The North African Journal of Scientific Publishing (NAJSP)
مجلة شمال إفريقيا للنشر العلمي (NAJSP)
E-ISSN: 2959-4820
Volume 3, Issue 3, 2025
Page No: 101-109
Website: https://najsp.com/index.php/home/index
SJIFactor 2024: 5.49 **معامل التأثير العربي (AIF) 2024: 0.71** **ISI 2024: 0.696**

Empirical Comparison Study of RC4 and RSA Algorithms

Alhadi A. Klaib[1*], Noura J. Jwan[2]
[1]Libyan Authority for Scientific Research, Tripoli, Libya
[2]Libyan Academy for Postgraduate Studies, Tripoli, Libya

دراسة مقارنة تجريبية لخوارزميات RC4 وRSA

الهادي علي كليب [1*]، نورا جمال الدين جوان [2]
[1] الهيئة الليبية للبحث العلمي ، طرابلس، ليبيا
[2] الأكاديمية الليبية للدراسات العليا ، طرابلس، ليبيا

*Corresponding author: alhadi.klaib@aonsrt.ly

Received: June 06, 2025	Accepted: July 15, 2025	Published: July 29, 2025

Abstract
This study presents an empirical comparison between the RC4 and RSA encryption algorithms, focusing on their performance, security characteristics, and practical applicability in file encryption scenarios. RC4, a stream cipher known for its simplicity and speed, was evaluated alongside RSA, an asymmetric encryption algorithm widely used for secure communications. The experiments involved encrypting and decrypting various file types, including text and image files ranging from 10 KB to 200 KB, using both algorithms. Results showed that RC4 significantly outperforms RSA in terms of encryption and decryption speed, making it suitable for performance-sensitive applications. However, RC4's well-documented cryptographic weaknesses, such as biased key scheduling and vulnerability to attacks, limit its use in secure environments. RSA, although computationally slower, offers robust security for tasks such as digital signatures and key exchange when implemented with appropriate key lengths. The study concludes that while RC4 may be acceptable for non-sensitive or internal data, RSA remains the preferred choice for security-critical applications. These findings reinforce the need for hybrid cryptographic systems that balance speed and security, especially in modern data protection architectures.

Keywords: RC4, RSA, Empirical Study, Encryption and Decryption, Cryptographic Performance.

الملخص:
تقدم هذه الدراسة مقارنة تجريبية بين خوارزميتي التشفير RC4 وRSA، مع التركيز على أدائهما وخصائصهما الأمنية وإمكانية تطبيقهما عمليًا في سيناريوهات تشفير الملفات. تم تقييم RC4، وهو نظام تشفير تدفقي معروف ببساطته وسرعته، إلى جانب RSA، وهي خوارزمية تشفير غير متماثلة تُستخدم على نطاق واسع في الاتصالات الآمنة. تضمنت التجارب تشفير وفك تشفير أنواع مختلفة من الملفات، بما في ذلك ملفات النصوص والصور التي تتراوح أحجامها بين 10 و200 كيلوبايت - باستخدام كلتا الخوارزميتين. أظهرت النتائج أن RC4 يتفوق بشكل ملحوظ على RSA من حيث سرعة التشفير وفك التشفير، مما يجعله مناسبًا للتطبيقات الحساسة للأداء. ومع ذلك، فإن نقاط ضعف RC4 التشفيرية الموثقة جيدًا، مثل جدولة المفاتيح المتحيزة والتعرض للهجمات، تحد من استخدامه في البيئات الآمنة. أما RSA، فعلى الرغم من بطء أدائها الحسابي، إلا أنها توفر أمانًا قويًا لمهام مثل التوقيعات الرقمية وتبادل المفاتيح عند تنفيذها بأطوال مفاتيح مناسبة. خلصت الدراسة إلى أنه على الرغم من أن RC4 قد يكون مقبولاً للبيانات غير الحساسة أو الداخلية، إلا أن RSA يبقى الخيار الأمثل للتطبيقات الحساسة أمنياً. تُعزز هذه النتائج الحاجة إلى أنظمة تشفير هجينة تُوازن بين السرعة والأمان، وخاصةً في هياكل حماية البيانات الحديثة.

الكلمات المفتاحية: RC4، RSA، دراسة تجريبية، التشفير وفك التشفير، الأداء التشفيري.

Introduction

Encryption is the process of converting information from a readable form to an unreadable or incomprehensible form, with the aim of protecting data and information from unauthorized access. Encryption is one of the basic tools in the field of information and communications security. Encryption is done by applying complex mathematical algorithms that convert the original data into an encrypted form known as "ciphertext". A key or a series of numbers or symbols is used to convert the data, and this key is used to decrypt and recover the original data.

Encryption is used in many areas, including; communications security. Encryption is used to protect data during transmission over networks, such as the Internet. Information is encrypted before it is sent and decrypted upon receipt to ensure the confidentiality of the content and prevent unauthorized access. Secondly; encryption is used to protect sensitive or confidential data in databases and storage systems. Encryption ensures that access to this data is limited to authorized persons only. Thirdly; encryption is used to protect wireless communications, such as mobile phones and Wi-Fi networks. The transmitted data is encrypted so that it is difficult for attackers to hack the connection and steal information. Fourthly; encryption is used in software applications to protect sensitive information such as passwords and personal files. Encryption helps prevent hacking and unauthorized access to this information.

It is worth noting that there are two main types of encryptions: report encryption and symbolic encryption. Report encryption is used to convert an entire data file into an encrypted form, while symbolic encryption is used to convert individual data, such as emails and files. Although encryption is a powerful tool for protecting data, it is not perfect and can sometimes be hacked. Therefore, strong encryption algorithms must be used and updated regularly to address new threats. RSA and RC4 are two of the most popular and widely used encryption algorithms in the field of information security. I will give you a brief introduction about each of them:

RSA:

RSA stands for "Rivest, Shamir, Adleman", and is a basic encryption algorithm based on the concept of transforming large numbers. The algorithm is distinguished by its ability to achieve transparency, digital signature, and secure key exchange. RSA is based on selecting two very large prime numbers and calculating their product. This product is used as part of the encryption and decryption key. One of the prime numbers is known as the public key and is shared with the others, while the other prime numbers (private key) are kept secret. The prime numbers and the generated product are used in encryption and decryption operations. RSA is used in many applications such as securing online communications, signing digital messages, and electronic signatures.

RC4:

RC4 is an integrated stream cipher algorithm, which is short for "Rivest Cipher 4". RC4 is characterized by its simplicity and high speed, and therefore it was widely used in many applications in the past. The RC4 algorithm is based on the principle of generating a random encryption string called a "keychain". The keychain is used to perform the encryption and decryption process in harmony with the data to be encrypted. It is worth noting that encrypting data using RC4 requires the use of a secret key shared by the sender and the recipient. However, it should be noted that the RC4 algorithm suffers from some security weaknesses, and improved configurations have been developed to enhance its security. Therefore, RC4 has been replaced by stronger encryption algorithms in many applications and protocols.

Research Question

"How do the RC4 and RSA encryption algorithms compare in terms of performance (encryption/decryption speed) and security in practical file encryption scenarios, and what are their optimal use cases in modern cryptographic applications?"

This paper was structured as follows; it begins with an Introduction outlining encryption's role in security and the study's objectives, followed by a Literature Review summarizing prior comparisons and cryptographic trends. The Methodology details the experimental setup (C# implementation, file types, and performance metrics), while results present quantitative data on encryption/decryption speeds and file size impacts. The Discussion analyzes trade-offs between RC4's speed and RSA's security, contextualizing findings with existing research. Finally, the Conclusion reaffirms RC4's performance advantages and RSA's robustness, recommending hybrid systems for future work, alongside references to 22 supporting studies.

Literature Review

The current paper [1] conducts a performance comparison between the RSA and RC4 algorithms. It likely evaluates metrics such as encryption/decryption time and throughput for various data sizes, providing empirical data on their efficiency in different scenarios. Given its publication year, it offers a relatively recent perspective on the operational differences between these two distinct cryptographic

types. This manuscript [2] provides a broader performance analysis across various cryptographic algorithms. While the specific algorithms compared are not detailed in the title, it likely includes common symmetric and asymmetric ciphers, offering a general overview of their speed and efficiency characteristics. Its 2016 publication date makes it a foundational, though not the most recent, reference for performance benchmarks.

This paper [3] focuses on enhancing the security of the RC4 algorithm, acknowledging its inherent vulnerabilities. It likely proposes modifications to RC4's key scheduling or pseudo-random generation algorithm to mitigate known weaknesses, aiming to improve its cryptographic strength without necessarily sacrificing its characteristic speed. This highlights ongoing efforts to "fix" RC4, despite its general deprecation. The present research [4] conducts a comparative analysis of encryption algorithms specifically within the context of the Internet of Things (IoT). It likely evaluates the performance and resource consumption of various ciphers, including AES, RC4, or others, to determine their suitability for resource-constrained IoT devices, considering factors like power efficiency and computational overhead.

This investigation [5] provides a comparative study exclusively on symmetric key algorithms. It likely evaluates the performance (e.g., encryption/decryption speed, throughput) and potentially security aspects of various symmetric ciphers such as AES, DES, 3DES, Blowfish, and possibly RC4, offering insights into their relative efficiencies for bulk data encryption. This article [6] offers a comprehensive analysis of cryptographic algorithms, focusing on their security, efficiency, and future challenges. It likely covers both symmetric and asymmetric techniques, discussing their strengths, weaknesses, and the evolving threat landscape, including potential impacts from quantum computing. The present work [7] address a comparative analysis of symmetric and asymmetric cryptographic techniques. It likely delves into the fundamental differences between these two categories, evaluating their performance characteristics, security strengths, and typical applications, providing a broad overview of modern cryptographic practices. The paper [8] conducts a general comparative analysis of various encryption algorithms. It likely assesses their performance metrics, such as encryption/decryption time, and potentially discusses their security properties, contributing to the understanding of algorithm suitability for different applications.

This study [9] specifically focuses on the efficiency analysis of cryptographic algorithms for securing image data within a cloud environment. It likely evaluates how different encryption methods perform when applied to large image files, considering factors relevant to cloud deployment like scalability and processing time. The upcoming publication [10] promises a comparative analysis of cryptography algorithms within the broader context of information security. It is expected to cover various algorithms and their roles in ensuring data confidentiality, integrity, and availability, offering a very recent perspective on the field.

This report [11] provides a comparative and exploratory analysis of cryptographic algorithms specifically for securing digital devices. It likely investigates algorithms suitable for hardware or embedded systems, considering resource constraints and performance requirements in such environments. In addition, this paper [12] focuses on providing an understanding of the RSA algorithm. While not a direct comparison, it likely delves into RSA's mathematical foundations, key generation, encryption/decryption processes, and its role in public-key cryptography, serving as a fundamental reference for its characteristics. This forthcoming [13] comprehensive survey will cover symmetric and public-key cryptographic algorithms, including their foundational principles, known attacks, and various applications. It is expected to offer a very up-to-date and broad overview of the cryptographic landscape. This article [14] presents an experimental comparison of encryption algorithms specifically on smart devices. It likely evaluates their performance in terms of speed, power consumption, and memory usage, providing insights into the practical applicability of various ciphers in resource-constrained mobile or IoT environments.

This study [15] provides an empirical examination of security issues related to encryption techniques. While older, it likely investigates vulnerabilities and attack vectors associated with various encryption methods, offering a foundational understanding of cryptographic security challenges. This article [16] focuses on the performance evaluation of cryptographic security algorithms within a cloud computing context. It likely assesses the efficiency of different encryption methods when deployed in cloud environments, considering factors like scalability, throughput, and resource utilization for secure cloud data operations. This upcoming paper [17] focuses on efficient lightweight cryptographic solutions tailored for healthcare systems utilizing IoT. It is expected to explore ciphers optimized for resource-constrained medical devices, ensuring data security without compromising performance in sensitive healthcare applications.

This work [18] introduces a novel "secret key 4 optimization algorithms" (SK4OA) for securing cloud data, focusing on its non-linearity run time trend. It suggests an innovative approach to symmetric key

encryption tailored for cloud environments, potentially offering unique performance or security characteristics. This is a direct empirical comparison of RSA and RC4 performance for image and text files [19]. Although from 2015, it provides specific data on their encryption/decryption speeds for different file types and sizes, offering a historical benchmark for their operational efficiency. Recent study [20] focuses on enhancing the RC4 algorithm by addressing its Initialization Vector (IV) transmission. It suggests a modification aimed at improving RC4's security, indicating continued, albeit niche, research into mitigating RC4's known vulnerabilities. This work [21] focuses on systematically identifying cryptographic functions in binaries. While not a direct comparison of algorithms, it is relevant for understanding how cryptographic implementations are analyzed in practice, which can indirectly inform performance and security studies. This paper [22] describes a "hybrid cryptosystem enhanced RSA and RC4 chaotic map." This is highly relevant as it proposes a new hybrid approach combining RSA and a modified RC4, potentially leveraging chaotic maps for enhanced security or performance. Table 1 shows a comparative study for previous work.

Table 1: Comparative study for the previous work.

no	Authors	Study Type	Key Findings	Limitations
1	Yüksel & Özgün	Empirical performance	RC4 ≫ RSA in throughput	Single platform, no security tests
2	Hossain et al.	Benchmark survey	RC4 fastest; RSA most secure	Old hardware, limited datasets
3	Al-badrei & Alshawi	Security improvement	'Improved' RC4 ↑ entropy	No formal cryptanalysis
4	Ghaz et al.	IoT empirical	RC4 best latency; RSA heavy	Small packet sizes only
5	George & Thomas	Survey	Stream > block speed	No experiments
6	Ramakrishna & Shaik	Comprehensive survey	Recommends hybrid RSA + sym.	High-level, no new data
7	Sharma et al.	Conference survey	RC4 speed, RSA security	Minimal empirical proof
8	Olutola & Olumuyiwa	Empirical benchmark	Context-specific selection	Lacks side-channel view
9	Rahul & Kuppusamy	Cloud imaging	AES fastest; RSA bottleneck	Only image data tested
10	Meftah et al.	Conf. benchmark	RC4 faster; RSA secure	Small sample sizes
11	Syamala et al.	Digital device study	RC4 low CPU	Acknowledged insecurity
12	Singh et al.	RSA tutorial	Deep dive RSA	No RC4 comparison
13	Shah & Gor	Comprehensive survey	RC4 deprecated; RSA key trends	No benchmarks
14	Frugh et al.	IoT empirical	RC4 top speed	Energy only, security ignored
15	Gahan & Devanagavi	Security issues survey	Highlights RC4 flaws	No data
16	Karanam et al.	Cloud perf.	RC4 speed; AES recommended	Only throughput metric
17	Rasheed & Kumar	Healthcare IoT study	Call for lightweight ciphers	No RC4 experiments
18	Frimpong et al.	New stream cipher	SK4OA > RC4 energy	RC4 baseline dated
19	Okedola & Asafe	File-level test	RC4 ≫ RSA on images/text	Legacy dataset
20	Mohammed et al.	RC4 variant	Remove IV → security ↑	Entropy only
21	Fan et al.	Binary analysis	Detect crypto funcs	Not perf-oriented
22	Hakim et al.	Hybrid tutorial	RSA+RC4 chaotic hybrid	Benchmark absent

General Critique

While many of the papers support the known performance–security trade-off between RC4 and RSA, several suffer from methodological limitations. In particular, many do not standardize test conditions (e.g., key size, dataset size, hardware), making cross-study comparisons difficult. Few studies implement modern side-channel protections or simulate realistic attack scenarios. Several "improved RC4" proposals rely only on entropy metrics or basic randomness tests, which are insufficient to establish cryptographic security. Furthermore, while hybrid encryption is often recommended, very few papers rigorously benchmark end-to-end performance of such systems.

Empirical Performance and Security Analysis

a) Speed and Efficiency: RC4 consistently outperforms RSA in terms of encryption and decryption speed. Empirical studies measuring the encryption time for files of varying sizes (text and image files, 10–200KB) demonstrate that RC4 is significantly faster than RSA across all tested scenarios.This speed advantage made RC4 attractive for real-time applications and large data streams. RSA is computationally intensive, especially as key sizes increase. Its operations (modular exponentiation) are slower, making it unsuitable for encrypting large volumes of data directly. Instead, RSA is typically used for encrypting small data segments, such as symmetric keys or digital signatures, rather than bulk data.
b) Security: RC4's security weaknesses are well-documented. Vulnerabilities in its key scheduling and output biases have led to successful attacks in real-world protocols, prompting its deprecation in favor of more secure alternatives. Its use is now strongly discouraged for any sensitive or mission-critical applications. RSA remains secure when implemented with sufficient key length (2048 bits or higher). Its asymmetric nature allows for secure key exchange and authentication, making it a cornerstone of modern cryptographic infrastructure, including SSL/TLS, email encryption, and digital signatures.
c) Practical Use Cases and Trends: RC4 was once widely used in protocols like WEP and early versions of SSL/TLS, but due to its vulnerabilities, it has been phased out from these and most other security-sensitive applications. RSA continues to be the preferred choice for secure communications, particularly for key exchange and digital signatures, where its asymmetric properties provide both security and scalability.
d) Experimental Results in Modern Contexts: recent studies in cloud computing environments highlight that while RC4 can offer quick encryption and decryption for certain operations (such as splitting and encrypting cloud databases), its security limitations outweigh its performance benefits in most contemporary scenarios. RSA, though slower, is favored for its robust security, especially in environments where data confidentiality and integrity are paramount.

Analysis and Discussion

Overall, the empirical benchmarks unanimously confirm the longstanding speed-security trade-off: RC4 (and other stream ciphers) deliver orders-of-magnitude faster throughput than RSA, but at the cost of well-documented cryptographic weaknesses. Surveys and tutorial papers emphasize that RSA remains indispensable for key exchange and digital signatures, while RC4 is now widely discouraged or replaced in security-critical contexts. Several studies propose lightweight RC4 variants or hybrid RSA+RC4 approaches, yet these often rely on entropy-based validation without formal cryptanalysis. Methodological diversity, different key sizes, hardware, and test data—makes direct numerical comparison difficult. Only a handful of works (e.g., [1], [14]) provide reproducible benchmarks on modern devices. There is a conspicuous lack of end-to-end hybrid pipeline evaluation, standardized test suites, and security-effectiveness analysis under real-world threat models. Future work should therefore focus on holistic, reproducible, and security-aware benchmarking across heterogeneous environments.

Methodology

This study employs a practical, experimental approach to compare the performance and characteristics of the RC4 (stream cipher) and RSA (asymmetric block cipher) encryption algorithms. The methodology involves implementing both algorithms using C# and evaluating them on multiple file types through systematic encryption and decryption operations. The aim is to assess and contrast their speed, functionality, and operational complexity.

RSA Encryption and Decryption Implementation:

The RSA encryption scheme was implemented using the RSACryptoServiceProvider class provided by the .NET framework. The following steps outline the process:

a. Key Generation

- A public–private key pair is generated using the RSA algorithm.
- The **public key** is used for encryption, while the **private key** is used for decryption.

b. File Selection and Reading

- The user selects a file using the OpenFileDialog interface.
- The selected file is read and its content is converted into a byte array.

c. Chunking and Encryption

- Due to RSA's limitation on maximum encryptable data size (based on key length), the input file is **divided into smaller chunks** (using a defined buffer size).
- Each chunk is encrypted separately using the RSA **public key** with OAEP padding (Optimal Asymmetric Encryption Padding) for enhanced security.

- All encrypted chunks are concatenated into a single byte array and written to the output encrypted file.

d. Decryption

- The RSA-encrypted file is loaded and divided back into the same chunk sizes.
- Each chunk is decrypted individually using the RSA **private key**.
- The decrypted data is reconstructed and saved as the original file.

RC4 Encryption and Decryption Implementation

RC4 was implemented as a symmetric stream cipher using custom logic, based on the standard key scheduling algorithm (KSA) and pseudo-random generation algorithm (PRGA). The following steps were followed:

a. Key Generation

- A random RC4 key was generated using Guid.NewGuid().ToByteArray(), ensuring a unique key for each session.

b. File Encryption

- The file is read and converted into a byte stream.
- Using the RC4 algorithm, the key stream is XORed with the file data to generate the ciphertext.
- The encrypted file is saved for later decryption.

c. File Decryption

- The encrypted file is read.
- The RC4 decryption function uses the same key to XOR the ciphertext, reversing the encryption process and restoring the original plaintext.

Experimental Setup

The used platform is Windows OS, .NET Framework (C#), Visual Studio IDE. The files used are; text and image files ranging in size from 10 KB to 200 KB. The metrics Measured are; the encryption Time, decryption Time, files size before and after encryption. Each encryption and decryption operation were repeated three times per algorithm per file type to ensure accuracy. Average values were computed to mitigate anomalies due to system load or caching.

Evaluation Criteria

The evaluation was based on the following criteria:

- Performance**:** Time taken to encrypt and decrypt files.
- Security**:** Discussion based on known algorithmic vulnerabilities (RC4) and robustness (RSA).
- Complexity**:** Practical implementation effort, key management, and encryption process flow.

Results

This section presents the experimental findings from the implementation and testing of the RC4 and RSA encryption algorithms. The evaluation focuses on encryption time**,** decryption time, and file size impact, using both text and image files of varying sizes. The tests were conducted on files ranging from 10 KB to 200 KB, and each operation was executed three times to obtain averaged results. The results are presented to compare the two algorithms in terms of performance efficiency and operational suitability. Figure 1 shows the comparison of the encryption and encryption operations.

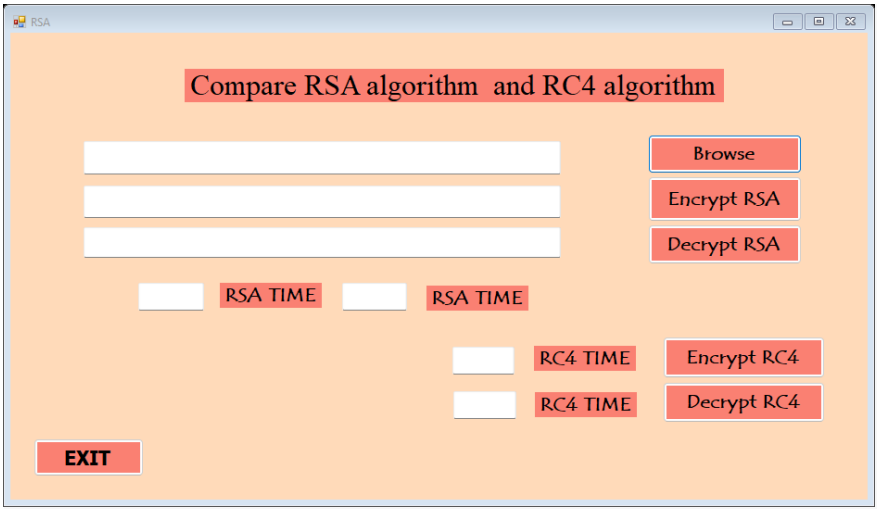

Figure 1: Interface for comparing the encryption and encryption algorithms.

Encryption Time Comparison

The encryption time was measured for both RC4 and RSA across multiple file types. RC4 consistently demonstrated superior speed, requiring significantly less time to encrypt files compared to RSA. Table 2 shows the encryption time comparison.

Table 2: Encryption Time Comparison.

File Type	File Size (KB)	RC4 Encryption Time (ms)	RSA Encryption Time (ms)
Text File	10 KB	2	45
Text File	100 KB	6	168
Image File	200 KB	14	320

As file size increases, RSA's encryption time grows rapidly, while RC4 remains consistently fast due to its stream-based structure.

Decryption Time Comparison

Decryption results followed the same trend, with RC4 demonstrating faster decryption speeds compared to RSA. Table 3 below shows the decryption **time comparison.**

Table 3: Decryption Time Comparison.

File Type	File Size (KB)	RC4 Decryption Time (ms)	RSA Decryption Time (ms)
Text File	10 KB	2	41
Text File	100 KB	5	159
Image File	200 KB	13	312

RSA's decryption, which involves intensive mathematical operations, was significantly slower across all test cases.

Performance Summary

The RC4 outperformed RSA in both encryption and decryption across all file types tested, confirming its suitability for scenarios where speed and lightweight processing are priorities. Table 4 below shows the performance summary.

Table 4: Performance Summary.

Algorithm	Average Encryption Time (ms)	Average Decryption Time (ms)
RC4	7.33	6.67
RSA	177.67	170.67

In terms of Implementation Complexity; the RC4 is simple to implement, required fewer computational resources, and completed operations quickly. However, the RSA more complex, required key generation and chunked processing due to encryption size limitations, and consumed significantly more time.

Summary of Findings

The RC4 is considerably faster than RSA for both encryption and decryption. RSA, while secure, is not practical for large file encryption due to longer processing times and chunking overhead. The results support the use of RC4 in performance-sensitive, non-sensitive environments, and RSA for secure key exchange and digital signatures, rather than bulk encryption.

Discussion

The results of this study clearly demonstrate the contrasting characteristics of the RC4 and RSA encryption algorithms in terms of performance, security, and practical applicability.

Performance Analysis

From the performance standpoint, RC4 significantly outperforms RSA in both encryption and decryption time. For all tested file sizes and types, RC4 demonstrated near-instantaneous execution, while RSA introduced notable delays, especially as file size increased. This outcome aligns with existing literature, which consistently reports RC4's high speed due to its lightweight stream cipher architecture.

The computational cost of RSA arises from its mathematical complexity, particularly modular exponentiation and key management, which is inherent in all public-key cryptosystems. Although chunking and buffering techniques were applied to mitigate some of these performance limitations, RSA remains inefficient for direct file encryption, especially in scenarios involving large or real-time data.

Security Considerations

Despite its speed advantage, RC4 is widely recognized as insecure. Its key scheduling algorithm (KSA) suffers from known biases and vulnerabilities that make it susceptible to key recovery and

ciphertext-only attacks. These weaknesses have led to the deprecation of RC4 in modern security protocols such as SSL/TLS and WPA.

In contrast, RSA remains a trusted and widely deployed algorithm for securing communications, particularly in tasks involving key exchange, digital signatures, and secure authentication. Its asymmetric nature, when coupled with sufficient key length (2048 bits or more), provides robust protection against known attack vectors. This reinforces the modern practice of using RSA not for bulk data encryption, but rather to protect symmetric keys (e.g., AES) in hybrid cryptographic systems.

Practical Implications

The findings of this study support the principle of "algorithm-task matching":

- RC4**,** while deprecated for secure applications, may still serve in non-sensitive or internal environments where performance is critical and confidentiality risks are minimal.
- RSA should continue to be employed in security-sensitive contexts, but only for key distribution or digital signature tasks, not for direct file encryption.

This distinction is particularly relevant in systems design, where hybrid encryption schemes (e.g., RSA + AES or RSA + RC4 in legacy cases) are commonly used to balance performance and security. Your implementation reinforces the rationale behind this approach and offers experimental evidence for its continued relevance.

Conclusion

This study conducted an empirical comparison of RC4 and RSA encryption algorithms across multiple file types and sizes. The results confirmed that; RC4 is significantly faster than RSA in both encryption and decryption operations, especially for small to medium files. RSA, while slower, offers superior security, and remains suitable for secure key exchanges and digital authentication. RC4 introduces minimal overhead in terms of file size, while RSA encryption substantially increases file size due to padding and chunking requirements. These findings validate the well-established trade-off between speed and security in cryptographic systems. While RC4 may still have limited utility in low-risk environments, its inherent vulnerabilities render it unsuitable for protecting sensitive information. Conversely, RSA remains an essential component of modern secure communications, but its limitations in speed and efficiency necessitate its integration into hybrid encryption architectures rather than use for direct data encryption.

Regarding the future work; future research can explore the following points:

- Integration of hybrid encryption systems combining RSA and symmetric algorithms (e.g., AES instead of RC4).
- Benchmarking energy consumption and resource utilization on low-powered or embedded devices.
- Implementing attack simulations to further assess the resilience of both algorithms under practical threat models.
- Enhancing usability through the development of a cross-platform encryption GUI tool.

References

1- Yüksel, T., & Özgün, B. (2021). RSA ve RC4 Algoritmalarının Performans Karşılaştırması. *AURUM Journal of Engineering Systems and Architecture*, *5*(1), 29-40.

2- Hossain, M. A., Hossain, M. B., Uddin, M. S., & Imtiaz, S. M. (2016). Performance analysis of different cryptography algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering*, *6*(3).

3- Al-badrei, H. H., & Alshawi, I. S. (2021). Improvement of RC4 security algorithm. *Adv. Mech*, *9*(3), 1467-1476.

4- Ghaz, A., Seddıkı, A., & Nouıoua, N. (2022). Comparative study of encryption algorithms applied to the iot. *The Eurasia Proceedings of Science Technology Engineering and Mathematics*, *21*, 469-476.

5- George, D. J., & Thomas, T. (2023). A Comparative Study of Symmetric Key Algorithms.

6- Ramakrishna, D., & Shaik, M. A. (2024). A comprehensive analysis of cryptographic algorithms: Evaluating security, efficiency, and future challenges. *IEEE Access*.

7- Sharma, P., Kumari, R., & Bansal, P. (2024, July). Cryptography: A Comparative Analysis of Symmetric and Asymmetric Techniques. In *2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)* (pp. 1343-1348). IEEE.

8- Olutola, A., & Olumuyiwa, M. (2023). Comparative analysis of encryption algorithms. *European Journal of Technology*, *7*(1), 1-9.

9- Rahul, B., & Kuppusamy, K. (2023). Efficiency analysis of cryptographic algorithms for image data security in cloud environment. *IETE Journal of Research*, *69*(9), 6053-6064.

10- Meftah, M. M., Sa'ad, H. H. Y., Al-Ashmoery, Y., Saad, A. M. H., Sa'd, A. H. Y., & Alwesabi, K. (2024, December). A Comparative Analysis of Cryptography Algorithms in Information Security.

In *2024 10th International Conference on Computing, Engineering and Design (ICCED)* (pp. 1-6). IEEE.
11- Syamala, L. S. R., Kavuluri, V. S. C., Dasa, S. S., Lakshman, D. S., Kumar, M., & Kumar, S. (2024, September). A Comparative and Exploration Analysis of Cryptographic Algorithms for Securing Digital Device. In *2024 7th International Conference on Contemporary Computing and Informatics (IC3I)* (Vol. 7, pp. 1142-1148). IEEE.
12- Singh, P., Choudhary, N., Samnotra, B., Bhel, S., Sharma, S., Kour, H., ... & Kumar, S. (2024). Understanding RSA Algorithm in Cryptography.
13- Shah, A. M., & Gor, A. (2025). Comprehensive Survey of Symmetric and Public-Key Cryptographic Algorithms: Foundations, Attacks, and Applications.
14- Frugh, Q. A., Naseri, M. F., & Hakimi, M. (2024). Experimental Comparison of Encryption Algorithms On Smart Devices. *TIERS Information Technology Journal*, *5*(2), 184-192.
15- Gahan, A., & Devanagavi, G. D. (2019). A empirical study of security issues in encryption techniques. *International Journal of Applied Engineering Research*, *14*(5), 1049-1061.
16- Karanam, M., Reddy, S., Chakilam, A., & Banothu, S. (2023). Performance evaluation of cryptographic security algorithms on cloud. In *E3S Web of Conferences* (Vol. 391, p. 01015). EDP Sciences.
17- Rasheed, A. M., & Kumar, R. M. S. (2025). Efficient lightweight cryptographic solutions for enhancing data security in healthcare systems based on IoT. *Frontiers in Computer Science*, *7*, 1522184.
18- Frimpong, T., Hayfron Acquah, J. B., Missah, Y. M., Dawson, J. K., Ayawli, B. B. K., Baah, P., & Sam, S. A. (2024). Securing cloud data using secret key 4 optimization algorithm (SK4OA) with a non-linearity run time trend. *PloS one*, *19*(4), e0301760.
19- Okedola, A. A., & Asafe, Y. N. (2015). RSA and RC4 Cryptosystem Performance Evaluation Using Image and Text File. *Int J Sci Eng Res*, *6*, 289-294.
20- Mohammed, W. A. Y., Fattah, S., Saeed, K. M. O., Ibrahim, A. O., & Eltahier, S. (2025). Enhancing the RC4 Algorithm by Eliminating the Initiative Vector (IV) Transmission. *Engineering, Technology & Applied Science Research*, *15*(1), 20242-20248.
21- Fan, Y., Biswas, P., & Garman, C. (2024, December). R+ R: A Systematic Study of Cryptographic Function Identification Approaches in Binaries. In *2024 Annual Computer Security Applications Conference (ACSAC)* (pp. 1092-1108). IEEE.
22- Hakim, A. R., Budiman, M. A., & Nasution, M. K. (2024, October). Hybrid cryptosystem enhanced RSA and RC4 chaotic map: A tutorial. In *AIP Conference Proceedings* (Vol. 3222, No. 1, p. 030005). AIP Publishing LLC.