PRIVACY-PRESERVING CIPHERTEXT MULTI-SHARING CONTROL FOR BIG DATA STORAGE

Bindu R[1], Chandana K A[2],Divya R[3] ,Pallavi TS[4],Manjula M[5]

Student, Department of Computer Science & Engineering, Atria Institute of Technology, Bangalore, India[1,2,3,4]
Assistant Professor, Department of Computer Science & Engineering, Atria Institute of Technology ,Bangalore, India[5]

---***---

Abstract -The demand for a safe big data storage service is significantly higher than it has ever been. The services most basic demand is that the data be kept confidential.However, one of the most important features of privacy, the anonymity of service clients, should be taken into account at the same time.Furthermore, the service should enable for realistic and fine-grained encrypted data sharing, so that a data owner can share a cipher text of data with others under certain conditions.For the first time, a privacy-preserving ciphertext multi-sharing mechanism is proposed in this study to achieve the following qualities. It combines the benefits of proxy re-encryption with an anonymous mechanism that allows a cipher text to be safely and conditionally transmitted numerous times without revealing both the underlying message and the senders/recipients' identities.Additionally, this proposed model is more secure against chosen cipher-text attacks.

***Key Words*: Privacy,Multisharing,Ciphertext,Proxy re-encyption.**

1.INTRODUCTION

The growing number of individual users, as well as public and private companies, who opt to store their data in the cloud forces us to make data more secure against hacking. Individual user data should be kept private, with only authorised users having access to it. Before storing data, the most crucial element to consider is the anonymity of the service providers. The data storage services should be able to deliver high-quality encrypted data sharing.These services let data owners to share only the encrypted text of their data with authorised users under certain restricted and predefined situations. The properties listed above are frequently required for safe processing, and they are provided using a novel technology known as ciphertext multi sharing method.A proxy re-encryption approach is used in this method, which allows just the ciphertext to be exchanged safely and conditionally several times. It also assures that the original message and information identity of ciphertext senders and recipients are not exposed, as well as that it is not subject to ciphertext assaults.

Consider the case below. Assume that a hospital maintains its patients medical data in a cloud storage system, and that the records are all encrypted to prevent the cloud server from viewing any of the patient's medical data. Only the doctors named in the record can access it after it has been encrypted and transferred to the cloud.The confidentiality of the record can be adequately secured using standard PKE, Identity-Based Encryption (IBE), or Attribute-Based Encryption (ABE).We cannot, however, prevent certain sensitive personal information from being leaked to the cloud server, as well as the general public, by just using typical encryption measures. This is due to the fact that typical encryption techniques ignore the sender/anonymity receiver's.As a result, someone, who may be anybody with the capacity to get a ciphertext (e.g., a cloud server), may be able to determine whose public key the ciphertext is encrypted under, i.e., who is the owner of the ciphertext, allowing the patient connected with the ciphertext to be readily recognised.Similarly,ciphertext's recipient/destination, such as Neurology Dept., may be determined without effort from the ciphertext. This is an embarrassment to the patient's privacy.

Furthermore, throughout different stages of therapy, a patient may be moved to more than one medical department. The accompanying medical record must subsequently be transformed into ciphertexts for various receivers so that it may be shared between departments. As a result, it is desirable to update the ciphertext recipient.In order for a ciphertext to be conditionally shared with others, a fine-grained ciphertext update for receivers is required. The owner of a medical record, such as a patient, has the authority to select who has access to the record and what data is accessible.For example, a patient might request that just the medical records including the word "eye" be viewed by a Opthalmologiest. This fine-grained control avoids a data sharing method from being restricted to "all-or-nothing" form of sharing.The goal of this project is to address the issues mentioned above. Some well-known encryption algorithms, such as attribute-based encryption or identity-based encryption, have been proposed in the literature to protect anonymity. The source and destination of data can be safeguarded privately using these primitives. The primitives, on the other hand, are unable to enable ciphertext receiver updates

2. PROBLEM STATEMENT

Security is the most important factor for any type of storage services. Because of its efficient data processing capabilities, cloud computing plays a vital role in conserving big data. Many individuals and organisations can see, change, and update their data stored in the cloud using remote access.

There's a risk that a number of common issues, such as privacy and security will arise.

Earlier, for privacy preserving purposes identity-based encryption and PKE techniques were used.It is impossible to keep some secret sensitive information from being leaked to the public and to the cloud server using standard procedures. This is because traditional procedures ignore a ciphertext sender's or receiver's anonymity. As a result, anyone who knows how to get the encrypted text can do so. obtain the text's public key, implying that the hacker will know the owner of the message.

Existing privacy-preserving solutions in Big Data Storage services fail on the following fronts:

- Data confidentiality and anonymity of server clients.
- A trusted third party with knowledge of the data owner's decryption key may be entrusted to handle the task
- Decryption and re-encryption take a long time.
- It will not satisfy all needs of big data storage

.

3. PROPOSED SYSTEM

A concrete construction for unidirectional AMH-IBCPRE is proposed in Proposed System, which provides multiple ciphertext receiver updates, conditional data sharing, anonymity, and collusion-safe (i.e., holding against collusion attacks) in an asymmetric bilinear group at the same time. The functionality of our system is depicted in broad strokes in the Figure . We claim that the new primitive can be used in a variety of situations.

In the proposed system we have considered an example of a hospital database, where doctors act as data owners and nurses, and insurance companies act as data users.

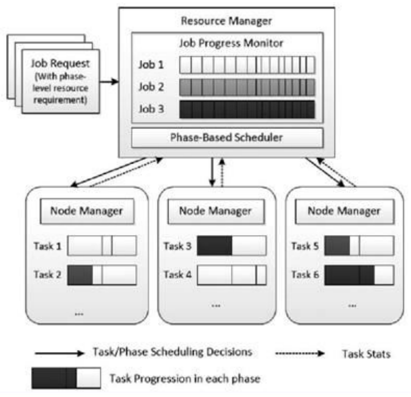

Figure 3.1 Architecture Diagram

3.1 WORKFLOW

A flowchart depicts sequences, decision points, and beginning and ending points. When the power is turned back on, the procedural flow begins with the initialization of the modules concerned.

The project is divided into three modules:

1. User.
2. Cloud server
3. Proxy server

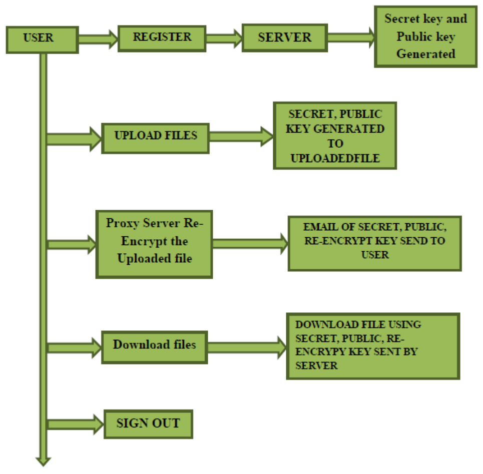
USER
REGISTER
SERVER
Secret key and Public key Generated
UPLOAD FILES
SECRET, PUBLIC KEY GENERATED TO UPLOADEDFILE
Proxy Server Re-Encrypt the Uploaded file
EMAIL OF SECRET, PUBLIC, RE-ENCRYPT KEY SEND TO USER
Download files
DOWNLOAD FILE USING SECRET, PUBLIC, RE-ENCRYPY KEY SENT BY SERVER
SIGN OUT

Figure 3.2 Flow Diagram

1. USER:

DESCRIPTION AND PRIORITY :

The system will check the user check first. If a user is not registered, he or she must first register and login to the server, after which the server will provide access to upload or download files

STIMULUS/RESPONSE SEQUENCE:

When a user requests that the cloud server approve their registration, the cloud server generates both the public and private keys and sends them to the relevant user.

The user logs in to the system with their registered user id and password.

The next user might be given permission to upload the file to the server.When a user uploads a file, the secret key and public key are generated, and an encrypt code is appended to the file name as a prefix.

After creating a group, the user must become an admin. After becoming an admin, the user must upload a file. After uploading the file, the user receives an email with the encrypted file name, secret key, and public key to the user's registered email address.

For the sake of security, we produce both a secret and a public key.

The uploaded file was then sent to the server, but it was unapproved, so we couldn't download it until the server administrator approved it.

2.CLOUD SERVER:

DESCRIPTION AND PRIORITY:

After successfully logging in with a valid username and password, the Cloud Server allows the user to upload or download the file..

STIMULUS/RESPONSE SEQUENCES :

The proxy server retrieves the newly approved users, generates a new set of keys, and saves them in the user account. The Advanced Encryption Standard (AES) Algorithm generates fresh keys.

The server receives the user-uploaded file, and the server administrator can approve it based on certain criteria, such as file size, check for potential space issues, and so on.

This server only allows users to download files that have been approved by the server.

3. PROXY-SERVER :

DESCRIPTION AND PRIORITY :

The Proxy-Server will only function when the main server is unavailable.

STIMULUS/RESPONSE SEQUENCES :

Re-encryption is performed for the aim of increased security. The data supplied by the user is re-encrypted during re-encryption to ensure that security is double-checked.

4. REQUIREMENTS:

4.1 Hardware Requirements:

- System : Intel I5.
- Hard Disk : 120 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB

4.2 Software Requirements:

- Windows 7 is the operating system.
- Java 1.8
- Drive HQ
- MySql is the database.
- Neatbeans is an integrated development environment.

5. ALGORITHMS USED

5.1 ABE(Attribute Based Encryption)

Sahai and Waters were the first to propose attribute based encryption (ABE), which uses public key cryptography to impose access control. Security and access control are the key goals of these approaches. Flexibility, scalability, and fine-grained access control are the most important features. Only when the user and the server are in a trusted domain can this be accomplished in the traditional paradigm.Both the user secret key and the cipher-text are connected with a set of characteristics in the ABE scheme. A user can decode the cipher-text if and only if the cipher-text and the user secret key share at least a certain number of characteristics.
In data sharing applications, ABE promises to give fine-grained access control over encrypted files, allowing the data owner to choose who can access the encrypted data. Key Policy Attribute Based Encryption (KP-ABE) and Ciphertext Policy Attribute-Based Encryption (CPABE) are the two primary types (CP-ABE).

ABE algorithm have four parts:

1) The setup phase, also known as the system initialization phase, is when security settings are entered and public parameters (PK) and master keys (MK) are produced.
2) At the KeyGen step, also called as key generation stage data owners submit their own characteristics to the system in order to get the private key associated with those attributes.
3) During the encryption phase, the data owner uses his or her public key to encrypt the data and obtain the ciphertext (CT), which he or she then transmits to the recipient or to the public cloud.
4) At Decryption phase, users receive ciphertext and use their own private key to decrypt.

5.2 AES Algorithm(Advance Encryption Standard)

It is built on the basis of a substitution–permutation network. It consists of a sequence of connected processes, some of which require substituting particular outputs for inputs (substitutions) and others involving shuffling bits about (permutations).Surprisingly, AES uses bytes rather than bits for all of its calculations. As a result, AES interprets a plaintext block's 128 bits as 16 bytes. For matrix processing, these 16 bytes are organised into four columns and four rows.
In contrast to DES(Data Encryption Standard), the number of rounds in AES is variable and dependent on the key length. For 128-bit keys, AES employs 10 rounds, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Each of these rounds use a unique 128-bit round key derived from the original AES key.

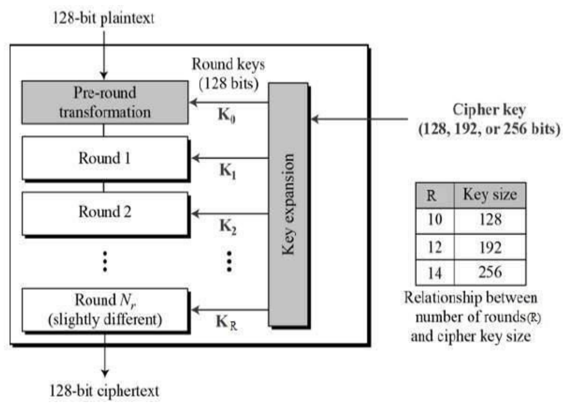

Figure5.2 AES Structure

Encryption Process: We'll stick to describing a typical cycle of AES encryption here. There are four sub-processes in each round. The first round is represented in the diagram below.

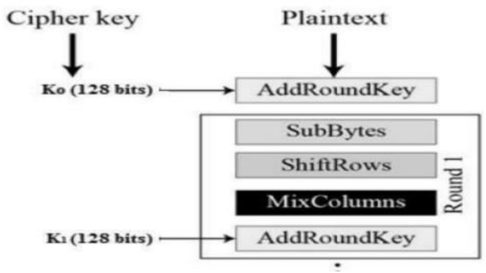

Figure5.3 Encryption process

Byte Substitution

By searching up a fixed table (S-box) provided in design, the 16 input bytes are replaced. The end result is a four-row, four-column matrix.

Shiftrows

Each of the matrix's four rows is moved to the left. Any entries that fall off the right side of the row are reinserted. The following is how the shift is carried out:

- The first row has not been moved.
- The second row has been relocated one byte to the left.
- the third row has been shifted two places to the left.
- The fourth row has been moved three spaces to the left.
- The outcome is a new matrix made up of the same 16 bytes that have been shifted in relation to each other.

MixColumns

A specific mathematical function is now used to convert each column of four bytes. This method takes four bytes from one column as input and returns four entirely new bytes that replace the original column. As a consequence, a new matrix with 16 additional bytes is created. It's worth noting that this stage is skipped in the final round.

Addroundkey

The matrix's 16 bytes are now treated as 128 bits, and they are XORed with the round key's 128 bits. The ciphertext is the output if this is the final round. Otherwise, the 128 bits are interpreted as 16 bytes, and the process repeats again.

Decryption Process

The reverse process of decrypting an AES ciphertext is identical to the reverse process of encryption. Each round consists of the four operations in reverse order:
Add round key ,Mix columns, Shift rows,Substitution of bytes

6. CONCLUSION

This study introduces the Advanced Encryption Standard (AES) algorithm, which combines several rounds of substitution-permutation, or SP network, to produce cipher text for secure data transfer in cloud computing.

This research focused on the recipient's anonymity, multiple recipient - update and multiple cipher text of recipient, which are essential for securing some secret information while delivering it.

This technique also guarantees data exchange consistency and efficiency in a time-consuming and cost-effective manner. It also took steps to protect against specific cipher text attacks.

5. REFERENCES

[1] G. Ateniese, K. Benson, and S. Hohenberger. Key-private proxy encryption. In CT-RSA '09, vol. 5473 of LNCS, pp. 279–294. Springer, 2009.
[2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In NDSS '05, pp. 29–43. Springer, 2005.
[3] X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In CRYPTO, vol. 4117 of LNCS, pp. 290–307. Springer, 2006.
[4] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. ACM TISSEC, 9(1):1–30, 2006.
[5] M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In PKC, vol. 4450 of LNCS, pp. 201–216. Springer, 2007.
[6] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In EUROCRYPT '98, pp. 127–144. Springer, 1998.
[7] D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In EUROCRYPT '04, vol. 3027 of LNCS, pp. 223–238. Springer, 2004.
[8] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In EUROCRYPT '05, vol. 3494 of LNCS, pp. 440–456. Springer, 2005.
[9] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In PKC, vol. 5443 of LNCS, pp. 196–214. Springer, 2009.
[10] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In Eurocrypt '04, vol. 3027 of LNCS, pp. 207–222. Springer, 2004.

[11] R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re encryption. In CCS, pp. 185–194. ACM, 2007.
[12]A. Sahai and B. Waters, ‘‘Fuzzy identity-based encryption,’’ in Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2005, pp. 457–473.