International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
ISSN: 2457-0362
www.ijarst.in

CLOUD FILE STORAGE SECURITY USING HYBRID CRYPTOGRAPHY

Dr. J. Ramesh Babu[1], Mohammad Nadeem[2], Kadambala Ashish Kumar[3], Dr. Amjan Shaik[4]

[1]Associate Professor, Department of Computer Science and Engineering, St. Peter's Engineering College, Dullapally, Maisammaguda, Medchal, Hyderabad, Telangana 500043.

[2,3]UG students, Department of Computer Science and Engineering, St. Peter's Engineering College, Dullapally, Maisammaguda, Medchal, Hyderabad, Telangana 500043

[4]Head of the Department, Department of Computer Science and Engineering, St. Peter's Engineering College, Dullapally, Maisammaguda, Medchal, Hyderabad, Telangana 500043.

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, although 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)

INTRODUCTION

Cryptography is the use of codes to safeguard data and communications so that only the intended receivers can interpret and process them. In this way, unauthorised individuals are prohibited from accessing information. Cryptography and crypto meaning, respectively, "written" and "hidden." The techniques used in cryptography to protect data are drawn from mathematical concepts and a collection of computations based on rules known as algorithms to change communications in ways that make them challenging to understand.

When encrypting and decrypting communications, symmetric key cryptography uses the same key by the message's sender and recipient. Symmetric Key Systems are speedier and simpler, but there is a requirement for safe key exchange between the sender and recipient. The most common symmetric key encryption system is data encryption technology (DES). This method makes no use of any keys at all. Since a fixed-length hash value is generated based on plain text, it is challenging to reconstruct the contents of plain text. Operating systems frequently employ hash methods to protect passwords. Information is encrypted and decrypted using a pair of keys in asymmetric key cryptography. A public key is used to encrypt data, and a private key is used to decode data. The public Key and Private Key are separate. Even if everyone has access to the public key, since only the intended recipient has access to the private key, nobody else can decode the message. Cloud computing has several advantages, including inexpensive prices and simple Internet access to knowledge. Security is the main concern in the cloud computing environment since users keep sensitive data with cloud providers, who might occasionally be unreliable. A well-organized method of accessing data from anywhere at any time is offered by cloud storage, which is why the majority of organisations are moving away from traditional data storage. Data security is the main concern when approving cloud computing for any organisation. Data splitting securely while securing data from an unreliable cloud is still a challenging problem. This project outlines a file/document security architecture that offers a practical fix for the core cloud security problems. Hybrid encryption combines the efficiency of symmetric encryption with the usefulness of public-key (asymmetric) encryption. Individuals who possess the private key are the only ones who can decode the material. To encrypt a transmission, a new symmetric key is created and applied to the plaintext material. The symmetric keys are encrypted with the recipient's public key. The final ciphertext has the symmetric ciphertext and the encrypted symmetric key. Important data may be protected with cryptography, whether it is being sent between computers or stored on a computer. In the cryptographic process, plaintext is transformed into ciphertext and then back into plaintext. Python's support for a cryptography module makes data encryption and decryption feasible. Using the encrypt and decrypt techniques, the fernet module of the cryptography package has built-in functions for creating keys, converting plaintext to ciphertext, and regaining plaintext from the ciphertext. Without the key, the Fernet module assures that the data it has encrypted cannot be altered or decrypted.

The idea of RSA is predicated on the difficulty of factoring large numbers. One of the two integers that make up the public key is the result of two gigantic prime numbers. The private key is likewise generated using the same two prime integers. Therefore, if someone can factorise the enormous number, the secret key is compromised. As a result, the strength of encryption is solely determined by the size of the key, and increasing the key size by two or thrice increases encryption strength. Most RSA keys are 1024 or 2048 bits in size, but experts believe that 1024-bit keys will soon be compromised. But at this point, it seems difficult to do. To provide a standard for the encryption of electronic data, (NIST) created the Advanced Encryption Standard (AES) in 2001. AES is still often used despite being more challenging to construct since it is significantly stronger than DES and triple DES. things to consider; A block cypher is AES.Key sizes range from 128 to 256 bits. Data is encrypted in chunks of 128 bits.

As a consequence, given 128 bits of input, it produces 128 bits of encrypted cypher text as an output. AES uses a network of connected processes known as a substitution-permutation network to replace and shuffle the input data as it functions.

LITERATURE SURVEY

[1]Shweta Kaushik proposed a hybrid symmetric encryption approach in this paper to provide more security for the owner's data than any single symmetric encryption algorithm. This hybrid approach secures data and protects it from any malicious activity carried out by an intruder. Using the proposed approach, brute force attacks are also rendered impossible. The use of symmetric encryption also increases processing capability because it is faster and more efficient than the asymmetric approach.

[2] Even though cloud-based services provide numerous benefits, data owners are still hesitant to store their data with a third party. The major concerns of outsourced data are confidentiality, integrity, privacy, and non-repudiation. Many traditional security approaches are proposed to secure data exchange between users and the cloud. In this paper, a hybrid encryption technique for image security is proposed. The scheme generates the secret key using Elliptic Curve Cryptography, which is then used by the DES and AES algorithms.

[3] There are numerous encryption techniques available to protect sensitive information from unauthenticated users. To protect the data, encryption, and decryption methods are used, and only authorized users can access the data. However, the brute force method can sometimes detect hidden data. A proposed method is used to enhance data confidentiality and authentication problems by combining AES and proxy re-encryption with Honey encryption. The system increases the security of outsourced data. Honey encryption combined with hybrid cryptography allows unauthorized users to access only messages that appear plausible.

[4] To secure data exchange between users and the media cloud, many traditional security approaches are available. However, there are still instances of security breaches. This paper discusses security concerns. To secure the images, Pallavi Kulkarni

proposes a hybrid encryption technique. The scheme generates the secret key using Elliptic Curve Cryptography, which is then used by the DES and AES algorithms.

[5] A hybrid cryptography study has been conducted in this paper from 2015 to early 2019. In this study, papers related to the problem were searched, and about 20 were considered based on filtering. Eight of these are based on a user-friendly tabular survey, while the other twelve are in-depth surveys. The primary goal of this review paper is to provide more and more information to new researchers, students in this field, and those who are unfamiliar with cryptography. The identified research gaps are the omission of user authentication and the improper implementation of hybrid algorithms.

[6] This paper proposed a multilevel cryptography-based cloud computing security system. The model combines symmetric and asymmetric key cryptography algorithms. The Data Encryption Standard (DES) and RSA are used in this implementation to provide multilevel encryption and decryption at both the sender and receiver sides, increasing the security of cloud storage. To reduce security threats, this security model provides transparency to both cloud users and cloud service providers. The proposed model is written in Java and uses the clouds cloud simulator tool. This model maximizes data security and saves time when uploading and downloading text files when compared to the existing system.

EXISTING SYSTEM

The existing system using SHA for hybrid cryptography has certain disadvantages that need to be considered. One major disadvantage of SHA is that it is vulnerable to collision attacks. This means that it is possible to find two different messages that produce the same hash value. This vulnerability makes SHA less secure as an encryption algorithm, especially when dealing with sensitive information. Another disadvantage of SHA is that it is a one-way function, which means that it is not possible to reverse the hash value back to the original message. This can be problematic if the original message needs to be recovered or decrypted. Additionally, SHA is not suitable for use in the key generation or key exchange, which are important components of hybrid cryptography. Key generation and key exchange require algorithms that are specifically designed for those tasks, and SHA is not optimized for these purposes. Finally, SHA is a widely used and well-known algorithm, which makes it a target for attackers. This means that there is a greater risk of attacks on systems that use SHA, as attackers may be more likely to target systems that use well-known algorithms. Overall, while SHA is a useful algorithm for certain applications, it has several disadvantages when it comes to hybrid cryptography. As such, it is important to carefully consider these disadvantages and evaluate whether SHA is the best choice for a given application.

PROPOSED SYSTEM

The proposed system for hybrid cryptography using Fornet has several advantages over the existing system that uses SHA. One major advantage of Fornet is its resistance to collision attacks. Fornet uses the FORS tree-based hash function, which is designed to be collision-resistant. This makes Fornet a more secure algorithm for encryption, especially when dealing with sensitive information. Another advantage of Fornet is its suitability for key generation and key exchange, which are essential components of hybrid cryptography. Fornet uses the FORS key agreement protocol, which allows for secure key exchange between parties. This means that Fornet is well-suited for use in hybrid cryptography systems, where secure key exchange is essential. Additionally, Fornet is a relatively new algorithm, which means that it is less well-known and less likely to be a target for attackers. This can help to increase the security of systems that use Fornet, as attackers may be less likely to target less well-known algorithms. Finally, Fornet is designed to be efficient, with a small memory footprint and fast execution times. This means that it can be used in a wide range of applications without negatively impacting system performance. Overall, the proposed system for hybrid cryptography using Fornet offers several advantages over the existing system that uses SHA. Fornet is more secure, more suitable for key generation and key exchange, less likely to be a target for attackers, and more efficient. As such, it is a promising choice for applications that require hybrid cryptography.

IMPLEMENTATION

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 produced 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 a 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 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 the 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). To encrypt 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 utilised ASCII encoding, we will also employ ASCII decoding.

Two ways for signing and validating our message utilising 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 utilise a technique based on signs (message, key). After preparing the message, the checkmark, and the key, we will implement a confirmed technique to verify the message. While the hash computation utilised 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 MODEL

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 utilising 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 to retrieve the first record, but once they do, they will have the secret keys.

MODULE 5: OUTPUT

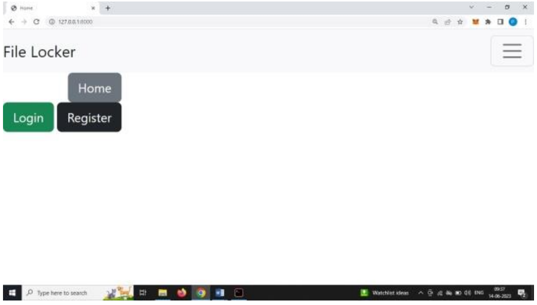
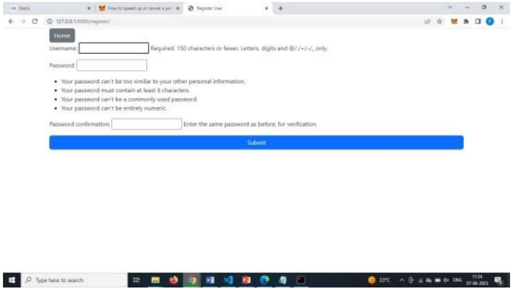


Fig 1 : Home Page



Username: Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

Fig 2 : Registration Page

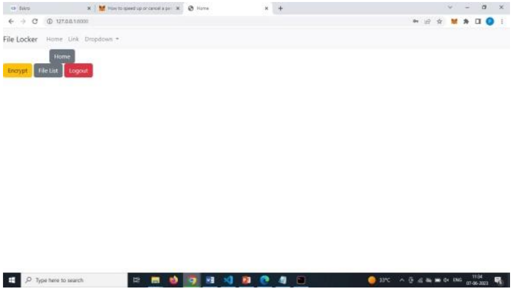
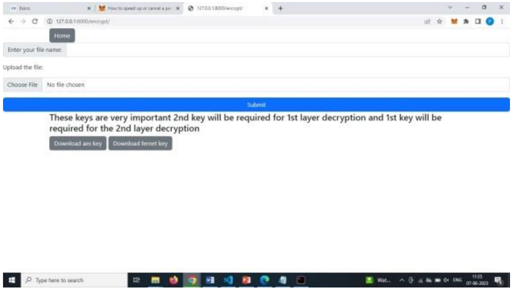


Fig 3 : User Page



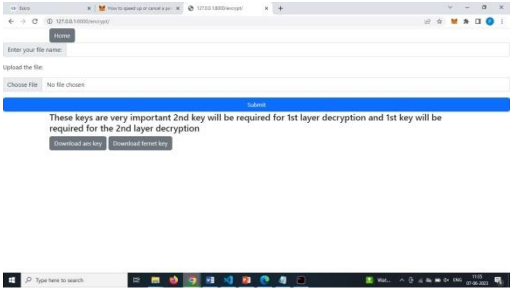
Enter your file name:

Upload the file:

No file chosen

These keys are very important 2nd key will be required for 1st layer decryption and 1st key will be required for the 2nd layer decryption

Fig 4 : Encryption Page



Enter your file name:

Upload the file:

No file chosen

These keys are very important 2nd key will be required for 1st layer decryption and 1st key will be required for the 2nd layer decryption

Fig 5 : Decryption Page

CONCLUSIONS

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.

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 Khalil Ibrahim, "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