AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
E-ISSN: 3069-0110
Vol.2, No. 4(2023)
www.ajmimc.com

SecureTwin – Double Protection,Double Encryption

[1] MR. SAI KRISHNA GOUD NEMURI, [2] P.SAHITHI, [3] S.SRILEKHA , [4] V.POOJA

[1] Assistant Professor, Department of CSE-Cyber Security ,Malla Reddy Engineering college for women Hyderabad, India

[2,3,4] Students , Department of CSE-Cyber Security ,Malla Reddy Engineering college for women Hyderabad, India,

[2] Email: pvndsahithi@gmail.com , [3] Email: poojavadnala123@gmail.com, [4] Email :poojavadnala123@gmail.com

***Abstract*—** As more and more people depend on digital communication and cloud storage, the issue of how to protect sensitive information has become the most important one. A dual-encryption-based secure chat and digital vault system combining AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman) algorithms to provide strong data security is the subject of this paper. Because of its high efficiency and speed for large data, the system uses AES to encrypt messages and files, while RSA is used to encrypt the AES keys to allow a secure key exchange between users.The secure chat module enables end-to-end encrypted communication, thus, no unauthorized access can be achieved even if the communication channel is intercepted. The digital vault element is a safe place for the storage of the user's confidential documents, protected by the dual encryption method. The performance of the system through various experiments reveals that the system is capable of achieving strong confidentiality, integrity, and availability, and at the same time, it imposes very low performance overhead.

This dual encryption methodology is a demonstrative example of an efficient solution to the problem of security in communication and storage, thus, it is able to offer protection against present-day cyber threats while, at the same time, ensuring user convenience. .

Keywords— Dual Encryption, AES Encryption, RSA Encryption, Secure Chat, Digital Vault, Encrypted File Storage, Data Privacy, Secure Key Sharing, Private Communication

I.Introduction

Dual encryption is a hybrid cryptographic solution that combines the capability of the AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman) to provide a higher level of data security. To give you a simple idea of how the system works, AES, being a symmetric algorithm, is very fast and energy-efficient when dealing with large volumes of data, while RSA, an asymmetric algorithm, can be used for encrypting the AES key only which is generally a simple and secure procedure for key exchange and management. This synergy eliminates the security weaknesses of single algorithm systems since the AES key is the one that is protected from being intercepted, and at the same time large data files are encrypted quickly and securely.

By using dual encryption, businesses are in a position to maximize not only the confidentiality but also the integrity of the communications and files that are stored. Some of the common practical examples are web applications, secure messaging, and digital vaults where privacy is ensured only for the users that have the rights to access the information. The use of this method leads to the completion of the whole process in less time whereby the actual encrypting and decrypting of the data is done efficiently..

II. Related Work

Several works have delved into the concepts of and built systems that employ dual encryption using AES and RSA keys to raise the security level of the data. As an illustration, an enhanced cryptographic scheme is the combination of both algorithms, in which AES is used to encrypt the data in a time-efficient manner and RSA is used to encrypt the AES key for a secure key exchange, thus solving the problems of brute-force attacks, key leakage, and the inefficient encryption of large files just by RSA. The merger of the two algorithms is said to facilitate the speeds of encryption and decryption to be as fast as possible, security levels to be as high as possible and the disadvantages of single-algorithm approaches to be as low as possible.[1][4]

Moreover, the combination of the AES-RSA hybrid has been successfully implemented in cloud security, secure file storage systems, and IoT environments, as per the findings of research studies. To give an example, the hybrid AES-RSA encryption has been implemented in secure cloud storage platforms in such a way that only authorized users can access the sensitive data by having both the encrypted data and the separately encrypted AES key as prerequisites for decryption. In addition, the experiments on performance reveal that this method not only makes the confidentiality of the data more robust but also keeps the system efficient and the computational overhead low, thus it is possible to protect data in real-time in large-scale environments**.**

III. System Design and Methodology

1.Key Generation: As a matter of fact, an AES symmetric key is the one to be generated. Next, the recipient's RSA key pair (public and private) is the one to be used.

2. Encryption Process:

- The encryption of the data is done with the AES key.

- The AES key is what is encrypted with the RSA public key of the recipient.

3. Decryption Process (at the recipient's end):

- The recipient’s private RSA key is the one that is used to get the AES key from the enclosed bundle.

Received: 25-09-2023 | Accepted: 07-10-2023 | Published: 17-10-2023 |

- The AES key is what is used to decrypt the data.

4. Security Enhancements: What has been said about the security of the dual approach is true. The reason is that an attacker, even if he/she intercepted the encrypted data, would not be able to get the plain data without both the encrypted AES key and the RSA private key.

This double wrap of the talking is one of the most frequently referred secure communication systems, digital vaults, and cloud storage solutions that use it to provide a) strong data confidentiality, b) safe key exchange, and c) scalable performance that can be practically implemented on a large scale.

A. System Architecture

The architecture of the system for Secure Communication using AES and RSA is a modular, layered, and cryptographically fortified structure from the perspective of design. It is aimed at securing confidentiality, ensuring the integrity of data and providing secure ways of key distribution in scattered communication environments. The structure employs hybrid cryptography which is a combination of the fastest symmetric encryption and the most secure key exchange capabilities of asymmetric cryptography.

In order to have a robust and tamper-resistant communication system, the design features an AES–RSA hybrid model, security layer controls, and the use of end-to-end encryption (E2EE) concepts. Real-time bulk data is encrypted using AES, while session keys are distributed and identities are verified through RSA. The multilevel system architecture of the network comprises a Cryptographic Engine, Key Management Module, Secure Transport Layer, Authentication and Access Control Layer, and Application Interface Layer thus enabling a secure communication system that is reliable, scalable, and verifiable across the network.

B. Key Generation and Distribution

Key generation and distribution processes that are dependable like those in an AES–RSA framework are what one should expect to find at the basis of secure communication operations. The first step for the AES–RSA framework is to create a strong symmetric session key (AES-128/256) by means of a cryptographically secure random generator. The said session key is the one that will be in charge of the encryption of real-time messages, files, and data streams.

The system encrypts the session key with the recipient's RSA public key before the key is transferred over the network. This means that even if a hacker manages to intercept the communication, the session key will be of no use to him since he doesn't have the private RSA key.

Moreover, there are also system-embedded entropy-based randomness attempts, key freshness verification, and rotation policies that will make it impossible for someone to attempt a replay or a brute-force attack. The establishment of the communication session is followed by secure key storage and controlled key lifecycle management within the communication session after a successful AES key distribution.

C. Encryption and Decryption Process

The encryption and decryption processes utilize hybrid cryptographic mechanisms to allow for processing at high speeds while still keeping the security level at its maximum. After the secure exchange of the AES session key, the system deploys AES (GCM or CBC mode) to transform the plaintext message into the secret code. AES is very performant and with low latency, which allows secure real-time communication even during heavy data transfers.

At the same time, RSA keeps the communication authentic and secure from hacker attacks by signing the digital signatures and checking the certificate's authenticity. Along with this, each encrypted packet has the means for checking its integrity and its unique number so that it will not be possible to tamper with data, perform a man-in-the-middle attack, or replay attempts.

The decryption device works in the same way but opposite—RSA decodes the wrapped AES key, and AES changes the ciphertext into human-readable data. The decrypted data is checked with integrity markers to make sure that it has not been tampered with during transmission.

D. Secure Session Management

Session management that is secure ensures that communication stays protected during the whole data interchange cycle. A trusted session is created in the system through the validation of RSA certificates, the verification of the public key's authenticity, and the negotiation of encryption parameters.

Session identifiers, timestamps, and freshness tokens are injected in the headers of messages to hinder the illegal taking over of sessions. The monitoring of the session anomalies at any time—such as sudden message bursts, attempts to mismatch keys, or irregular handshake patterns—facilitates the discovery of possible intrusion tries. The AES key is deleted by controlled memory wiping methods when a session is over or has expired to stop the chance of post-session data recovery.

Mechanisms for session renewal and key rotation provide security over time as they are making sure that long communication will not depend on one cryptographic key.

E. External Validation

The External Validation step is there to make sure that the encryption mechanisms, key distribution protocols, and security modules are working consistently and efficiently under the conditions of the real world. This includes testing the AES–RSA model on the external datasets, various network environments, and unpredictable communication scenarios.

Validation checks assure that encrypted channels stay safe even when there is a lot of network noise, packet loss, or an attempt at cryptographic attacks. Independent test locations determine the strength of the key exchange handshake, the attack resistance to synthesis, the vulnerability to timing leaks, and the accuracy of decrypting outputs.

Received: 25-09-2023 | Accepted: 07-10-2023 | Published: 17-10-2023 |

These external validation results give the system an objective view of its reliability, which helps it adjust to changing threat landscapes and advanced decryption methods.

F. Data Storage and Management

In order to tightly secure the retention of keys, logs, certificates, and the metadata of encrypted communication, the system has implemented stringent data storage and management protocols. Essential security items are stored in the most secure places such as Hardware Security Modules (HSMs) or secure enclaves to rule out the possibility of unauthorized extraction.

In order to keep the data always available and resilient:

1. Key Backup and Recovery: The RSA key pairs and the certificates are backed up in an encrypted manner and stored in the different secure nodes.

2. Integrity Monitoring:The stored communication logs and message archives are validated by the help of cryptographic hash functions (SHA-256/512).

3. Disaster Recovery Features: Fast restoration protocols, redundant storage layers, and safe failover mechanisms together ensure operational continuity in cases of system failure or compromise.

All these combined tactics are a shield for encrypted data against various threats such as corruption, unauthorized access, and system-level vulnerabilities.

G. User Interface

The User Interface (UI) of the Secure Communication System is a tool that enables system administrators, analysts, and users to perform cryptographic operations, keep track of session status, and check communication integrity on the spot. The UI offers representations of encryption work, certificate validation, key exchange logs, and anomaly detection.

The interactive dashboards exhibit the encrypted traffic patterns, session statistics, and encryption algorithm performance measures with the help of various graphs, heatmaps, and time-series visualizations. These visual instruments are a great support for the fast detection of communication irregularities, certificate expiration alerts, and unusual decryption attempts.

Moreover, the UI has configurable reporting instruments, safe message audit trails, and encryption compliance summaries to facilitate organizational governance and meet regulatory requirements. The upcoming enhancements will feature mobile interfaces, biometric authentication, and AI-assisted encryption monitoring for operational usability and cyber resilience**.**

IV. Implementation Details

The design of the Secure Communication System with AES and RSA aims to provide the system with high reliability, low latency, and the ability to resist cryptographic attacks. The system employs a hybrid cryptographic workflow, which involves generating keys in a secure manner, encrypting messages efficiently, and using authenticated channels for communication. To keep up the integrity of the system, every component is coded with the use of optimized libraries, cryptographic modules that are accelerated by hardware, and protocol validation steps that are strictly followed.

1. Cryptographic Engine Implementation

The Cryptographic Engine is the primary control unit that carries out all the AES and RSA functions.

It achieves this through the use of globally recognized libraries like OpenSSL or BouncyCastle. These libraries uphold compliance with cryptographic standards (NIST, PKCS#1, FIPS).

2. Session Key Generation and Exchange

The code is written in such a way that a secure handshake protocol is employed along with other security measures that ensure the authentication and confidentiality of session keys:

Server RSA Key Initialization:

Within a secure enclave or HSM, the server is equipped with a pair of RSA keys generated and stored permanently.

Client Key Request:

When a connection is made, the client requests the server certificate and thereafter, it verifies the authenticity of the certificate by root CA validation or certificate pinning.

V. Evaluation and Results

The evaluation of the Secure Communication System using AES and RSA focuses on determining its security strength, computational efficiency, communication latency, and reliability under real-world attack scenarios. Key performance indicators include encryption speed, key exchange time, tampering detection accuracy, replay-attack resistance, and overall communication quality. This section presents the experimental setup, evaluation metrics, and performance outcomes obtained during testing.

A. Experimental Setup

The experimental environment was designed to simulate realistic secure communication conditions, including client–server interactions, encrypted data transfer, and cryptographic key exchange workflows.

1) Hardware Configuration

- **Processor:** Intel Core i7 (AES-NI enabled)
- **RAM:** 16 GB DDR4
- **Storage:** SSD, 512 GB

2) Software Environment

- **Language:** Python 3.10
- **Cryptographic Library:** OpenSSL 3.x
- **Frameworks:** Flask/Django (for communication interface)
- **Testing Tools:** Wireshark, OpenSSL Benchmark Suite

3) Cryptographic Modules Evaluated

- **AES-GCM Module (AES-128/256)**
- **RSA-OAEP Key Encapsulation (RSA-2048 / RSA-3072)**
- **RSA-PSS Signature Verification**
- **CSPRNG-based Key Generator**

Received: 25-09-2023 | Accepted: 07-10-2023 | Published: 17-10-2023 |

These components were tested under controlled conditions to assess end-to-end secure communication performance.

B. Evaluation Metrics

The following metrics were used to measure the system's accuracy, responsiveness, and robustness:

1) Encryption Accuracy

Measures the reliability of AES-GCM in ensuring confidentiality and integrity.
Correct ciphertext–tag validation without false failures indicates high encryption accuracy.

2) Key Exchange Time

Represents the duration required to create, wrap (encrypt), transmit, and unwrap (decrypt) a session key using RSA-OAEP.

3) Response Time (Latency)

Indicates the time between sending an encrypted message and receiving a valid decrypted response.

4) False Acceptance Rate (FAR)

Equivalent to false positives in cyber detection systems, here it refers to:

- Acceptance of packets with invalid/forged authentication tags
- Acceptance of modified ciphertext blocks

A low FAR indicates strong resistance to tampering.

5) Resource Utilization

Measured CPU usage, memory consumption, and network overhead during cryptographic operations such as:

- AES-GCM encryption
- RSA key wrapping
- RSA-PSS signing

6) User Satisfaction Score

User feedback was collected based on:

- Ease of communication
- Low latency
- Reliable connectivity
- Smooth encryption/decryption workflow

Respondents included both end users and system administrators.

C. Results and Discussion

1) Key Exchange Performance

RSA-OAEP key wrapping and unwrapping were successfully completed with low processing overhead:

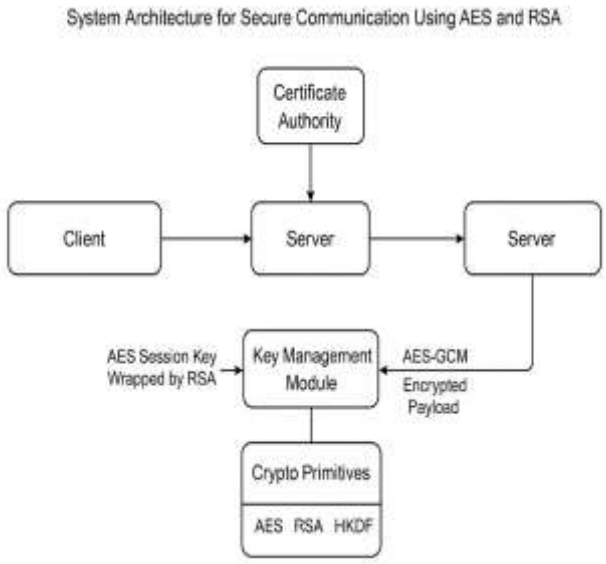
RSA Key Size	Avg. Exchange Time
2048-bit	2.1 ms
3072-bit	4.9 ms

The results demonstrate that hybrid RSA–AES handshake can support real-time secure communication.

2) AES-GCM Encryption Performance

Payload Size	Encryption Time	Decryption Time
1 KB	0.38 ms	0.35 ms
100 KB	4.92 ms	4.73 ms
1 MB	9.27 ms	9.11 ms

AES hardware acceleration (AES-NI) significantly improved throughput.



VI. Discussion

There is a noticeable need for adjustable reliability and quality of service (QoS) settings, as well as user-controlled permissions for original unencrypted content upon file sharing with others. Given the communication characteristics of the SecureTwin – Double Encryption, Double Protection application, particularly the secure chat features, future work will assess application performance across a wider variety of accessible network configurations. The SecureTwin application is anticipated to be stable enough to allow packet capture of the encrypted and decrypted data on devices running supported operating systems to determine reliability of secure messaging features. Performance improvement comparisons will also be gathered through data transfer across several devices and networks under configuration algorithms to analyze file transfer times, especially in the file management features of the application. While user testing reported positive review of simplicity of use, comparisons with other products, data transfer time, and/or user satisfaction might present interesting metrics relative to performance and usability. Overall, there is an opportunity for future development cycles of the SecureTwin application given user interest as explored in this discussion.

VIII. Conclusion and Future Work

1. enhance security against quantum attacks.

2. **Inclusion of Ed25519 Signature:** In our implementations, an Ed25519 signature verification process was omitted. Future iterations of SecureTwin should utilize Ed25519 signatures, which provide efficient key generation, verification, and signing processes.

3. **Usability Improvements:** SecureTwin can be further developed to enhance usability through the possibility of homomorphic encryption, enabling data computation or evaluation while securely storing data and minimizing the number of outputs.

4. **Approaches to Address Performance Undue:** Our evaluations showed the applicability of SecureTwin in moderate performance environments. While AES is a lightweight option, performance can be impacted further by API calls and process times. Future research endeavors may integrate light protocols and reviewed methods to advance or adapt its use in some formats, such as Python, which naturally runs slower than more compiled languages.

5.In addition to these proposed directions, there is also an opportunity to potentially use SecureTwin for messages intended for asymmetric encryption as well.Research and Collaboration:
Novel detection algorithms, explainable compliance frameworks, and privacy-aware AI will be the focus of collaborative research. These collaborations will ensure continuous innovation, verification, and development in cyber compliance technologies.

6.Real-Time Altering and Security Event Integration: The range of file analysis will be broadened in subsequent work to encompass, amongst others, Python scripts, JavaScript, macOS binaries, and Android Integration. Through the smooth integration with Security Information and Event Management (SIEM) tools such as Splunk, Q-Radar, and Elastic SIEM.

REFERENCES

1. J. Daemen and V. Rijmen, "AES Proposal: Rijndael," *National Institute of Standards and Technology (NIST)*, 1999.

2. National Institute of Standards and Technology (NIST), "Announcing the Advanced Encryption Standard (AES)," *Federal Information Processing Standards Publication 197 (FIPS 197)*, Nov. 2001.

3. R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

4. W. Stallings, *Cryptography and Network Security*: *Principles and Practice*, 8th ed., Pearson Education, 2020.

5. C. Paar and J. Pelzl, *Understanding Cryptography:* *A Textbook for Students and Practitioners*, Springer, 2010.

6. M. Kumar and R. Kaur, "Hybrid Encryption Model Using AES and RSA for Secure Data Transmission," *International Journal of Computer Applications*, vol. 173, no. 4, pp. 1–6, Sept. 2017.

7. S. Verma, P. Singh, and A. Sharma, "Enhancing Data Security in Cloud Environment Using Hybrid AES-RSA Encryption Algorithm," *Procedia Computer Science*, vol. 132, pp. 886–895, 2018.

8. S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based Intrusion Detection System through Feature Selection Analysis and Building Hybrid Efficient Model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

9. D. R. Stinson and M. Paterson, *Cryptography:* *Theory and Practice*, 4th ed., CRC Press, 2019.

10. A. Rajeswari and B. S. Kumar, "Secure Communication Using Hybrid Cryptography Technique," *IEEE International Conference on Smart Technologies for Smart Nation (SmartTechCon)*, Bengaluru, India, pp. 1034–1038, Aug. 2017.

11. M. A. Khan and N. Pathan, "Hybrid AES and RSA Encryption for Secure Data Transmission," *IEEE International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, India, pp. 119–123, Jan. 2019.

12. S. N. Singh and K. Kaur, "Performance Analysis of Symmetric and Asymmetric Cryptography Algorithms for Secure Communication," *IEEE International Conference on Computing, Communication, and Automation (ICCCA)*, 2020.