WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Four Swapped Primes and Hidden Modulus RSA

Getaneh Awulachew Zimbele[1*], Samuel Asferaw Demilew [1], Aminu Bashir Suleiman[2]

[1]Department of Information Technology, Debre Berhan University, Debre Berhan, Ethiopia
[2]Department of Cyber Security, Federal University Dutsin-Ma, Katsina, Nigeria
Email; get.awulachew@gmail.com

A B S T R A C T

RSA is the most popular and reliable asymmetric key cryptographic mechanism for data security. Cryptanalysis attacks could compromise its security, which depends on the difficulty of factoring a large common modulus, which is public. This paper proposes Four Swapped Primes and Hidden Modulus RSA (SPHM-RSA), which adds a key pair and the public mask modulus, a pseudo-random masking number derived from the real modulus, to address these weaknesses. The second public-key component uses the derived public mask modulus instead of the second real modulus. Unlike RSA, the second real modulus is hidden. A second encryption process uses an extra public key exponent and a public mask modulus. In contrast, the initial decryption process uses an additional private key and a real modulus for data security. Spider 6 is used to compare the proposed SPHM-RSA algorithm with state-of-the-art algorithms based on key generation speed, encryption speed, decryption speed, time complexity, attack resistance, and avalanche effect. Performance analysis shows that SPHM-RSA is fast and more secure than current algorithms. SPHM-RSA improves the key generation speed of ESRKGS, MRSA, and SNA-RSA by 298%, 31%, 19978.8%; encryption time of RSA, ESRKGS, MRSA, HRM-RSA, SNA-RSA by 34%, 263%, 1331%, 33%, 19%; decryption speed of ESRKGS, MRSA by 51%, 404%, respectively. The experiment Python code and results are publicly available at https://doi.org/10.5281/zenodo.15464909.

Keywords: Cryptography, double key, Hidden Modulus, Masking, RSA, Security

INTRODUCTION

Although data security is the major priority of every company and individual, cryptographic encryption is one of the principal approaches to secure the security of data during its transmission across an untrusted channel, and the storage (Zimbele & Demilew, 2023; Stallings, 2020). In general, Cryptography may be classed into three categories a symmetric key, asymmetric key and cryptographic protocol systems. While distinct keys will be utilized for ciphering and decipherment procedures in asymmetric cryptosystems, a single shared key is used in the symmetric key cryptosystem. Crypto protocol is the application of cryptographic techniques such as the Transport Layer Security (TLS) scheme. The popular asymmetric key cryptosystems include Diffie-Hellman, RSA, Rabin, ElGamal, Elliptic Curve Cryptography (ECC), and others (Zimbele & Demilew, 2023; Stallings, 2020; Panda & Chattopadhyay, 2017).

RSA has gained widespread adoption due to its applicability in encryption, digital signatures, and key exchange mechanisms, utilizing one key for message encryption and the corresponding key for decryption, thereby ensuring the integrity, authenticity, confidentiality, and non-repudiation of electronic data communications. It was developed by Rivest, Shamir, and Adelman in 1978 (Zimbele & Demilew, 2023; Panda & Chattopadhyay, 2017; Jintcharadze & Abashidze, 2023; Surajo et al., 2023). The complexity of computing large numbers is a fundamental aspect of its security. RSA is a deterministic algorithm, indicating that the ciphertext remains consistent for a specific plaintext and key. This trait permits assailants to execute several forms of effective indirect assaults, including common modulus factorization attacks, known-plaintext attacks, chosen-plaintext attacks, and sophisticated timing attacks (Zimbele & Demilew, 2023; Chaudhury et al., 2017; Santhosh et al., 2018).

The primary objective of Swapped Primes and Hidden Modulus RSA (SPHM-RSA) is to improve the security and efficiency of advanced algorithms. This study's principal contributions encompass the following objectives:

a. Improved resilience against cryptanalysis threats, including quantum factorization, multiple exponents, lattice-based reduction attacks, and DEA. Our improved approach employs an additional key pair and a random masking modulus during the encryption process, obscuring clues from potential attackers; hence, we have determined it to be more secure than current solutions.

b. Improved resilience against factorization attacks. The actual modulus is kept secret, making the original encrypted cipher concealed by a further encryption process that utilizes a public mask modulus. The independence of the keys from the public mask modulus renders it theoretically difficult for adversaries to factor the moduli and obtain a private key.

c. Enhanced avalanche effect to make confusion and diffusion as a result, making it more secure.

d. Enhanced key generation, encryption, and decryption efficiency.

e. Improved security by concealing the mathematical connections between the public modulus and the prime numbers.

The remainder of the paper is structured as follows: Section 2 examines significant related research on the RSA cryptosystem, including its methodologies, contributions, and deficiencies. Section 3 introduces the proposed SPHM-RSA. Section 4 provides mathematical proofs for SPHM-RSA. Section 5 offers a performance analysis of SPHM-RSA in relation to existing studies. Lastly, Section 6 concludes with recommendations for future research. The preprint of this paper is published at (Getaneh & Samuel, 2024).

RELATED WORKS

Dalal et al. (2024) introduced an innovative cryptosystem method known as RSA for safeguarding data secrecy. RSA is the inaugural cryptosystem employed for both digital signatures and data encryption. It employs substantial prime numbers p and q to produce asymmetric key pairs. RSA encompasses key generation, encryption, and decryption as its fundamental methods (Zimbele & Demilew, 2023). The decryption key exponent differs from, yet is mathematically related to, the encryption key exponent. The primary limitation of this approach is its prevalent modulus. Factoring a common modulo "n," characterized by being the product of two large prime integers, is straightforward with the use of extensive parallel computational quantum computers. Consequently, the entire RSA will be decrypted, allowing for the straightforward generation of the private key. Additional RSA attacks encompass: quantum annealing integer factorization, quantum polynomial-time fixed-point attack for RSA, large decryption exponent utilizing lattice basis reduction, a combined attack on RSA via SAT approach, and double encryption attack (DEA) (Zimbele & Demilew, 2023; Stallings, 2020; Jintcharadze & Abashidze, 2023; Shahid et al., 2020; Wang et al., 2022; Susilo et al., 2020; Mumtaz & Ping, 2021; Mumtaz & Ping, 2019).

A study by Gandhi et al. (2022) introduced a third prime number to propose an improved technique named "Enhanced method for RSA cryptosystem algorithm," aimed at augmenting the security of the conventional RSA. Its encryption and decryption speeds surpass those of traditional RSA, albeit without any enhancement in security efficacy. Consequently, the original message can be readily retrieved. Therefore, assaults on RSA may also compromise its modified variant (Zimbele & Demilew, 2023).

In a study conducted by Zimbele and Demilew (2023) and Thirumalai et al. (2020), an improved methodology called "An Enhanced and Secured RSA Key Generation Scheme (ESRKGS)" was proposed, utilizing four randomly generated prime numbers p, q, r, and s to augment the time required to factor the common modulus n. The computation of the public and private keys is contingent upon the value of n, which is the product of four prime numbers. It augmented the security of RSA by prolonging the "key generation time" and diminishing direct assaults compared to conventional RSA through the utilization of higher exponents. This technique is limited by longer encryption and decryption times compared to the original RSA, and all attacks applicable to RSA are also relevant to this algorithm (Zimbele & Demilew, 2023; Thirumalai et al., 2020; Islam et al., 2018).

Delpech de Saint Guilhem, Cyprien, et al. (2021) presented "The Return of Eratosthenes," a technique for safe RSA modulus generation that use distributed sieving to augment security. This method encounters difficulties due to its complexity, substantial processing demands, and dependence on secure communication, rendering it less suitable for environments that necessitate low latency or cost-effective key generation. Conversely, contemporary cryptographic instruments prefer the Miller-Rabin Primality Test in conjunction with Secure Randomized Search, thereby enhancing efficiency, scalability, and security while mitigating the dangers of weak primes and replay attacks (Shacham & Vadhan, 2021; Bilgin & Altun, 2021; Burkhardt et al., 2023). While distributed sieving enhances scalability for substantial RSA moduli, it fails to tackle significant issues like factorization attacks or encryption efficiency, domains in which the Miller-Rabin test excels.

Gandhi et al. (2022) presented the "Enhanced RSA Cryptosystem: A Secure and Nimble Approach (SNA-RSA)" to fortify RSA in 2022. The proposed key generation technique incorporated cryptographic characteristics to mitigate modulus assaults and counter quantum computing threats. The encryption and decryption methods are altered for defense purposes. This paper asserts enhancements in performance and security using analytical comparisons and experimental validation. Notwithstanding the authors' assertions, their key generation mechanism is less efficient than RSA. The suggested approach requires greater processing resources than existing encryption techniques but provides superior security compared to RSA. The authors fail to demonstrate that decryption is the mathematical inverse. The several processes necessary render decryption more challenging and time-consuming than RSA. The simplistic calculations of the start1 and start2 parameters diminish security, performance, and memory efficiency when compared to alternative RSA variations, rather than employing secure random generation of Big Integers for a designated bit size. The bit size remains unspecified; nevertheless, the authors seem to employ smaller bit sizes that contravene public key encryption standards, notably RSA. The approach additionally disseminates n1 and n2 prime-product moduli. Insufficient focus on modulus-based factorization assaults and advancements in quantum computing. When adversaries possess knowledge of n1 and n2, quantum computers can factorize without requiring additional parameters.

Zimbele and Demilew (2023) introduced an advanced technique termed "Hidden Real Modulus RSA Cryptosystem (HRM-RSA)" to augment the security and efficiency of the conventional RSA algorithm by employing a novel security parameter known as the public mask modulus M, derived from an unpredictable random integer m and a real modulus n. In contrast to previous RSA and comparable systems that utilize a shared real modulus n for both encryption and decryption, this cryptosystem maintains a private real modulus n exclusively for decryption, while encryption employs a distinct security parameter known as the public mask modulus M. The unexpected nature of the public mask modulus M will be a problem for cryptanalysts attempting to decrypt HRM-RSA. Utilizing this strategy, the sender employs a public mask modulus M to encrypt plaintext T and produces a deceptive cipher. A deceptive cipher conceals an authentic cipher. Consequently, only the recipient with a concealed real modulus n may derive an authentic cipher from a

deceptive cipher obtained from the sender and retrieve the original plaintext T. Consequently, a man-in-the-middle attacker cannot capture an authentic cipher and a legitimate modulus. However, it is entirely feasible to enhance the security efficacy of this cryptosystem.

As discussed by Zimbele & Demilew in 2023, prior to the development of these more recent secure methods, earlier approaches such as the Enhanced and Secured RSA Key Generation Scheme (ESRKGS) and the Modified and Secured RSA (MRSA) were proposed to strengthen RSA cryptosystems. ESRKGS, introduced by Thangavel et al., used four distinct prime numbers to form the modulus, thereby increasing the difficulty of factorization and enhancing key generation security. However, this also led to higher encryption and decryption times, with vulnerabilities still comparable to traditional RSA. Similarly, Islam et al. (2018) proposed MRSA, which also employed four primes and introduced two pairs of public and private keys to improve security. Despite these enhancements, MRSA faced limitations such as increased computational overhead and a ciphertext size double that of standard RSA, raising concerns for bandwidth efficiency.

An analysis of pertinent literature indicates that classic RSA, ESRKGS, MRSA, SNA-RSA, and HRM-RSA are more dependable algorithms compared to other comparable studies. Nonetheless, these algorithms exhibit security and execution performance deficiencies that could be rectified. This study seeks to rectify these deficiencies, and the simulation outcomes were juxtaposed with contemporary relevant works: RSA, ESRKGS, MRSA, and HRM-RSA.

PROPOSED METHODOLOGY

Typically, all current methodologies, with the exception of HRM-RSA, employ a common modulus for both encryption and decryption. The security of these algorithms relies on the complexity of the large integer factorization problem, which poses no challenge for quantum annealing integer factorization, quantum polynomial-time fixed-point attacks on RSA, large decryption exponents utilizing lattice basis reduction techniques, and double encryption attacks (Zimbele & Demilew, 2023; Wang et al., 2022; Mumtaz & Ping, 2021; Mumtaz & Ping, 2019). Furthermore, all current studies, with the exception of MRSA, employ a singular key pair for encryption and decryption, so compromising their security. To circumvent these constraints, an unexpected public mask modulus M, derived from a real modulus n, and dual key pairs [16] are employed in our proposed approach.

Similar to RSA and associated research, the fundamental procedures in SPHM-RSA, including key generation, encryption, and decryption, have been modified. The second, third, and fourth steps in our key generation algorithm diverge from those in RSA and other related studies. Additionally, it employs two separate pairs of randomly generated large prime numbers (p, q) and (r, s) to compute two distinct key pairs, utilizing a random masking modulus M in the first public key component rather than a singular common modulus n.

During the key generation process, the receiver does the following fundamental sequential steps: Initially, two unique pairs of huge prime numbers, (p, q) and (r, s), are produced randomly. Secondly, if the product of the second prime pair (r, s) exceeds that of the first prime pair (p, q), interchange p with r and q with s. This enhances speed performance and reduces ciphertext size to optimize bandwidth usage. Third, the initial genuine modulus n, referred to as the hidden real modulus n in this study, is derived from the product of the first prime pair (p, q), while the product of the second prime pair (r, s) yields another common modulus variable N. Fourth, Euler's Ø(n) is determined by multiplying p-1 by q-1, while Euler's Ø(N) is computed by multiplying r-1 by s-1. Fifth, the initial prime public key exponent e is calculated using Ø (n). The second prime public key exponent f is calculated using Ø(N). Seventh, the first prime private key exponent d is calculated using e and Ø(n). The second prime private key exponent g is calculated using f and Ø(N). A huge multiplayer number m will be produced randomly in the ninth instance. The integer m can possess any bit size, albeit an increased bit size enhances security. The sum of the real modulus n and the random multiplier m yields an unpredictable public mask modulus M. The masking procedure conceals the actual modulus n from public view, maintaining its confidentiality, in contrast to the standard modulus in RSA. Consequently, in this study, a real modulus n is designated as a concealed real modulus n. The receiver publicly discloses the initial public key pairs (e, M) and the subsequent public key pairs (f, N) to the correspondents, while retaining the first private key pairs (d, n) and the second private key pairs (g, N) in confidentiality from the correspondents.

In the encryption process, the sender initially encrypts the plaintext T utilizing the components of the second public key (the second public exponent f and the common modulus N). Secondly, it re-encrypts the intermediate result cipher utilizing the initial public key components (the first public key exponent e and the public mask modulus M). The transmitter ultimately conveys a deceptive ciphertext to the receiver. This procedure conceals the authentic cipher from adversaries. This enhances the security efficacy of SPHM-RSA.

In the decryption process, the receiver initially encrypts the received erroneous ciphertext with a concealed authentic modulus n, the first component of the private key, to generate a legitimate ciphertext. This unmasking procedure also enhances decryption speed efficiency. The receiver decrypts a valid ciphertext using the first components of the private key (the first private key exponent d and the concealed modulus n). Ultimately, the receiver reconstructs the original text by decrypting the intermediate cipher obtained from the second stage utilizing the components of the second private key (the second private key exponent g and the common modulus N).

The exponents of both the private and public keys are contingent upon the actual modulus n, which remains concealed from the public to prevent mathematical cryptanalysis. The absence of a direct mathematical correlation between the public mask modulus M and the key exponents e and d enhances the security of our cryptosystem, rendering it more challenging to decrypt.

Initially, the plaintext is encrypted utilizing the components of the second public key, followed by a subsequent encryption with the components of the first public key. The resulting encrypted ciphertext will undergo masking using the mask number M. This method enhances the complexity of

cryptanalysis concerning modulus n and known-ciphertext attacks. Consequently, these techniques enhance the security of the proposed SPHM-RSA cryptosystem compared to current algorithms. The bit length of the plaintext must be less than the bit length of the modulus N.

The masking modulus M, characterized by a large integer number, decreases encryption size, hence enhancing transmission efficiency between communicating entities. To enhance security, it conceals the mathematical correlation among the actual modulus N, the keys (e and d), and the prime numbers (p and q) from the correspondents. The speed performance of MRSA is diminished due to its reliance on huge de/encryption exponents, whereas ESRKGS further suffers from several exponentiation and modulation phases. To mitigate these restrictions, we employed a substantial public mask modulus M for the encryption process, leading to a reduction in the number of steps in the proposed technique.

Key Generation and En/Decryption processes of the SPHM-RSA are shown in Algorithm 1. Key Generation and En/Decryption architectures of SPHM-RSA are further illustrated in Fig. 1 and Fig. 2, respectively.

Algorithm 1: SPHM-RSA Algorithm

SPHM-RSA_Key_Generation ()

INPUT:

Four randomly distinct prime numbers p, q, r, and s within bit-length/2, and a randomly generated mask multiplier m within bit-length.

OUTPUT:

Find public key exponents (e, f), private key exponent (d, g), and modulus numbers (n, M, N) in bit length.

Begin

Procedure (p, q, r, s, m, e, d, f, g, n, N, M)

1. Randomly generate distinct prime numbers p, q, r, and s.
2. If(r * s > p * q)
   a. swap = p, p = r, r = swap
   b. swap = q, q = s, s = swap
3. Compute a private real modulus n and common modulus N such that,
   c. $n \leftarrow p * q$
   d. $N \leftarrow r * s$
4. Calculate Euler Ø (n), and Ø (N) such that
   a. $Ø(n) \leftarrow (p - 1) * (q - 1)$
   b. $Ø(N) \leftarrow (r - 1) * (s - 1)$
5. Randomly generate the first prime public key exponent e, such that,

$GCD(e, Ø(n)) = 1, 1 < e < Ø(n)$

6. Randomly generate the second prime public key exponent f, such that,

$GCD(f, Ø(N)) = 1, 1 < f < Ø(N)$

7. Compute the first private key exponent d, such that,

$d \leftarrow e^{-1} \bmod Ø(n)$

8. Compute the second private key exponent g, such that,

$g \leftarrow f^{-1} \bmod Ø(N)$

9. Generate a random number m, such that,

$m \leftarrow RNG(m.length, rand), m>1;$

10. Compute a public mask modulus M for n, such that,

$M \leftarrow m * n;$

End Procedure

End

SPHM-RSA_Encryption ()

Input:

Plain text (T) < N, public key exponents (e, f), and public modulus numbers (M, N).

Output:

Cipher text (C).

Begin

Procedure (T, e, f, M, N, C)

$C \leftarrow (T^{f} \bmod N)^{e} \bmod M$

End Procedure

End

SPHM-RSA_Decryption ()

Input:

Cipher text (C), private key exponents (d, and g), and real modulus numbers (private modulus (n), and N).

Output:

Find plain text (T).

Begin

Procedure (C, d, g, n, N, and T)

$T \leftarrow ((C \bmod n)^{d} \bmod n)^{g} \bmod N$ // modular distributive property. The first mod removes the mask and improves decryption speed performance.

End Procedure

End

Key generation process in Fig. 1. shows that SPHM-RSA algorithim randomly generates four large prime numbers p, q, r, and s by Random Prime Number Generator function (RPNG), and random multiplier number m by Random Number Generator function (RNG) to use as input, and computes double key pairs: Public Keys KU = [(e, M), (f, N)], and Private keys KR =[(d, n), (g, N)] as output.

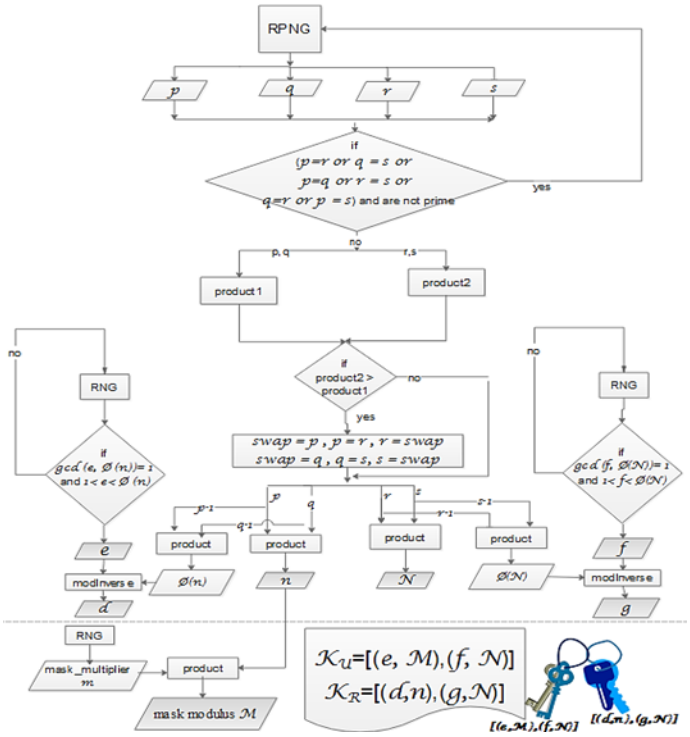

Fig. 1. SPHM-RSA key generation architecture

Figure 2 illustrates a revised flowchart of the SPHM-RSA method for encryption and decryption procedures. The flowchart illustrates that during the encryption process, Alice utilizes Bob's public key components, K_U= [(e, M), (f, N)], along with her plaintext T, which has a bit length shorter than that of N, to generate a doubly encrypted false ciphertext output C for transmission to the recipient Bob via the SPHM-RSA encryption algorithm. The public mask modulus M is employed for masking both the actual ciphertext C, calculated as C = (T^f mod N) ^{e} mod n in the context of MRSA, and the genuine modulus n. When Alice employs mask modulus M to encrypt intermediate cipher C_1, A real cipher C becomes hidden in a false cipher C_2.

Upon receiving the erroneous cipher C2, Bob initiates the decryption process with our SPHM-RSA decryption algorithm alongside his private key components K_R = [(d, n), (g, N)]. During the decryption procedure, he initially calculates the authentic cipher C utilizing the erroneous cipher C_2 obtained from Alice and his concealed true modulus n. Subsequently, he employs his private key component (d, n) and authentic ciphertext C as input to calculate the intermediate cipher C_1 as output. Ultimately, he retrieves the original plaintext T from the intermediate cipher C_1 utilizing his private key components (g, N) and our SPHM-RSA decryption technique.

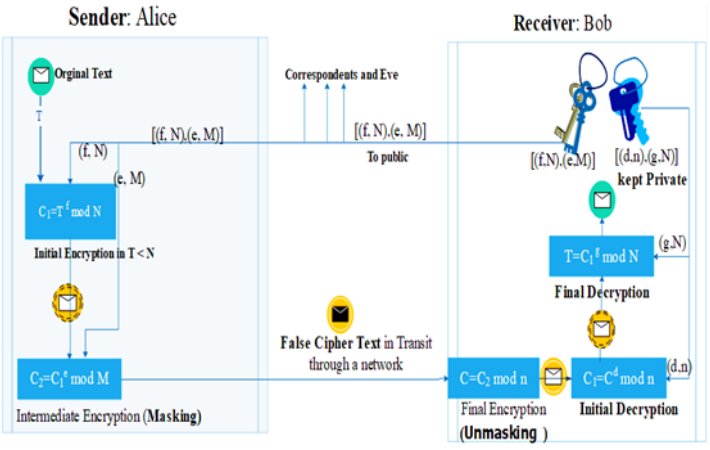

Fig. 2. SPHM-RSA encryption/decryption architecture

RESULTS AND DISCUSSION

SPHM-RSA is executed utilizing Python on Spyder 6, operating on an Intel(R) Core(TM) i5-6200U CPU at 2.30GHz (4 CPUs) with 12 GB of RAM. For our experiment, four unique random prime numbers are created from each of six possible bit sizes: 28-bit, 56-bit, 128-bit, 256-bit, 1024-bit, and 2048-bit. For the simulation, we utilized six distinct bit sizes and randomly generated unique prime numbers.

To enhance the reliability of our result analysis, we conducted the algorithms five times for each input, considering the average execution time for key creation, encryption, and decryption. The implementation code for SPHM-RSA is available at Error! Reference source not located. We generate the graphs with Spyder 6, as it employs Python code with comprehensive libraries such as matplotlib, seaborn, and pandas for producing high-quality visualizations.**Key**

Generation Time of Algorithms

In terms of average key generation time, the key generation performance of SPHM-RSA is notably inferior to that of RSA and HRM-RSA by 18% and 8%, respectively, hence enhancing security performance. Nonetheless, it enhances the key generation efficiency of ESRKGS, MRSA, and SNA-RSA by 298%, 31%, and 19978.8%, respectively.

According to the key generation time in Fig. 3, SPHM-RSA has marginally inferior performance compared to RSA and HRM-RSA, aimed at enhancing security against the factorization problem; however, it significantly outperforms ESRKGS, MRSA, and SNA-RSA. Consequently, SPHM-RSA key production is less intricate than MRSA, ESRKGS, and SNA-RSA due to its utilization of a concealed modulus instead of incorporating superfluous exponentiation and modulation processes.

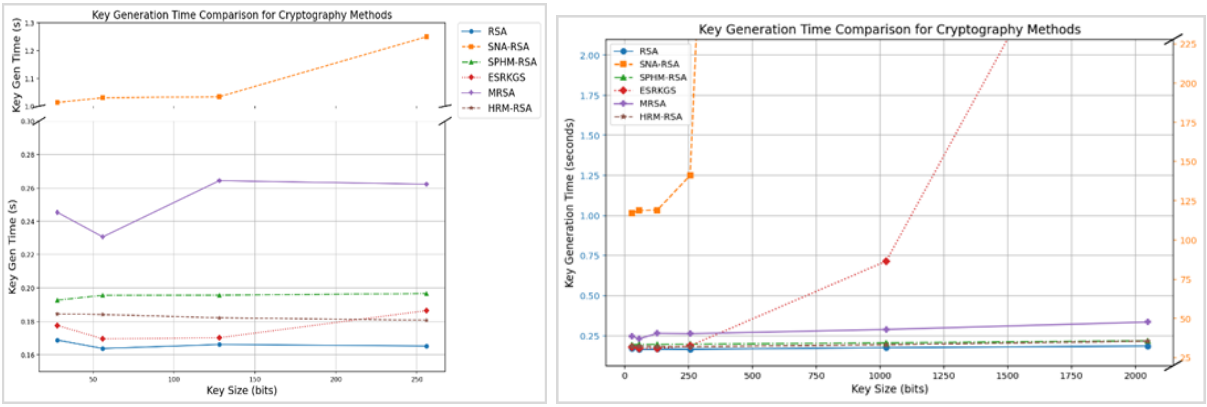

Fig. 3. Analysis of key generation performance (seconds)

Encryption Time of Algorithms

According to average encryption time, SPHM-RSA demonstrates enhanced encryption performance over standard RSA, ESRKGS, MRSA, HRM-RSA, and SNA-RSA by 34%, 263%, 1331%, 19%, and 3170.6%, respectively. This demonstrates that our algorithm surpasses existing state-of-the-art algorithms.

Figure 4 illustrates that SPHM-RSA surpasses all contemporary methods in terms of average encryption time. As the bit length escalates, our method exhibits enhanced performance.

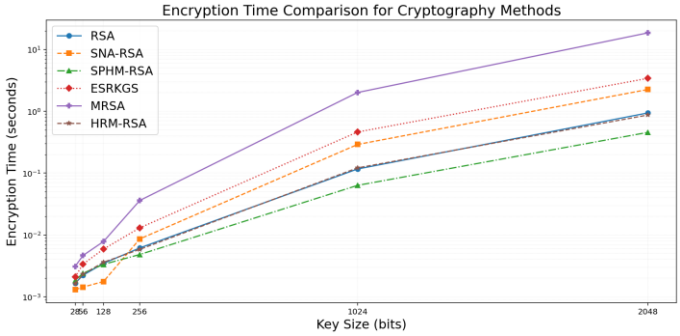

Fig. 4. Analysis of Encryption Performance (seconds)

Decryption Time of Algorithms

According to the average decryption times of algorithms, SPHM-RSA exhibits a decryption speed that is 51% superior to ESRKGS and 404% superior to MRSA. Owing to the dual decryption mechanism employed to enhance its security, the decryption speed of SPHM-RSA is inferior by 27%, 29%, and 78.3% compared to RSA, HRM-RSA, and SNA-RSA, respectively. Figure 5 illustrates that the decryption performance of SPHM-RSA markedly surpasses that of other ESRKGS and MRSA, however it is marginally inferior to RSA and HRM-RSA due to the dual decryption procedure implemented to enhance security. The decryption efficiency of SNA-RSA surpasses that of its counterparts. The decryption procedure employs the Chinese Remainder Theorem (CRT) approach.

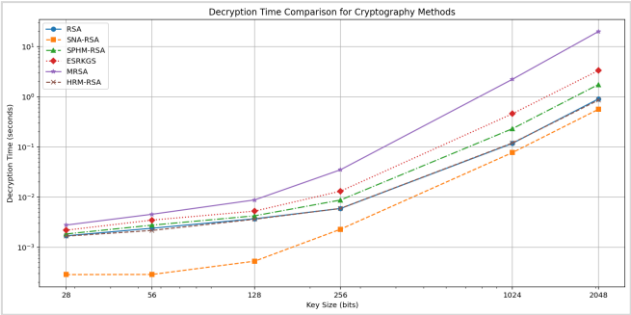

Fig. 5. Analysis of Decryption Performance (seconds)

Time Complexity of Algorithms

Based on the complexity of MILLER-RABIN computed in (Zimbele & Demilew, 2023; Islam et al., 2018), and Fig. 3, SPHM-RSA is less complex than MRSA, ESRKGS, and SNA-RSA. Therefore, it requires fewer computing resources than other existing works. However, it is significantly complex than RSA, and HRM-RSA to improve its security.

Security Strength of Algorithms

Various techniques exist to compromise RSA, except HRM-RSA, such as DEA, multiple private exponent attacks, and the application of mathematical theories (Zimbele & Demilew, 2023).

Double Encryption Attack

Our SPHM-RSA technique conceals the actual modulus n from the public while publicly disclosing the masked modulus M. In our approach, the plaintext is initially encrypted with a common modulus N, after which the resultant ciphertext undergoes a second encryption with the public mask modulus M, yielding a false ciphertext. These complicate the retrieval of plaintext from a deceptive cipher with DEA (Zimbele & Demilew, 2023).

Factorization Attack

To decrypt RSA, we can factor the common modulus “n” into prime numbers P1, P2, … Pn via the Sieve of Eratosthenes (Zimbele & Demilew, 2023). In our technique, "n" remains confidential, and the public mask modulus "M" is unrelated to prime numbers; thus, attackers are unable to discern any information to compromise our cryptosystem.

Multiple Private Exponent Attack

The key's size must not exceed the actual modulus "n," resulting in a distinct key pair. Consequently, a genuine modulus remains confidential, but a public mask modulus bears no correlation to prime numbers; this complicates attackers' efforts to execute factorization and multiple key generation assaults.

4) Avalanche Effect (AE) of Algorithms

AE = (Number of Flipped bits in cipher text) / (Number of bits in cipher text)*100% s (1)

We employed three scenarios for comparison against the standard cipher (Original Plain Text and Public Key (PT&PK)); Original Plain Text with a single bit alteration in the Public Key (PT&OPK), a single bit alteration in the Plain Text with the Original Public Key (OPT&PK), and a single bit alteration in both the Plain Text and the Public Key (OPT&OPK) (Raju & Kiran, 2021).

The average AE Fig. 6 shows that our SPHM-RSA algorithm has a relatively higher percentage of AE, which shows it is highly secured as compared to other state-of-the-art algorithms.

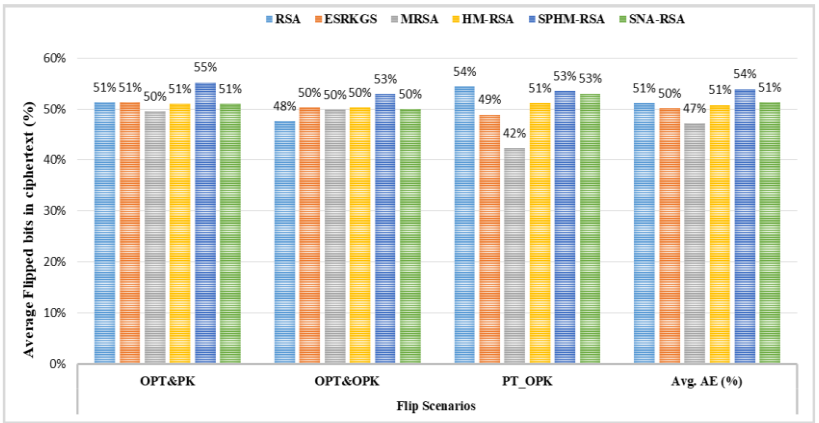

Fig. 6. Average AE Analysis of Algorithms (%)

CONCLUSION

This work proposes the Swapped Primes and Hidden Modulus RSA Cryptosystem (SPHM-RSA). Current cryptosystems rely on a shared modulus, rendering them susceptible to several forms of cryptanalysis, including quantum factorization, multiple exponents, lattice-based reduction attacks, DEA, and others. Our improved approach employs an additional key pair and a random masking modulus during the encryption process, effectively concealing clues from attackers; hence, we have determined it to be more secure than current solutions. The real modulus remains confidential, rendering the initial encrypted cipher obscured by a subsequent encryption process utilizing a public mask modulus. The keys are independent of this public mask modulus, making it mathematically challenging for attackers to decipher our algorithm and derive a private key.

The suggested algorithm exhibits enhanced attack resistance, an avalanche effect, and improved speeds for key generation, encryption, and decryption, rendering it more safe and efficient for implementation across many devices and in high-security contexts such as e-business and e-governance applications.
Future endeavors may enhance the decryption performance of our SPHM-RSA with the implementation of CRT and its application in security-sensitive domains such as email, bitcoin, mobile technology, IoT, and medical imaging security, among others.

COMPLIANCE WITH ETHICAL STANDARDS

Since this article does not contain any studies with living things, ethical approval is not required.

DATA AVAILABILITY

The simulation data utilized in this study consists of randomly produced huge prime numbers, with their sizes specified in the publication. The authors' randomly generated numbers are available upon request from the appropriate author. Randomly generated sample data can be accessed using the published Java code of this research at https://codeocean.com/capsule/9779950/tree/v1.

CODE AVAILABILITY

A comprehensive step-by-step Python code for this paper is accessible at https://doi.org/10.5281/zenodo.15464909, and its Java code is published

FUNDING STATEMENT

This research work is not funded by any organization.

CONFLICT OF INTEREST

We affirm that we possess no recognized competing financial interests or personal affiliations that may have seemingly influenced the work presented in this publication.

References

B. Wang, X. Yang, and D. Zhang, “Research on quantum annealing integer factorization based on different columns,” Frontiers in Physics, vol. 10, no. June, pp. 1–10, 2022. https://doi.org/10.3389/fphy.2022.914578.

C. Thirumalai, S. Mohan, and G. Srivastava, "An efficient public key secure scheme for cloud and IoT security," Computer Communications, vol. 150, pp. 634-643, 2020. https://doi.org/10.1016/j.comcom.2019.12.015.

Cyprien Delpech de Saint Guilhem, Eleftheria Makri, Dragos Rotaru, and Titouan Tanguy, “The Return of Eratosthenes: Secure Generation of RSA Moduli using Distributed Sieving,” In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), Association for Computing Machinery, New York, NY, USA, 594–609, 2021. https://doi.org/10.1145/3460120.3484754.

D. G. Raju, and K. (n.d.) Kiran, “Analysis of Avalanche Effect in Asymmetric Cryptosystem Using NTRU & RSA,” 7884.

E. Jintcharadze and M. Abashidze, "Performance and Comparative Analysis of Elliptic Curve Cryptography and RSA," 2023 IEEE East-West Design & Test Symposium (EWDTS), Batumi, Georgia, 2023, pp. 1-4, doi: 10.1109/EWDTS59469.2023.10297088.

F. Shahid, et al., “PSDS–proficient security over distributed storage: A method for data transmission in cloud,” IEEE Access, vol. 8, pp. 118285-118298,2020. https://doi.org/10.1109/ACCESS.2020.3004433.

G. A. Zimbele, and S. A. Demilew, “Hidden Real Modulus RSA Cryptosystem,” vol. 22, no. 2, pp. 238–247, 2023. https://doi.org/10.47839/ijc.22.2.3094.

Getaneh Awulachew, Samuel Asferaw. “Double Key Pair and Hidden Modulus RSA Cryptosystem,” Research Square, 19 September 2024, PREPRINT (Version 1), https://doi.org/10.21203/rs.3.rs-4655782/v1

H. Shacham and S. Vadhan, "Secure Primality Testing: The Role of Distributed Algorithms in Cryptography," IEEE Transactions on Information Theory, vol. 67, no. 8, pp. 5000-5012, 2021.

Jakob Burkhardt, Ivan Damgård, Tore Kasper Frederiksen, Satrajit Ghosh, and Claudio Orlandi, “Improved Distributed RSA Key Generation Using the Miller-Rabin Test,” In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), Association for Computing Machinery, New York, NY, USA, 2501–2515, 2023. https://doi.org/10.1145/3576915.3623163.

M. A. Islam, M. A. Islam, N. Islam, and B. Shabnam, “A Modified and Secured RSA Public Key Cryptosystem Based on ‘n’ Prime Numbers,” J. Comput. Commun., vol. 06, no. 03, pp. 78–90, 2018.

M. Mumtaz and L. Ping, “Forty years of attacks on the RSA cryptosystem: A brief survey,” Journal of Discrete Mathematical Sciences and Cryptography, vol. 22, issue 1, pp. 9-29, 2019. https://doi.org/10.1080/09720529.2018.1564201.

M. Mumtaz, and L. Ping, “Cryptanalysis of a special case of RSA large decryption exponent using lattice basis reduction method,” Proceedings of the IEEE 6th International Conference on Computer and Communication Systems (ICCCS), Chengdu, China, 2021, pp. 714-720, https://doi.org/10.1109/ICCCS52626.2021.9449268.

N. Bilgin and M. Altun, "Enhanced Primality Testing for RSA Key Generation," Journal of Cryptographic Engineering, vol. 11, no. 3, pp. 303–314, 2021.

P. Chaudhury, S. Dhang, M. Roy, S. Deb, J. Saha, A. Mallik, R. Das, “ACAFP: Asymmetric key based cryptographic algorithm using four prime numbers to secure message communication. A review on RSA algorithm,” 2017 8th Industrial Automation and Electromechanical Engineering Conference, IEMECON 2017, 332–337. https://doi.org/10.1109/IEMECON.2017.8079618.

P. K. Panda, and S. Chattopadhyay, “A hybrid security algorithm for RSA cryptosystem,” 2017 4th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2017, 2017.

Santhosh Kumar B J, Roshni V K, and Anjali Nair, “Comparative Study on AES and RSA,” International Conference on Communication and Signal Processing, Electronic ISBN. 978-1-5090-3800-8, pp. 0501-0504, IEEE, India, 2018.

Surajo, Y., Suleiman, A. B., & Yahaya, U. (2023).Exponential-Self-Adaptive Random Early DetectionScheme for Queue Management in Next GenerationRouters. FUDMA Journal of Sciences, 7(3), 33-39.https://doi.org/10.33003/fjs-2023-0703-1763.

T. Gandhi, M. Navlakha, R. Raheja, V. Mehta, Y. Jhaveri and N. Shekokar, "Enhanced RSA Cryptosystem: A Secure and Nimble Approach," 2022 5th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 2022, pp. 388-392, doi: 10.1109/ICAST55766.2022.10039627.

W. Stallings, Cryptography and Network Security: Principles and Practice, 8th ed., Pearson Education, 2020, ISBN. 978-0-13-670722-6.

W. Susilo, W. Susilo, J. Tonien, and G. Yang, “Institutional knowledge at Singapore Management University – A generalised bound for the Wiener attack on RSA,” vol. 2020, pp. 1–4, 2020. https://doi.org/10.1016/j.jisa.2020.102531.

Y. M. Dalal, S. S, A. K, T. Y. Satheesha, A. PN and S. Somanath, "Optimizing Security: A Comparative Analysis of RSA, ECC, and DH Algorithms," 2024 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Bagalkote, India, 2024, pp. 1-6, https://doi.org/10.1109/NKCon62728.2024.10775183.

Zimbele, Getaneh Awulachew, Demilew, Samuel Asferaw(PhD) (2020) DKPHM-RSA Cryptosystem [Source Code]. https://doi.org/10.24433/CO.3817501.v1.