International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307

OPEN ACCESS

Available Online at : www.ijsrcseit.com

doi : https://doi.org/10.32628/CSEIT2410263

Comparative Analysis of AES and RSA with Other Encryption Techniques for Secure Communication

Prashant[1], Md Sohail Haque[1], Amrinder Kaur[1], Pankaj Yadav[2]

[1]School of Computer Science and Engineering, Lovely Professional University Phagwara, Punjab, India

[2]Assistant Professor, School of Computer Science and Engineering, Lovely Professional University Phagwara, Punjab, India

ARTICLEINFO

Article History:

Accepted: 10 April 2024

Published: 19 April 2024

Publication Issue

Volume 10, Issue 2

March-April-2024

Page Number

565-574

ABSTRACT

In today's digitized era, where the whole world is interconnected and every information about individuals are collected, it is important to process and store the data such that it is safe from unauthorized access. Encryption is used to turn the plain text into cipher text which makes the data unreadable, thus maintaining its confidentiality and integrity. Among the several encryption methods available, Advanced Encryption Standard (AES) and Rivest Shamir Adleman (RSA) are popularly used due to their effectiveness and efficiency. However, alternative encryption techniques exist, each offers different level of security and performance. This study presents a thorough comparative analysis of AES and RSA alongside other encryption methods to assess their suitability for secure communication. Factors such as encryption strength, computational complexity, key management, scalability, and versatility are examined to provide a comprehensive understanding of each technique's strengths and weaknesses. By scrutinizing these aspects, this research aims to offer insights for decision-makers in selecting the most suitable encryption method tailored to specific requirements and constraints.

Keywords : Encryption, Advanced Encryption Standard (AES), Rivest-Shamir-Adleman (RSA), Secure Communication, Comparative Analysis, Key Management, Computational Complexity, Scalability

I. INTRODUCTION

In today's digital era, sharing sensitive information across networks is pervasive, spanning from personal exchanges to critical business dealings. However, this heightened connectivity also exposes data to numerous security risks, including interception, eavesdropping, and unauthorized access. Encryption serves as a fundamental defense against several threats by converting plain text into cipher text, which makes the data unreadable to unauthorized parties. AES and RSA are highly regarded encryption methods due to their established security and effectiveness, making them popular choices among users. Nonetheless, given the technological advancement and rising cyber threats, it is important to conduct a comparative evaluation of

AES and RSA alongside other encryption methods to ascertain their effectiveness in ensuring secure communication.

A. Objective

The primary objectives of this research paper is to:

To provide a comprehensive understanding of encryption algorithms, symmetric and asymmetric encryption techniques

Analyze the strengths and weaknesses of AES (Advanced Encryption Standard) and RSA (Rivest Shamir Adleman) for secure communication.

Compare AES and RSA with alternative encryption techniques, considering factors such as encryption strength, computational complexity, key management, scalability, and versatility.

Explore real-world use cases and case studies highlighting the practical applications and performance of different encryption algorithms.

Identify emerging challenges in secure communication and potential future directions for encryption techniques.

B. Scope

The scope of this research paper encompasses the following areas:

A detailed exploration of encryption principles, including symmetric and asymmetric encryption algorithms.

In-depth analysis of AES (Advanced Encryption Standard) and RSA (Rivest Shamir Adleman), covering their encryption processes, strengths, weaknesses, and real-world applications.

Comparative evaluation of AES and RSA with alternative encryption techniques, such as ECC (Elliptic Curve Cryptography), Diffie-Hellman Key Exchange, Blowfish, and ChaCha20.

Consideration of various factors influencing the selection and implementation of encryption techniques, including encryption strength, computational complexity, key management strategies, scalability, and versatility.

Examination of case studies and performance evaluations to assess the practical implications and effectiveness of different encryption algorithms in diverse scenarios.

Discussion on emerging challenges in secure communication and potential avenues for advancing encryption techniques to address evolving threats and technological advancements.

II. ENCRYPTION FUNDAMENTALS

Encryption is one of the most important method that maintains the confidentiality, integrity, and authenticity of the transmitted data. The main component of encryption is that it transforms plain text data into cipher text by using various cryptographic algorithms. There are two main categories of encryption i.e. symmetric and asymmetric.

A. Symmetric Encryption:

Symmetric encryption, uses a single key for encryption and decryption, and then that key is shared between sender and receiver. Symmetric encryption algorithms like Advanced Encryption Standard (AES), Data Encryption Standard (DES) and 3DES are generally faster and more efficient than asymmetric encryption algorithms. While symmetric encryption is quick and efficient, secure key distribution is one of the major drawback, as a compromised key could lead to a security breach.

Strengths:

Efficient: Symmetric encryption algorithms are faster and consume less resources than asymmetric encryption.

Suitable for large data volumes: Symmetric encryption efficiently encrypts substantial amounts of data.

Easy to implement: Symmetric encryption algorithms are often simpler to implement and manage.

Limitations:

Key distribution: Securely distributing and managing the shared secret key among communicating parties is the primary challenge in symmetric encryption.
Lack of key exchange mechanism: Symmetric encryption lacks an inherent mechanism for securely exchanging keys between parties.

B. Asymmetric Encryption:

Unlike symmetric encryption, asymmetric encryption uses two key, public key for encryption and private key for decryption. Public key is distributed via public channel and be utilized by anyone for encryption while private key can only be used by the receiver decrypt the encrypted message, which ensuring confidentiality. Widely used asymmetric encryption include AES and ECC. Asymmetric encryption is beneficial for key exchange and digital signatures, but it tends to be slower and more computationally demanding than symmetric encryption.
Strengths:
Key distribution: Asymmetric encryption eliminates shared secret key method, as both party has a public-private key pair.
Key exchange: Asymmetric encryption provides secure key exchange, allowing parties to establish secure communication channels[14].
Digital signatures: Asymmetric encryption enables secure transmission of digital signatures, ensuring data integrity and authenticity[14].

Limitations:
Computational complexity: Asymmetric encryption operations, particularly key generation and decryption, consumes more resources and are slower as compared to symmetric encryption.
Key size: Asymmetric encryption requires larger key sizes for equivalent security levels compared to symmetric encryption, resulting in increased computational overhead.

III. ADVANCED ENCRYPTION STANDARD (AES)

Advanced Encryption Standard (AES) is a symmetric encryption algorithm which was developed and adopted by the U.S. government for securing sensitive data. While DES only operates on 64 bits block size and 48 bits key size, AES operates on 128 bits keys and three different key length i.e. 128, 192, 256 bits, offering strong protection against various cryptographic attacks.

A. Encryption Process

The encryption process in AES involves several key steps:
Key Expansion: The original encryption key undergoes a key expansion process, generating a set of round keys used in the subsequent encryption rounds.
Initial Round: The plaintext block is XORed with the initial round key.
Rounds: Every round in AES consists of four rounds in itself, namely: SubBytes, ShiftRows, MixColumns, and AddRoundKey.
Final Round: In the final round, the MixColumns stage is omitted, simplifying the process.
Output: The resulting ciphertext block represents the encrypted data.

B. Strengths:

Security: AES offers robust protection against various cryptographic attacks, such as brute force, differential and linear cryptanalysis.
Efficiency: AES is commonly used for encrypting large chunk of data and is computationally efficient.
Standardization: AES is a widely adopted standard, ensuring interoperability and compatibility across different platforms and systems.

C. Weaknesses:

Key Management: Although AES itself is considered secure, the overall security of encrypted data greatly depends on effective key management procedures,

encompassing tasks such as generating, distributing, and securely storing keys.

Vulnerability to Side-channel Attacks: AES implementations may be vulnerable to side-channel attacks, such as timing and power analysis attacks, if not properly secured.

D. Use Cases:

AES finds extensive application in various domains due to its strong security and efficiency. Some common use cases include:

Data Encryption: AES is employed to secure confidential and integrity of information stored on various devices such as hard drives, USB drives, and smartphones.

Secure Communication: AES is employed in secure communication protocols, including SSL/TLS, VPNs, and secure messaging applications, to protect data transmitted over networks.

File and Disk Encryption: AES is utilized to encrypt files and disk volumes, providing secure storage and confidentiality.

Cloud Security: AES encryption is integral to securing data stored in cloud environments, safeguarding it from unauthorized access and data breaches[12].

IV. RIVEST SHAMIR ADLEMAN (RSA)

Rivest Shamir Adleman (RSA) is an asymmetric encryption algorithm developed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977. Even after all these years it isl one of the most used and studied public-key encryption schemes. RSA relies on the mathematical properties of large prime numbers and modular arithmetic for its security. The algorithm uses public and private key for encryption and decryption, which ensures secure communication and transfer of digital signatures.

A. Encryption Process:

The encryption process in RSA involves several key steps:

i. Key Generation:

- Select two large prime numbers, p and q , calculate,

$$n = p*q,$$ which serves as the modulus

- Select an integer e such that $1 < e < phi(n)$, where

$$phi(n) = (p-1)(q-1)$$

where e is coprime with phi(n).

- Calculate private key d such that

$$d = inverse(e)*phi(n)$$

- Generated public key (n, e)
- Generated private key (d,n)

ii. Encryption: Cipher text is calculated using the previously generated public key (n,e)

$$c = m^e \bmod n.$$

iii. Decryption: on receiver side plain text is calculated using the previously generated private key (d,n)

$$m = c^d \bmod n.$$

B. Strengths:

Security: RSA uses large prime numbers that increases the difficulty to generate factors of the numbers which makes it computationally hard.

Key Exchange: RSA provides secure key exchange and digital signatures without requiring pre-shared secrets.

Versatility: RSA is versatile and can be used for encryption, digital signatures, and key exchange in various cryptographic protocols.

C. Weaknesses:

Computational Complexity: RSA encryption and decryption operations involve modular exponentiation, that are computationally intensive, especially for large key sizes.

Key Size: As computing power increases, larger key sizes are required to maintain security, leading to increased computational overhead.

Vulnerability to Quantum Computing: RSA encryption is vulnerable to quantum computing attacks, particularly Shor's algorithm, that can efficiently factor large integers.

D. Use Cases:

RSA is widely employed in numerous applications across various domains due to its strong security and versatility. Some common use cases include:

Secure Communication: RSA is mainly used in secure communication protocols, such as SSL/TLS, HTTPS, and SSH, to establish secure channels and exchange symmetric encryption keys.

Digital Signatures: RSA is utilized for creating and verifying digital signatures, providing data integrity and authenticity in electronic transactions, digital documents, and software distribution.

Key Exchange: RSA facilitates secure key exchange in key management systems and cryptographic protocols, enabling secure communication between parties without prior arrangements[13].

Identity Authentication: RSA-based authentication mechanisms, such as SSH key pairs and digital certificates, are used to authenticate users, devices, and services in network environments.

V. OTHER ENCRYPTION TECHNIQUES

A. Elliptic Curve Cryptography (ECC):

Overview: ECC is an asymmetric encryption technique grounded in elliptic curve mathematics over finite fields. It offers robust security with shorter key lengths compared to traditional asymmetric algorithms like RSA, making it particularly suitable for devices and environments with limited resources[5].

Strengths: ECC ensures high security with shorter key lengths, resulting in decreased computational and storage requirements. It demonstrates strong resilience against various cryptographic attacks, including brute force and quantum computing attempts.

Weaknesses: Implementation of ECC can be intricate due to factors like curve selection, key sizes, and cryptographic protocol considerations. Additionally, ECC may be vulnerable to exploitation and side-channel attacks if not adequately safeguarded.

Use Cases: ECC finds widespread usage in applications requiring strong security within resource-constrained environments, such as mobile devices, smart cards, IoT devices, and cryptographic protocols like TLS/SSL and PGP.

B. Diffie-Hellman Key Exchange:

Overview: Diffie-Hellman Key Exchange method allows two parties to setup a shared private key for secure communication over an insecure channel. It allows for secure key exchange without requiring pre-shared secrets and serves as the foundation for numerous cryptographic protocols such as SSL/TLS and SSH[6].

Strengths: Diffie-Hellman Key Exchange offers secure key exchange without dependence on pre-shared secrets, providing forward secrecy and resilience against passive eavesdropping attacks. It is computationally efficient and widely supported in cryptographic libraries and protocols.

Weaknesses: Without authentication mechanisms, Diffie-Hellman Key Exchange is vulnerable to man-in-the-middle attacks. Certain versions, like the original non-ephemeral variant, may be vulnerable to cryptographic attacks.

Use Cases: Diffie-Hellman Key Exchange is utilized in various cryptographic applications to establish secure communication channels, including SSL/TLS for secure web browsing, VPNs for secure remote access, and secure email protocols like PGP and S/MIME.

C. Blowfish:

Overview: Developed by Bruce Schneier in 1993, Blowfish is a symmetric block cipher designed as an alternative to encryption algorithms like DES. Operating on variable-length blocks and key sizes, Blowfish ensures swift encryption and decryption alongside robust security[4].

Strengths: Blowfish offers strong security through a straightforward and efficient algorithm, rendering it suitable for a broad spectrum of applications. It withstands known cryptographic attacks and accommodates key size that ranges from 32 to 448 bits.

Weaknesses: Despite its resilience, Blowfish has been surpassed by newer encryption algorithms such as AES, which offer standardized security and performance advantages. Additionally, Blowfish may not have undergone as comprehensive scrutiny as widely adopted encryption standards.

Use Cases: Blowfish has found utility in diverse applications requiring encryption and data safeguarding, including file and disk encryption, secure communication protocols, and password hashing.

D. ChaCha20:

Overview: ChaCha20, conceived by Daniel J. Bernstein, is a symmetric encryption algorithm known for its efficiency and security[3]. Derived from the ChaCha stream cipher, a variation of the Salsa20 cipher family, ChaCha20 is often paired with the Poly1305 authenticator for authenticated encryption[11].

Strengths: Known for its exceptional efficiency, ChaCha20 excels in software implementations, making it particularly well-suited for mobile and IoT devices. With a 256-bit key size, it offers strong security and resilience against established cryptographic attacks.

Weaknesses: ChaCha20 may encounter limitations in adoption and standardization compared to more prevalent encryption algorithms like AES, potentially affecting interoperability in certain contexts. Additionally, the security of ChaCha20 relies on the integrity of its underlying cryptographic components.

Use Cases: ChaCha20 finds widespread application in cryptographic protocols and scenarios demanding both high performance and security. It is commonly employed in TLS/SSL, VPNs, disk encryption, and secure messaging platforms like Signal and WhatsApp.

VI. CASE STUDIES:

Cryptographic Algorithm	Encryption Strength	Computational Complexity	Key Management	Scalability	Versatility
AES	High	Medium	Moderate	High	High
RSA	High (with adequate key length)	High	Complex	Moderate	Moderate
ECC	High (with adequate key length)	Medium	Complex	High	High
Diffie-Hellman	High (with adequate key length)	Medium	Moderate	High	Moderate
Blowfish	High (with adequate key length)	Low	Simple	High	High
ChaCha20	High	Low	Simple	High	High

Fig 1.1

Fig 1.1 Shows comparison of the RSA and AES with other discussed algorithms

A. AES and RSA: Real-World Applications:

Case Study 1: Secure Communication in Health-care

Scenario: A health-care organization implements secure communication protocols to protect patient

health information (PHI) transmitted between medical professionals, hospitals, and insurance providers[1].

AES Implementation: AES is utilized for symmetric encryption in virtual private networks (VPNs) and secure socket layer/transport layer security (SSL/TLS) connections. It ensures confidentiality and integrity of PHI during transmission.

RSA Implementation: RSA is used for key exchange and digital signatures in SSL/TLS handshakes. It facilitates secure authentication and establishes trust between communication endpoints[2].

Outcome: The combination of AES and RSA enables secure communication channels, safeguarding PHI against unauthorized access.

Case Study 2: Financial Transactions Security

Scenario: A vendor deploys encryption to secure the online transactions and customer data.

AES Implementation: AES is used for encrypting sensitive financial data, such as account numbers and transaction details, during online banking sessions. It provides strong encryption to prevent unauthorized interception and tampering of financial information.

RSA Implementation: RSA is employed for securing the digital signatures used in financial transactions. It ensures the authenticity of transaction requests and protects against fraudulent activities, such as unauthorized fund transfers.

Outcome: By leveraging AES and RSA, the financial institution ensures the confidentiality, integrity, and authenticity of online banking transactions, enhancing customer trust and confidence in the security of their financial data.

B. Performance Evaluation:

Case Study 1: Secure Messaging Application

Scenario: A messaging application provider evaluates the performance of encryption algorithms to ensure fast and secure message delivery.

AES Performance: AES encryption and decryption speeds are benchmarked on various platforms, including mobile devices and servers. It demonstrates fast performance with minimal latency, making it suitable for real-time messaging applications.

RSA Performance: RSA key negotiation and cryptographic operations are evaluated for secure communication establishment. While RSA offers strong security guarantees, its computational overhead results in higher latency, especially for large key sizes.

Outcome: Based on performance evaluation, the messaging application provider selects AES for symmetric encryption of message contents due to its fast and efficient performance. RSA is highly esteemed for facilitating key exchange and digital signatures, effectively managing security concerns while ensuring reasonable latency for timely message delivery.

Case Study 2: Cloud Data Encryption

Scenario: A cloud service provider assesses encryption algorithms for securing data-at-rest stored on cloud servers.

AES Performance: AES encryption and decryption speeds are measured for encrypting and decrypting large files stored in cloud storage. AES demonstrates high throughput and low latency, ensuring minimal impact on file upload and download speeds.

RSA Performance: RSA key generation and cryptographic operations are tested for secure key exchange and access control in the cloud environment. While RSA offers secure key management, its computational complexity results in slower performance compared to AES.

Outcome: The cloud service provider leverages AES for encrypting data-at-rest on cloud servers due to its high performance and efficiency. RSA is employed for secure key exchange and access control mechanisms, prioritizing security over performance in key management operations.

These case studies highlight the real-world applications and performance considerations of AES and RSA in various scenarios, showcasing their strengths uses in security, efficiency, and scalability. Organizations can utilize this information to take

decisions while choosing encryption techniques for their needs and performance requirements.

VII. CHALLENGES AND FUTURE DIRECTIONS

A. Emerging Threats

As encryption technologies evolve, so do the threats posed by adversaries. Some emerging challenges include:

Quantum Computing: The advancement of quantum computing threatens the security of traditional encryption algorithms, such as RSA and ECC, that uses mathematical problems which can solved efficiently using quantum computers. Post-quantum cryptography research aims to develop encryption algorithms resistant to quantum attacks[10].

Side-channel Attacks: Attackers utilize leakage from physical implementations of encryption algorithms to gain information. Mitigating side-channel attacks requires robust cryptographic implementations and secure hardware/software design practices.

Cryptanalysis Advances: Cryptanalysts continuously develop new attack techniques and exploit vulnerabilities in encryption algorithms, underscoring the need for ongoing research and scrutiny of cryptographic primitives.

B. Advances in Encryption Techniques

To mitigate security threats, advancements in encryption techniques are crucial. Some areas of innovation include:

Post-Quantum Cryptography: the research aims on developing encryption algorithms that are unsusceptible to quantum attacks, such as lattice-based cryptography, hash-based cryptography, and code-based cryptography[9].

Homomorphic Encryption: this encryption allows computation on cipher text, which allows secure outsourced computation and privacy-preserving data analysis. Advancements in homomorphic encryption techniques enhance data privacy and security in cloud computing and data analytics.

Fully Homomorphic Encryption (FHE): FHE enables arbitrary computations on encrypted data without decrypting it, offering unprecedented levels of privacy and security. Research efforts focus on improving the efficiency and practicality of FHE for real-world applications.

Blockchain and Cryptocurrency Security: Encryption is the fundamental building block for securing blockchain networks and cryptocurrency transactions. Innovations in cryptographic primitives, zero-knowledge proofs, and consensus mechanisms enhance the security and scalability of blockchain-based systems.

C. Standardization Efforts

Standardization plays a vital role in promoting interoperability, compatibility, and security across cryptographic systems. Some ongoing standardization efforts include:

NIST Cryptographic Standards: The National Institute of Standards and Technology (NIST) develops security frameworks and encryption techniques (e.g., AES), asymmetric encryption (e.g., RSA, ECC), hash functions, and random number generators[7]. NIST's Cryptographic Algorithm Validation Program (CAVP) ensures compliance and interoperability of cryptographic implementations[8].

IEEE P1363 Working Group: the group focuses on standardizing public-key cryptographic techniques, including RSA, ECC, and digital signatures. IEEE standards facilitate interoperability and ensure the integrity and security of cryptographic systems.

IETF (Internet Engineering Task Force): The IETF develops and maintains cryptographic rules and standards for secure communication over the Internet, including SSL/TLS, IPsec, and DNSSEC. IETF's rigorous review process ensures the security, privacy, and interoperability of Internet protocols.

VIII. CONCLUSION

In an age characterized by extensive digital interconnections and the proliferation of data, encryption plays an important role in communication and data transmission by ensuring confidentiality and privacy of data. This research has thoroughly examined encryption algorithms such as AES (Advanced Encryption Standard), RSA (Rivest Shamir Adleman), and emerging cryptographic approaches, providing a comprehensive analysis of their strengths, weaknesses, and practical applications.

Through comparative analysis, we have investigated crucial factors such as encryption strength, computational complexity, key management, scalability, and versatility, offering insights into the trade-offs inherent in different encryption methods. While AES demonstrates efficiency and robustness in symmetric encryption tasks, RSA and Elliptic Curve Cryptography (ECC) provide exceptional security for asymmetric encryption and key exchange.

Furthermore, this paper has addressed the challenges and future directions in encryption, discussing emerging threats, advancements in encryption techniques, and standardization efforts aimed at enhancing the security and resilience of cryptographic systems. From addressing quantum computing threats to advancing post-quantum cryptography, from exploring homomorphic encryption to driving standardization initiatives, the encryption landscape continues to evolve in response to the dynamic cybersecurity environment.

II. REFERENCES

[1]. Daemen, J., & Rijmen, V. (2002). The design of Rijndael: AES - the Advanced Encryption Standard. Springer Science & Business Media.

[2]. Rivest, R. L., Shamir, A., & Adleman, L. M. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[3]. Bernstein, D. J. (2007). The ChaCha stream cipher. In New Stream Cipher Designs (pp. 90-171). Springer, Berlin, Heidelberg.

[4]. Schneier, B. (1994). Description of a new variable-length key, 64-bit block cipher (Blowfish). In Fast Software Encryption (pp. 191-204). Springer, Berlin, Heidelberg.

[5]. Lange, T., & Bernstein, D. J. (2006). Faster addition and doubling on elliptic curves. In Advances in Cryptology-EUROCRYPT 2006 (pp. 29-50). Springer, Berlin, Heidelberg.

[6]. Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. IEEE Transactions on Information -Theory, 22(6), 644-654.

[7]. National Institute of Standards and Technology (NIST). (2001). FIPS PUB 197: Advanced Encryption Standard (AES). Retrieved from https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

[8]. National Institute of Standards and Technology (NIST). (2016). FIPS PUB 186-4: Digital Signature Standard (DSS). Retrieved from https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

[9]. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V., Schwabe, P., & Seiler, G. (2019). Lattice Signatures and Bimodal Gaussians. In Advances in Cryptology–CRYPTO 2019 (pp. 537-566). Springer, Cham.

[10]. National Institute of Standards and Technology (NIST). (2019). NISTIR 8105: Report on Post-Quantum Cryptography. Retrieved from https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8105.pdf

[11]. Rogaway, P. (2011). The sponge construction. In Fast Software Encryption (pp. 443-457). Springer, Berlin, Heidelberg.

[12]. Daemen, J., & Rijmen, V. (2002). The design of Rijndael: AES - the Advanced Encryption Standard. Springer Science & Business Media.

[13]. Rivest, R. L., Shamir, A., & Adleman, L. M. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[14]. Morteza SaberiKamarposhti, Amirabbas Ghorbani, Mehdi Yadollahi. "A comprehensive survey on image encryption: Taxonomy, challenges, and futuredirections", Chaos, Solitons & Fractals, 2024