A Seminar on

Web based application for secure data storage using hybrid symmetric algorithm

Team Details

1. P Shivani (20eg105237)
2. Juhi Mohta (20eg105221)
3. K Hemasri (20eg105240)

Project Supervisor
Name : DR.V.Subrahmanyam
Designation : Asso.professor

Introduction

The protection of sensitive data and information is of utmost importance in the modern digital age. As cyberattacks and data breaches become more frequent, it is essential to provide safe file storage on a local host. By using hybrid cryptography techniques to safeguard files kept on a local host, this project tries to address this problem.

To create a strong security solution, hybrid cryptography combines the benefits of symmetric and asymmetric encryption. Asymmetric encryption enables reliable key management and distribution whereas symmetric encryption is quick and effective. This project offers a reliable and effective solution for securing local file storage by combining these two techniques.

Concept Tree

- Secure File Storage
|
- Local Host
|
- File Storage System
|
- Cryptography
|
- Hybrid Cryptography
|
- Symmetric Encryption
- Asymmetric Encryption
|
- Key Management
|
- Key Generation
- Key Distribution
- Key Storage
- Key Revocation
|
- Secure Communication
|
- Data Confidentiality
- Data Integrity
- Authentication
- Authorization
|
- Implementation
|
- Integration of Symmetric Encryption (e.g., AES)
- Integration of Asymmetric Encryption (e.g., RSA)
- Key Management Implementation
- Secure Communication Protocols (e.g., TLS)
- User Authentication and Authorization
|
- Maintenance and Updates
|
- Key Rotation
- Patch Management

Literature

Author(s)	Strategies	Advantages	Disadvantages
M. Malarvizhi	The system uses a database that stores the files that need to be protected and their hash codes	The system focuses on ensuring the integrity of files, providing a reliable method to detect any unauthorized modifications.	Complexity increases the risk of implementation errors and may require more resources for maintenance and troubleshooting.
Jerzy Kaczmarek	This uses a pattern of each protected file to determine its modification. Methods used for pattern generation are cryptographic hash functions.	Storing file names and hash codes in a database provides a structured and organized way to manage and retrieve integrity information efficiently.	Continuous hash code generation and file verification processes may introduce performance overhead, especially in systems with a high volume of file operations.
Tulip Dutta	A secret key can be shared with other users to whom access needs to be given.	The use of key aggregation allows for different keys to be used for encrypting different data files, enhancing access control.	Proper key management practices are essential to avoid security risks.

Literature(cont..)

Author(s)	Method	Advantages	Disadvantages
Bilal Habib	A new method to implement the public key infrastructure.	Secure Data Sharing in Cloud Storage: The paper addresses the specific context of secure data sharing in cloud storage.	the mathematical relation between public and private between the public and the private key is maintained.
Rohit Barvekar	The proposed security mechanisms will prevent confidential data from being misused making the system more reliable.	The proposed security mechanisms aim to prevent the misuse of confidential data, enhancing the reliability of the system.	The effectiveness of the proposed security mechanisms needs to be rigorously validated through security analysis and testing.

Problem Statement

User selects the file from the local storage. The file will be uploaded to the cloud after getting encrypted. It uses only AES algorithms. It needs some more time for encryption and decryption process. Security provided by AES algorithm is less therefore, it is time taking to upload and download the files.

Objective

Our methodology guarantees better security and protection of customer data by storing encrypted data on a single cloud server, using AES, DES, and RC6 algorithms.

Problem Illustration

Scenario:

•Alice, a user, has a confidential document containing sensitive information.
•She wants to store this document securely in the cloud.
•Alice logs into a web-based application for secure file storage.
•She selects the document from her local storage to upload it.

AES Encryption Process:

The application employs the AES algorithm to encrypt the entire document.A secure symmetric key is generated for the encryption process.
The encrypted document is now ready for upload to a cloud storage service.

Download and Decryption by Authorized User:

When Alice needs to access the document, she securely retrieves and decrypts it.
Alice logs into the secure file storage application. She selects the encrypted document for download.
The application retrieves the encrypted document from the cloud and uses the stored symmetric key to decrypt it, providing Alice with the original document.

The process of encrypting and decrypting large files can introduce performance overhead, especially on resource-constrained devices or networks.

Proposed Method

The project aims to address the critical need for secure file storage on a local host by implementing a robust solution based on hybrid cryptography techniques. This project seeks to develop a hybrid cryptography method that combines the strengths of both symmetric and asymmetric encryption techniques, for secure file storage on a local host. Scalability is another key consideration, as the solution must be adaptable to accommodate various file types and sizes while maintaining performance and security standards. Lastly, error handling and data recovery mechanisms are of paramount importance to prevent data loss or corruption.

Proposed Method Illustration

Scenario: Bob wants to securely store a sensitive spreadsheet on his local server, ensuring confidentiality and integrity. The proposed method involves hybrid cryptography for enhanced security.
Sample Values:
Spreadsheet: sensitive_data.xlsx
Local Host: Bob's server
Encryption Algorithms: AES (Symmetric Encryption), RSA (Asymmetric Encryption)
Keys:
Symmetric Key (AES_Key)
Public and Private Key Pair (RSA_Public_Key, RSA_Private_Key)
Steps in the Proposed Method:
Key Generation:
Generate a Symmetric Key (AES_Key) for AES encryption.Generate a pair of Public and Private Keys (RSA_Public_Key, RSA_Private_Key) for RSA encryption.
File Encryption:
Encrypt the sensitive spreadsheet (sensitive_data.xlsx) using the Symmetric Key (AES_Key) with AES encryption.
Key Distribution:
Use the recipient's public key (RSA_Public_Key) to securely share the Symmetric Key (AES_Key) for decryption.
File Storage:
Store the encrypted spreadsheet on Bob's local server using the file storage system.

Proposed Method Illustration(con…)

File Decryption:
When an authorized user requests access to the spreadsheet, use their private key (RSA_Private_Key) to decrypt the Symmetric Key (AES_Key).Use the decrypted Symmetric Key (AES_Key) to decrypt the sensitive spreadsheet.
Sample Execution:
Bob encrypts sensitive_data.xlsx using AES_Key to get encrypted_spreadsheet_aes.Bob shares AES_Key encrypted with RSA_Public_Key with Alice.
Alice, an authorized user, decrypts AES_Key using her RSA_Private_Key.
Alice uses the decrypted AES_Key to decrypt encrypted_spreadsheet_aes and access the original sensitive spreadsheet.
This example illustrates the proposed method for secure file storage on a local host using hybrid cryptography, showcasing the process and interactions with sample values.

Parameter

1 Security Strength (SS):

- Formula:Strength of Symmetric Algorithm×Strength of Asymmetric AlgorithmSS=Strength of Symmetric Algorithm×Strength of Asymmetric Algorithm

2.Encryption Time (ET):

Formula:Time for Symmetric Encryption+Time for Asymmetric EncryptionET=Time for Symmetric Encryption+Time for Asymmetric Encryption

3.Decryption Time (DT):

- Formula:Time for Symmetric Decryption+Time for Asymmetric DecryptionDT=Time for Symmetric

Decryption+Time for Asymmetric Decryption

4.Time for Symmetric Key Generation+Time for Asymmetric Key Pair GenerationKGT=Time for Symmetric Key Generation+Time for Key Generation Time (KGT):

5.Key Storage Space (KSS):

- Formula:Space for Symmetric Key+Space for Asymmetric Public and Private KeysKSS=Space

Experiment Environment

Software Requirements

- Operating system: Windows Family.
- Coding Language: J2EE (JSP,Servlet,Java Bean)
- Data Base: mySQL.
- Web Server: Wamp server
- Tools: Django and python

Hardware Requirements

- Hard Disk: 40 GB. 8
- Floppy Drive: 1.44 MB.
- Monitor: 14’ Color Monitor.
- Ram: 4 GB.

Project status

S.No	Functionality	Status (Completed /in-progress/Not started)
1	Data Collection	completed
2	Modules Description	In-progress
3	System Architecture	In-progress
4	Implementation	Not yet started

References

[1] The OpenStack Project. (2015). OSSA-2015-006: Unauthorized Delete of Versioned Swift Object. Retrieved Dec 14, 2022, from https://security.openstack.org/ossa/ OSSA-2015-006.html

[2] The OpenStack Project. (2015). OSSA-2015-016: Information Leak Via Swift Tempurls. Retrieved August 26, 2022, from https://security.openstack.org/ossa/OSSA2015-016.html

[3] The OpenStack Project. (2015). Possible Glance Image Exposure Via Swift. Retrieved February 23, 2023, from https://wiki.openstack.org/wiki/OSSN/OSSN0025 [4] Cloud Security Alliance. (2018). Top Threats to Cloud Computing: Deep Dive. Retrieved August 8, 2022, from https://downloads.cloudsecurityalliance.org/assets/ research/topthreats/top-threats-to-cloudcomputing-deepdive.pdf

[5] The OpenStack Project. (2015). OpenStack Security Advisories. Retrieved February 2, 2023, from https://security.openstack.org/ossalist.html

[6] Common Vulnerabilities and Exposures. (2015). CVE-2015-5223. Retrieved July 1, 2022, from https://cve.mitre.org/cgi-bin/cvename.cgi?name$=$CVE-2015-5223

[7] Common Vulnerabilities and Exposures. (2016). CVE-2016-9590. Retrieved November 23, 2022, from https://cve.mitre.org/cgi-bin/cvename.cgi?name$=$CVE2016-959

Thank you

Project seminar–I Evaluation

S.No	Rubrics	Marks
1	Concept Introduction	4
2	Literature and Parameter	5
3	Problem and Problem Illustration	8
4	Proposed Method and Illustration	8
Total		25