© May 2024| IJIRT | Volume 10 Issue 12 | ISSN: 2349-6002

Secure File Storage using Hybrid Cryptography

Ishika Shroff[1], Kabeer Gupta[2], Kartik Ghanani[3]
[1,2,3]*Dept. of Computer Science and Information Technology Indore, India*

Abstract: Conducting a comprehensive comparative study of all cryptographic algorithms for secure file storage using hybrid cryptography involves analyzing various aspects of each algorithm. The choice of cryptographic algorithms depends on factors such as security, performance, and suitability for specific use cases.

Aim: The aim is to do comparative study on selection and implementation of cryptographic algorithms on a file storage system.

Objectives: The study aims to achieve these key objectives:

- **Security Assessment: Determine the cryptographic algorithms' advantages and disadvantages as well as how well they withstand known cyberattacks.**
- **Performance Analysis: Examine the cryptography algorithms' performance characteristics, taking into account things like resource usage, computational overhead, and the speed at which data is encrypted and decrypted.**
- **Suitability for Specific Use Cases: Examine how well cryptographic techniques fit the requirements of safe file storage.**
- **Hybrid Cryptography Integration: Examine whether incorporating hybrid cryptography— which mixes symmetric and asymmetric encryption—for secure file storage is feasible and effective. Examine how hybrid cryptography provides an efficient solution to the shortcomings of individual algorithms.**
- **Key Management Considerations: Examine each algorithm's key management features, such as key creation, distribution, and storage. Assess the degree to which each algorithm facilitates safe and efficient key management procedures.**

Keywords: Here a
re some potential keywords for the research paper on study on Secure File Storage

- **Secure File Storage**
- **Encryption**
- **Cloud Storage Security**
- **Access Control**
- **Authentication**
- **Data Integrity**
- **Compliance**
- **Secure File Deletion**
- **Audit Trails**
- **Data Loss Prevention (DLP)**
- **Risk Management**
- **Mobile Security**
- **User Education and Training**

INTRODUCTION

OVERVIEW: Cloud computing is a popular technique that enables users to access files, storage, software, and servers through their internet-connected devices such as smartphones, tablets, computers, and wearables. Cloud storage is widely used by organizations such as universities and military institutions due to its efficiency. However, there are some security and privacy concerns related to cloud computing. Data stored in the cloud can be accessed by cloud providers and can be easily altered. Additionally, the stored data could also be shared with third parties if necessary as per the privacy policy. Data leaks are also one of the potential risks associated with cloud storage. To address these issues, cryptographic solutions are implemented to ensure data security and privacy. Cryptography uses algorithms to encrypt and decrypt data to prevent unauthorized access. Transport Layer Security (TLS) or Secure Socket Layer (SSL) is a standard protocol that ensures the transfer of data between server and client over the internet remains secure. Authentication and non-repudiation are other security objectives besides confidentiality that are also taken into consideration. This paper proposes a hybrid cryptography scheme in a cloud environment that combines symmetric key cryptography and asymmetric key cryptography to improve data security and reduce risk. A combined hybrid algorithm can lead to better security and efficiency in cloud computing.

PROBLEM STATEMENT

I completely agree with your concerns regarding the security of information stored on the cloud. It is indeed a crucial issue that needs to be addressed. The confidential nature of the data stored on the cloud makes it essential to ensure that the data can only be accessed by authorised users.

Data integrity is another significant problem that needs to be addressed. Data corruption can occur during data processing or transmission, and it can result in unexpected results to the authorised user when they access the data. It is, therefore, crucial to ensure the accuracy and integrity of the data using common methods such as hashing.

The authentication process is also a major security concern in cloud computing. Cyber attacks such as phishing and credential theft can expose the data stored in the cloud if the hackers possess the credentials of the authorised users. It is imperative to have robust cloud authentication procedures that verify the identity of the users effectively and protect the cloud applications from such security concerns.

Literature Review- Through the prism of hybrid cryptography—a combination of symmetric and asymmetric encryption techniques—this literature review investigates the field of secure file storage. The text explores the essential ideas of cryptography, clarifying the benefits and drawbacks of both symmetric and asymmetric encryption, and highlighting the need for encryption in file storage.

The study ends with a discussion of the obstacles and future possibilities for research, highlighting the significance of ongoing innovation to improve the efficiency and security of secure file storage cryptography.

Information about algorithms tested: Symmetric encryption is a type of cryptographic techniques known as symmetric encryption employs the same key for data encryption and decryption. A single secret key is used in symmetric encryption to reverse the process of converting ciphertext back into plaintext and to turn plaintext into ciphertext.

Key Concepts in Symmetric Encryption:

- Key Generation: To ensure safe communication, a secret key is created and shared among the participants. When entities need to share encrypted data, the key needs to be kept private.
- Encryption Process: The communication is converted from plaintext to ciphertext by the sender using the secret key to encrypt it. The encryption algorithm generates the ciphertext by using the secret key and the plaintext as inputs.
- Decryption Process: The ciphertext is decrypted by the recipient using the same secret key, restoring the original plaintext. The plaintext is generated by the decryption algorithm using the ciphertext and secret key as inputs.
- Security Relies on the Secrecy of the Key: The confidentiality of the key determines how strong symmetric encryption is. An unauthorized person can access the original data by decrypting the ciphertext if they manage to obtain the key.

Symmetric Encryption Algorithms:

- Data Encryption Standard (DES): Developed in the 1970s, DES is a historic symmetric encryption algorithm that uses a 56-bit key. Due to its short key length, it is considered insecure for modern applications.
- Triple DES: An improvement upon DES that applies the DES algorithm three times to each data block. It uses either two or three keys for enhanced security. Triple DES is still used in legacy systems, but it is being phased out in favor of more modern algorithms.
- Advanced Encryption Standard (AES): AES is widely adopted as a secure replacement for DES and Triple DES. It offers key sizes of 128, 192, or 256 bits, and is efficient and secure, suitable for a broad range of applications.
- Blowfish: These are block ciphers designed for efficient encryption and decryption. Blowfish has a variable key length, while Twofish operates on 128-bit blocks. These algorithms provide alternatives to AES in certain contexts.

Asymmetric Encryption: Asymmetric encryption is a cryptographic system that employs a pair of keys, a public key and a private key, for safe communication. Asymmetric encryption uses two mathematically related but different keys, in contrast to symmetric encryption, which uses the same key for both encryption and decryption.

Key Concepts in Asymmetric Encryption:

Key Generation: Every communication entity possesses a set of two keys, a public key and a private key. While the private key is kept confidential, the public key is shared with everyone.

Encryption Process: The plaintext communication is encrypted by the sender using the recipient's public key. The recipient's public key and the plaintext are the inputs used by the encryption algorithm to create the ciphertext.

Decryption Process: The original plaintext can be recovered by the recipient using their private key to decrypt the ciphertext. The recipient's private key and the ciphertext are inputs to the decryption algorithm, which outputs the plaintext. Security Relies on the Complexity of Mathematical Problems: The security of asymmetric encryption depends on the intricacy of mathematical difficulties, such as factoring big composite numbers or figuring out discrete logarithms.

Asymmetric Encryption Algorithms:
RSA (Rivest–Shamir–Adleman): The difficulty of factoring the product of two large prime numbers provides security. mostly used for digital signatures, key exchange, and communication security.

Elliptic Curve Cryptography (ECC): Compared to RSA, Elliptic Curve Cryptography (ECC) uses the principles of elliptic curves to offer comparable security with shorter key lengths. Greater computing efficiency qualifies it for resource-constrained contexts, such as the Internet of Things and mobile devices.

Hybrid cryptography is a cryptographic methodology that uses symmetric and asymmetric encryption methods to effectively address data security and secure communication. In hybrid cryptography, symmetric encryption is used to encrypt data, and asymmetric encryption is used to exchange keys. This hybrid solution tackles some of the shortcomings of both symmetric and asymmetric encryption by striking a balance between security and processing efficiency.

Key Components of Hybrid Cryptography:
Symmetric Encryption: Because it is quick and easy to use, symmetric encryption works well for encrypting large amounts of data. Both encryption and decryption are accomplished using the same secret key. The Advanced Encryption Standard (AES) and Triple Data Encryption Standard (TDES) are two popular symmetric encryption methods.

Asymmetric Encryption: Two sets of keys are needed for asymmetric encryption: public and private. While the private key is kept confidential, the public key is shared with everyone. Asymmetric encryption techniques that are frequently used are Diffie-Hellman key exchange, RSA, and Elliptic Curve Cryptography (ECC).

AES- The Advanced Encryption Standard (AES) is a symmetric encryption algorithm widely used for securing sensitive data. AES operates on blocks of data and supports key sizes of 128, 192, or 256 bits. Here's a high-level overview of how the AES algorithm works:

Rounds of Encryption:
First Round: The first round key is XORed with the input block (plaintext).

Principal Rounds: The number of rounds in the encryption process is determined by the size of the key.
A 128-bit key is worth ten rounds.
A 192-bit key is worth 12 rounds. for a
256-bit key, 14 rounds.
The four primary operations in each round are Add Round Key, Mix Columns, Sub Bytes, and Shift Rows.

a. Sub Bytes: A corresponding byte from the S-box, a predetermined substitution table, is used to replace each byte in the block.
The algorithm now has non-linearity as a result.

b. Shift Rows: Each row's bytes are moved to the left. There is no shift in the first row, one byte shift in the second, two byte shift in the third, and three byte shift in the fourth.
The diffusion across rows is provided by this stage.

c. Mix Columns: A linear transformation involving matrix multiplication with a fixed polynomial is used to mix the columns.
This stage makes sure that every byte is dependent on several other bytes from other columns.

d. Add Round Key: The state (the result of the Mix Columns step's output) and the round key are XORed.
The current round key is introduced into the state in this stage.

Final Round: The Mix Columns procedure is not used in the final round, which is otherwise identical to the preceding rounds. Add Round Key, Shift Rows, and Sub Bytes make up this structure.

Decryption: In AES, decryption is the opposite of encryption. The decryption algorithm reverse engineers the Sub Bytes, Shift Rows, Mix Columns, and Add

Round Key inverse operations in reverse order using an inverse key schedule.

Output: The plaintext for decryption and the ciphertext for encryption make up the final output.

DES: Due to its tiny key size, the Data Encryption Standard (DES), a symmetric key block cipher that was once widely used for encryption, is today regarded as unsecure for use in major cryptographic applications. DES uses a 56-bit key and works with 64-bit data blocks. An outline of the DES algorithm's operation is provided below:
Important Generating Important Compression and Permutation: Initially, the 56-bit key is compressed and permuted to produce two 28-bit halves (C0 and D0). The rounds of key creation that follow make use of these halves.
Key Schedule: The sixteen subkeys that make up the 56-bit key are individually generated by rotating, permuting, and compressing the original key.
K1, K2,..., K16 are the 16 48-bit subkeys generated by this key schedule.

Rounds of Encryption:
The first permutation (IP): The first permutation is applied to the 64-bit plaintext block. Block components are reconfigured using a predetermined permutation table.

Feistel Network: The 64-bit block in DES is split into two 32-bit halves (L0 and R0) using a Feistel network topology. The Feistel function includes four steps: substitution using S-boxes, XOR with a subkey, expansion of the R half to 48 bits, and permutation. To get the next R, the output of the Feistel function is XORed with the L half.

Feistel Operations: 16 Rounds: The Feistel function is performed 16 times, utilizing a distinct subkey produced by the key schedule in each round. After every round, the L and R halves are switched.
Inverse Initial Permutation (IP-1): The inverse initial permutation is performed to the final L and R halves after 16 rounds of combination. The 64-bit ciphertext block is the result of this.
A more secure version of the original Data Encryption Standard (DES) is offered by 3DES, sometimes referred to as TDEA or Triple DES, which is a symmetric key block cipher. 3DES uses a total key length of 168 bits (three 56-bit DES keys) to apply the DES algorithm three times to each data block. This is a summary of how the 3DES algorithm works.
Keying Options:
3DES can be utilized with several keying configurations:

2-Key 3DES (2TDEA): This algorithm applies the DES algorithm twice using two separate 56-bit keys.

Three-Key 3DES (3TDEA): This algorithm applies the DES algorithm three times using three separate 56bit keys.

Initial Permutation (IP): Encryption The first permutation is applied to the 64-bit plaintext block.

Three DES Operational Rounds: In 2TDEA, the plaintext block is first encrypted using the first key, then it is decrypted using the second key, and finally it is encrypted once more using the first key.
In 3TDEA, the first key encrypts the plaintext block, the second key decrypts it, and the third key encrypts it once more.

Inverse Initial Permutation (IP-1): Following three rounds of DES operations, the ciphertext block is the outcome, and the inverse initial permutation is applied.

Decryption: For 2TDEA, the ciphertext block is first encrypted with the second key, then decrypted with the third key. This is the opposite procedure of 3DES decryption.
The ciphertext block for 3TDEA is first encrypted with the second key, decrypted with the third key, and finally encrypted with the first key.

Blowfish: This versatile method is appropriate for a range of applications since it works with variable length blocks (32 to 448 bits) and accepts key lengths between 32 and 448 bits. One of the things that sets Blowfish apart is its key scheduling, which is implemented using a Feistel network structure. An outline of the Blowfish algorithm's operation is provided below:

Initialization: P-array and S-box setup is where Blowfish starts.
There are multiple 32-bit words in the S-boxes and eighteen 32-bit subkeys in the P-array.

Key Expansion: To establish a baseline state, the first P-array and S-boxes are XORed with pi's hexadecimal digits (3.141592653589793).
The input key is then used to modify the subkeys iteratively.

Encryption:

Initialization: The plaintext block is divided into two 32-bit halves, left (L) and right (R).
16 Rounds of Feistel Operations:
Blowfish uses a Feistel network structure with 16 rounds of operations.
Each round consists of the following steps:
Substitution (S-boxes): The S-boxes are used to substitute the values of the right half (R).
XOR with P-array: The result is XORed with the left half (L).

Swap: The left and right halves are swapped.
Final Swap: After 16 rounds, a final swap is performed to ensure that the left and right halves are in the correct order.

Decryption: Using the subkeys in the opposite sequence from encryption, Blowfish's decryption procedure is basically the same as encryption. The subkeys are applied in reverse order, but the ciphertext block still goes through the same 16 rounds of Feistel operations.

RSA: The RSA algorithm is a popular asymmetric key scheme for digital signatures and secure data transmission. The security of RSA is predicated on the difficulty of factoring the product of two large prime numbers, and it requires the use of public and private key pairs. An outline of the RSA algorithm's operation is provided below:

Key Generation:
Choose p and q, two huge prime numbers.
Determine $n = p \times q$ and $\phi(n) = (p-1) \times (q-1)$.
Determine the private exponent d and select a public exponent e.
Using encryption, show the message as m.
With the public key, compute $c \equiv me \pmod{n}$.
Decryption: With the private key, calculate $m \equiv cd \pmod{n}$.

ECC-
The Elliptic Curve Cryptography (ECC) algorithm is a form of public-key cryptography that uses elliptic curves over finite fields. Compared to traditional public-key algorithms, ECC provides strong security with shorter key lengths. Here's an overview of how the ECC algorithm works: An elliptic curve is a plane curve with the equation y^2 = x^3 + ax + b, where a and b are constants, and x and y are variables. Due to their unique mathematical properties, elliptic curves are ideal for cryptography. For instance, given two points P and Q on an elliptic curve, there is a third point R such that P + Q = R. This property is known as "point addition." Another valuable property of elliptic curves is "point doubling." This involves taking a point P on the curve and finding another point, 2P, such that P + P = 2P. We can repeat this process of doubling points until we reach the "infinity point," which we refer to as O. The infinity point represents the limit where the distance between P and 2P approaches zero. Thus, we can add and double points on an elliptic curve endlessly and never get the same result twice (except for O).
When any point P is given on an elliptic curve, an infinite number of points can be obtained by adding and doubling P, including O. As a result, an endless number of possible keys can be generated using elliptic curve.

COMPARATIVE ANALYSIS

Algorithm/ parameter	AES	DES	3DES	Blowfish	RSA	ECC
Type	symmetric	symmetric	symmetric	symmetric	asymmetric	asymmetric
Key size (in bits)	128,192,256	56	168,112	32-448	1024	160
Rounds of encryption	10,12,14	16	48	16	1	16
Block size (in bits)	128	64	64	64	Minimum 512	64
Effectiveness	High	Moderate	Low	Low	High	Low
Security	Sufficient	Not sufficient	Not sufficient	Least secure	Least secure	Sufficient

Power consumed	Low	Low	Low	Low	High	Low
Advantages	High security and key size	High no of rounds	Easy implementation	More efficient thanDES and 3DES	Private key computation infeasible from public key	Uses less resources
Disadvantage	Difficult implementation	Short key size	Low efficiency due to 48 rounds	Prone to birthday attacks	Slow computation	Complex compared to RSA

METHODOLOGIES OR PROBLEM SOLVING

Customers' stored data at cloud service providers are vulnerable to various threats. In this work, we consider four types of threat models.

The first is the single point of failure, which affects the data availability if a server at the cloud service provider fails or crashes, making it harder for the customer to retrieve their stored data from the server. The availability of data is also an important issue that could be affected if the cloud service provider shuts down their service.

Our second threat is data integrity. Integrity is the degree of confidence that the data in the cloud is what is supposed to be there, and is protected against accidental or intentional alteration without authorization. Such worries are no longer beneficial issues; therefore, a cloud service customer cannot entirely rely on a cloud service provider to ensure the storage of their data. Security is a necessary service for both wired networks and wireless network communication to improve what is offered in the cloud. Most of the businesses that have held back from adopting the cloud have done so due to the fear of having their data leaked. This fear stems from the fact that the cloud is a multi-user environment where all the resources are shared. It is also a third-party service, which means that data is potentially at risk of being viewed or mishandled by the provider. Several external threats can lead to data leakage, including cloud provider hacking or compromised cloud user accounts. The best strategy is to depend on file encryption and stronger passwords, instead of relying solely on the cloud service provider.

CONCLUSION

After studying all the trending hybrid cryptographic models, we concluded that data security is the most significant topic related to cloud computing technology. To overcome security limitations, we employed the integration between symmetric and asymmetric cryptosystems. By applying the hybrid of different encryption algorithms such as DES, 3DES, AES, Blowfish, RSA, and SHA, we would try to secure sensitive data on the cloud. Our study also concludes that hybrid cryptography enhances performance and adds more security levels to the data compared to applying these algorithms individually. All examined studies have benefits and some drawbacks.

ACKNOWLEDGEMENT

We express our gratitude to all the participants who took part in this study and provided valuable insights into various cryptographic algorithms. We also extend our thanks to our institution, which helped in the data collection process.

We would like to acknowledge the contributions of our research team who assisted with data collection and analysis. Their hard work and dedication were critical to the success of this study. Finally, we would like to express our gratitude to our families, friends, and colleagues for their support and encouragement throughout the research process. Without their support, this study would not have been possible.

REFERENCE

[1] Aws.amazon.com “AWS Cloud Security,”: https://aws.amazon.com/security/?nc1=f_cc

[2] Madhumala RB et al, International Journal of Computer Science and Mobile Computing, Vol.10 Issue.5, May- 2021, pg. 49-59

[3] 1Student, School of Computer Science & Engineering, Vellore Institute of Technology, Tamil Nadu, India 2Assistant Professor, School of Computer Science & Engineering, Vellore Institute of Technology, Tamil Nadu, India

[4] Singh V, Pandey SK (2020) Cloud computing: vulnerability and threat indications. In: Performance management of integrated systems and its applications in software engineering Springer, Singapore, pp 11–20.

[5] Madhu Sharma, Dr. Ashish Sharma, A secret file sharing scheme with chaos-based encryption. Institute of Electrical and Electronics Engineers.2019.