Menemui Matematik
(Discovering Mathematics)

journal homepage: https://persama.org.my/dismath/home

Fast Decryption Methods For The Somsuk-RSA Cryptosystem

Nur Adira Mohamad Azlan[1], Aniza Abd Ghani[2], Faridah Yunos[3] and Muhammad Asyraf Asbullah[4,*]

[1,2,3]*Department of Mathematics and Statistics, Universiti Putra Malaysia, 43400 UPM Serdang, Malaysia*
[4]*Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Malaysia*
[4]*Centre for Foundation Studies in Science of Universiti Putra Malaysia, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia*
[4]*Malaysia Cryptology Technology and Management Centre, Universiti Putra Malaysia, 43400, Serdang, Selangor, Malaysia*

ma_asyraf@upm.edu.my
*Corresponding author

Received:4 March 2025
Accepted:3 September 2025

ABSTRACT

In the digital era, ensuring the security of data transmission and storage remains a critical concern. The RSA cryptosystem plays a pivotal role in safeguarding information through asymmetric encryption. While the Somsuk-RSA variant enhances security, it introduces computational challenges, particularly during decryption. This study proposes innovative methods to accelerate decryption by substituting the Euler function with the more efficient Carmichael function, outperforming the original RSA and Somsuk-RSA systems. The analysis highlights the inherent slowdown in the Somsuk-RSA decryption process, emphasizing the need for optimization. The study presents effective strategies for improvement. A comparative evaluation with prime key sizes of 512, 1024, and 2048 bits demonstrates that the enhanced Somsuk-RSA cryptosystem achieves significantly faster decryption times than the original Somsuk-RSA approach.

Keywords: Somsuk-RSA, cryptography, decryption, Euler function, Carmichael function

INTRODUCTION

A public-key cryptosystem known as RSA (Rivest-Shamir-Adleman) was first published in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. A user of RSA generates and disseminates a public key based on two significant prime integers and an extra value. The prime numbers are not disclosed. With the public key, anyone can encrypt communications, but only someone who knows the prime numbers can decrypt them. (Al Hasib and Haque, 2008).

After RSA was proposed, numerous researchers attempted to improve its algorithm. Researchers frequently enhance the RSA algorithm using various mathematical techniques without altering the program's fundamental core. As a result, we will examine the RSA variant algorithms that have been put out in the past to enhance the RSA cryptosystem.

To overcome the shortcomings of the original RSA cryptosystem and boost its efficiency and security, researchers have put forth several improvements and modifications over time. This literature review aims to give a general overview of significant RSA cryptosystem variants, emphasizing their special traits, benefits, and possible concerns.

Islam et al. (2018) propose a Modified RSA (MRSA) proposal that aims to reduce the main weaknesses of the RSA system. The main worry in most cases is that it is easily breakable because keys can be calculated easily based on N . Since the RSA modulus N is the only product of two prime integers, it can be easily tracked. So, instead of using two huge primes, MRSA utilized four. The key generation time for MRSA is longer since it depends on a large factor value N . Additionally, compared to the RSA approach, encryption and decryption take more time.

Raghunandan et al. (2019) propose an enhanced RSA approach to overcome the integer factorisation attack's drawback by making the factorisation process more complex. They do this by utilizing a phony/fake public key exponent f instead of e and a phony modulus instead of N . By reducing the time required for encryption and decoding, this technique will offer more security than RSA.

The factoring challenge put forth by Ismail et al. (2018) served as the foundation for a novel, straightforward ESF-RSA public key cryptosystem. The secret key generation process for the newly suggested cryptosystem does not call for an inverse modular operation. Additionally, the size and value of the secret key are decreased if the public key and modulus in ESF-RSA are fixed. Therefore, compared to RSA, ESF-RSA encryption and decryption procedures are more effective.

Although RSA is a strong encryption technique, a factorization attack can be used to break it. Puneeth et al. (2022) enhanced the RSA algorithm, emphasizing the security feature of RSA's resilience to factorization attacks, is presented. The common modulus N is replaced with a third variable that the algorithm offers and serves as the network's public key.

Shah et al. (2023) have established a groundbreaking generalized RSA cryptosystem based on $2N$ prime numbers. This innovative approach significantly enhances security in the digital landscape by employing $2N$ distinct primes for factoring the variable N . This method not only facilitates a higher encryption exponent derived from the massive product of these primes but also fortifies overall security. In contrast to the traditional RSA algorithm, the time required for factoring dramatically increases when utilizing multiple prime integers and larger encryption exponents. The study clearly demonstrates that RSA and generalized RSA (GRSA) differ fundamentally in their security measures and operational speeds.

It is important to note that RSA is inherently slow in single-precision multiplication and actual running times. As a direct result, it is rarely employed for encrypting user data. Therefore, a multitude of researchers have proactively developed various variants of RSA-based cryptosystems to enhance the algorithm without compromising security.

Therefore, we want to overcome this deficiency. To increase the effectiveness of the Somsuk-RSA cryptosystem, we suggest a unique decryption technique that switches the Euler function for the Carmicheal function. The performance of this new decryption technique will then be compared to that of the original RSA cryptosystem and Somsuk-RSA cryptosystem.

PRELIMINARIES

2.1 Fundamental Concept of Number Theory

Cryptography heavily relies on number theory, especially in the design and analysis of cryptographic algorithms. Below are some fundamental concepts from number theory relevant to this study.

Definition 2.1 (Euler Function) *The notion of this function is $\phi(N)$ and defined as follows, where $p_0 \ldots p_k$ are the prime factors of N. Given*

$$N = p_0^{e_0} \cdot p_1^{e_1} \cdot \ldots \cdot p_k^{e_k}$$
$$\phi(N) = (p_0 - 1)\, p_0^{e_0 - 1} \cdot (p_1 - 1)\, p_1^{e_1 - 1} \cdot \ldots \cdot (p_k - 1)\, p_k^{e_k - 1}$$

The totient function describes the number of values less than N, which are relatively prime to N. For RSA, we are only concerned with values of N which are the product of two primes, p and q, so $\phi(N)$ is always just $(p-1)(q-1)$.

Definition 2.2 (Carmichael Funtion) *The notion of this function is $\lambda(N)$ and defined the smallest positive integer m such that*

$$a^m \equiv 1 (\operatorname{mod} N)$$

for every a that is coprime to N. The Carmichael function is also known as the reduced totient function or the least universal exponent function. By the fundamental theorem of arithmetic, any $N > 1$ can be written in a unique way

$$N = p_1^{a_1} \quad N = p_2^{a_2} \quad \ldots \quad N = p_{\omega(N)}^{a_{\omega(N)}}$$

where $p_1 < p_2 < \ldots < p_\omega$ are primes and the $a_i > 0$. For this project, only use this formula:

$$\lambda(N) = lcm(p-1, q-1) = \frac{\phi(N)}{\gcd(p-1, q-1)}$$

2.2 Rivest-Shamir-Adleman (RSA) Cryptosystem

A public-key cryptosystem known as RSA (Rivest-Shamir-Adleman) was first published in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. A user of RSA generates and disseminates a public key based on two significant prime integers and an extra value. The prime numbers are not disclosed. Anyone can encrypt communications With the public key, but only someone who knows the prime numbers can decrypt them. (Al Hasib and Haque, 2008).

The three steps of the RSA algorithm are key generation, encryption, and decryption. These terms define the RSA cryptosystem.

```
Algorithm 1 RSA Key Generation Algorithm
Require: The size $k$ of the security parameter
Ensure: The public key $(e, N)$ and the private key $(d, N)$
    1. Choose two random and distinct prime $p$ and $q$ such that $2^k < p$, $q < 2^{k+1}$
    2. Compute $N = pq$ and $\phi(N) = (p-1)(q-1)$
    3. Choose $e$ such that $3 \le e < \phi(N)$ and $\gcd(e, \phi(N)) = 1$
    4. Compute $d$ such that $ed \equiv 1 (\bmod N)$
    5. Return the public key $(e, N)$ and the private key $(d, N)$
```

```
Algorithm 2 RSA Encryption Algorithm
Require: The plaintext $m$ and the public key $(e, N)$
Ensure: A ciphertext $c$
    1. Choose integer $m$ such that $0 \le m < N$
    2. Compute $c \equiv m^e (\bmod N)$
    3. Return the ciphertext $c$
```

```
Algorithm 3 RSA Decryption Algorithm
Require: Ciphertext $c$ and private key $(d, N)$
Ensure: A plaintext $m$
    1. Compute $m \equiv c^d (\bmod N)$
    2. Return the plaintext $m$
```

2.2.1 Proof of Correctness for RSA Decryption

Proposition 2.1 Rivest et al. (1978). *Let* $N = pq$ *and* $\phi(N) = (p-1)(q-1)$. *For every integer* m *such that* $\gcd(m, N) = 1$, *and* $c \equiv m^e (\bmod N)$. *Then* $m \equiv c^d (\bmod N)$.

Proposition 2.2 Rivest et al. (1978). *Let* (e, N) *and* (d, N) *be the public and private keys for the RSA cryptosystem, respectively. Suppose* $0 < m < N$ *such that* $\gcd(m, N) = 1$ *and* $c \equiv m^e (\bmod N)$. *Then* $m \equiv c^d (\bmod N)$.

2.3 Somsuk-RSA Cryptosystem

To shorten the computation time for the RSA decryption procedure, Somsuk (2017) introduces a new equation. The private key is typically produced with a larger value than the public key to prevent simple access by outside parties. However, the decryption process takes longer when the private key is huge. The new exponent shrinks to a little integer when a high private key is employed in the suggested method. The experimental results show that the suggested approach can quickly finish the RSA decryption procedure when the private key is large, especially close to the Euler value.

Here is Somsuk-RSA cryptosystem:

```
Algorithm 4 Somsuk-RSA Key Generation Algorithm
Require: The size $k$ of the security parameter
Ensure: The public key $(e,N)$ and the private key $(x,N)$
    1. Choose two random and distinct prime $p$ and $q$ such that $2^k < p$, $q < 2^{k+1}$
    2. Compute $N = pq$ and $\phi(N)=(p-1)(q-1)$
    3. Choose $d$ such that $2^{\frac{k}{2}} < d < 2^{\frac{k}{2}+1}$
    4. Compute $e$ such that $ed \equiv 1(\bmod\ \phi(N))$
    5. Obtain $x$ such that $x + d = \phi(N)$
    6. Return the public key $(e,N)$ and the private key $(x,N)$
```

```
Algorithm 5 Somsuk-RSA Encryption Algorithm
Require: The plaintext $m$ and the public key $(e,N)$
Ensure: A ciphertext $c$
    1. Choose integer $m$ such that $0 < m < N$
    2. Compute $c \equiv m^e \pmod N$
    3. Return the ciphertext $c$
```

```
Algorithm 6 Somsuk-RSA Decryption Algorithm
Require: The ciphertext $c$ and the public key $(e,N)$
Ensure: A plaintext $m$
    1. Compute $m \equiv (c^{-1})^x \pmod N$
    2. Return the plaintext $m$
```

THE PROPOSED FAST DECRYPTION METHOD FOR NEW RSA-SOMSUK CRYPTOSYSTEM

3.1 Methodology

It suggests that the tiny value of the private key should not be selected because it is simple for third parties to calculate using Wiener's attack. Consequently, it needs to be large to evade this assault. However, the private key has a big value and directly impacts the decryption process, which has a high computational cost.

As a result, the revised equation for RSA's decryption process in this study substitutes the Carmichael function for the Euler function. The comparative analysis will use mathematical software with 512-bit, 1024-bit, and 2048-bit prime sizes. The mathematical software is used to compare computing time using different methods, which are 512-bit, 1024-bit, and 2048-bit. For every n, the prime number being used is the same, and every three cryptosystems run about a hundred times to get the best average result time.

3.2 New Design Algorithm: New Somsuk-RSA Cryptosystem

Here is the new Somsuk-RSA cryptosystem:

```
Algorithm 7 New Somsuk-RSA Key Generation Algorithm
Require: The size $k$ of the security parameter
Ensure: The public key $(e,N)$ and the private key $(x,N)$
    1. Choose two random and distinct prime $p$ and $q$ such that $2^k < p$, $q < 2^{k+1}$
    2. Compute $N = pq$ and $\lambda(N) = \text{lcm}(p-1)(q-1)$
    3. Choose $d$ such that $2^{\frac{k}{2}} < d < 2^{\frac{k}{2}+1}$
    4. Compute $e$ such that $ed \equiv 1(\text{mod } \lambda(N))$
    5. Obtain $x$ such that $x + d = \lambda(N)$
    6. Return the public key $(e,N)$ and the private key $(x,N)$
```

```
Algorithm 8 New Somsuk-RSA Encryption Algorithm
Require: The plaintext $m$ and the public key $(e,N)$
Ensure: A ciphertext $c$
    1. Choose integer $m$ such that $0 < m < N$
    2. Compute $c \equiv m^e (\text{mod } N)$
    3. Return the ciphertext $c$
```

```
Algorithm 9 New Somsuk-RSA Decryption Algorithm
Require: The ciphertext $c$ and the public key $(e,N)$
Ensure: A plaintext $m$
    1. Compute $m \equiv (c^{-1})^x (\text{mod } N)$
    2. Return the plaintext $m$
```

3.3 Proof of Correctness

Computing $m \equiv (c^{-1})^x (\text{mod } N)$ using the Algorithm 9, m will be certainly recovered. The reason is as follows.

Proposition 3.1 *Suppose e, d, x and N as defined in Algorithm 7. Let $c \equiv m^e (\text{mod } N)$ be the ciphertext where $0 < m < N$. Then, the plaintext m can be recovered from Algorithm 9.*

Proof: Suppose e, d, x and N as defined in Algorithm 7. Let $\lambda(N)=\frac{\phi(N)}{\gcd(p-1,q-1)}$, $x=\lambda(N)-d$, and $ed\equiv 1(\bmod\,\lambda(N))$ and $c\equiv m^e(\bmod\,N)$ be the RSA parameter. Thus, we have

$$\begin{aligned}
\left(c^{-1}\right)^x &\equiv \left(m^e\right)^{-x} \\
&\equiv m^{-ex} \\
&\equiv m^{-e(\lambda(N)-d)} \\
&\equiv m^{-e\lambda(N)+ed} \\
&\equiv m^{-e\lambda(N)}\cdot m^{ed} \\
&\equiv m^{-e\lambda(N)}\cdot m^{1+k\lambda(N)} \\
&\equiv m^{-e\lambda(N)}\cdot m^{1}\cdot m^{k\lambda(N)} \\
&\equiv m^{-e\lambda(N)}\cdot m^{1}\cdot 1 \\
&\equiv m^{-e\lambda(N)}\cdot m^{1} \\
&\equiv 1\cdot m^{1} \\
&\equiv m(\bmod\,N)
\end{aligned}$$

From Definition 2.2, it follows that $m^{k\lambda(N)}=1$. Since $m<N$, then we have $m=\left(c^{-1}\right)^x(\bmod\,N)$.

Example 3.1 *Consider a scenario where two parties, a sender and a recipient, are involved in communication. The security parameter is set to* $n=62$.

Key Generation: Recipient generates two distinct primes $p=5270450229848425421$ and $q=7484594782929751261$.

1. Compute $N=pq=39447184294014433304172066596039205881$
2. Compute $\lambda(N)=\frac{\phi(N)}{\gcd(p-1,q-1)}$ and get $\lambda(N)=1972359214700721664570851079163051460$
3. Choose that $2^{\frac{n}{2}}<2^{\frac{n}{2}+1}$, which is $d=1972359214700721643946140955353248551$
4. Compute $e=\frac{1}{d}(\bmod\,\lambda(N))=798236715144239558053344617163202871$
5. Obtain x such that $x+d=\lambda(N)$, then x = 20624710123809802909

Encryption: The sender receives the recipient's public key. The sender wants to send a message m = 7908767739938060921.

1. Compute $c\equiv m^e(\bmod\,N)=9602523284462970954021074644467656554$

Decryption: The recipient receives a ciphertext $c = 960252328446297095402107464446765655 4$ from the sender. To decrypt c, the recipient then performs:

1. Compute $m \equiv \left(c^{-1}\right)^{x} \left(\operatorname{mod} N\right) = 7908767739938060921$

PERFORMANCE ANALYSIS

4.1 Comparative Analysis

This section thoroughly analyses the findings, focusing on the decryption performance of various RSA cryptosystem varieties in second. Each algorithm was executed 100 times through the code. The evaluation utilized prime sizes of 541, 1024, and 2048 bits for each cryptosystem. This analysis primarily clarifies the relevance of the found patterns and differences in decryption times across the three algorithms. There are RSA, Somsuk-RSA, and new Somsuk-RSA.

Table 1: Comparative of Key Generation

	Running time (sec)		
Bit length	RSA	Somsuk-RSA	New Somsuk-RSA
512	1.625	1.797	2.674
1024	2.020	2.496	1.832
2048	1.529	1.679	2.197

From Table 4.1, the Original RSA technique demonstrated the fastest key generation among the three in bit length (2048), taking 1.529 seconds. However, the new Somsuk-RSA algorithm is substantially faster, taking only 1.832 seconds for a bit length of 1024. Additionally, the quickest RSA for 512 is 1.6254, the original version. Thus, the optimal cryptosystem for key generation is the original RSA key generation.

Table 2: Comparative of Encryption

	Running time (sec)		
Bit length	RSA	Somsuk-RSA	New Somsuk-RSA
512	0.551	0.381	0.430
1024	0.710	0.589	0.471
2048	1.101	1.000	1.086

Table 4.2 shows that, in terms of bit length (2048), the Somsuk-RSA technique was the fastest at the encryption part, requiring 1.000 seconds. But the original RSA technique is much faster. It takes just 1.101 seconds for a 1024-bit bit length. Furthermore, the original 0.43 version of RSA is the fastest for 512. Therefore, Somsuk-RSA is the best cryptosystem for the encryption portion.

Table 3: Comparative of Decryption

	Running time (sec)		
Bit length	RSA	Somsuk-RSA	New Somsuk-RSA
512	0.481	0.319	0.309
1024	0.583	0.566	0.327
2048	0.386	0.380	0.314

According to Table 4.3, the new Somsuk-RSA technique took 0.314 seconds to decrypt the data, which is the fastest bit length (2048). Furthermore, the new Somsuk-RSA algorithm is significantly faster, requiring only 0.327 seconds to process a 1024-bit length. Also, the new Somsuk-RSA, 0.309, is the fastest RSA for 512. Therefore, the new Somsuk-RSA cryptosystem is best for the decryption section.

4.2 Graph Comparison

To have a clearer view of the comparison across the three algorithms which are the RSA, Somsuk-RSA, and new Somsuk-RSA, the bit length is represented by the x-axis, while the y-axis represents the running duration in seconds. The following figure is the graph comparison of key generation times.

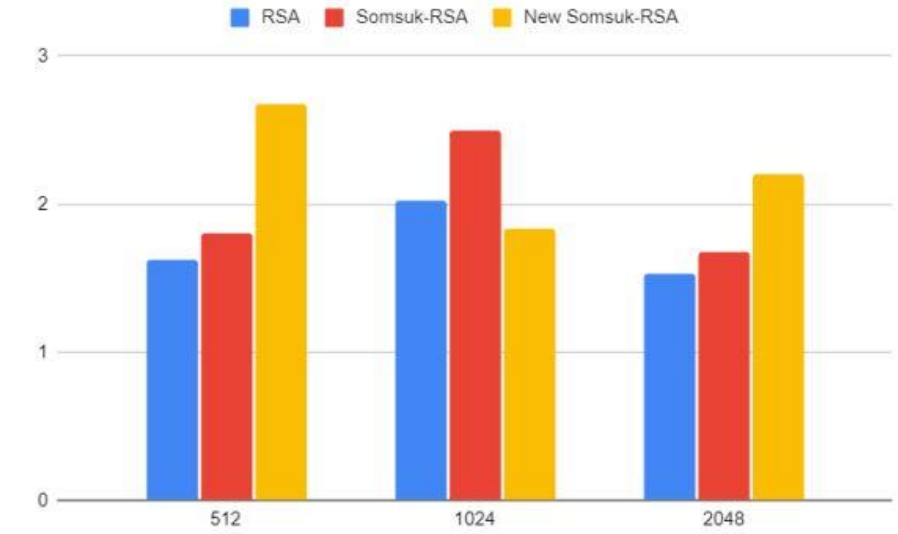

Figure 1: Graph running time vs bit length for key generation

Next is the graph comparison of encryption times.

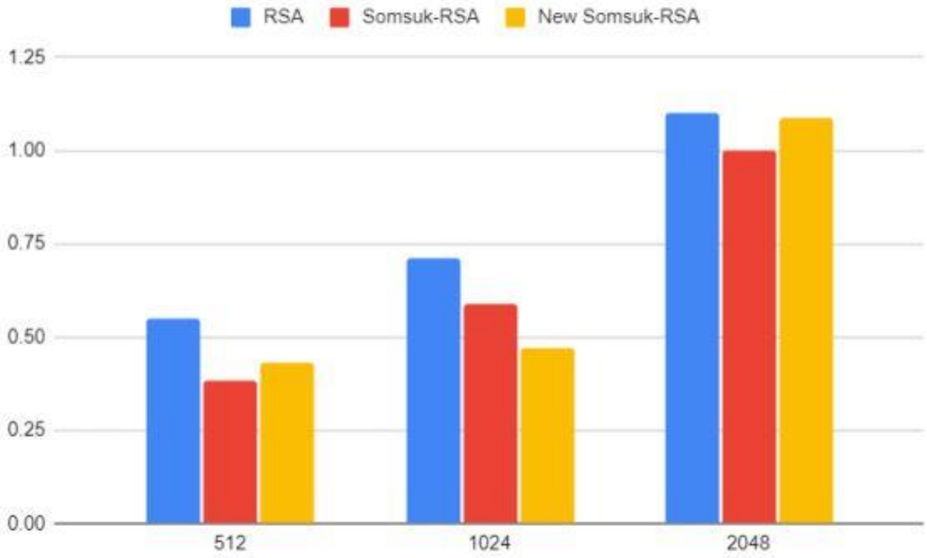

Figure 2: Graph running time vs bit length for encryption

The last one is the graph comparison of decryption times.

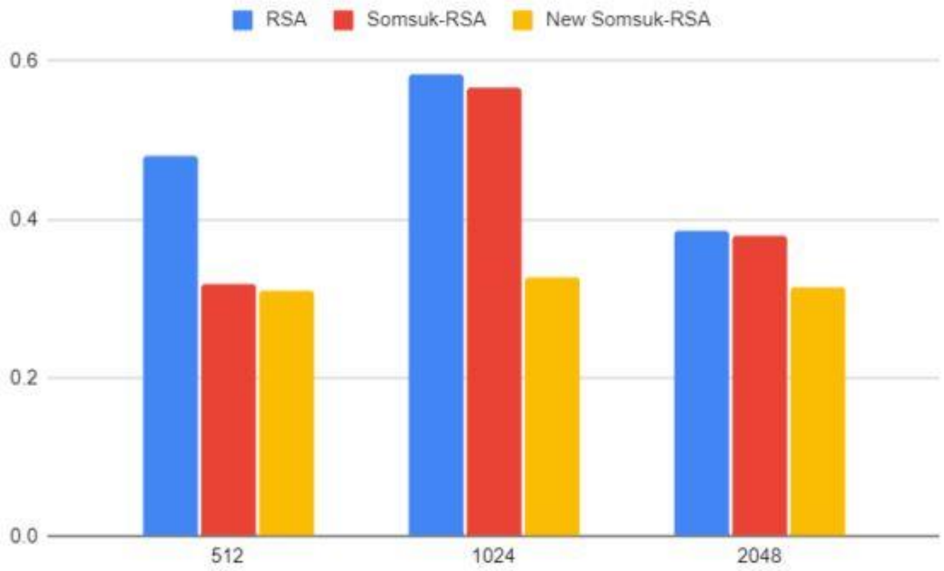

Figure 3: Graph running time vs bit length for decryption

CONCLUSION

This study presents an innovative enhancement to the Somsuk-RSA cryptosystem by replacing the Euler function with the Carmichael function. It is well-known that the risk of unauthorized access can be reduced by ensuring that the private key is larger than the public key. However, a larger private key often leads to slower decryption processes. The proposed technique addresses this challenge by significantly improving decryption efficiency.

The effectiveness of this approach is evaluated by comparing its performance with the original Somsuk-RSA and the classical RSA cryptosystems. A comparative analysis is conducted using mathematical software, examining key sizes of 512, 1024, and 2048 bits, with 100 trials performed for each algorithm. The test results are generated using an optimized algorithm, and decryption speeds are recorded in seconds based on the corresponding bit lengths. The findings consistently demonstrate that the modified Somsuk-RSA achieves faster decryption times than both the original RSA and the conventional Somsuk-RSA.

In summary, the enhanced Somsuk-RSA outperforms the existing RSA and Somsuk-RSA schemes in terms of decryption speed. Decryption efficiency is significantly improved by employing the Carmichael function instead of the Euler function. This study successfully develops and implements the improved Somsuk-RSA cryptosystem, achieving its intended objectives.

REFERENCES

Al Hasib, A. and Haque, A. A. M. M. (2008), A comparative study of the performance and security issues of aes and rsa cryptography. In *2008 third international conference on convergence and hybrid information technology*, volume 2, pp. 505–510.

Islam, M. A., Islam, M. A., Islam, N., and Shabnam, B. (2018), A modified and secured RSA public key cryptosystem based on "n" prime numbers, *Journal of Computer and Communications*, **6(03)**: 78.

Ismail, E. S., Zaharidan, M. Z., and Samat, F. (2018), ESF: Suatu kriptosistem mudah ringkas berasaskan masalah pemfaktoran, *Journal of Quality Measurement and Analysis JQMA*, **14(2)**: 81–89.

Puneeth, B., Raghunandan, K., Bhavya, K., Shetty, S., NS, K. R., Dodmane, R., and Islam, S. M. a. (2022). Preserving confidentiality against factorization attacks using fake modulus approach in RSA and its security analysis. In *2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pp. 1–6.

Raghunandan, K., Aithal, G., and Shetty, S. (2019), Comparative analysis of encryption and decryption techniques using mersenne prime numbers and phony modulus to avoid factorization attack of RSA. In *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pp. 152–157.

Rivest, R. L., Shamir, A., and Adleman, L. (1978), A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, **21(2)**: 120–126.

Shah, T., Zohaib, M., Xin, Q., Almutairi, B., and Sajjad, M. (2023), Generalization of rsa cryptosystem based on 2n primes, *AIMS Mathematics*, **8(8)**: 18833–18845.

Somsuk, K. (2017), The new equation for RSA's decryption process appropriate with high private key exponent. In *2017 21st International Computer Science and Engineering Conference (ICSEC)*, pp. 1–5.