A Study on the Quantum Resistant Hybrid Cryptographic Scheme using AES and RSA Algorithms.

P. Maduni, Taein Kwon, Seungwan Je, Juyoung Jung, and Kyeongjun Ko

Dong-A Univ.

2372907@donga.ac.kr, kwontaein9@gmail.com, ngxy99@naver.com , 2024317@donga.ac.kr, kkj8000@dau.ac.kr

AES 와 RSA 알고리즘을 이용한 양자저항성 하이브리드 암호화 기법에 관한 연구

P. Maduni, 권태인, 제승완, 정주영, 고경준
동아대학교

Abstract

With the rapid advancement of Quantum Computing, there is a huge threat to traditional cryptographic schemes. This urges the need to develop a quantum resistant encryption method. This paper presents a comprehensive study on a novel Quantum Resistant Hybrid Encryption Scheme (QRHES) that integrates Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) algorithms. The proposed scheme aims to mitigate the vulnerabilities posed by quantum computing while leveraging the efficiency of classical encryption algorithms.

Ⅰ. Introduction

Hybrid encryption is a cryptographic technique that integrates both symmetric and asymmetric encryption methods to address the limitations of each approach individually.

`1. Symmetric Encryption: This encryption mechanism involves using a single shared key for both encryption and decryption processes. While it offers fast encryption and decryption speeds, it suffers from the challenge of securely sharing the secret key between the communicating parties. If this key is compromised, the security of the entire communication is jeopardized.

2. Asymmetric Encryption: This encryption, on the other hand, employs a pair of keys: a public key for encryption and a private key for decryption. This method solves the key distribution problem inherent in symmetric encryption since the public key can be freely distributed. However, asymmetric encryption tends to be slower and computationally intensive compared to symmetric encryption.

Hybrid encryption addresses the limitations of symmetric and asymmetric encryption by combining their strengths. It offers improved security through secure key exchange facilitated by asymmetric encryption while maintaining efficiency and performance by utilizing symmetric encryption for data transmission.

Ⅱ. Method

1. Database Setup and Key Management

Database Creation: An SQLite database is established to store public and private keys.

Table Creation: A table is created to store public-private key pairs.

Key Insertion: Functions are devised to insert generated keys into the database and retrieve them when needed.

2. Key Generation

Post-Quantum Key Generation: Utilizing a function, post-quantum RSA key pairs are generated for encryption and decryption processes. [1]

3. Encryption Process

Symmetric Key Generation: A random symmetric key is created.

AES Encryption: The plaintext message is encrypted using AES symmetric encryption), generating a ciphertext.

RSA Encryption: The AES symmetric key is encrypted using RSA public key encryption resulting in a ciphertext.

4. Decryption Process

RSA Decryption: The encrypted AES symmetric key is decrypted using RSA private key decryption.

AES Decryption: The ciphertext message is decrypted using the decrypted AES symmetric key, revealing the original plaintext message.

5. Integration (AES and RSA Algorithms)

This algorithm implements a hybrid encryption scheme to secure messages. It utilizes both symmetric and asymmetric encryption techniques for efficiency and security. At first, post-quantum public and private keys are generated for asymmetric encryption (RSA), and a random symmetric key is generated for symmetric encryption (AES). Then the plaintext message is encrypted using AES with the symmetric key, ensuring efficient encryption for large messages.

The symmetric key is then encrypted using RSA with the post-quantum public key to secure it during transmission. Afterwards the encrypted symmetric key is decrypted using the post-quantum private key. Using the decrypted symmetric key, the ciphertext message encrypted with AES is decrypted. At the end, the algorithm measures the time taken for encryption and decryption processes for various message sizes and plots them to evaluate efficiency.[3]

6. Execution and Analysis

Execution: The encryption and decryption processes are executed for each message size.

Data Collection: Encryption and decryption times are collected for performance analysis.

Analysis: The obtained data is analyzed to assess the efficiency of the hybrid encryption scheme under varying message sizes.

7. Result Interpretation

Encryption and Decryption Efficiency: The efficiency of the encryption and decryption processes are evaluated concerning encryption / decryption time and message size.

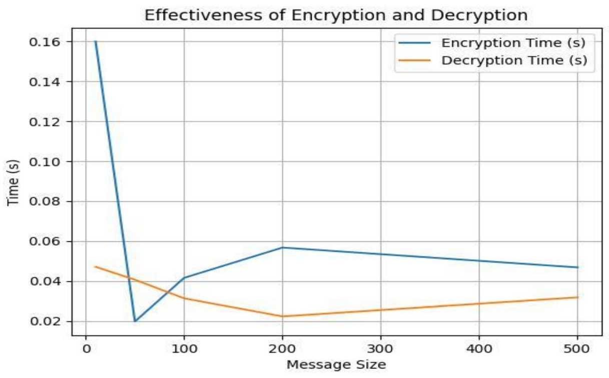

Figure 1 - Efficiency of the developed Quantum Resistance Algorithm.

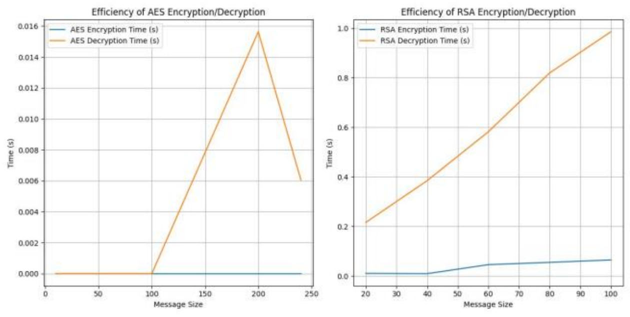

Figure 2 - Efficiency of the existing AES and RSA Algorithms.

Ⅲ. Conclusion

The paper explores the effectiveness of a QRHES merging AES and RSA algorithms. Its aim is to create a cryptographic method resilient against quantum computing threats while ensuring efficient data transmission.

Hybrid Encryption Approach: QRHES combines AES for symmetric encryption and RSA for key exchange and digital signatures.

Quantum Resistance: QRHES's use of RSA for key exchange addresses vulnerabilities to quantum algorithms, ensuring robust protection against quantum attacks.

Key Management and Database Integration: The scheme includes functionalities for secure key generation, encryption, decryption, and storage within an SQLite database, ensuring seamless operation in practical scenarios.

Future Directions: While this study provides a solid foundation for the implementation and evaluation of the QRHES, there are several avenues for further research and development. Future work could explore optimizations to enhance the performance of the encryption scheme, investigate alternative post-quantum cryptographic primitives, and assess the scheme's resilience against advanced quantum attacks.

REFERENCE

[1] Adu-Kyere, A., Nigussie, E., & Isoaho, J. (2022). Quantum key distribution: Modeling and simulation through bb84 protocol using python3. *Sensors*, *22*(16), 6284.

[2] Edward Keitaro Heru, F. H. (2023). File Encryption Application using Menezes-Vanstone Elliptic Curve Cryptography Based on Python. *Procedia Computer Science, Volume 227*, 51-658

[3] Dunmore, A., Samandari, J., & Jang-Jaccard, J. (2023). Matrix encryption walks for lightweight cryptography. *Cryptography*, *7*(3), 41.