Comparative Assessment of Attack Strategies and Their Efficacy on the RSA Encryption Algorithm

Zhengli Hao

School of mathematics and physics, University of Science and Technology Beijing, Beijing, 100083, China

ndavis81569@student.napavalley.edu

Abstract. The RSA encryption system, standing as the preeminent public key cryptographic mechanism, has managed to sustain its robust security profile despite the manifold attack algorithms that have emerged over time. In fact, to this day, no attack strategy has genuinely undermined or imperiled the inherent security of the RSA algorithm. This manuscript embarks on a meticulous exploration of the contemporary, predominant RSA attack algorithms. It elucidates the foundational principles underpinning each algorithm, offering readers an in-depth understanding. Furthermore, to lend a practical dimension to this theoretical exposition, simulated attack experiments are designed. These are tailored to suit specific environments, thereby facilitating a robust comparative analysis. Drawing upon empirical evidence harvested from these experimentations, this document offers critical appraisals of each algorithm. This assessment juxtaposes their theoretical prowess against tangible deployability, illuminating their strengths and potential vulnerabilities. A pivotal aspect underscored in this treatise is the necessity of judiciously architecting RSA algorithms. Only by acknowledging and circumventing potential pitfalls can one truly ensure the algorithm's resilience. The insights and evaluations furnished herein aim not only to serve as a beacon for those navigating the vast sea of cryptographic research but also to inspire and guide future endeavors aimed at developing potent attack algorithms against the RSA framework.

Keywords: RSA, Attack algorithm, Public Key Cryptography, Boneh-Durfee, Wiener.

1. Introduction

The RSA algorithm, distinguished as the most prevalent asymmetric encryption technique, falls under the umbrella of public key encryption systems. Since its inception in the 1970s, the algorithm has been the focal point of immense academic scrutiny and rigorous research, leading to the emergence of numerous attack strategies against it [1]. Commencing with integer decomposition algorithm assaults, transitioning to those targeting private keys and smaller-scale public keys, and culminating in encrypted ciphertext attacks, a plethora of attack methodologies have sequentially surfaced. This continuous evolution has simultaneously propelled the RSA algorithm towards heightened security and complexity.

Remarkably, despite the diverse array of attack modes now at play, none has yet managed to pose a genuine, substantive threat to the RSA algorithm. Its security integrity remains overwhelmingly acknowledged within academic circles, and its wide-ranging applications span sectors from manufacturing to finance, and even the military [2]. Consequently, dissecting the array of attack algorithms aimed at RSA not only advances the realm of public key cryptosystems and cryptography but also potentially offers a strategic edge in any contest centered around security. This manuscript embarks on a journey tracing the chronological progression of attack algorithms against the RSA algorithm, delving deep into the underpinning principles and unique attributes of each. A comparative analysis will be furnished, shedding light on the merits and shortcomings of each strategy, as well as their optimal operational environments. The insights presented here aim to pave the way for future research trajectories in this vital field.

2. Principle Overview

The RSA algorithm is a public-key cryptosystem proposed in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman, RSA is a combination of the initials of the three founders [3]. The algorithm is a cryptosystem that uses different encryption and decryption keys, based on the theory that it is computationally infeasible to derive a decryption key from a known encryption key.

The principles of the RSA public key cryptosystem are: Arbitrarily choose two different large prime numbers p and q, calculate the product $n = pq$,such that $\varphi(n) = (p-1)(q-1)$;

Arbitrarily choose a large integer e as the key, which needs to satisfy $\gcd(e, \varphi(n)) = 1$; A determined solution key d that satisfies $de \bmod \varphi(n) = 1$, $de = k\varphi(n) + 1$, where $k \geq 1$ and is an arbitrary integer; Make public the integers n and e, and keep d in secret; Encrypt the plaintext m (which is an integer) into a ciphertext c with the encryption algorithm [4].

$$c = E(m) = m^e \bmod n \tag{1}$$

Decrypt the ciphertext c into plaintext m, the decryption algorithm is

$$m = D(c) = c^d \bmod n \tag{2}$$

In theory, it is extremely difficult to compute d based only on n and e. Thus, anyone can encrypt the plaintext, but only the person who has the key can decrypt the ciphertext.

3. Formatting the text

According to the principle of the RSA algorithm, since there is no effective and universal RSA attack algorithm yet, basically all the mainstream attack algorithms of various types are only applicable to specific cases, which means in the construction of the RSA algorithm itself there is a risk of vulnerability. In the absence of private key leakage and misplaced error construction algorithms (e.g., the public key n is a number of powers of a large prime number p, n can be factored into 3 or more prime numbers, etc.), according to these specific cases and the corresponding principles of attack algorithms, RSA attack algorithms can be roughly categorized into the following four categories: the integer factorization attack on the public key n, the attack on the low-bit public key e, the attack on the low-bit private key d and the attack on the ciphertext c.

Table 1. Introduction to Partial Integer Factorization Algorithms.

Name	Year	Time Complexity	Digital Scale
TD	-	$O(\sqrt{N})$	Low
p-1	1974	$O(B \times \log B \times \log^2 N)$	Low
Rho	1975	$O(N^{\frac{1}{4}} \log^2 N)$	Low
ECM	1987	$O(\exp((\sqrt{2} + O(1))\sqrt{\log p}\sqrt{\log\log p}))$	Middle
QS	1991	$O(\exp((1 + O(1))\sqrt{\log N}\sqrt{\log\log N})$	Middle
NFS	1993	$O(\exp(\sqrt[3]{\left(\frac{64}{9} + O(1)\right)}\sqrt[3]{\log N}\sqrt[3]{\log\log N^2})$	High

3.1. Attack on Public Key n

This attack is the most direct attack on the RSA public key encryption system, which can realize the complete cracking of this public key encryption system [5]. The idea of the algorithm is to use a series of integer factorization algorithms to factor the large integer n, to find out the composition of the two large prime factors p and q, then solve the Euler function $\varphi(n)$ and the private key d. At present, integer factorization has formed a number of mature factorization algorithms, including trial division method (TD), p-1 method (P-1), Pollard Rho method (Rho), Brent method (Brent), elliptic

curve decomposition method (ECM), quadratic screening method (QS), numerical field screening method (NFS) and so on [6]. The ECM, QS and NFS are the most common integer factorization algorithms, which can cope with most of the medium and high scale integer factorization problems. ECM and QS are suitable for integer factorization above 64 bits, while when n has more than 200 bits, NFS is the most efficient factorization algorithm. Tables 1 and 2 present data and characteristics for a comparative analysis of each algorithm.

The efficiency of this attack pattern is often equivalent to the efficiency of the new algorithm for large integer factorization, so it is important that the digits of the prime numbers p and q are similar and high enough when both selecting [7].

Table 2. Comparison of Partial Integer Factorization Algorithm.

Name	Advantage	Disadvantage
TD	Simple, good parallelism, good for randomly selecting or integers with small prime factors	Small scope of use and high time complexity
p-1	Suitable for integers with small factors	Poor generalization, weakly practical
Rho	Suitable for integers with small factors	Poor generalization
ECM	Simple structure, small memory footprint, fastest running among smoothness-based algorithms	Average generalization
QS	Well parallelism and generality	Very high memory requirements
NFS	Current optimal algorithms for integer factorization, high generality parallelism and practicality	Complex algorithms, high memory requirements, difficult to implement

3.2. Attack on Low-bit Public Key e

As mentioned earlier, because the RSA algorithm has the problem of long encryption and decryption time, in order to ensure the efficiency of information transmission, the number of bits will be appropriately reduced when selecting the public key e or the private key d. The public key e will be used to realize fast encryption, while the private key d will be used for decryption. The general RSA algorithm encryption system will select a smaller public key e to achieve fast encryption, part of the RSA algorithm for digital signature will select a smaller private key d for decryption [8]. Obviously, this change brings risks to the security of cryptosystems, in which there are two forms of exponential attack and broadcast attack for rather smaller public keys e ($e = 3$ for example). Avoiding such attacks is as simple as not choosing the public key that is too small. For the general RSA public key cryptography system, we usually choose the Fermat number F_m when $m = 4$, exactly $e = 65537$, as the public key.

3.2.1. Exponential Attack.

Take $e = 3$ for example, e is a very small integer while n is very large, according to the encryption formula $c = m^e \bmod n$, if $m^e < n$, it can directly calculate the ciphertext m by computing $\sqrt[e]{c}$. On the other hand, if the ciphertext m satisfying $m^e > n$, it could be obtained by exhaustively enumerating k to find the integer result of $\sqrt[e]{c - kn}$.

Despite the simplicity of the algorithm, exponential attack requires a large number of exhaustive and open-square calculation, which makes it extremely inefficient and not commonly used as a primary attack method against small public key encryption [9].

3.2.2. Broadcast Attack.

Compared to exponential attack, broadcast attacks is more common and efficient method. The attackers need to collect several plaintexts c and public keys n with the same encryption key e. Then the attack is transformed into a mathematical problem of solving a one-dimensional linear congruence equation. Assuming the ciphertext is m, which is encrypted by different users as c_1, c_2,

... , the system of simultaneous equations can be solved to obtain the ciphertext according to the Chinese remainder theorem as follow:

$$\begin{cases} c_1 = m^e \bmod n_1 \\ c_2 = m^e \bmod n_2 \\ \quad\cdots \\ c_j = m^e \bmod n_j \end{cases} \quad (j < 1, j \in N_+) \tag{3}$$

Let

$$N = \prod_{i=1}^{j} n_i, \;\; N_i = \frac{N}{n_i}, \;\; t_i = N_i^{-1} \bmod n_i \qquad (i \le j, \;\; i, j \in N_+) \tag{4}$$

where t_i is the inverse of n_i. Then the general solution of the equation is,

$$m = \sum_{i=1}^{j} c_i t_i N_i \bmod n_i \qquad (i \le j, \;\; i, j \in N_+) \tag{5}$$

According to China's Residual Theorem, the necessary condition for the above system of equations to have a unique solution is $j \le e$, and only then the ciphertext can be solved [10]. Therefore, in order to realize this attack, the attacker must need to collect the ciphertext and public key information of more than j users, and this situation will instead force the user to avoid using too small e.

3.2.3. Attack on Low-bit Private Key d.

The above section has shown that choosing a too small encryption key e is insecure, and similarly, choosing a too small decryption key d is risky too. The attack against a low-digit private key d was initially introduced in the 1990s, an American mathematician named Wiener used the method of concatenated fractions to prove that a private key d can be recovered in polynomial time when it satisfies $d < N^{0.25}$ and $p < q < 2p$. In 1999, Boneh and Durfee improved on Wiener's result by raising the upper bound of the small-exponent attack to $N^{0.292}$.

As in the previous section, such attacks can be solved by picking a decryption key d that avoids being too small.

3.2.4. Wiener Attack.

The Wiener attack works with the mathematical concept of continued fractions, that is for any positive integer ξ, it can be expressed as

$$\xi = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ddots + \cfrac{1}{a_K}}}$$

Figure 1. continued fraction

with $a_0 \ge 0, \; a_1, \; a_2, \;$ ... are integers. According to the principle of RSA algorithm, it has

$$\varphi(n) = (p-1)(q-1) = pq - (p+q) + 1 = n - (p+q) + 1 \tag{6}$$

Since the prime integers p and q are very large, so $pq \gg p+q$, which means $\varphi(n) \approx n$. Then due to $de \bmod \varphi(n) = 1$, it can easily know that:

$$de - 1 = k\varphi(n) \;\; (k \in N_+) \tag{7}$$

Dividing both sides of this equation identically $d\varphi(n)$, it can be obtained

$$\frac{e}{n} = \frac{k}{d}(1-\delta), \quad \delta = \frac{p+q-1-\frac{1}{k}}{n} \tag{8}$$

Since $\varphi(n) \approx n$, so $\frac{k}{d} \approx \frac{e}{n}$, then d can be obtained from the continuous fraction expansion of $\frac{e}{n}$.

3.2.5. B-D Attack.

Boneh and Durfee have proved that most of the time it can successfully factor n if the private key d is less than $N^{0.292}$ in 1999.

Since

$$\begin{aligned} & e \cdot d = 1 \bmod \varphi(n) \\ & \Rightarrow e \cdot d = k \cdot \varphi(n) + 1 \\ & \Rightarrow k \cdot \varphi(n) + 1 = 0 \bmod e \\ & \Rightarrow k \cdot (N + 1 - p - q) + 1 = 0 \bmod e \end{aligned} \tag{9}$$

Here the unknowns are k and $(-p-q)$.Then it can write that problem as a polynomial with root x_0 and y_0:

$$f(x, y) = x \cdot (A + y) \tag{10}$$

such that $f(x_0, y_0) = 0 \bmod e$ with $A = N + 1$ and $y = -p - q$. Then whole algorithm as the figure below:

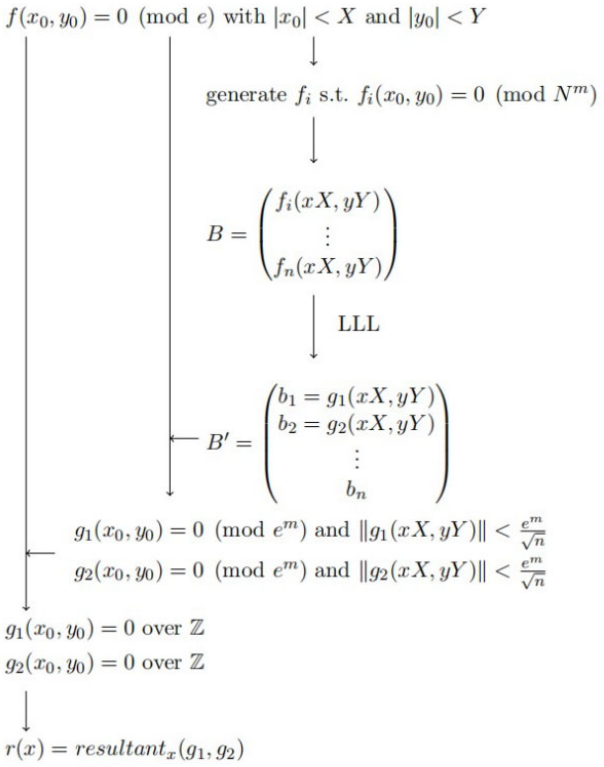

Figure 2. Boneh-Durfee algorithm (Photo/Picture credit: Original).

Boneh and Durfee proposed a construction of the f_i polynomials as follow:
for

$$k = 0, \ \ldots, m \begin{matrix} g_{i,k}(x) = x^i \cdot f^k(x,y) \cdot e^{m-k} \\ h_{j,k}(x) = y^i \cdot f^k(x,y) \cdot e^{m-k} \end{matrix} \qquad \text{for } i = 0, \ldots, m, \ j == 0, \ldots, t \tag{11}$$

They called the $g_{i,k}$ as x-shifts and the $h_{j,k}$ as y-shifts. By using these polynomials to build the lattice, carefully balancing the variables so that the determinant of the triangular basis doesn't exceed e^{mn} , Boneh and Durfee showed that LLL successfully yielded useful results if $d < N^{0.284}$. And finally to achieve their improved results of $d < N^{0.292}$, they showed that using a sublattice by ignoring some of the y-shifts, the bounds on the shortest vectors found by LLL were improved.

3.3. Attack on Ciphertext

Unlike the last three types of algorithms, the attack on ciphertexts avoids public and private keys, and instead obtains the plaintext directly without breaking the cryptosystem. Compared with cracking the public and private keys of RSA, such algorithms have higher operational efficiency, but require the attackers to disguise as a user and integrate into the cryptosystem to obtain the information, which may be even more difficult in real conditions. These attack methods include and select ciphertext attack and common-mode attack.

3.3.1. Select Ciphertext Attack.

Since RSA plaintexts are disseminated through public channels, it is very possible for an attacker to obtain the ciphertext. Supposing an attacker has intercepted a ciphertext c if he needs to obtain the plaintext $m = c^d \bmod n$. To recover m, he finds a random number $r\ (r < n)$ and the recipient's public key (e, n), encrypts r as

$$x = r^e \bmod n \tag{12}$$

Then multiply the temporary ciphertext x with c as

$$y = x \cdot c \bmod n \tag{13}$$

and find

$$t = r^{-1} \bmod n \tag{14}$$

Since $x = r^e \bmod n$ if $x = r^e \bmod n$. Therefore, it is sufficient to have the recipient sign y with the recipient's own private key and send the result

$$u = y^d \bmod n \tag{15}$$

back to the attacker. The attacker can simply compute:

$$m = u \cdot t \bmod n \tag{16}$$

to restore the plaintext. The usual way to prevent this attack is to not decrypt or sign random messages from the unknown origin.

3.3.2. Common-mode Attack.

When using the RSA algorithm for encryption, if the modulus n in the public keys used by different users are the same, the algorithmic system has the possibility of common-mode attack, which can be cracked without factoring n. The specific conditions for the realization of the common-mode attack are that two different users use the public key (n, e_1) and (n, e_2) to encrypt the same message m to obtain different ciphertexts (c_1, c_2) and $\gcd(, e_2) = 1$. At this time, according to the nature of the greatest common divisor, there must be two integers a and b so that $ae_1 + be_2 = 1$ is satisfied, then the plaintext can be computed as:

$$m = (c_1^a \bmod n) \cdot \left(c_2^b \bmod n\right) \bmod n \tag{17}$$

A general way to prevent this attack is to avoid the same large integer n as much as possible in the same cryptosystem.

4. Experimentation and Analysis

Since each algorithm has different application conditions and needs to satisfy its own conditions before the attack experiment can be carried out, a horizontal comparison of the performance of all the algorithms do not have such reference value and practical significance, so this experiment focuses on comparing the performance of algorithms of the same type, and comparing their performance with the change of system complexity (e.g., the change of the values of the public keys n and e or the private key d increases).

At the same time, since the security of the RSA public key cryptosystem can be approximately equated to the difficulty of the large integer factorization problem, the integer factorization attack in the above attack algorithms is undoubtedly the slowest, and its algorithmic performance depends on the integer factorization algorithm employed, so it is not considered as the object of the experiments. In addition, because the attack algorithms against ciphertexts can be compatible with the case of low-bit private key and low-bit public key, this experiment is divided into three: the low-bit public key experiment, the low-bit private key experiment and the ciphertext experiment. Experiment with each algorithm 5 times per case.

4.1. Experiments results

In order to explore the performance of each of the above algorithms, all the algorithms have been implemented based on program implementation in python programming environment using A 64-bit Windows 10 operating system PC with an Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz and 64GB of RAM was used for testing. The experimentally encrypted test plaintext is a combination of several numbers, letters and special symbols, which is "$114514qwer\alpha \in, \pm$".

4.1.1. Low-bit Public Key Experiment.

This experiment examines the exponential attack, broadcast attack, setting up several public keys and the number of bits of n is 512, and here are the results of the experiment. As shown in Table 3.

Table 3. Low-bit Public Key Experiment Results.

Private key e / Name	3	11	23	61
Exponential	0.97	229.79	365421.1	-
Broadcast	1.00	2.01	10.17	78.88

Average running times (ms).

4.1.2. Ciphertext Experiment.

This experiment examines the Wiener attack and B-D attack, setting up the private key $d \approx n^{0.25}$ or in the same order of bits, and here are the results of the experiment. As shown in Table 4.

Table 4. Low-bit Private Key Experiment Results.

Size of n (bits) / Name	256	512	1024	2048
Wiener	0.99	5.98	40.22	181.74
B-D	213.43	321.93	693.96	1679.40

Average running times (ms).

4.1.3. Ciphertext Experiment.

This experiment examines the select ciphertext attack and the common-mode attack, setting up the public key $e = 65537$ or in the same order of bits (for Common-mode attack, $e_1 = 65537$, $e_2 = 49999$), and here are the results of the experiment. As shown in Table 5.

Table 5. Ciphertext Experiment Results.

Size of n (bits) / Name	256	512	1024	2048
Select ciphertext	1.99	15.97	86.62	537.33
Common-mode	2.01	3.99	15.62	28.37

Average running times (ms).

4.2. Experiments results

Based on the algorithmic principles and the results of their respective algorithmic runs in the same programming environment, we can comment on the above algorithms.

The first is the attack against the low-bit encryption key e. It can be found that in the case that the encryption key is small enough, both the exponential attack and the broadcast attack have the same level of excellent performance, but when the value of e is out of a certain range, the effect of the exponential attack will be rapidly reduced, and eventually even in the case that the decimal size of e is in the two-digit range, it is not possible to arrive at the result in a certain amount of time. The reason for this is that the algorithm needs to exhaustively search for k that can enter the satisfy to perform the square operation when $m^e > n$ is satisfied, and this efficiency is undoubtedly extremely low. Similarly, if the encoding value of the plaintext m is too large, it may also make the exponential attack eventually transformed into agonistic attack, so this method is eliminated in reality. While broadcast attack has excellent attack effect, but also has the limitation of needing to collect extra e users' encrypting the same message using the same public key. So as of now, although smaller encryption keys are risky, there are still a lot of difficulties in actual deciphering, which is why some encryption regimes dare to use some very small public keys to encrypt information.

Next is the attack against the low-digit public key d. It can be seen that both algorithms perform well when the decryption conditions are satisfied. Although from the experimental results when $d \approx n^{0.25}$, wiener attack is better than B-D attack, but B-D attack has a larger attack range $d < n^{0.292}$ which is due to the fact that the algorithm itself is more complex than the wiener attack, and with the increase of the number of integer n digits, the B-D attack running time increases insignificantly, its algorithmic stability is better than the wiener' attack, the two algorithms are evenly matched.

Finally, the attack algorithm bypassing the public key and private key of this attack ciphertext attack algorithm, common mode attack in the algorithmic efficiency and stability are better than the choice of ciphertext attack, but its implementation in reality is significantly more difficult than the latter. Common mode attack requires the use of the same modulus n of two different encryption key user encrypted the same information, and in the construction of the RSA encryption algorithm in reality, due to the infinite number of prime numbers and irregular, so it often can avoid the production of common mode in the first step of selecting different large prime numbers p and q without consider encrypting the same information later. The choice of ciphertext attack is obviously easier to realize, the attacker only needs to obtain the ciphertext and public key, disguised as a user to send some meaningless information to request encryption can pose a threat, so for this type of attack, to avoid information leakage and deal with information from unknown sources is particularly critical.

To summarize, wiener attack and B-D attack are both excellent attack algorithms, and there exists a certain realization possibility of choosing the ciphertext attack, while the other attack modes have greater limitations in the real environment and algorithm realization, so as far as the current RSA public key cryptosystem is concerned, it is the most crucial to ensure the high digit number of the private key d and the difference of the public key and the difference of the public key when it is used by different users.

5. Conclusion

RSA stands as the paramount public key encryption system, holding the distinction of being the most prevalently employed across various domains. With its widespread application, a myriad of attack methodologies tailored for the RSA have consequently emerged. This treatise delineates the contemporary, leading-edge RSA assault algorithms, delving into their foundational principles. We have designed intricate simulation experiments to facilitate side-by-side comparisons of these algorithms in real-world scenarios. Drawing upon the empirical evidence harvested from our experimentations, we critically appraise each algorithm, juxtaposing their theoretical constructs

against practical deployability. A salient inference we've distilled is the indispensable role played by the bit-length of the private key in bolstering RSA's security. Equally pivotal is the caution against the repetitious employment of the public key. Such lapses can inadvertently render the system susceptible to breaches.

References

[1] Overmars A, Venkatraman S. Mathematical attack of RSA by extending the sum of squares of primes to factorize a semi-prime[J]. Mathematical and Computational Applications, 2020, 25(4): 63.

[2] May A, Nowakowski J, Sarkar S. Partial key exposure attack on short secret exponent CRT-RSA[C]//International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer International Publishing, 2021: 99-129.

[3] Sarna S, Czerwinski R. Small Prime Divisors Attack and Countermeasure against the RSA-OTP Algorithm[J]. Electronics, 2021, 11(1): 95.

[4] Boneh D. Twenty years of attacks on the RSA cryptosystem[J]. Notices of the AMS, 1999, 46(2): 203-213.

[5] Κουνής Κ Ε. Boneh-Durfee attack in RSA[D]. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, 2021.

[6] Harjito B, Tyas H N, Suryani E, et al. Comparative Analysis of RSA and NTRU Algorithms and Implementation in the Cloud[J]. International Journal of Advanced Computer Science and Applications, 2022, 13(3).

[7] Boneh D, Durfee G. Cryptanalysis of RSA with private key d less than N/sup 0.292[J]. IEEE transactions on Information Theory, 2000, 46(4): 1339-1349.

[8] Imam R, Areeb Q M, Alturki A, et al. Systematic and critical review of rsa based public key cryptographic schemes: Past and present status[J]. IEEE Access, 2021, 9: 155949-155976.

[9] Anwar M N B, Hasan M, Hasan M M, et al. Comparative study of cryptography algorithms and its' applications[J]. International Journal of Computer Networks and Communications Security, 2019, 7(5): 96-103.

[10] Raghunandan K R, Aithal G, Shetty S. Comparative analysis of encryption and decryption techniques using mersenne prime numbers and phony modulus to avoid factorization attack of RSA[C]//2019 International Conference on Advanced Mechatronic Systems (ICAMechS). IEEE, 2019: 152-157.