Design and implementation of hybrid Cryptographic systems for File security

[1]MEGHA, [2]GEETA GARAPATI, [3]MAYALURI KOWSALYA,
[4]VAKKALAGADDA BHARGAVI, [5]Dr.S. Neduncheliyan

Department of Computer Science and Engineering,
Bharath Institute of Science & Technology affiliated to Bharath Institute of Higher Education and Research,
Chennai, Tamilnadu, India

***Abstract*- To ensure private communication between parties, hybrid cryptography combines the best features of symmetric and asymmetric encryption techniques. An encrypted symmetric key is created for data transmission and then decrypted with the recipient's public key. After receiving the encrypted message, the recipient may use their private key to decode the symmetric key and access the original message. This method eliminates the necessity for a private key exchange channel by keeping the private key secret at all times. Hybrid cryptography is commonly employed in today's communication networks to protect the privacy and integrity of sensitive information during online transactions, instant messaging, and electronic mail. Key management is essential to the security of the system because, despite the fact that hybrid cryptography gives excellent security assurances, it is subject to assaults if the private key is compromised.**

***Keywords*- Secure Socket Layer (SSL), Transport Layer Security (TLS), Key Encapsulation Mechanism (KEM)**

I INTRODUCTION

In today's digital world, private conversations are more crucial than ever. Cryptography, the technique of secure communication in the presence of third parties, is one approach to guarantee confidentiality in online interactions. There are two primary methods upon which cryptography is based: symmetric key cryptography and asymmetric key cryptography.

In symmetric key cryptography, a single key serves as both the encryption and decryption mechanisms. Even if it's quick and effective, getting the secret key to everyone who needs it safely is a major issue. On the other hand, public and private keys are used in asymmetric key cryptography. While it effectively eliminates the issue of key distribution, it is computationally intensive.

Hybrid cryptography is a method for ensuring confidential communication that draws on the best features of both symmetric and asymmetric key encryption. For data encryption, hybrid cryptography employs symmetric encryption, while asymmetric cryptography is utilized for key exchange. To exchange keys securely, this method does not require a secure channel. Hybrid cryptography is commonly employed in today's communication networks to protect the privacy and integrity of sensitive information during online transactions, instant messaging, and electronic mail. The purpose of this paper is to introduce readers to hybrid cryptography and to discuss its benefits, drawbacks, and practical uses in today's advanced communication networks.

II LITERATURE REVIEW

[1] In order to offer more security for the owner's data than any single symmetric encryption algorithm, Shweta Kaushik proposed a hybrid symmetric encryption approach in this paper. This hybrid strategy secures data and guards against any nefarious actions taken by an intruder. The suggested method also makes brute force attacks impossible. Symmetric encryption is used to increase processing power because it is quicker and more effective than asymmetric encryption.

[2] Data owners are still hesitant to store their data with a third party, despite the fact that cloud-based services offer a number of advantages. The main issues with outsourced data are non-repudiation, confidentiality, integrity, and privacy. For the purpose of securing data transfer between users and the cloud, many conventional security measures are proposed. A hybrid encryption method for image security is put forth in this paper. The DES and AES algorithms use the secret key that was generated by the scheme using elliptic curve cryptography.

[3] To keep sensitive information safe from unauthorized users, many encryption techniques are available. Only authorized users are able to access the data, and encryption and decryption techniques are used to protect the data. The brute force approach, though, can occasionally find hidden data. By combining AES, proxy re-encryption, and Honey encryption, a suggested method is used to improve data confidentiality and authentication issues. The system improves the security of data that is outsourced. Only messages that seem plausible can be accessed by unauthorized users thanks to honey encryption and hybrid cryptography.

[4] There are numerous conventional security methods that can be used to secure data transfer between users and the media cloud. Security lapses do still happen occasionally, though. In this essay, security issues are covered. Pallavi Kulkarni suggests using a hybrid encryption method to secure the images. The DES and AES algorithms use the secret key that was generated by the scheme using elliptic curve cryptography.

[5] In this paper, hybrid cryptography research from 2015 to the beginning of 2019 has been done. In this study, problem-related papers were searched, and 20 or so were taken into consideration after filtering. Twelve of these are in-depth surveys, while the other eight are based on a user-friendly tabular survey. This review paper's main objective is to continuously enlighten new researchers, students studying these subject, and non-cryptographers. The failure to include user authentication and the improper application of hybrid algorithms are the identified research gaps.

[6] A multilevel cryptography-based cloud computing security system was proposed in this paper. Algorithms for symmetric and asymmetric key cryptography are combined in the model. This implementation increases the security of cloud storage by offering multilevel encryption and decryption at both the sender and receiver sides using the Data Encryption Standard (DES) and RSA. This security model offers both cloud users and cloud service providers transparency in order to lessen security threats. The suggested model makes use of the cloud Sim cloud simulator and is written in Java. When compared to the current system, this model maximizes data security and speeds up the uploading and downloading of text files.

[7] In this paper, the user offers some background information on those kinds of works, focusing primarily on the AES and FHE hybrid approach. This hybrid approach allows the user to maintain data that is more redundant and secure than other methods. Users can protect data integrity, personal privacy, and privacy from hackers by using this technique.

[8] In order to develop a thorough security framework for electronic health records, this paper focuses on the research challenges and directions in cyber security (EHRs). The following tasks were identified by the researchers through surveys, investigations, and reviews of various articles: EHR security and privacy are covered in the following four categories: 1) EHR cloud architecture; 2) e-health data in the cloud security and privacy requirements; and 3) various EHR cryptographic and non-cryptographic approaches. They discuss some important issues as well as the many possibilities for cutting-edge research in the area of EHR security and privacy. Big data offers e-Health applications a wealth of knowledge and information, so serious privacy and security issues demand immediate attention.

[9] P. Chinnasamy developed a secure cloud storage system for medical data using a hybrid cryptographic method. Asymmetric encryption is used to encrypt the keys while symmetric encryption is used to encrypt the data. Performance and security of the suggested method were evaluated and contrasted with a method already in use. The outcomes unequivocally demonstrate that in terms of security, this method outperforms all other hybrid algorithms.

[10] According to the paper's overview of cloud computing security issues, fully homomorphic encryption has disadvantages that make it unsuitable for secure cloud computing, including a large key size and poor calculation efficiency. The researchers create a hybrid homomorphic encryption scheme based on the multiplicative homomorphic RSA algorithm and the additively (single bit) homomorphic GM encryption algorithm. The speed of this hybridization of homomorphic encryption algorithms was increased by 2.9 times, the computation time was decreased by 66% compared to the prior one, and the security of the data stored in the cloud was improved (two layers of encryption methods used).

III. PROPOSED SYSTEM

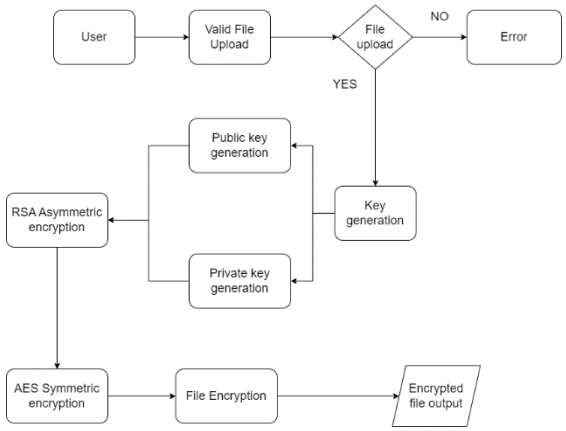
Many crucial parts make up the framework for hybrid cryptography that has been presented. In the first step, the system would have the sender and receiver create a secret key together using a secure key exchange protocol like the Diffie-Hellman key exchange. Symmetric encryption of the sent data would then be performed using the shared secret key.

The sender would then encrypt the secret key with the recipient's public key. A message including both the encrypted shared secret key and the encrypted data would then be sent to the receiver. While receiving encrypted data, the recipient would first use their private key to decode the shared secret key, and only then would they be able to use the shared secret key to decrypt the contents.

The system's security would be ensured by using robust encryption methods, such as Advanced Encryption Standard (AES), for symmetric encryption, and by safely storing and managing the private keys used for this purpose. In addition, the system would take necessary precautions to safeguard the privacy and authenticity of the information being communicated, warding off threats like Man-in-the-Middle assaults.

Together, the use of safe key exchange protocols, symmetric encryption, asymmetric encryption, and other relevant security measures would provide for a robust hybrid cryptography system capable of ensuring the confidentiality of communications between participants.

SYSTEM ARCHITECTURE



IV. MODULES

Module 1: Fernet Algorithm

An encrypted message sent via Fernet cannot be decrypted or manipulated without the key. Fernet is an example of a symmetric verified cryptography implementation. When it comes to cryptography, fernet and fernet are in the same class; they both offer encoding and decoding services (key).
The encrypt strategy is used to encrypt data before the produce key() class method generates a new fernet key (data). This encryption results in a secure transmission that can't be deciphered or altered without the key. It has excellent guarantees of security and authenticity, is base64-encoded, and may be used in URLs. The term "Fernet token" is used to describe it. You might think of the boundary of the message you wish to conceal as the data that must be in bytes, or else we'll have a "type error."

A prospective attacker may tell when a message was generated since the time of its creation is included in plaintext in an encrypted conversation.
The data is securely encrypted at the current time using the function encrypt at time(data, current time). This method may be used to enable testing for token expiration in the client code. It is important to always specify the correct time (int(time,time())) outside of testing, as this function may be used in a variety of circumstances.
decrypt(token, ttl=None) decrypts a Fernet token. As soon as the original plaintext has been decoded and returned, a unique case is normally raised. For this reason, the encrypt() method requires the boundary token, the only fernet token it generates, to be in byte format. TTL (int) is Discretionary; it determines the amount of time in seconds after which a message is no more urgent.

If the duration of the message in seconds exceeds the maximum allowed, an error will be thrown. Without the ttl field, the message's age is not considered.
AES in CBS mode with PKCS7 padding, a timestamp, and message markings using HMAC and SHA256 are all used by Fernet to make encryption more secure. Fernet gets around many of the problems and blunders that would be evident to an inexperienced engineer by offering a safe method for producing keys (a key is like a secret phrase) and by selecting a safe encryption algorithm and taking various other precautions.

Module 2: RSA Algorithm

In asymmetric cryptography, the RSA calculation is utilised. Asymmetric refers to the use of two distinct keys, such as the Public Key and Secret Key. The Public Key is shared openly whereas the Confidential Key remains private, as suggested by their respective names. The private key is used for decryption while the public key is used for encryption.
The first step is to generate the public and private keys. The keys, both public and private, will be saved to files. To store the keys safely with the files, we'll create a Keys envelope in our project planner. You'll be able to keep track of two separate keys—one private and one public—on the Keys organizer's two separate pages. As such, the next step should be to stack the keys. We return both the private and public keys after decrypting the previously generated data, which causes the keys to be stacked.

Create two separate methods of protecting data from being read and seen. First encrypt the encryption method (message, key). In order to encrypt a communication, both the message and the encryption key are required. Next, we'll go over the encryption technique, and then we'll send you the secret message. We'll supply both the key and the ASCII representation of the message. Afterwards, we use the decoding procedure to read it (ciphertext, key).
The approach relies on ciphertext and a decoding technique. We'll do our best to decipher the message and get you the translated version. As we utilized ASCII encoding, we will also employ ASCII decoding.

Two ways for signing and validating our message utilizing a key and sha1 hash capability will be developed at the end of this section. If we use this strategy, not only can we sign the message, but also the key that went along with it. To decipher the message, we'll apply our hashing technique and the key. Namely, SHA-1. While signing, we utilize a technique based on signs (message, key). After preparing the message, the checkmark, and the key, we will implement a confirm technique to verify the message. While the hash computation utilized in the mark is revealed by this check method, our goal is to use it to see if our message is authentic. We verify that this result is the same as the one produced by the SHA-1 hash function. If the logo is authentic, the information will be correct. If a unique situation occurs, it will report false, meaning the verification was unsuccessful. If the message or the mark was manipulated, then it is false.

Module 3: Two Tier Algorithm

In the two-tiered system, we employ an asymmetric code (RSA) to protect the security of the secret key and a symmetric code (Fernet) to protect the privacy of the data as a whole. Once the mystery symmetric key and RSA key pair have been generated, the information document is encrypted using fernet figures. Soon after, the secret key encryption is completed with the help of the RSA number and the public key. For further document decoding on the back end, the symmetric mystery key is retrieved with the RSA using the private key.

Module 4: THREE TIER MODEL

As part of the three-tiered approach, we've looked into including RSA as well as a comparable Fernet symmetric code for bidirectional encryption. Assuming a three-tiered structure, we generate an RSA key and two unknown symmetric keys.

Then the information is encrypted twice in rapid succession using the fernet figure.
The final step in the key exemplification process involves utilizing the public key to encrypt the secret keys using the RSA digest. Lastly, encrypted data and keys are both sent over the cloud. The receiver must initially decode the encoded data twice in order to retrieve the first record, but once they do, they will have the secret keys.

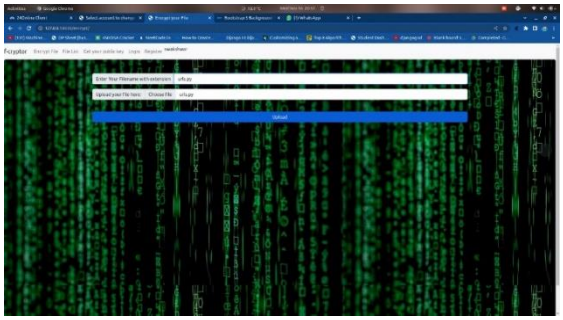
PERFORMANCE ANALYSIS

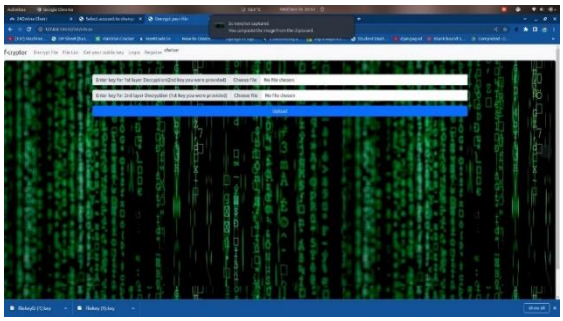
File Size	Encryption Time	Decryption Time
1 KB	0.002 seconds	0.001 seconds
10 KB	0.005 seconds	0.002 seconds
100 KB	0.035 seconds	0.011 seconds
1 MB	0.332 seconds	0.114 seconds
10 MB	3.385 seconds	1.146 seconds
100 MB	34.249 seconds	11.646 seconds
1 GB	345.236 seconds	114.678 seconds

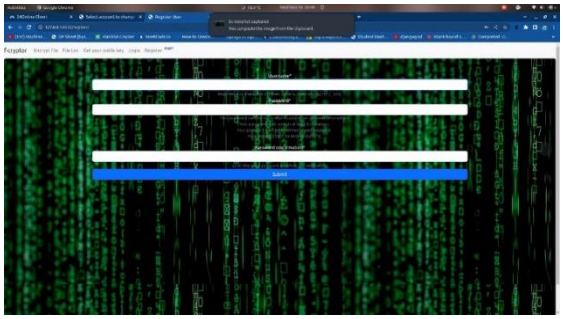
Fig 1: Performance analysis

Fig 2: performance graph

RESULT







CONCLUSION

By using the best features of both symmetric and asymmetric encryption approaches, hybrid cryptography is a formidable cryptographic tool. It allows for encrypted communication between parties without requiring a private key exchange route. Comparatively, asymmetric encryption ensures the safety of key distribution whereas symmetric encryption is quick to process and efficient.

Secure email, instant messaging, and online transactions are just a few examples of why hybrid cryptography is so vital in today's technological landscape, when protecting the privacy and authenticity of the information being transmitted is of the utmost

importance. Yet if the private key is compromised, hybrid cryptography is open to assaults, making key management an essential part of the security architecture.

Ultimately, hybrid cryptography is an essential tool for securing digital communications. It's a reliable and effective means of protecting data while in transit, and it has several uses. Because of this, knowing the benefits and drawbacks of the method and taking the necessary precautions to protect sensitive data is crucial for putting it into practice successfully.

FUTURE SCOPE:

For ensuring security we utilized double (AES-RSA) model. In future work, we can aim to overcome the limitation in the performance of 3-tier hybrid models for ensuring security in cloud environments as well.

REFERENCES:

[1] Shweta Kaushik, Ashish Pate, "Secure Cloud Data Using Hybrid Cryptographic Scheme", 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019
[2] Pallavi Kulkarni, Rajashri Khanai, Gururaj Bindagi, "A Hybrid Encryption Scheme for Securing Images in the Cloud", International Conference on Inventive Computation Technologies (ICICT), 2020
[3] B. Deepthi, G. Ramani, R. Deepika, Md Shabbeer, "Hybrid Secure Cloud Storage data based on improved Encryption Scheme", International Conference on Emerging Smart Computing and Informatics (ESCI), 2021
[4] Pallavi Kulkarni, Rajashri Khanai, Gururaj Bindagi, "A Comparative Analysis of Hybrid Encryption Technique for Images in the Cloud Environment", International Conference on Communication and Signal Processing (ICCSP), 2020
[5] Sadiq Aliyu Ahmad, Ahmed Baita Garko, "Hybrid Cryptography Algorithms in Cloud Computing: A Review", 15th International Conference on Electronics, Computer and Computation (ICECCO), 2020
[6] Sanjeev Kumar, Garima Karnani, Madhu Sharma Gaur, Anju Mishra, "Cloud Security using Hybrid Cryptography Algorithms", 2nd International Conference on Intelligent Engineering and Management (ICIEM), 2021
[7] Lalit Kumar, Neelendra Badal, "A Review on Hybrid Encryption in Cloud Computing", 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019
[8] Shekha Chenthara, Khandakar Ahmed, Hua Wang, Frank Whittaker, "Security and Privacy-Preserving Challenges of e-Health Solutions in Cloud Computing", IEEE Access (Volume: 7), 2019
[9] P. Chinnasamy, P. Deepalakshmi, "Design of Secure Storage for Health-care Cloud using Hybrid Cryptography", Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018
[10] Zainab Hikmat Mahmood, Mahmood Khalel Ibrahem, "New Fully Homomorphic Encryption Scheme Based on Multistage Partial Homomorphic Encryption Applied in Cloud Computing", 1st Annual International Conference on Information and Sciences (AiCIS), 2019