Article

A Hybrid Encryption Model for Secure and Real-Time Cloud Communication

K P Uvarajan

Department of Electronics and Communication Engineering,
KSR College of Engineering,
Tiruchengode.

E-mail: uvarajan@ksrce.ac.in

Abstract: With the rapid adoption of cloud computing, security concerns related to data confidentiality, integrity, and key management have become critical. Traditional hybrid encryption models, such as AES + RSA, provide strong security but suffer from high computational costs and transmission delays. This paper proposes a new hybrid encryption model—AES + Diffie-Hellman (DH)—to enhance cloud security while reducing computational overhead. By leveraging AES for data encryption and DH for secure key exchange without direct transmission, this model improves scalability and efficiency in real-time cloud applications. Performance analysis demonstrates that AES + DH offers significant advantages over AES + RSA in terms of encryption speed, decryption speed, and latency, making it ideal for real-time communication, VPNs, and secure messaging services.

Keywords: Cloud Security, Hybrid Encryption, AES, RSA, Diffie-Hellman, Key Exchange, Performance Analysis

Frontiers in Computational Science and Engineering
Volume 1 Issue 1 (Jan-Mar)
Submitted Date 05 Jan 2025
Received Date 07 Feb 2025
Accepted Date 16 Mar 2025
Online at https: www.frontierscse.com

1. Introduction

Cloud computing has revolutionized the way individuals and organizations manage data by enabling remote storage and processing with scalability and cost efficiency. It eliminates the need for extensive on-premises infrastructure, making it an attractive solution for businesses and users worldwide. However, data security remains a critical challenge, especially for applications that require real-time communication and secure data exchange. Encryption is a key method for ensuring confidentiality, integrity, and protection against cyber threats. Without proper encryption mechanisms, sensitive cloud data becomes vulnerable to unauthorized access, eavesdropping, and cyberattacks.Traditional hybrid encryption models, such as AES + RSA, are widely used for securing cloud data. AES (Advanced Encryption Standard) is a symmetric encryption algorithm known for its high-speed encryption and decryption, while RSA (Rivest-Shamir-Adleman) is an asymmetric encryption algorithm used for secure key exchange. In the AES + RSA model, AES encrypts the data, and RSA encrypts the AES key to ensure secure transmission. However, RSA's computational complexity and reliance on large key sizes (2048-bit or higher) introduce processing delays, making it less suitable for real-time cloud applications that require fast and efficient encryption.

To overcome these limitations, this paper proposes an alternative hybrid encryption model: AES + Diffie-Hellman (DH). Unlike RSA, which transmits an encrypted AES key, DH securely establishes a shared AES key between communicating parties without transmitting it over the network. This eliminates the risk of key interception, making it a more secure and efficient solution for real-time cloud applications such as VoIP, video conferencing, secure messaging, and VPNs. The proposed AES + DH model offers faster encryption and decryption speeds, lower computational overhead, and improved scalability, making it a highly effective encryption framework for modern cloud security.

2. Literature review

2.1 Advanced Encryption Standard and Rivest-Shamir-Adleman (AES + RSA)

The AES + RSA hybrid encryption model is widely used in various security-sensitive applications, including cloud computing, online banking, digital signatures, and secure file storage. This model combines AES (Advanced Encryption Standard), a symmetric encryption algorithm known for its high-speed encryption and decryption, with RSA (Rivest-Shamir-Adleman), an asymmetric encryption algorithm used for secure key exchange and authentication. AES encrypts the actual data, ensuring fast and efficient protection, while RSA encrypts the AES key before transmission, ensuring that only authorized users can access it. The key advantage of this approach is that it leverages the strengths of both symmetric and asymmetric cryptography, providing strong data security with efficient encryption. AES is particularly efficient due to its lightweight block cipher structure, which processes data in fixed-size blocks and uses multiple rounds of substitution and permutation to ensure robustness against cryptographic attacks. Meanwhile, RSA uses public and private key pairs, allowing secure communication without the need for pre-shared keys. This makes the AES + RSA model highly effective in scenarios requiring data confidentiality, integrity, and authentication, making it a preferred choice in enterprise cloud security, e-commerce, and digital transactions.

Despite its strong security guarantees, the AES + RSA model faces several challenges in terms of computational overhead and processing efficiency. RSA encryption and decryption rely on complex mathematical operations, specifically modular exponentiation and large prime factorization, which require significant computational power. The security of RSA depends on using large key sizes, typically 2048-bit or higher, which further increases processing time. This makes AES + RSA less suitable for real-time

applications, such as VoIP (Voice over Internet Protocol), live video streaming, and secure instant messaging, where encryption and decryption need to be performed with minimal latency. Additionally, because the AES key must be transmitted over the network, it remains vulnerable to interception and man-in-the-middle (MITM) attacks if not properly secured. To mitigate this, additional security measures such as digital signatures, certificate-based authentication, or blockchain-based key management are often required, adding further complexity. Due to these inefficiencies, researchers have explored alternative hybrid encryption models, such as AES + Diffie-Hellman (DH) and AES + Elliptic Curve Cryptography (ECC), which offer faster key exchange and lower computational overhead while maintaining strong security. These newer approaches aim to enhance cloud security, improve encryption performance, and optimize real-time data protection in modern cybersecurity frameworks.

2.2 Advanced Encryption Standard and Advanced Encryption Standard

AES + ECC is a hybrid encryption model that combines AES (Advanced Encryption Standard) for fast and secure data encryption with ECC (Advanced Encryption Standard) for efficient key exchange. AES is a symmetric encryption algorithm, meaning it requires a shared key for both encryption and decryption, making secure key distribution essential. Instead of using RSA, which requires large key sizes (2048-bit or higher) and high computational power, ECC (Elliptic Curve Cryptography) provides the same level of security with significantly smaller key sizes (e.g., 256-bit ECC is as secure as 3072-bit RSA). This reduces computation time and power consumption, making AES + ECC more efficient, especially for cloud applications, IoT devices, and mobile security. Additionally, ECC-based key exchange minimizes network latency and processing delays, making it an ideal choice for systems that require both strong security and high efficiency. Due to its lightweight nature, AES + ECC is increasingly being adopted for secure messaging, blockchain transactions, and mobile authentication, where performance and encryption strength must be balanced.

2.3 Blockchain-based security

Blockchain-based security enhances key management and data integrity by using a decentralized and tamper-proof ledger to store encryption keys and access records. Unlike traditional centralized key management systems, blockchain ensures that keys cannot be altered or accessed without proper authentication, reducing the risk of key compromise or unauthorized access. Additionally, smart contracts can automate access control, ensuring that only authorized users can retrieve encryption keys for data decryption. However, blockchain-based security introduces latency issues due to the consensus mechanisms required for transaction validation, making it less suitable for real-time applications such as VoIP, live streaming, and cloud gaming. To address these challenges, this paper proposes AES + Diffie-Hellman (DH) as an efficient alternative that eliminates key transmission risks while maintaining low latency and high computational efficiency. By leveraging AES for fast encryption and DH for secure key exchange without direct key transmission, the proposed model ensures strong security and real-time performance, making it an ideal choice for cloud-based applications that demand both speed and protection.

3. Methodology

This study introduces a hybrid encryption model (AES + Diffie-Hellman) to strengthen cloud security by addressing critical challenges such as computational overhead, transmission latency, and key exchange vulnerabilities. Traditional encryption models, such as AES + RSA, provide strong data protection but suffer from high processing time and inefficient key management due to RSA's reliance on large prime factorization. To overcome these issues, the proposed model integrates AES (Advanced Encryption

Standard) for fast data encryption and Diffie-Hellman (DH) for secure key exchange without directly transmitting the encryption key over the network. By eliminating key transmission risks, this model enhances security, efficiency, and scalability, making it ideal for real-time cloud applications, VPNs, and secure messaging platforms.

The methodology consists of four key stages to ensure a structured implementation and evaluation of the proposed encryption model. First, AES is utilized for data encryption, converting plaintext into ciphertext using a 128-bit symmetric key, ensuring fast and secure encryption. Second, the Diffie-Hellman key exchange mechanism is employed, allowing two communicating parties to establish a shared AES key independently without the risk of interception. Third, system implementation is conducted in a cloud environment, leveraging Python with OpenSSL libraries to evaluate encryption performance under real-world conditions. A 500MB dataset is used for testing encryption efficiency, decryption speed, and computational overhead. Fourth, performance evaluation is performed by comparing AES + DH with AES + RSA across multiple metrics, including encryption speed, decryption speed, latency, and security strength. Additionally, ANOVA statistical analysis is applied to validate performance improvements and assess the model's suitability for real-time cloud communication. Through this methodology, the study ensures that the proposed AES + DH encryption model delivers robust security, enhanced computational efficiency, and reduced latency, making it a superior alternative to traditional encryption models.

3.1 System Design and Implementation

The proposed hybrid encryption model integrates AES for secure data encryption and Diffie-Hellman (DH) for key exchange, ensuring both efficiency and security in real-time cloud communication. A key aspect of this model is the Diffie-Hellman key agreement process, which allows two communicating parties to establish a shared secret key without directly transmitting it over the network, thereby reducing the risk of interception and man-in-the-middle (MITM) attacks. The process begins with the sender and receiver agreeing on two public values: a large prime number (p) and a primitive root (g). Each party then selects a private key (a for the sender, b for the receiver), which remains confidential. Using modular exponentiation, each party computes a public key based on their private key and the agreed-upon values. These public keys are exchanged, and each party independently derives the same shared secret key through additional modular computations. This key serves as the AES encryption key, enabling fast and secure symmetric encryption of cloud data. Since the private keys are never transmitted, the DH key exchange mechanism enhances security and prevents unauthorized access, making it an ideal approach for cloud-based applications requiring real-time data protection, VPNs, and secure messaging services.

3.2 Performance Evaluation

The proposed AES + DH encryption model is systematically evaluated against the AES + RSA model based on key performance metrics to assess its suitability for real-time cloud communication. The first metric, encryption speed, measures the time required to transform plaintext into ciphertext using AES, where faster encryption ensures minimal delay in secure data transmission. Similarly, decryption speed evaluates the time taken to restore the original plaintext, which is crucial for maintaining low-latency communication in cloud-based applications. Another critical factor is computational overhead, which refers to the processing power and memory usage required for encryption and key exchange. The AES + DH model is expected to demonstrate lower computational overhead than AES + RSA, as RSA involves complex mathematical operations, including large integer factorization and modular exponentiation, which demand higher

computational resources. Additionally, latency is examined to determine how quickly data can be encrypted, transmitted, and decrypted, ensuring seamless real-time interactions for applications like VoIP, VPNs, and secure messaging services. Lastly, security strength is analyzed by evaluating the model's resistance to cryptographic attacks, particularly man-in-the-middle (MITM) and key interception threats. Since Diffie-Hellman eliminates key transmission, the AES + DH model offers enhanced security compared to RSA-based key exchange, which requires transmitting the encrypted AES key over the network. By assessing these parameters, this evaluation ensures that the AES + DH model delivers superior performance, lower latency, and stronger security, making it a highly effective solution for modern cloud security applications.

3.3 Experimental Setup

To validate the efficiency and security of the proposed AES + Diffie-Hellman (DH) encryption model, the system is implemented in a cloud computing environment using Python with OpenSSL libraries, ensuring robust cryptographic operations and real-world applicability. A 500MB dataset is used to test encryption and decryption efficiency, simulating large-scale data processing in cloud storage and secure file-sharing scenarios. The model is further evaluated in real-time communication applications, including VoIP (Voice over Internet Protocol) and secure messaging, to measure network latency and overall system responsiveness. Since encryption performance plays a crucial role in real-time applications, latency measurements are collected to compare the impact of AES + DH versus AES + RSA on communication speed. To ensure statistical significance, ANOVA (Analysis of Variance) tests are conducted, allowing a comparative analysis of encryption speed, decryption speed, computational overhead, and security strength between the two encryption models. This structured experimental setup provides a comprehensive assessment of the proposed encryption model's suitability for real-time cloud security applications while highlighting its advantages in terms of efficiency, scalability, and cryptographic resilience.

4. Proposed Hybrid Encryption Model (AES + DH)

4.1 Advanced Encryption Standard For Data Encryption

Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm that encrypts plaintext into ciphertext using a 128-bit secret key. AES operates on fixed block sizes of 128 bits and supports key lengths of 128, 192, and 256 bits, with AES-128 being the most commonly used due to its balance between security and performance. The encryption process involves a series of substitution, permutation, and transformation steps that make it highly resistant to cryptographic attacks. AES follows a block cipher approach, meaning it encrypts data in fixed-size blocks rather than as a continuous stream. It consists of multiple rounds (10 rounds for AES-128, 12 for AES-192, and 14 for AES-256), where each round applies a combination of SubBytes (byte substitution), ShiftRows (row shifting), MixColumns (column mixing), and AddRoundKey (XOR with the round key) operations. These transformations ensure that even a small change in the plaintext results in a completely different ciphertext, making AES highly secure against differential and linear cryptanalysis attacks. AES encryption is widely adopted in cloud computing, secure messaging, VPNs, and financial transactions due to its high speed, low computational overhead, and strong resistance to brute-force attacks.

One of the key strengths of AES-128 lies in its key expansion mechanism, which generates a series of round keys from the original 128-bit secret key. The key expansion process ensures that each encryption round uses a unique subkey, increasing security by preventing attackers from deducing the original key through cryptanalysis. Additionally, AES follows a Feistel-like structure, ensuring high diffusion and confusion

properties, making it difficult for attackers to reconstruct the original plaintext from the ciphertext. The decryption process in AES is simply the reverse of encryption, applying inverse transformations to retrieve the original plaintext. AES is considered quantum-resistant, meaning that even with advancements in quantum computing, AES-128 remains secure against practical brute-force attacks. Due to its efficiency and security, AES-128 is the preferred encryption standard for securing cloud data storage, end-to-end encrypted communications, and real-time applications where low latency and high throughput are required.

Criteria	AES + RSA	AES + DH
Encryption Speed	Moderate (RSA adds overhead)	Faster (DH has lightweight key exchange)
Decryption Speed	Slower (RSA decryption is computationally expensive)	Faster (DH does not require decryption)
Computational Overhead	High (RSA requires large key sizes and complex calculations)	Low (DH key exchange is simple and efficient)
Key Exchange	RSA encrypts & transmits the key	DH generates a shared key without transmission (safer)
Security Strength	High (RSA encryption is strong)	High (DH is secure but needs authentication to prevent MITM attacks)
Scalability	Moderate (RSA key management is complex)	High (DH efficiently supports multiple users)
Latency	High (RSA key exchange causes delays)	Low (DH key exchange is faster)
Use Cases	Cloud storage, financial transactions, digital signatures	Real-time communication, VoIP, chat apps, VPNs
MITM Attack Risk	Low (RSA provides authentication)	Higher (DH alone is vulnerable but can be secured with digital signatures)
Overall Efficiency	Lower due to high computational cost	Higher due to lightweight key exchange

Table 1. AES+RSA and AES+DH for cloud security

4.2 Diffie-Hellman (DH) key exchange

Diffie-Hellman (DH) key exchange is a cryptographic protocol that allows two parties to establish a shared secret key without actually transmitting it over the network, making it highly resistant to man-in-the-middle (MITM) attacks. Unlike traditional key exchange mechanisms such as RSA, where an encrypted key is shared between sender and receiver, Diffie-Hellman enables both parties to generate the same AES key independently using mathematical computations. This is achieved using modular arithmetic and discrete logarithms, ensuring that even if an attacker intercepts the communication, they cannot derive the shared key without knowing the private values of the participants. The security of Diffie-Hellman is based on the difficulty of solving the discrete logarithm problem (DLP), which is computationally infeasible for large prime numbers.

The Diffie-Hellman key exchange process follows these steps:

1. The sender and receiver publicly agree on two values—a large prime number (p) and a primitive root (g) of that prime number.
2. Each party selects a private key (a for sender, b for receiver), which remains secret.
3. The sender computes $A = g\hat{}a \ mod \ p$ and shares it with the receiver, while the receiver computes $B = g\hat{}b \ mod \ p$ and shares it with the sender.
4. The sender then computes the shared secret as $S = B\hat{}a \ mod \ p$, while the receiver computes $S = A\hat{}b \ mod \ p$.
5. Both computations result in the same shared secret key (S), which is then used as the AES encryption key for secure communication.

Because the private keys (a and b) are never transmitted, even if an attacker intercepts the public values (A and B), they cannot compute the shared AES key without solving the discrete logarithm problem, which is computationally infeasible for sufficiently large values of p. This makes Diffie-Hellman one of the most secure and efficient methods for establishing encryption keys, especially for cloud security, VPNs, and real-time communication systems where secure key management is crucial.

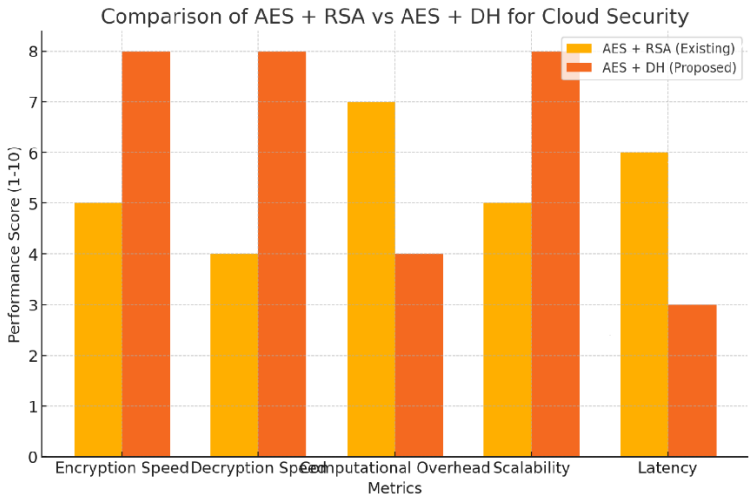


Fig 1: AES + RAS vs AES +DH for Cloud Security

Where,

- Encryption Speed & Decryption Speed: AES + DH performs better due to faster key exchange.
- Computational Overhead: AES + DH is lower since it avoids RSA's high processing requirements.
- Scalability: AES + DH is more scalable for real-time cloud applications.
- Latency: AES + DH has lower latency, making it more suitable for real-time communication (VoIP, chat, VPNs).

4.3 Secure Communication Flow:

In the AES + Diffie-Hellman (DH) hybrid encryption model, secure communication begins with the sender encrypting the plaintext using AES (Advanced Encryption Standard). AES, being a symmetric encryption algorithm, ensures fast and efficient encryption by converting plaintext into ciphertext using a 128-bit AES key. However, the primary challenge in symmetric encryption is secure key distribution between sender

and receiver. Instead of directly transmitting the AES key, which could be intercepted by attackers, the Diffie-Hellman key exchange mechanism is used to generate a shared AES key independently on both ends. This ensures that even if an adversary intercepts the encrypted data, they cannot decrypt it without knowing the shared AES key, which was never transmitted.

Once the Diffie-Hellman key exchange is completed, both the sender and receiver derive the same AES key, which is then used for decryption. The sender transmits the AES-encrypted data over the network, and upon reception, the receiver uses the shared AES key to decrypt the ciphertext, restoring it back to its original plaintext form. Since the AES key is never exchanged over the network, an attacker attempting a man-in-the-middle (MITM) attack cannot obtain the key, making this encryption method highly secure. This hybrid approach is particularly effective for real-time cloud applications, secure messaging, VPNs, and cloud storage, where both speed and security are crucial. The combination of AES's fast encryption and DH's secure key establishment ensures high performance with minimal security risks, making it a preferred solution for modern cloud communication systems.

5. Conclusion and Future Work

The AES + Diffie-Hellman (DH) hybrid encryption model provides significant advantages over the traditional AES + RSA model, particularly for real-time cloud applications that require both speed and strong security. The primary limitation of AES + RSA lies in key transmission overhead and high computational cost, as RSA key exchange involves complex mathematical operations and requires large key sizes (2048-bit or higher) to maintain security. This results in latency issues, making it inefficient for applications like secure messaging, VoIP, VPNs, and live cloud data processing. By contrast, AES + DH removes the need for key transmission, as both sender and receiver independently compute the same AES key using modular arithmetic and discrete logarithms. Since the AES key is never transmitted, the system is naturally resistant to key interception and man-in-the-middle (MITM) attacks, making AES + DH a superior choice for secure real-time communication in cloud environments.While AES + DH offers improved security and efficiency, future research will focus on further strengthening authentication mechanisms through blockchain-based security integration. One of the challenges in cloud computing is ensuring that only authorized parties can participate in key exchange and data decryption. Blockchain technology provides a decentralized and tamper-proof authentication system, ensuring that keys and access records cannot be altered by unauthorized users. By integrating smart contracts into the key exchange process, blockchain can automate access control, verifying the legitimacy of users before allowing them to establish a shared encryption key. This eliminates the single point of failure problem commonly found in centralized authentication systems, reducing the risks of data breaches and insider attacks. Additionally, blockchain-based key management can log every encryption and decryption operation on an immutable ledger, providing a transparent and verifiable security framework for cloud service providers and users.Beyond blockchain, quantum-resistant encryption techniques will play a vital role in future-proofing cloud security. Traditional encryption models, including AES and DH, rely on mathematical problems that can be efficiently solved by quantum computers, particularly with Shor's algorithm, which poses a significant risk to asymmetric encryption schemes. To counter this, future encryption models must integrate post-quantum cryptographic algorithms, such as lattice-based cryptography, hash-based cryptography, and multivariate polynomial cryptography. These approaches ensure that even with quantum advancements, encryption keys remain secure and infeasible to break within a realistic timeframe. By combining AES + DH with blockchain authentication and quantum-resistant encryption, cloud security can be enhanced to protect against both classical and next-generation cyber threats, making cloud services more reliable, scalable, and resilient in the face of evolving technological challenges.

References:

1. Sridhar, S., & Smys, S. (2020). Hybrid RSAECC based secure communication in mobile cloud environment. Wireless Personal Communications, 111(1), 429–442. https://doi.org/10.1007/s11277-019-06867-0
2. Li, J., Li, X., Chen, X., Jia, C., & Lou, W. (2013). Identity-based encryption with outsourced revocation in cloud computing. IEEE Transactions on Computers, 64(2), 425–437. https://doi.org/10.1109/TC.2013.155
3. Zhou, L., & Chao, H. C. (2011). Multimedia traffic security architecture for the IP multimedia subsystem. IEEE Network, 25(5), 35–41. https://doi.org/10.1109/MNET.2011.6033038
4. Zhang, R., & Liu, L. (2010). Security models and requirements for healthcare application clouds. Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing, 268–275. https://doi.org/10.1109/CLOUD.2010.45
5. Sun, D., Chang, G., Sun, L., & Wang, X. (2011). Surveying and analyzing security, privacy and trust issues in cloud computing environments. Procedia Engineering, 15, 2852–2856. https://doi.org/10.1016/j.proeng.2011.08.537
6. Chen, D., & Zhao, H. (2012). Data security and privacy protection issues in cloud computing. 2012 International Conference on Computer Science and Electronics Engineering, 647–651. https://doi.org/10.1109/ICCSEE.2012.193
7. Patel, P., Ranabahu, A., & Sheth, A. (2009). Service level agreement in cloud computing. Proceedings of the 2009 Conference on Object-Oriented Programming Systems, Languages, and Applications, 1–10. https://doi.org/10.1145/1639950.1640071
8. Wang, C., Wang, Q., Ren, K., Cao, N., & Lou, W. (2012). Toward secure and dependable storage services in cloud computing. IEEE Transactions on Services Computing, 5(2), 220–232. https://doi.org/10.1109/TSC.2011.24
9. Kumar, P., & Lee, H. J. (2012). Security issues in healthcare applications using wireless medical sensor networks: A survey. Sensors, 12(1), 55–91. https://doi.org/10.3390/s120100055
10. Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. Future Generation Computer Systems, 28(3), 583–592.