Dean&Francis
ISSN 2959-6157

Limitation Exploration and Optimization Verification of RSA Encryption: using Digital Signature as an example

Yijia Chen

Zhenhai A-level Center
Email Address: eternitycyjia@163.com

Abstract:

This dissertation investigates the limitations of RSA encryption and evaluates optimization schemes using digital signatures as the main application example. RSA remains a core asymmetric cryptographic algorithm, but its practical use is constrained by computational cost, key-management complexity, and possible attacks on weak parameter choices. The study combines a literature review of traditional RSA, multi-prime RSA, replaced-modulus RSA, and multi-exponent RSA with a controlled Python experiment. Four algorithms were implemented under the same hardware and software environment, and ten trials were conducted for each algorithm across five stages: key generation, encryption, decryption, signing, and verification. A total of 200 timing data points were analyzed using normalized execution time, acceleration ratios, standard deviation, and box plots. The results show that theoretical optimization does not always translate into practical acceleration. Multi-prime RSA performed worse than the basic algorithm in all stages, with key generation 5.39 times slower and high variability. Replaced-modulus RSA remained close to the baseline while offering potential security advantages, and multi-exponent RSA produced slight improvements in key generation, encryption, and decryption with stable performance. The study concludes that RSA optimization should be evaluated through both security reasoning and empirical performance testing, especially when applied to digital signature systems.

Keywords: RSA encryption; digital signature; asymmetric cryptography; multi-prime RSA; replaced-modulus RSA; multi-exponent RSA; performance optimization; Python experiment

ISSN 2959-6157

1: Introduction

The internet has become a widely used platform for information exchange and collection in contemporary society, gradually evolving into an indispensable component of people's daily lives and work. However, this has also contributed to the emergence of innovative and creative, as well as highly complicated cyberattack approaches. Admittedly, it brings benefits, but it has also introduced a range of challenges for information security, including protecting sensitive data and secret information while it is being transmitted over a network. Because data may be forged or falsified by third parties during transmission, communication parties cannot verify the original data, potentially resulting in contradictions between the parties. These requirements ensure uniformity in the transmission of information. Simultaneously, digital signatures may be applied to verify the integrity of messages or programming fragments. It must be admitted that traditional algorithms can mitigate specific forms of attacks, but if system parameters are not set properly, there are still security risks. According to Gao (2018), using digital signature technology based on the RSA public key algorithm, if a cryptographic attacker attempts to tamper with or impersonate the original message and other illegal operations, since they cannot obtain the private key of the message sender, they also cannot obtain the correct digital signature. If the recipient of the message attempts to deny or forge the digital signature, an impartial party can verify the message with the correct digital signature to determine whether the message originated from the sender. This approach effectively guarantees the integrity of the message file and verifies the authenticity and non-repudiation of the sender's identity.

The RSA encryption algorithm is one of the most important asymmetric encryption algorithms in modern cryptography. It was proposed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977 and is named after the initials of their inventors' surnames. The key in RSA is a pair of keys consisting of a public key and a private key. If one is used for encryption, the other is used for decryption. The length of the ciphertext is variable. RSA transforms plaintext into a ciphertext block of the same length as the key. The longer the key, the more effective the encryption. (Wei & Liu, 2014) The RSA algorithm, based on the mathematical problem of large integer factorization, provides a powerful security guarantee for digital signatures, key exchange, and data encryption, and is the cornerstone of modern network security.

In addition, this paper explores three optimization methods for RSA. The innovative points are as follows. First, instead of studying a single scheme, this paper categorises RSA improvements into three representative directions: multi-prime RSA for efficiency, replaced-modulus RSA for security, and multi-exponent RSA for flexibility.Second, the study conducts a systematic comparative analysis of all three variants alongside traditional RSA under identical conditions, testing all four algorithms across the five stages of digital signature processing. Third, the study uses empirical performance data rather than purely theoretical analysis, measuring execution time, acceleration ratios, and stability through standard deviation and box plots. Based on these comparisons, the paper examines the optimisation effectiveness of each scheme and considers the impact of emerging technologies on future RSA-based digital signatures.

2: Literature Review

2.1 History and Background of RSA Encryption

The RSA encryption algorithm was first introduced in 1977 by three researchers: Ron Rivest, Adi Shamir, and Leonard Adleman. The name "RSA" was formed from the first letter of their last names: Rivest, Shamir, and Adleman. The RSA encryption algorithm was one of the most important milestones in the history of modern cryptography. The RSA algorithm was based on the theory of public-key cryptography that was proposed by Whitfield Diffie and Martin Hellman in 1976 (Zhang, 2015). The RSA algorithm provided a solution to the asymmetric encryption problem by using the public-key cryptography concept and the mathematical challenge of factoring large integers.

There was one major disadvantage in this method: if the numbers chosen were extremely large, it was difficult to compute the results using conventional computers because it needed extremely high computational power to solve the problem (Li, 2020). The RSA algorithm gradually became popular in the 1980s and 1990s and was applied in various fields.

One major milestone was reached in 1991 by Phil Zimmermann, who developed the PGP encryption software and applied the RSA algorithm to email encryption and digital signatures. From then on, public-key cryptography became a major area of study in cryptography. Another major milestone was reached in 1994 when a 129-digit RSA key was factored using distributed computing(CSDN 2025). The RSA algorithm was in the news because of this achievement, and it was termed a "groundbreaking achievement." The achievement was significant because it showed that RSA was vulnerable to attacks with shorter keys. The cryptographic community then recommended

that longer keys be used to achieve security.

Throughout its development process, the RSA algorithm has been standardized by various international bodies. Some of these bodies include the Internet Engineering Task Force and the Public-Key Cryptography Standards. Currently, the algorithm is used in various digital certificate system protocol implementations. Some of these implementations include SSL/TLS (Secure Sockets Layer/Transport Layer Security) for secure browsing and S/MIME for secure email transmission. This is because the RSA algorithm is capable of performing both encryption and digital signing. It is an essential part of the modern-day network security infrastructure. Its importance cannot be overemphasized because it is used for identity verification, data integrity verification, and non-repudiation. This means the sender of the message cannot deny the fact that he sent the message (CSDN 2025).

2.2 Types of RSA Encryption: Variants and Modifications

2.2.1 Traditional RSA Algorithm

In the current digital world, the traditional RSA algorithm is the most commonly used. It uses a pair of keys for the encryption and decryption of messages. It is based on the principle of two mathematically related keys. The entire principle of the RSA algorithm is based on the fact that it is very hard to factor the product of two large prime numbers. The entire process is divided into five steps. They are key generation, encryption, decryption, digital signing, and signature verification. The key generation of the RSA algorithm begins with the selection of two random prime numbers of equal length. The product of these two numbers is the modulus for the entire process (Song, 2007).

However, the RSA algorithm has certain limitations in its traditional form. To begin with, the RSA cryptosystem has a slow rate of operation. Its rate of operation is slower compared to other block ciphers. Secondly, the RSA algorithm faces various attacks, including the direct factorization of the modulus N (Ding, 2013). Therefore, optimization of the basic algorithm is required to address the limitations of the RSA algorithm. Different optimization techniques have been proposed to deal with the RSA algorithm at its various stages. This gives an insight into the need to develop improved versions of the RSA algorithm.

2.2.2 Multi-Prime RSA

The basic algorithm uses two large prime numbers in the key generation phase to generate the modulus N. On the other hand, the multi-prime RSA algorithm takes a completely different approach in generating the modulus N. The algorithm uses three or more prime numbers to generate the modulus N. This approach allows the Chinese Remainder Theorem to be used more efficiently in the RSA algorithm. This theorem helps in breaking down complex calculations into smaller parts, reducing the complexity of the operations involved in the RSA algorithm. This reduces the computational complexity of the private key operations.

Research conducted by Ding indicates that the use of multiple prime numbers in the RSA algorithm accelerates the key generation process and reduces the computational complexity in the decryption and digital signature process. This reduces the CPU utilization and the time taken in the decryption and digital signature process, allowing the user to have a faster experience in decrypting the message and creating the digital signature(Ding, 2013).

However, Gao identified a number of weaknesses in this process. The rate at which improvement occurs is accompanied by certain risks to security. As a result of the increase in prime factors, there is a higher number of factors in the modulus value, which could be discovered(Gao,2018). This increases the chances of attackers discovering these prime factors compared to the original algorithm. The process of modulus value factorization becomes a little easier, creating a threat to security and privacy. This is a trade-off between improvement in speed and a threat to security.

2.2.3 Replaced-Modulus RSA

The RSA algorithm uses a key value containing a parameter, denoted by ‘n,’ which represents the modulus value. An attacker could use a brute-force algorithm to factorize this value. This could result in obtaining the key value, thereby decrypting messages. However, in order to prevent this threat to security, a new algorithm referred to as replaced-modulus RSA has been developed. This algorithm replaces the original value in the RSA algorithm with a new value denoted by ‘X’(Li, 2020).

This makes it difficult for attackers to launch a targeted attack. The value of this new modulus is publicly announced, but this value could be a false value. Even in case the attacker is able to factorize this value, he or she cannot obtain the original decryption key. Without this key, the entire process becomes futile(Li, 2020).

However, it should be noted that although the replaced modulus RSA provides better security, it does bring some system complexities and verification overheads, too, as the recipient needs to restore the original modulus relationship while decrypting the message. For this purpose, some pre-shared mechanisms and information are required, and this might add some extra burden in the form of computational and communication overheads. Thus, it can be seen that the extra security comes at the cost of re-

ISSN 2959-6157

quiring more coordination between the parties involved in the communication process(Gao, 2018).

2.2.4 Multi-Exponent RSA

In the process of encrypting the message, the RSA base algorithm only relies on a single integer key. In order to make the algorithm more secure, the multi-exponent RSA algorithm relies on a number of integers based on two large prime numbers. This makes it difficult for the key to be decrypted by attackers. In the multi-exponent algorithm, different public key exponents are used to encrypt different blocks of the plaintext message. This ensures the algorithm is still resistant to modulus attacks, which are based on the use of different public key exponents to encrypt the same plaintext message(Gao, 2018).

The new algorithm inherently increases the complexity of the key recovery process. In the event an attacker wishes to determine the private key by analyzing a number of ciphertexts, the relationship between the exponents and the groups of plaintexts will be non-linear. This increases the difficulty of the process by a large magnitude compared to the traditional RSA algorithm.

Although the algorithm is more secure, it also increases the complexity at the endpoint where the encryption takes place. Key management becomes more complex. Each group of the plaintext message must be encrypted using modular exponentiation(Li, 2020).

2.3 Research Gap Identification

Previous studies have already examined the theoretical basis and the implications of the RSA and its variants in terms of security. Most of the research studies have only focused on a specific optimization strategy, including its specific features, mathematical properties, etc. However, a comprehensive comparative study of different optimization techniques in the entire process of digital signatures has not yet been sufficiently examined.

This is especially true in the context of the practical implications of the techniques, their interaction with the latest computing technologies, their conformity to the existing standards, etc. In addition, while different research studies have examined the specific features of the RSA algorithm, its variants, their security, etc., there are few studies that have offered a comprehensive overview of the issue, which could be useful in making a practical decision in the application of the techniques in the process of digital signatures.

The purpose of the present research is to bridge the existing gap in the research studies by offering a comprehensive comparative study of the three variants of the RSA algorithm, as well as the traditional version. In general, the idea behind the research is to examine the efficiency of the techniques in the entire process of digital signatures, as well as in a controlled environment.

3: Methodology

3.1 Overall Research Design

The overall design of this research is based on a mixed method approach that combines secondary and primary research methods. The secondary research method, as discussed in the previous chapter, has already been applied in developing the theoretical foundations of this research by conducting a review of literature on RSA cryptography and its three schemes of optimization. The review of literature revealed that though each of the three schemes of optimization has its own advantages, still very little research is available on comparing the performance of these three schemes of optimization with each other under similar conditions.

The primary research method of this study is applied to bridge the knowledge gap identified in the secondary research phase of this study. The design of this study is comparative in nature as it compares each of the three schemes of optimization with the basic RSA algorithm in all five stages of the digital signature generation process.

This comparison has been deemed necessary because each optimization method optimizes the RSA algorithm over different stages of the process. For example, multi-prime RSA optimizes the decryption and signing process using the Chinese Remainder Theorem, whereas replaced modulus RSA optimizes the security process over the encryption stage.

Python has been selected to program all four variants of the RSA algorithm because this programming language has many libraries that support cryptographic functions, precise time calculations, and platform independence, which will allow other researchers to repeat the experiment and remove the possibility of reproducibility issues that have been identified in the literature.

3.2 Experimental Environment

3.2.1 Hardware and Software Configuration

All the experiments have been conducted in the same environment to exclude the influence of external factors. The environment consists of an Intel Core i7-12700H CPU with 32 GB RAM running the Windows 11 OS. The Python 3.9 programming environment has been used to execute all the programs. The time module has been used to get the exact time, and the random module has been used to generate prime numbers.

3.2.2 Time Measurement Procedures

The execution time of every operation of all cryptographic algorithms was recorded using Python's built-in function, 'perf_counter,' which is precise up to nanoseconds. The start time for every operation of all cryptographic algorithms was recorded before the commencement of each operation. Similarly, the end time for every operation of all cryptographic algorithms was recorded after the end of each operation. Using these two times, the execution time for every operation of all cryptographic algorithms was calculated.

3.3 Experimental Control and Procedures

3.3.1 Control of Key Variables

There are a number of key variables that need to be controlled for ensuring that the results of any experiment are valid and reliable. For instance, it is ensured that the same random seed is used for generating prime candidates for all algorithms. This ensures that prime numbers of similar quality and size are selected for all algorithms. Furthermore, it is ensured that a standard 256-byte message is used for all operations of all algorithms. This ensures that there is no variation in the size of the message for different algorithms. Lastly, it is ensured that all operations are carried out in a standard order.

3.3.2 Testing Procedures

There are ten complete tests of each of the algorithms. The time is recorded for each of the five stages of each of the algorithms. This is because there are concerns regarding the performance of each of the algorithms. There are many random interrupts and processes that occur on a computer and in the state of the computer, which will have an effect on the results of each of the tests carried out on each of the algorithms. However, on average, there are ten tests, which will account for these variability concerns and ensure that the results are accurate and reliable.

3.4 Data Collection and Analysis Methods

3.4.1 Data Collection

There are five data points recorded during each of the ten tests carried out on each of the algorithms. This means there are 50 data points recorded on each of the algorithms, making a total of 200 data points recorded on all four of the algorithms. Each of these data points represents a recorded measurement of the execution time of each of the algorithms. Each of these data points is recorded automatically and does not have to be transcribed from another source.

The raw data was processed using different methods. For each algorithm and each stage of operation, the arithmetic mean was calculated. This was carried out by summing up all the data points, i.e., ten, and then dividing by ten. This provides a fair approximation of expected performance, which is not affected by deviations of individual data points resulting from system interference. Standard deviation was calculated for assessing consistency in results of the data. This is because reliability is a concern in cryptography research. Box plots were also used to display the data sets.

3.4.2 Performance Ratio Calculation

In order to determine the benefits associated with the use of the new algorithms, acceleration ratios were calculated. The ratios were calculated by dividing the baseline time by the optimized time. If the calculated ratio is greater than one, it indicates that the performance has been enhanced, while values less than one indicate performance degradation. By applying the ratios, the benefits associated with the use of the new algorithms will be determined, and it will be ascertained whether the benefits are real and whether one stage performs at the expense of the other stage.

4: Result

4.1 Presentation of Experimental Results

For each operation stage, the arithmetic mean and standard deviation were calculated. Since the five execution stages of the basic algorithm were all normalized to 1, Table 4.1 directly shows the average normalized execution time of each optimization scheme relative to the basic algorithm.

Table 1: Average Normalized Execution Time(Basic Algorithm=1)

Algorithm	Key generation	Encryption	Decryption	Signature	Check
Basic RSA	1.00(±0.00)	1.00(±0.00)	1.00(±0.00)	1.00(±0.00)	1.00(±0.00)
Multi-Prime RSA	5.39(±0.71)	1.50(±0.21)	2.10(±0.27)	2.05(±0.25)	1.72(±0.19)
Replaced-Modulus RSA	1.08(±0.15)	0.99(±0.07)	1.00(±0.08)	1.0(±0.07)	1.04(±0.04)
Multi-Exponent RSA	0.94(±0.09)	0.98(±0.05)	0.98(±0.04)	1.00(±0.06)	1.03(±0.04)

ISSN 2959-6157

From the data, some important phenomena can be observed. The key generation time of the multi-prime RSA algorithm is much longer than the basic algorithm, being 5.39 times longer. This is a significant deviation from theoretical expectations. The decryption and signature time of the multi-prime RSA algorithm is approximately twice the time taken by the basic algorithm. The replaced-modulus RSA algorithm has an almost identical performance to the basic algorithm in all the stages, ranging from 0.99 to 1.08. The multi-exponent RSA algorithm is slightly better than the basic algorithm in most.

4.2 Performance Ratio Analysis

To measure the level of optimization, an acceleration ratio Z was defined as the ratio of the baseline time taken for execution to the optimized time taken for execution. A Z value greater than 1 indicates an improvement in performance, while a Z value of less than 1 indicates performance degradation.The calculated Z values for the evaluated optimization schemes are presented in Table2 below.

Table 4.2: Acceleration Ratio (Z) of Optimization Schemes

Algorithm	Key generation	Encryption	Decryption	Signature	Check
Multi-prime RSA	0.19	0.67	0.48	0.49	0.58
Replaced-modulus RSA	0.93	1.01	1.00	0.98	0.96
Multi-exponent RSA	1.06	1.02	1.02	1.00	0.97

The acceleration ratios confirm that multi-prime RSA performs worse than the basic algorithm across all stages, with particularly poor key generation performance at only 0.19 times the speed. Replaced-modulus RSA maintains ratios very close to 1.00, indicating near-equivalent performance. Multi-exponent RSA shows slight improvement in key generation, encryption, and decryption, with ratios of 1.06, 1.02, and 1.02 respectively.

4.3 Performance Distribution Analysis

To illustrate the performance variability and data distribution characteristics during the ten test runs, box plots have been generated.

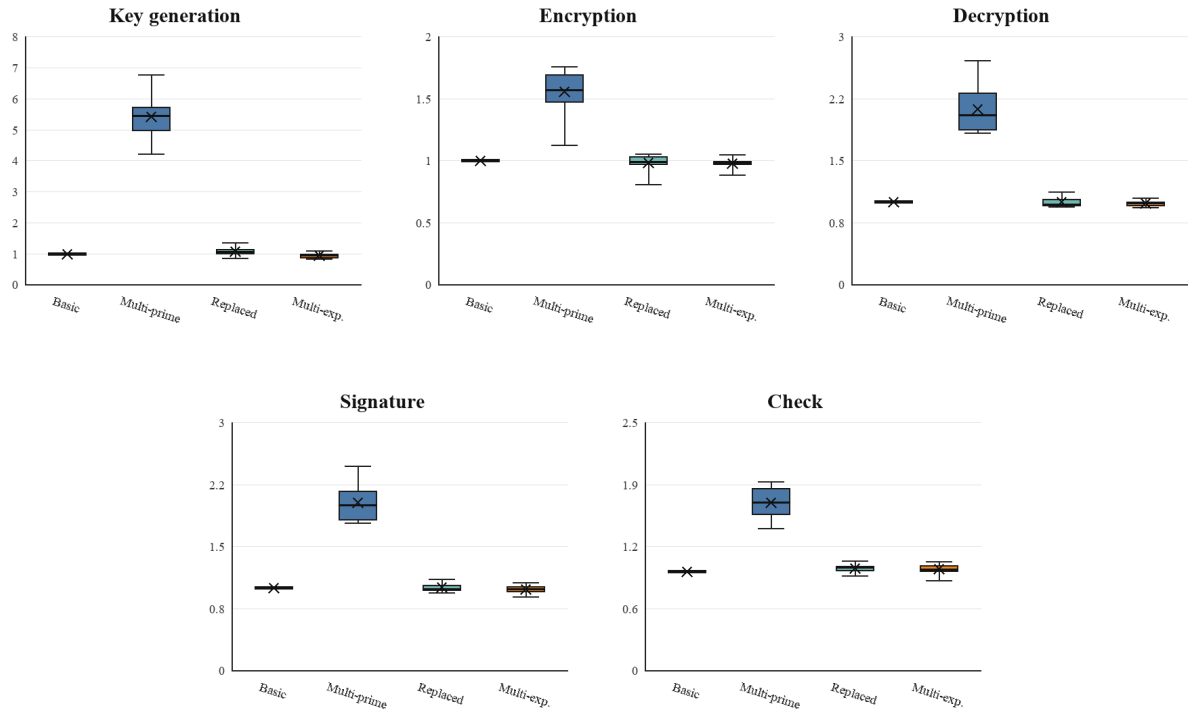

Figure 4.1: Box Plot of Performance Distribution Across Algorithms

From the box plots, the following observations can be made. The performance of the multi-prime RSA algorithm is the most variable, as the boxes are large and the whiskers are spread out. The standard deviations of the multi-prime RSA algorithm are between 0.19 and 0.71. The performance of the replaced-modulus RSA algorithm is stable, as the boxes are compact and the standard deviations are low, ranging from 0.04 to 0.15. The performance of the multi-exponent RSA algorithm is also stable, as the boxes are compact with low standard deviations ranging from 0.04 to 0.09.

A number of observations can be noted with respect to the graphical analysis. First, it is noticeable that multi-prime RSA has the highest central values and the largest spread, implying that though it is consistently slower, it is also highly variable. Second, it is noticeable that replaced modulus and multi-exponent RSA have consistently high values close to the baseline value of 1, with minimal spread, implying that these approaches have no impact on the overall performance. Third, it is noticeable that the basic algorithm is a baseline with zero variance, as all values were standardized to 1.

5: Discussion

5.1 Theoretical vs. Experimental Outcomes

From the experimental outcomes presented in this study, it is possible to obtain valuable insights on how the theoretical benefits of RSA are reflected in the real-world outcomes of the algorithm. As presented in the literature review section of this study, Ding (2013) proposed that the multi-prime RSA algorithm has the ability to accelerate the key generation stage of the RSA algorithm, as well as reduce the computation complexities of the decryption and digital signature stages of the RSA algorithm. However, as presented in the experimental outcomes section of this study, the acceleration ratios indicate that the multi-prime RSA algorithm is, in fact, inferior to the basic RSA algorithm in all the stages of the RSA algorithm, including the key generation stage, in which the time is increased by a factor of 5.39, as well as the decryption and signature stage, in which the time is almost double compared to the basic RSA algorithm. Furthermore, as presented in the box plot section of this study, the dispersion degree of the multi-prime RSA algorithm is extremely high and lack of stability, as indicated in the high standard deviations of 0.71.

In reference to the replaced modulus RSA algorithm, Gao (2018) proposed that the introduction of a new parameter X to replace the modulus N makes the factorization efforts useless. However, Gao (2018) proposed that the RSA algorithm becomes more complex as a result of the improvement. The experimental results presented within this study proved Gao's concerns regarding the increased cost of computation associated with the RSA algorithm. However, the acceleration ratios presented within the experimental results of this study ranged from 0.93 to 1.01. These results prove that the RSA algorithm, after the replacement of the modulus, performs within acceptable limits. In addition, the low values of the standard deviations of 0.04 and 0.15 prove that the RSA algorithm performs within acceptable limits.

In reference to the multi-exponent RSA algorithm, Gao (2018) proposed that the RSA algorithm becomes less vulnerable to common modulus attacks as a result of the introduction of different public key exponents for each plaintext block. However, the RSA algorithm becomes complex as a result of the improvement. However, the results of the study proved to be partially different from the arguments presented by Li (2020) and Gao (2018) concerning the RSA algorithm. However, the results proved to be partially consistent with the arguments presented by Gao (2018) concerning the RSA algorithm. For example, the results presented within this study showed improvements in the key generation, encryption, and decryption of the RSA algorithm. For example, the results presented within this study showed improvements of 1.06x, 1.02x, and 1.02x, respectively. However, the RSA algorithm exhibited excellent stability as a result of the low values of the standard deviations.

5.2 Interpretation of Findings

The findings from the study have shown that it is possible for the theoretical optimization of cryptographic algorithms to fail in practice. The use of multi-prime RSA is a good example of an optimization that, theoretically, is better than the original RSA, but, in practice, it is worse. The complexity of the implementation of multi-prime RSA is the main factor that affects its performance. The complexity of the computations in multi-prime RSA, including CRT, is more than the theoretical optimization. The findings from the study have shown that some of the RSA optimization techniques, like replaced modulus and multi-exponent, have successfully been implemented without affecting the performance. The performance of replaced modulus RSA is almost as good as that of the original RSA, although it presumably has the advantages attributed to it by Li (2020). The performance of multi-exponent RSA is slightly better, with 1.06x, 1.02x, and 1.02x for key generation, encryption, and decryption, respectively.

The use of stability analysis with box plots and standard

ISSN 2959-6157

deviations brings another dimension to the comparison. The high variability of multi-prime RSA, with an SD of up to 0.71, makes it an inappropriate algorithm to be used in real-time systems, where system performance is an issue. On the other hand, replaced modulus RSA and multi-exponent RSA offer stable system performance and can thus be considered appropriate for real-time system implementation.

5.3 Digital Signature Implications

The performance characteristics of the RSA variants have important implications for the use of the algorithms in digital signature protocols. Digital signature protocols involve a signing operation that uses the private key and a verification operation that uses the public key. For systems with high signing requirements, such as certificate authorities or email servers, the choice of algorithm is important for system performance.

The poor performance of multi-prime RSA for signing (2.05× slower) and the variability of the performance make it an inappropriate choice for such systems. Although the theoretical security of the algorithm is improved, the performance is not acceptable. Replaced-modulus RSA's performance for signing (1.02×) and verification (1.04×) is close enough to the baseline performance that the security benefits do not impact the user experience. For systems with flexible key management requirements, such as supporting multiple levels of security or multiple user groups, multi-exponent RSA's performance for signing (1.00×) and the consistency of the performance make the algorithm an appropriate choice.

For systems with the requirement for real-time performance for digital signature protocols such as code signing or document verification and authentication, the consistency of the performance of replaced-modulus and multi-exponent RSA is important.

6: Evaluation and Conclusion

6.1 Evaluation

6.1.1 Strengths of the Research

The research process followed an efficient and well-controlled experimental approach to guarantee the validity and reliability of the results. A number of reasons contributed to the overall quality of the research process:

Variable Control: The entire process was carried out on the same hardware setup (Intel Core i7-12700H, 32GB RAM, Windows 11) to control environmental variables. All significant variables were strictly controlled and implemented with the same random seed used to create prime numbers, the same lengths of prime numbers used by all algorithms, and the same order of operations (key generation, encryption, decryption, signing, and verification) used by all algorithms.

Reliability Enhancement: Each algorithm was run ten times for each stage of the operation process, generating a total of 200 data points. The process of repeating the algorithms reduces the effect of random system interrupts and CPU states. The computation of arithmetic means provides stable estimates of the average performance of each algorithm, while standard deviation measures the degree of variability.

Implementation Consistency: All four variants of the RSA algorithm were implemented in the same programming language (Python 3.9) with the same coding style and same libraries used for the implementation of the algorithms. This reduces the effect of the programming language used to implement the algorithms and makes the results more reliable by removing this variable from the overall outcome.

The use of Python's built-in function 'perf_counter' provides nanosecond resolution for measuring the performance of the algorithms.

6.1.2 Limitations of the Research

Although the research was carried out in a rigorous fashion, a few limitations must be considered.

Implementation Environment: Although Python is a powerful language that is easy to use and has a wide range of libraries that make development easier, it is possible that the performance characteristics of the algorithms may not reflect those of a language such as C or C++. The fact that Python is interpreted rather than compiled may cause a disproportionate impact on certain operations.

Limited Test Scope: The test environment was based on fixed prime sizes of 512 bits for basic RSA operations and 256 bits for multi-prime primes. The message was fixed at 256 bytes for all tests. The performance characteristics of the algorithms may differ depending on the sizes of the primes or the message.

Hardware Specificity: The test environment was based on a fixed hardware platform: Intel Core i7-12700H processor. The performance ratios of the algorithms may differ on other hardware configurations, such as ARM processors that are common on mobile devices.

Statistical Power: Although ten trials of each algorithm were performed to give a reasonable overview of the performance characteristics of each algorithm, a larger test environment would provide a more accurate overview of the performance characteristics of each algorithm. The differences between the performance characteristics of the algorithms may not be as pronounced for the replaced

modulus RSA or multi-exponent RSA algorithms, as the values are close to 1.00.

6.2 Conclusion

6.2.1 Research Findings

The research employed a mixed method of approach in the systematic study and experimental evaluation of the traditional RSA algorithm and optimization algorithms in Limitation Exploration and Optimization Verification of RSA Encryption: Using Digital Signature as an Example. The researcher found that the algorithms performed differently in all the stages of the digital signature processing. The researcher obtained 200 data points from ten measurements of each algorithm.

The results obtained from the research indicate that there is a divergence between theory and practice. In the research, it is theoretically proven that the multi-prime RSA algorithm has the potential for acceleration in the decryption and signing process through the Chinese Remainder Theorem. However, the algorithm performed poorly in all the stages of the digital signature processing compared to the traditional RSA algorithm with a speedup of 5.39 times slower in the key generation with a high variability (SD up to 0.71), indicating that the algorithm is more complex than the benefits obtained from the theory in solving the mathematical problem. The replaced modulus RSA algorithm performed well in all the stages of the digital signature processing with a speedup ranging from 0.93 to 1.01, indicating that the theory is correct in improving the security without compromising the speed. The multi-exponent RSA algorithm performed with a slight speedup in the key generation (1.06 times) and encryption (1.02 times) with an excellent stability, indicating that the theory is correct in improving the flexibility without compromising the speed.

This study would fill the gap that was identified in the existing literature, as it would provide systematic and comparative empirical data on the RSA optimization schemes that have been tested and compared under a unified framework. It is evident from this study that testing and validating a theoretical model under a unified framework is a must.

6.2.2 Future Research Directions

Some future research directions were identified in this study, which need to be explored in more detail in the future. It is evident from this study that it would be beneficial to test and compare these algorithms using compiled languages such as C and Rust, in order to evaluate whether there is a significant difference in the performance characteristics of the algorithms. It may also be necessary to test and compare these algorithms using different hardware platforms, such as ARM processors, in order to increase the scope and applicability of this study. It may also be necessary to combine security validation with performance measurements in order to obtain a comprehensive evaluation of the efficacy of the algorithm for use in a real-world application such as digital signatures.

References

Bai, Y. and He, L. and Chen, Y. (2021), 'The Design and Implementation of RSA Cryptographic Algorithm Based on Python', *Electronic Design Engineering* (Accessed: 18 March 2026)

CeLaMbDa (2025) ' An In-Depth Understanding of RSA Encryption and Decryption Techniques ', CSDN, 25 December. Available at: https://blog.csdn.net/weixin_34438187/article/details/150144405
(Accessed: 18 March 2026)

Ding, Z. (2013) *Research and analysis of the RSA algorithm*. Available at: https://cdmd.cnki.com.cn/Article/CDMD-90008-1011057301.htm (Accessed: 17 March 2026).

Gao, L. (2018) *Research on digital signature technology based on the RSA public key algorithm*. Available at: https://www.zhaokaiti.com/searchdetail?dk=bylw&pid=16680 (Accessed: 17 March 2026).

Gao, L. and Zhou, J. (2018) 'Research on digital signature based on multiple keys and CRT', *Internet of Things Technologies*, 8(4), pp. 39–41.

Gao, L. and Zhou, J. (2019) 'Research on the improved RSA algorithm based on multiple prime numbers and parameter substitution', *Application Research of Computers*, 36(2), pp. 495–498. doi: 10.19734/j.issn.1001-3695.2017.09.0879.

Hu, F. (2008) 'Improved RSA Algorithm and Its Application in Digital Signatures', *Northeast University* (Accessed: 18 March 2026)

Li, J. (2020) *Optimization and improvement of RSA encryption algorithm*. Available at: https://www.cnki.com.cn/Article/CJFDTotal-HLKX202117041.htm (Accessed: 17 March 2026).

Li, Q. and Fang, J. (2025) 'A Secure Sharing Method for Computer Network Data Based on Improved RSA Algorithm', *Industrial Control Computer* (Accessed: 18 March 2026)

OpenHiTLS Open-Source Community(2025) 'An In-Depth Analysis of Asymmetric Cryptographic Algorithm Analysis Techniques and Future Prospects ', CSDN, 25 December. Available at: https://openhitls.blog.csdn.net/article/details/153677851 (Accessed: 18 March 2026)

Song, S. (2007) *The application of RSA algorithm in digital signatures*. Available at: https://www.cnki.com.cn/Article/CJFDTotal-WLAQ202205023.htm (Accessed: 17 March 2026).

Wei, Y. and Liu, X. (2014) 'Research on digital signature system based on RSA', *Journal of North China Institute of Aerospace

ISSN 2959-6157

Engineering*, 24(5), pp. 20–22.
Zhang, W. and Feng, M. and Li, Q. and Jiang, B. (2020) 'Research on RSA Encryption Algorithm Based on python and Several Cracking Methods', *Information System Engineering*(Accessed: 18 March 2026)
Zhang, X. (2017) 'Implementation and Research of Encryption and Decryption Algorithms Based on python', *Textile Report* (Accessed: 18 March 2026)
Zhang, Y. (2015) *Research on RSA algorithm and SM2 algorithm*. Available at: https://pdf.hanspub.org/csa20211100000_36474466.pdf (Accessed: 17 March 2026).

Appendix

Table for all data points.

Run	Algorithm	Key generation	Encryption	Decryption	Signature	Check
1st	Basic	1	1	1	1	1
1st	Multi-prime	6.79	1.76	2.72	2.48	1.91
1st	replaced-modulus	1.21	1.03	1.13	1.11	1.11
1st	multi-exponent	0.97	0.92	0.94	0.90	0.91
2nd	Basic	1	1	1	1	1
2nd	multi-prime	5.77	1.71	2.35	2.09	1.90
2nd	replaced-modulus	0.87	0.97	0.95	0.99	1.06
2nd	multi-exponent	0.85	0.98	0.95	0.95	1.02
3rd	Basic	1	1	1	1	1
3rd	multi-prime	4.89	1.13	1.87	1.79	1.66
3rd	replaced-modulus	1.09	0.81	0.97	0.97	1.05
3rd	multi-exponent	1.11	1.03	1.01	1.04	1.10
4th	Basic	1	1	1	1	1
4th	multi-prime	5.36	1.54	1.84	1.81	1.57
4th	replaced-modulus	1.37	1.05	0.95	0.95	1.01
4th	multi-exponent	1.02	1.05	1.00	0.99	1.10
5th	Basic	1	1	1	1	1
5th	multi-prime	4.24	1.45	1.89	1.88	1.56
5th	replaced-modulus	0.98	1.00	1.06	1.04	1.05
5th	multi-exponent	0.84	1.00	1.02	0.99	1.06
6th	Basic	1	1	1	1	1
6th	multi-prime	4.79	1.45	1.84	1.81	1.44
6th	replaced-modulus	1.17	1.04	0.97	1.02	1.01
6th	multi-exponent	0.93	0.99	0.97	1.01	1.02
7th	Basic	1	1	1	1	1
7th	multi-prime	5.63	1.74	2.33	2.27	1.87
7th	replaced-modulus	1.07	0.98	0.98	0.99	1.03
7th	multi-exponent	0.90	0.99	1.00	1.02	1.06
8th	Basic	1	1	1	1	1
8th	multi-prime	5.18	1.66	2.22	2.09	1.74
8th	replaced-modulus	1.03	0.98	0.97	0.97	0.96
8th	multi-exponent	0.99	0.98	0.97	0.98	0.99
9th	Basic	1	1	1	1	1
9th	multi-prime	6.08	1.59	2.29	2.21	1.75
9th	replaced-modulus	0.96	1.06	1.09	1.10	1.07
9th	multi-exponent	0.86	0.89	1.05	1.07	1.02

Run	Algorithm	Key generation	Encryption	Decryption	Signature	Check
10th	Basic	1	1	1	1	1
10th	multi-prime	5.55	1.55	1.88	1.91	1.60
10th	replaced-modulus	1.01	0.95	0.95	0.95	0.97
10th	multi-exponent	0.97	0.97	0.94	0.95	0.99

Reviewer comments:

1. Please provide the author's affiliation (school), region (postal code), country, email address, and corresponding author.
2. Supplement page numbers.