:02 (UTC+05:30) |

**Fig.5 Uploaded file saved in S3**

Fig.5 represents the files that are stored in the Amazon S3 cloud storage. Before uploading the file to cloud we need to create the Bucket in S3 and we should give the access of the bucket to the users. When the user clicks the upload button in the encryption webpage the file will be encrypted and uploaded to the S3. Uploaded file will be updated in the Amazon S3 as shown in the figure.



127.0.0.1:2000/decryption

FILE DECRYPTION!!!

Select file: enc_Home.jpg

Enter decryption key:

**Fig.6 File download user interface**

Fig.6 represents the user interface for downloading files to the cloud. The interface consists of a form where user needs to select the file that need to be downloaded and user needs to enter the decryption key which is shared key generated in previous step and that key is used in AES decryption technique for decrypting file.

## VI. CONCLUSION

Larger businesses are increasingly adopting cloud computing in greater numbers. Regarding privacy, data integrity, and individualization, cloud computing presents several security concerns. Effective and methodical techniques that may be utilised to increase the security of cloud data are needed. For encryption and decryption operations, the project makes use of the AES (Advanced Encryption Standard) algorithm. The key generation procedure adheres to the DHKE principles, enabling safe key exchange even while a possible spy is present. This project makes it possible to encrypt and decrypt files securely, protecting the confidentiality and integrity of the data. AES is used to guarantee strong encryption, while the DHKE protocol offers a safe way to exchange keys.

As a future work, to maintain the long-term security of encryption keys, implement key management features like key rotation and safe key storage. This might entail studying cryptographic key vaults or connecting with secure key management solutions.

## REFERENCES

[1]. Lata Gadhavi, Madhuri Bhavsar, Monica Bhatnagar, Shivani Vasoya, “Design of Efficient Algorithm for Secured Key Exchange over Cloud Computing””, In proceedings of 6th International Conference - Cloud System and Big Data Engineering (Confluence), IEEE, 2016, pp. 180-187.

[2]. Bindhu Raj L, Vandana R, Santhosh Kumar B J, “Integrity based Authentication and Secure Information Transfer Over Cloud for Hospital Management System”, In proceedings of the International Conference on Intelligent Computing and Control Systems, IEEE, 2020, pp. 139-144.

[3]. Narander Kumar, Jitendra Kumar Samriya, “EAAP: Efficient Authentication Agreement Protocol Policy for Cloud Environment”, In proceedings of International Conference on Next Generation Computing Technologies, Springer, 2019, pp. 311-320.

[4]. Kaustubh Purohit, Avanish Kumar, Mayank Upadhyay, and Krishan Kumar, “Symmetric Key Generation and Distribution Using Diffie-Hellman Algorithm”, In proceedings of Soft Computing: Theories and Applications, Springer, 2020, pp. 135-141.

[5]. Samrat Mitra, Samanwita Das, and Malay Kule, “Prevention of the Man-in-the-Middle Attack on Diffie–Hellman Key Exchange Algorithm: A Review”, In proceedings of International Conference on Frontiers in Computing and Systems, Springer, 2021, pp. 625-635.

[6]. Nazmun Naher, Asaduzzaman and Md. Mokammel Haque, “Authentication of Diffie-Hellman Protocol Against Man-in-the-Middle Attack Using Cryptographically Secure CRC”, In proceedings of International Ethical Hacking Conference, Springer, 2019, pp. 139-150.

[7]. Rokesh Kumar Yarava, Rajendra Prasad Singh, “Efficient and Secure Cloud Storage Auditing Based on the Diffie-Hellman Key Exchange”, In International Journal of Intelligent Engineering and Systems, Vol.12, No.3, pp. 50-58, 2019.

[8]. Durbadal Chattaraj, Monalisa Sarma, Debasis Samanta, “An Efficient Two-Server Authentication and Key Exchange Protocol”, In proceedings of International Conference on New Trends in Computing Sciences, IEEE, 2017, pp. 127-132.

[9]. Arjun Singh Rawat, Maroti Deshmukh, “Efficient Extended Diffie-Hellman Key Exchange Protocol”, In proceedings of International Conference on Computing, Power and Communication Technologies, IEEE, 2019, pp. 447-451.

[10]. M. Somasundara Rao, Dr. K. Venkata Rao, Dr. M.H.M. Krishna Prasad, “Hybrid Security Approach for Database Security using Diffusion based cryptography and Diffie-Hellman key exchange Algorithm”, In proceedings of Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), IEEE, 2021, pp. 1608-1612.