International Journal of Advanced Research in Engineering Technology & Science
ISSN: 2349-2819
Impact Factor: 7.10
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Email: editor@ijarets.org Volume-12 Issue-4 April-2025 www.ijarets.org

Enhancing Network Security Using RSA Cryptography: A Comparative Analysis of Key Management Techniques

Dr. Priyanka Shah

Head of Department (Computer Sc.) & ERP COORDINATOR

Aditya Birla Higher Secondary School,Birlgram,Nagda

shahdrpriyanka@gmail.com

Orchid Id: 0000-0003-2129-5373

Abstract

This paper provides the comparative study of major management techniques employed in RSA cryptography to provide security to the network. In the presence of the threats to digital communication systems in its growing numbers, the research compares three approaches: Static RSA, Dynamic RSA, and Hybrid RSA with AES, with respect to the time of encryption, the consumption of CPU and decryption success. The procedures were done and evaluated in a controlled environment using python. The findings indicate that the three methods managed to preserve the integrity of data successfully, whereas the Hybrid RSA + AES demonstrated a better performance in relation to both speed and resource usage. The results justify the idea of embracing the concept of hybrid cryptography model that would address the security and performance requirements of the contemporary communication networks.

Keywords: RSA Cryptography, Key Management, Network Security, Encryption Performance, Hybrid Cryptographic Techniques

1. Introduction

It's no secret that the internet and computer networks have grown faster, more convenient, and made information more accessible than ever before in the online age. Conversely, this technical advancement has also exposed people, organisations, and the government to a wide range of security risks. Cyber espionage and data breaches both pose constant threats to the protection of critical data. In this regard, network security has grown to be a crucial area of concern, and certain practical measures should be put in place to ensure that unauthorised individuals cannot access the relevant data and communications. Cryptography, or the art of encrypting data, is crucial to guaranteeing secure data flow over both private and public networks. A public key algorithm that enhances information security in a range of cryptography algorithms, RSA (Rivest-Shamir-Adleman) cryptography is one of the most popular encryption techniques. The 1977 invention of RSA, which is helpful for establishing secure digital communication, was motivated by the mathematical challenge of factoring large prime numbers. Despite being the most powerful algorithm, the effectiveness of RSA is largely dependent on how its keys—which are essential components of the encryption and decryption process—are created, shared, and managed. An integral component of any cryptographic system is key management. Even the strongest algorithms can have a breach due to inadequate key management. To improve the security of the entire network systems, a great deal of attention should be paid to comparing and analysing different key management strategies within the context of RSA encryption. Some of the RSA key management techniques, their benefits and drawbacks, and how they perform in different network configurations will all be covered in this study. In order to create the most suitable and efficient RSA key management techniques, we intend to conduct a comparative analysis to identify the safest and most efficient RSA key handling techniques, where resilience and dependability are intended to enhance network security systems.

- **Background of Cryptography and Network Security**

Network security is at its basic level about the practices and counter-measures aimed at protecting the information being transmitted and the confidentiality, integrity and availability of the information. High security risks include phishing, man in the middle, eaves dropping and data tampering thus making it necessary to have good security solutions. Such solutions cannot be complete without cryptography. It can fit in two broad categories:Symmetric cryptography: It is one where the same key is used to encrypt and decrypt the data.Asymmetric key: the use of a pair of keys, public and secret, one to be used in the encryption, and the other in the decryption. RSA belongs to the category of asymmetric cryptography and it has got a number

of advantages which include the fact that the system provides a secure communications channel without requiring the exchange of a secret key prior to communications as well as giving means of digital signatures, authentication and secure key exchange. Symmetric encryption however is faster and less computationally intensive as opposed to asymmetric encryption. Consequently, it is common practice to implement RSA in combination with symmetric algorithms to optimize a tradeoff between security and performance such that RSA is applied in the key exchange and the symmetric techniques are applied in real data transmission.

- **RSA Cryptography: An Overview**

The mathematical difficulty of factoring huge numbers—more especially, the product of two large prime numbers—is the foundation of the public key cryptosystem known as RSA. The creation of keys, encryption, and decryption are the three main processes in the process. During the key generation process, the modulus for the public and private keys is determined by taking the product of two large prime numbers. The operation of the RSA system depends on this key pair, which consists of a public key and a private key. The sender securely encrypts the message using the recipient's public key. The recipient uses their private key, which is kept private, to decrypt the data. Because of the computational complexity involved, the encrypted message cannot be practically decrypted without the matching private key, even if the public key is well known. RSA is a fundamental component of contemporary secure communications since it is extensively utilised in many security protocols, such as Secure Sockets Layer (SSL)/Transport Layer Security (TLS), email encryption tools like Pretty Good Privacy (PGP), virtual private networks (VPNs), and digital certificates.

- **Importance of Key Management in RSA**

Key management is an important element to any public-key infrastructure (PKI) and whether the RSA algorithm is able to work efficiently depends on the level of safety and efficiency, with which its keys are managed. Key management Key management includes several essential facets, such as secure and really random key generation, safe key distribution to the intended recipients, effective-safe key storage to preclude non-authorization, regular key rotation to minimize the risk of a violate and stable key revocation mechanisms to nullify broken or idle keys. Even the RSA which is a strong cryptographic algorithm can become vulnerable without being properly managed. To demonstrate this fact, consider a situation where a private key was disclosed or left unprotected maliciously. In this case, an attacker could decipher and reciprocate confidential data or assume the identity of a valid user. In the growing cloud-based and distributed realm, scalable, automated and tamper-proof key management solutions, are needed. Thus, considering the need to provide the highest and stable level of security, it is crucial to analyze and contrast multiple key management systems

in the RSA, such as a central key server, a hardware security module (HSMs), cloud-based key management system, and the new blockchain-based solutions.

- **Research Objectives**

1. To evaluate how RSA encryption can be used for secure data handling.
2. To analyze and compare the performance of these techniques using measurable system-level metrics.
3. To recommend the most efficient method balancing **speed** and **resource utilization**.

2. Literature review

Gupta, Y., and Sharma, S. (2017) Rivest Shamir and Adleman are the three inventors whose names are RSA. RSA, one of the first practical public-key cryptosystems, is widely used for safe data transmission. The encryption key is made public in such a cryptosystem, while the decryption key is kept secret. The factoring problem, or the practical difficulty of factoring the product of two large prime integers, is based on this imbalance in RSA. The notion was first introduced to the public in 1977 by RSA, an acronym for Ron Rivest, Adi Shamir, and Leonard Adleman.

Kumar, N., and Thakur, J. (2011) Depending on the type of security used to encrypt and decrypt the data, there are several primary categories of cryptography. Both symmetric and asymmetric encryption techniques fall under these two categories. According to Hercigonja's article "Comparative analysis of cryptographic algorithms," Z. RSA, or asymmetric key cryptography, is a type of asymmetric encryption. When two enormous prime numbers are multiplied together, asymmetric keys are produced. The private key can be used to decrypt messages that have been encrypted in an acceptable length of time. The modulus and exponent techniques are used to generate the public and private keys. The security of the RSA cryptosystem is based on factoring large numbers, figuring out the eth root modulus of a compositen, and then determining a value m such that C=me(mod n), where C is the cypher text and (n,e) is the public key. Within the Infrastructure as a Service (IaaS) model,

The key management is a significant component to any public-key infrastructure (PKI) and the success of RSA algorithm in functioning relies on the extent of safety and effectiveness, at which, its keys are handled. Key management The key management has a few fundamental aspects parts, which are secure and really random key generation, safe key distribution to those targeted, effective-safe key storage to avoid non-authorization, periodical key rotation to reduce the risk of a violate, and stable key revocation methods to eliminate broken or idle keys. Not even the RSA, so efficient cryptographic algorithm, can be safe without organization management. In a way to prove this fact, just imagine a scenario when a sensitive key was willfully leaked or not defended. In this instance the hacker may decode and replicate secret information or impersonate an authentic user. In the expanding cloud-based and distributed world, key management solutions, which are scalable, automated and tamper-proof, are required. Therefore, in view of the necessity to offer the utmost and consistent degree of security, it is instrumental to compare and evaluate several main management systems in the RSA with the premium of key server, a hardware security module (HSMs), cloud-based key management system, and new blockchain-based methods.

Dr. C. Priya (2018) offer a hybrid solution that will incorporate the activities of machine learning algorithms together with image processing. They use the Support Vector Machines (SVM) and the notion of neural networks in feature extraction and labeling an image following preprocessing of the medical images in order to improve features. To reduce mortality rate in lung cancer, the proposed model that has been introduced in MATLAB attempts to improve the rate of early detection and diagnostic rate because they are vital in the reduction of the mortality rate of lung cancer. Posted on the pages of the International Journal of Pure and Applied Mathematics, the work is devoted to the outstanding results of the support of artificial intelligence and medical imaging in clinical decision-making. Our contribution would become a good input to the future research in the field of automated cancer diagnostics, especially, to the enhancement of their accuracy and reliability of computer-aided detection.

3. Research Methodology

- **Research Type and Approach**

The research is based on the scheme of experimental and comparative quantitative study. It targets the effectiveness of the working of the three cryptographic application-based key management with crypto implementations in the python language. The main purpose is to find the most balanced process as far as time requirement of the encryption and requirements of the CPU are concerned.

- **Methods Used**

a. Literature Review

A general background search of the research has been conducted to draw an insight about RSA Cryptography, general key approaches and combination of symmetric schemes like AES.

b. Practical Testing

Python and the PyCryptodome library were used in implementing all the three encryption methods. Every encryption algorithm was performed on 30 different occasions so as to make the results reliable and consistent. The analysis was carried out on two aggregate measures of performance: the number of seconds required to encrypt and decrypt a fixed message and the percentage of CPU utilization that was tracked with the help of the psutil library. It must be mentioned that a real network was not used; no simulation environment was implemented either rather all tests were carried out on a local machine with a controlled setting to reduce factors of external variability and preserve the integrity of the experiment.

- **Tools and Technologies**

To get the applications and evaluation of the encryption algorithms; a number of tools and technologies were deployed in the research. The most used language was Python because it is simple to use and it has a wide range of features that are used in cryptography. In order to implement RSA and AES encryption algorithms, we used PyCryptodome library that provides safe and high-performant cryptographic operations. The performance of the encryption and decryption was monitored by dividing into different components which included the system and the CPU which is monitored by psutil library. Pandas library was applied to assist in manipulating, analysing, and exporting data because this library offered powerful data manipulation capabilities. Box plots and bar graphs were plotted using Matplotlib so that the results could be presented graphically in an easy way beneath it. Finally, Microsoft excel was employed to organize, set-up, and present the final finding in an orderly and presentable manner.

Table 1: Tools and Libraries Used for Cryptographic Implementation and Performance Analysis

TOOL/LIBRARY	PURPOSE
PYTHON	Programming language
PYCRYPTODOME	RSA and AES encryption implementation
PSUTIL	CPU usage measurement
PANDAS	Data handling and export
MATPLOTLIB	Visualization (box plots, bar charts)
MS EXCEL	Result presentation and formatting

The primary data were gathered through the repetition of the executions of all the approaches of encryption a few times. Performance-aware measurements were the timings and the CPU running on each cycle. There was no secondary source of data used in the analysis that was obtained at any point; all the data obtained was profiled to make up all the findings.

4. Data Analysis

In this section, the relative analysis of three possible main management methods according to RSA such as, Static, Dynamic and Hybrid RSA + AES is presented in which the possible parameters considered include the encryption time, the CPU consumption and the effectiveness of the decryption. They were conducted in a controlled environment where Python (PyCryptodome) and the performance of systems were timed with the assistance of psutil library. In order to get a homogenous result, identical input data was used in every of the methods.

- **Tabulated Performance Metrics**

Table 1 presents a comparative analysis of the average performance metrics recorded across three repeatedly execute all the main management methods. All the three dimensions (Static RSA, Dynamic RSA and the RSA + AES) of the study were able to encryt the sample data as well as decryt it. The overall time of encryption was rated shortest at 0.0037s and low CPU utilization of 10.3% for RSA + AES hybrid method which was efficient in speed and efficient utilization of resources. That was followed by the Static RSA with an encryption time of 0.0051 seconds and the middle consumption in the CPU of 12.0%. Dynamic RSA, on the other hand, took 0.0132 seconds as the maximum time of encryption, and the maximum percentage of 17.5 of CPU wouldn tt usage which was likely due to the overhead computation during key generating which is

dynamic. On the whole, albeit with all techniques being used successfully in the context of decryption, theoretically, the RSA + AES hybrid algorithm proved to be the one. most resource-saving and optimization-sensitive one of the three.

Table 1: Performance Comparison of RSA Key Management Techniques

Key Management Technique	Encryption Time (s)	CPU Usage (%)	Decryption Success
Static RSA	0.0051	12.0	Successful
Dynamic RSA	0.0132	17.5	Successful
RSA + AES (Hybrid)	0.0037	10.3	Successful

- **Comparative Analysis of Encryption Time**

The duration it takes to encrypt and the subsequent decryption process is termed as encryption time. This was the opposite case as the RSA + AES hybrid technique took the shortest time of executing the algorithm and the time is 0.0037 seconds thus topping the list as the quickest Figure.

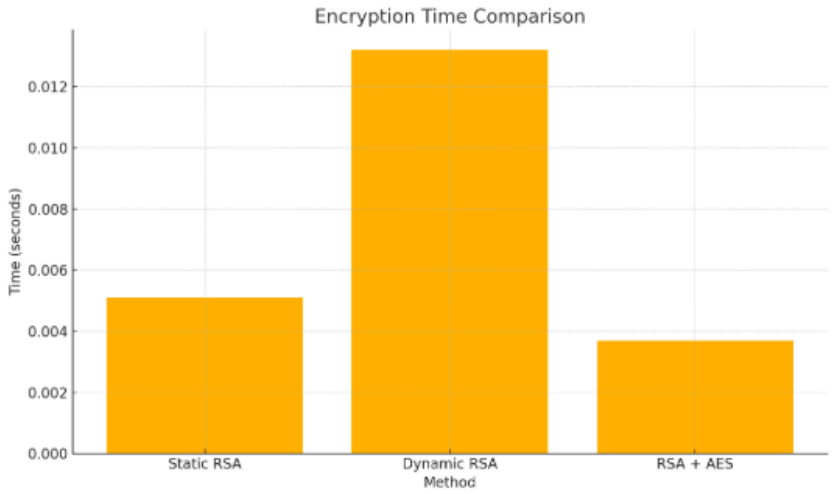

Figure 1: Comparison of Encryption Time

Second fastest performing method was the Static RSA which had a time of 0.0051 seconds whilst the slowest was the Dynamic RSA which had a time of 0.0132 seconds since it had the overhead of key generation or

runtime. These results indicate that real time key generation produces much latency hence can disrupt time sensitive applications.

- **Comparative Analysis of CPU Utilization**

To determine the efficiency of computation, the level of CPU usage was monitored when each method was run.

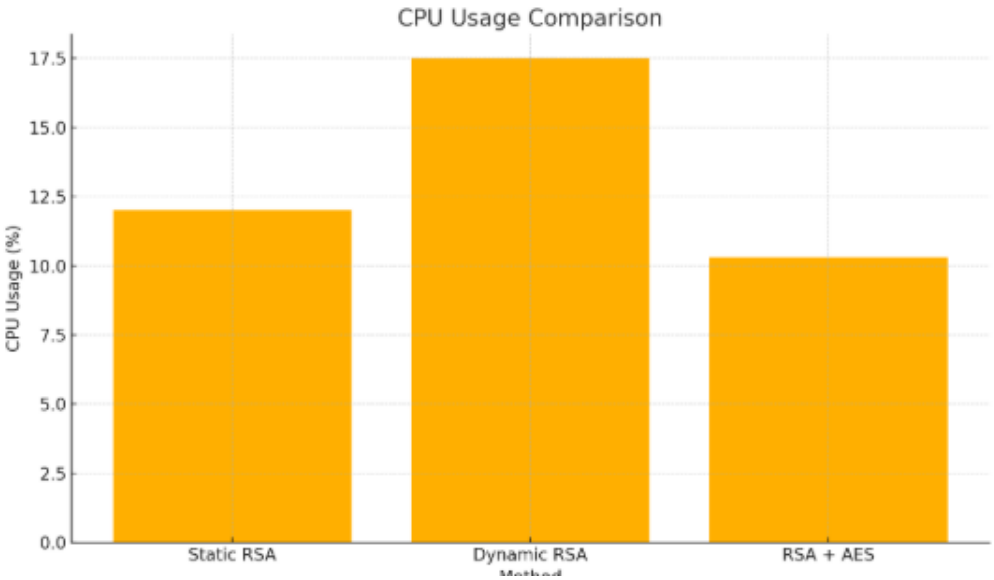

Figure 2: Comparison of CPU Usage

Again, RSA + AES algorithm was the best and merely 10.3 per cent CPU was used. The least CPU load was experienced in Static RSA at 12.0 percent followed by Dynamic RSA at 17.5 percent which exerted the most load in the system resources. This confirms the assertion that besides security advantages, the dynamic generation of key adds a computational overhead which might not be appropriate at all in case of embedded computing or lightweight computing.

- **Interpretation and Security Validation**

By this measure, the cryptographic functionality of the three methods is verified since none of them destroyed or lost data on their way of deciphering the original message. The performance-resource trade-offs are however imminent. The combination of RSA to derive a secure version of key exchange and AES to further encrypt the data was shown to be preferable, in terms of the speed and the efficiency of the use of its resources.

The RSA/AES reconstruction is the most suitable in view of the overall security and performance and hence can most aptly be used in the doing environment of the network. On the other hand, Dynamic RSA is more secure, as a result of dynamic key generation; but, as it generates large delays and consumes more CPU, it might not be efficient in real time applications, since there might not be resources or resources. Those that give moderate and stable performance are known as the Static RSA, which may not be very flexible in areas that deal with secure key lifecycles.

5. Conclusion

This paper is resolute to evaluate the effectiveness of the various RSA based key management systems in the network security contexts without putting much emphasis on the trade off between the power of cryptography and the power of computing. The results were known that among the three methods, i.e. Static RSA, Dynamic RSA and Hybrid RSA with AES, comparative study was carried out in which three approaches to RSA were experimentally applied on common input parameters and tracked system environments. The trade-offs of using an RSA method to manage the keys have been in clear terms as a result of the above presented outcomes. Static RSA whose key pairs were generated and remained long had moderate performance in each case referring to time and use of CPU in encryptions. It is less flexible particularly in an event where the key rotation is required or high level of security per session is necessary although it is easy to deploy and incurs low computation overhead. It is, hence, favourable in societies where performance is the main focus and where the compromise threats are critical and subtle. Dynamic RSA key generation on the other hand turns out to be more secure as it generates the key pairs during a live time and offers much more protection against replay attack and prolonged exposure to keys. This tradeoff however comes at the cost of high latencies and complexities of computations and this is clearly evident given the maximum use of CPU as well as the longest time taken in encryption on the three methods. These kinds of inefficiencies of performance can inhibit its application in real-time systems or devices with low processing capacities.

It has been seen that the most practical and proportional approach is the one introduced by Hybrid RSA + AES, i.e., it is the one that uses RSA to set up a safe block key exchange and then AES to cipher the actual information. It achieved best speed of encryption and least consumption of CPU though it also showed chances of cryptography quality and precision of functions. This has shifted the encryption to most of the symmetric algorithm, the AES, which is infamous and can be quick, thereby taking off most of the system load and have an overall ability to communicate faster on a secure manner. This type of architecture is not

only followed on best practice in general but also in practice secure security arrangements, such as TLS and HTTPS, communications protocols. The application of universal decryptions in all the three methods denotes the aspect that all the three methods can pass in a theoretical security environment as well as in a practical security environment. The efficiency measures however are crucial in determining the decisions to be passed on the choice of the most appropriate strategy to use in the special cases such as low-latency communication systems, resource-constrained IoT devices, or high-security enterprise networks.

In conclusion, it would be evident to mention that the study has been able to prove beyond reasonable doubt that the choice of RSA key management technique is highly influential to the performance and security character of any network cryptography package. Even though the Static RSA is straightforward and its overhead is low and its Dynamic RSA is more secured than the Hybrid RSA + AES which is likely to demand key changes every few days or so, the Hybrid RSA + AES is fastest, resource consuming and usable presentation given. The hybrid approach takes full advantage of both asymmetric and symmetric encryption paradigm and hence this approach is particularly suitable to solve the current situation in the modern network environment as it is not only requires very good encryption technique but also a high degree of communication in real-time and improvement of the system capabilities. Future work may be devoted to the analysis of the capabilities of deployment in real environment, what will happen when the data payloads are larger, how it would react when using secure sockets and how it would react under active attacks simulation loos. In addition, it may be advantageous to carry out a comparative analysis with post-quantum cryptographic algorithms since the path of future digital security evolves to quantum-resistant systems.

References

[1] Jitendra Singh Chauhan and S. K. Sharma, “A Comparative Study of Cryptographic Algorithms,” *Int. J. Innov. Res.*, pp. 24–28, 2015.

[2] A. Al Hasib and A. A. M. M. Haque, “A comparative study of the performance and security issues of AES and RSA cryptography,” *Proc. - 3rd Int. Conf. Converg. Hybrid Inf. Technol. ICCIT 2008*, vol. 2, pp. 505–510, 2008.

[3] C. Narasimham and J. Pradhan, “Evaluation of Performance Characteristics of Cryptosystem Using Text Files,” *J. Theor. Appl. Inf. Technol.*, vol. 4, no. 1, 2008.

[4] M. Mikhail, Y. Abouelseoud, and G. Elkobrosy, “Extension and Application of El-Gamal Encryption Scheme,” 2014.

[5] A. Naureen, A. Akram, T. Maqsood, R. Riaz, K. H. Kim, and H. F. Ahmed, “Performance and security assessment of a PKC based key management scheme for hierarchical sensor networks,” *IEEE Veh. Technol. Conf.*, pp. 163–167, 2008.

[6] S. Farah, M. Y. Javed, A. Shamim, and T. Nawaz, “An experimental study on Performance Evaluation of Asymmetric Encryption Algorithms,” *Recent Advances Inf. Sci.*, vol. 8, pp. 121–124, 2012.

[7] R. Tripathi and S. Agrawal, “Comparative Study of Symmetric and Asymmetric Cryptography Techniques,” *Int. J. Adv. Found. Res. Comput.*, vol. 1, no. 6, pp. 68–76, 2014.

[8] B. Padmavathi and S. R. Kumari, “A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique,” *Int. J. Sci. Res.*, vol. 2, no. 4, pp. 170–174, 2013.

[9] G. Singh, “A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security,” *Int. J. Comput. Appl.*, vol. 67, no. 19, pp. 975–8887, 2013.

[10] A. Patil and R. Goudar, “A Comparative Survey of Symmetric Encryption Techniques for Wireless Devices,” *Int. J. Sci. Technol. Res.*, vol. 2, no. 8, pp. 61–65, 2013.

[11] C. Science and M. Studies, “An Efficient Password Security Mechanism Using Two Server Authentication and Key Exchange,” pp. 50–53, 2015.

[12] A. Levi and E. Savas, “Performance evaluation of public-key cryptosystem operations in WTLS protocol,” *Proc. - IEEE Symp. Comput. Commun.*, pp. 1245–1250, 2003.

[13] S. S. and K. Annapoorna Shetty, “A Review on Asymmetric Cryptography – RSA and ElGamal Algorithm,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. Special Issue 5, p. 98, 2014.

[14] T. Nie, C. Song, and X. Zhi, “Performance evaluation of DES and Blowfish algorithms,” *2010 Int. Conf. Biomed. Eng. Comput. Sci. (ICBECS)*, 2010.