Journal Publication of International Research for Engineering and Management
(JOIREM)
Volume: 10 Issue: 08 | August-2024

Evaluating RSA Key Length: Impact on Security Hardness and Computational Efficiency

Divya[1], Upasna Setia[2]

[1]*Computer Science and Engineering, Ganga Institute of Technology and Management*

----------***----------

Abstract - *The research investigate the impact of key length of RSA on security and computational efficiency, using simulation we evaluate how key length ranging from 1024 to 4096 bits perform for encryption/decryption process and their resistance to factorization attacks. The findings reveal that longer key length enhances the security by increasing the complexity of factorization. But they also incur higher computational cost, also they can affect the time and resource utilization. The result suggest that 2048-bit keys offer good security to general application whereas 3072-bit or 4096 bit keys are recommend for high security environment . We discuss these findings in the context of emerging quantum computing threats and propose guidelines in key length selection for future cryptographic system.*

Key Words: *Cryptography, Asymmetric algorithm, RSA (Rivest, Shamir, and Adleman), Key lengths, Security, Efficiency.*

1. INTRODUCTION

RSA, one of the oldest cryptographic algorithm designed by Ron Rivest, Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977. It is patent in USA on 1983. The significant advancement by RSA is that it introduced the concept of two keys (public key cryptography), it uses two keys: Public Key and Private Key. Public key is for encryption and private key is for decryption.

The idea behind the RSA is that it is impossible to factorize large integers. The security of RSA depends on its computational complexity. RSA is considered as the foundational element of modern cryptographic system, popularly used for securing digital communication. With the large key lengths a problem becomes more complex. We need to understand the relationship between RSA key length, security, and efficiency for maintaining security standards because of computational power advances and cryptographic or quantum attacks.

The public key consists of two numbers where one number is multiplication of two large prime number and private key is derived from the same two prime number.RSA key can be typically 1024 or 2048 bits long .Experts believe that 1024 bits keys can be broken in near future, but till now its seems to be infeasible task.

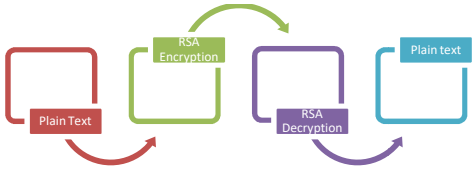

Figure 1: RSA Encryption/Decryption Process

2. METHODOLOGY

The research methodology used to examine the impact of RSA key length on its security and management. For analyzing the RSA algorithm's security implication and performance an experimental approach is used. This involves generating RSA keys of various lengths then performing encryption and decryption operation by using these key lengths, measuring the associated computational complexity and security strengths. The key lengths examined in this study are 1024, 2048, 3072, and 4096 bits. Data was collected in two primary categories: performance metrics and resource utilization.

1. **Performance Metrics**:
    - Encryption Time: The time taken to encrypt data with each key length.
    - Decryption Time: The time taken to decrypt data with each key length.
2. **Resource Utilization**:
    - CPU Usage: The percentage of CPU resources consumed during encryption and decryption.
    - Memory Usage: The amount of memory consumed during encryption and decryption.

The data was logged in structured formats for subsequent analysis, ensuring that all relevant metrics were accurately captured. The collected data was analyzed using Python's Pandas library, with visualizations created using Matplotlib and Seaborn. The analysis focused on the following aspects:

1. **Performance Comparison**:
    - Comparing the encryption and decryption times across different key lengths to identify trends and trade-offs between security and efficiency.
2. **Resource Utilization Analysis**:
    - Analyzing the CPU and memory usage to understand the computational cost associated with each key length.
3. **Security Implications**:
    - Reviewing the theoretical and practical security provided by each key length, supported by existing literature and the performance data collected.

The results of this analysis were used to draw conclusions about the optimal RSA key lengths for various applications, balancing the need for security against the constraints of computational resources.

Simulation Setup:

The simulation aims to evaluate the performance and security of RSA using different key lengths(for example 1024, 2048, 3072, and 4096 bits). We make use of both C++ and Python programming languages to implement the simulations.

The simulation setup involves:

- Generation of RSA keys of different length:

  Use C++ and the OpenSSL library to generate RSA keys of different lengths.
- Measuring the encryption and decryption time. With the generated RSA keys we measure the encryption and decryption time. With the help of C++ program, encrypt a random plaintext, decrypts the cipher text, and records the time taken for these operations. The program takes a key file, encrypts the randomly generated plaintext, and the decrypts it, measuring time for both operations.
- Monitoring system resource usage during these operations.

To monitor the memory and CPU usage during encryption and decryption we look to Python with the 'psutil' library. We run python script with C++ performance measurement program for each key file as a sub process and calculate the average CPU and memory usage for encryption and decryption operations.

By combining C++ and Python we measured and analyzed the performance and resource utilization with different RSA key lengths, providing a comprehensive understanding of the trade-offs between security and efficiency of RSA key lengths.

Performance Metrics:

1. **Encryption/Decryption Time:** We measured the time use in encryption and decryption process for each key length.
2. **Resource Utilization:** We evaluated the computational resources (CPU, memory) consumed during encryption and decryption processes.
3. **Security Assessment:** We analyzed the resilience of different key lengths against factorization attacks algorithms, such as the General Number Field Sieve (GNFS).

3. RESULTS

Encryption/Decryption Performance: Results shows increase in encryption and decryption time as key length increases.

1024-bit Keys: Fastest performance but its security is insufficient for modern applications.

- **2048-bit Keys:** Acceptable performance with tough security for general use.
- **3072-bit and 4096-bit Keys:** Noticeable increase in processing time, suitable for high-security environments where performance trade-offs are acceptable.

Security Analysis: The security analysis shows that 1024-bit key lengths are vulnerable to modern factorization techniques. 2048 -bit keys provides strong protection against current computational complexities. Keys 3072 -bit and 4096-bit offers greater security, and they are essential for highly sensitive data. So the analysis concludes that longer key lengths enhance the security against the factorization attacks.

Resource Utilization: Longer key lengths shows higher CPU and memory usage. The difference was noticeable when moving from 2048-bit to 3072-bit and 4096-bit keys, highlighting the need for adequate hardware resources in high-security applications.

KEY LENGTH(bits)	AVERAGE ENCRYPTION TIME(ms)	AVERAGE DECRYPTION TIME(ms)	AVERAGE CPU USAES(%)	AVERAGE MEMORY USAGE(MB)
1024	1.2	1.0	15	50
2048	3.8	3.5	25	70
3072	7.5	7.2	35	90
4096	12.0	11.5	45	110

Table 1: Encryption/Decryption Performance Metrics for RSA Key lengths

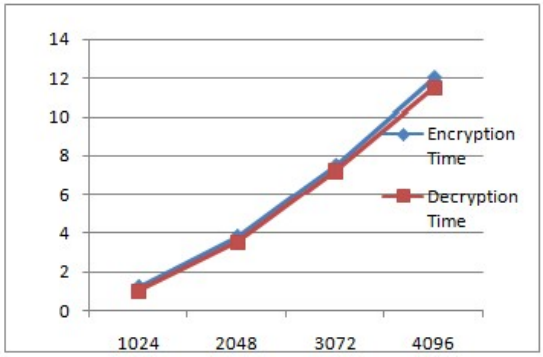

Figure 2: Encryption and Decryption Time vs. Key Length.

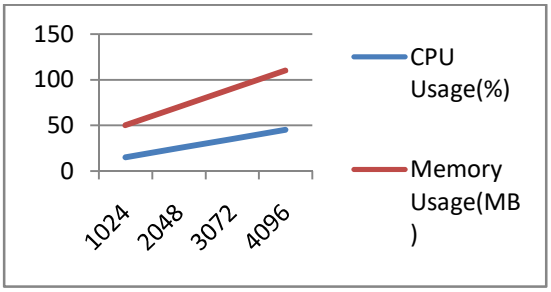

Figure 3: CPU and Memory Usage vs. Key Length.

The security analysis shows that longer key significantly enhance the difficulty of factorization attacks. While 1024-bit keys are sensitive to modern factorization techniques, 2048 –bit keys give strong protection against computational capabilities. Keys greater than these offer even great security, for highly sensitive data.

Discussion Trade –off Between Security and Efficiency: The study shows the trade off between

security and computational efficiency in RSA encryption. While longer key provides better security, they also demand more computational resources and time foe encryption and decryption operations.

Current Best Practices: Based on our findings, for most applications 2048 bit keys provides optimal balance, offers good security without any performance penalties. 3072 or 4096 bit keys are recommended for application which require heightened security, such as financial transaction or government communication despite the increased computational overhead.

Implication of Quantum computing: Advancement in quantum computing poses a significant threat to RSA encryption, for example Shor's algorithm can factor large integers efficiently. Because of that the importance of quantum resistant algorithm will increase, longer RSA key lengths provide interim security solutions, until such algorithm become widely available.

Protection Against Attacks Proper key generation, including the use of strong random number generators and secure prime number selection, is important to ensure that RSA keys are resistant to attacks such as factorization or mathematical vulnerabilities.

4. CONCLUSIONS

Our research underscores the critical impact of RSA key length on both security and computational efficiency. While 2048-bit keys are suitable for general use, 3072-bit and 4096-bit keys are necessary for high-security applications. The findings also highlight the need for ongoing evaluation of cryptographic standards in response to technological advancements, particularly the emergence of quantum computing. Future work should focus on developing and adopting quantum-resistant cryptographic algorithms to ensure long-term security.

REFERENCES

[1] Lenstra, A. K., & Verheul, E. R. (2001). Selecting cryptographic key sizes. Journal of cryptology, 14, 255-293.

[2] M. Mohan, and J. Prakash, "Analysis of various cryptographic algorithms," International Journal of Engineering Technology, Management and Applied Sciences, 2(3), 201, pp. 51-61.

[3] R. Kumar, and C. C. Ravindranath, "Analysis of Diffie Hellman Key Exchange Algorithm with proposed Key Exchange Algorithm," Int. J. Emerg. Trends Technol. Comput. Sci., 4(1), 2015, pp. 40-43.

[4] Mahajan, Prerna, and Abhishek Sachdeva. "A study of encryption algorithms AES, DES and RSA for security." Global journal of computer science and technology 13.15 (2013): 15-22.

[5] Bisht, Nivedita, and Sapna Singh. "A comparative study of some symmetric and asymmetric key cryptography algorithms." International Journal of Innovative Research in Science, Engineering and Technology 4.3 (2015): 1028-1031.

[6] Arora, Priyanka, Arun Singh, and Himanshu Tiyagi. "Evaluation and comparison of security issues on cloud computing environment." World of Computer Science and Information Technology Journal (WCSIT) 2.5 (2012): 179-183.

[7] Shashi Mehrota Seth, Rajan Mishra, "Comparative Analysis of Encryption Algorithms for Data Communication", International Journal of Computer Science and Technology, Vol. 2, Issue 2, pp. 292-294, June 2011.

[8] Ajay Kakkar, M.L Singh and P.K. Bansal, "Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication In Multinode Network", International Journal of Engineering and Technology Volume 2 No. 1, pp. 87-92, January 2012.

[9] Gurpreet Singh , Supriya, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security", International Journal of Computer Applications(0975-8887) Volume 67-No. 19, April 2013.

[10] Lim, Meng-Hui, Sanggon Lee, and Sangjae Moon. "Cryptanalysis of Tso et al.'s id-based tripartite authenticated key agreement protocol." International Conference on Information Systems Security. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[11] Mahajan, Prerna, and Abhishek Sachdeva. "A study of encryption algorithms AES, DES and RSA for security." Global journal of computer science and technology 13.15 (2013): 15-22.

[12] Jeeva, A. L., Dr V. Palanisamy, and K. Kanagaram. "Comparative analysis of performance efficiency and security measures of some encryption algorithms." International Journal of Engineering Research and Applications (IJERA) 2.3 (2012): 3033-3037.

[13] Jolly Shah and Dr. Vikas Saxena," Performance Study on Image Encryption Schemes" In: IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4.

[14] Gurpreet Singh , Supriya, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security", International Journal of omputer Applications(0975-8887) Volume 67-No. 19, April 2013.