ISSN: 2182-2069 / E-ISSN: 2182-2077

An Enhanced Cryptographic Method for Securing and Accelerating Digital Signatures

Thanapat Chiawchanwattana[1], and Kritsanapong Somsuk[2*]

[1]Department of Computer and Communication Engineering, Faculty of Technology and Engineering, Udon Thani Rajabhat University, Udon Thani, Thailand. thanapat.ch@udru.ac.th, https://orcid.org/0009-0003-3719-6430

[2*]Department of Computer and Communication Engineering, Faculty of Technology and Engineering, Udon Thani Rajabhat University, Udon Thani, Thailand. kritsanapong@udru.ac.th, https://orcid.org/0000-0002-1311-8222

Received: October 07, 2025; Revised: November 17, 2025; Accepted: January 06, 2026; Published: February 27, 2026

Abstract

This study aims to propose the improvement of RSA to secure the digital signature. This algorithm, which is called FE-RSA, focuses on two primary challenges: security and computing efficiency. A secret key and a fake public key are selected for the implementation to provide an additional security layer. In fact, FE-RSA attempts to protect against digital signature forgery. Moreover, the other main point of FE-RSA is that although an attacker can factor the modulus, FE-RSA is still secure. In addition, 512-bit, 1024-bit, and 2048-bit key sizes across four cryptography algorithms, including RSA, Multi-Prime RSA, the application of RSA and ElGamal (RSA-ElGamal), and FE-RSA, are selected for the experiment. Performance evaluations focused on processing time for the generation and validation processes. The experimental results demonstrate that FE-RSA regularly outperforms Multi-Prime RSA and RSA-ElGamal in terms of signing and verifying speed. In the signing process, FE-RSA is approximately 92.47%, 64.50%, and 3.82% faster than RSA-ElGamal for key sizes of 512-bit, 1024-bit, and 2048-bit, respectively. However, in the verification process, this method reduces processing time by about 97.12%, 87.53%, and 34.33% for identical key sizes. Then, it implies that FE-RSA provides the most significant performance benefit in the verification process. Although the processing time required in FE-RSA is slower than that of RSA, the security is higher than that of RSA. The reason is that RSA is based on only the Integer Factorization Problem (IFP). On the other hand, FE-RSA is based on both the Discrete Logarithm Problem (DLP) and IFP. However, FE-RSA is effectively applied in situations when a specific and reliable verifier has the secret key. In fact, FE-RSA integrates enhanced computing efficiency with multi-layered security. Therefore, the proposed method is established as a viable and resilient option for digital signature systems.

Keywords: FE-RSA, RSA-ElGamal, Integer Factorization Problem, Discrete Logarithm Problem, RSA, Multi-Prime RSA.

1 Introduction

Cryptography (Sasikumar & Nagarajan, 2024) is one of the significant techniques for securing information transmitted over the communication channel. It can secure the secret information by using

Journal of Internet Services and Information Security (JISIS), volume: 16, number: 1 (February-2026), pp. 624-641. DOI: *10.58346/JISIS.2026.I1.035*

*Corresponding author: Department of Computer and Communication Engineering, Faculty of Technology and Engineering, Udon Thani Rajabhat University, Udon Thani, Thailand.

encryption and decryption processes. Based on modern computers, cryptography is categorized into three groups. The first group is symmetric key cryptography (Khudhair et al., 2024), using the secret key for encryption and decryption processes. The advantage is the processing time to finish the process on the sender and receiver sides. However, the secure channel for exchanging the secret key is the problem for this group. The second group is asymmetric key cryptography (Halak et al., 2022), which employs a mathematically correlated key pair for encryption and decryption processes. The public key is always disclosed to everyone in the group. However, the private key must be kept secret. In addition, asymmetric key cryptography is known as public key cryptography. In fact, many public key cryptography algorithms have been developed. Moreover, some algorithms can be implemented for both data encryption and the digital signature process. Currently, many applications require the digital signature process for authentication. The example applications are the incorporation into online transactions for user identification in cloud systems and the maintenance of data integrity in dispersed environments. The third group is post-quantum cryptography (PQC) (Joseph et al., 2022; Dam et al., 2023), which is developed to resist attacks from quantum computers. However, PQC is in the development stage. Therefore, many applications still select public key cryptography algorithms to secure the secret information. In fact, the aim of developing algorithms in PQC is to resist attacks from quantum computers. However, PQC remains in the developmental stage and continues to be studied alongside the public key cryptography algorithms.

RSA is a well-known algorithm in public key cryptography that is still widely used today (Shivaramakrishna & Nagaratna, 2023). It achieves high security levels when it is executed by using a large modulus. In fact, RSA can be applied to secure the secret information and sign the digital signature (Qiang, 2025). The security is based on integer factorization (Somsuk, 2022). If the modulus is factorable, the private key may be retrieved, and the secret information can be disclosed by an attacker who has the private key. To address this problem, many enhancements to RSA (Sethukarasi et al., 2025; Kaliyamoorthy & Ramalingam, 2022; Ochoa-Jiménez et al., 2020; Srivastava & Gupta, 2024) have been suggested to improve the security. Multi-Prime RSA (Zhao et al., 2025) was presented to expand the size of the modulus, including several prime factors, to significantly improve the difficulty of the factorization procedure. Nevertheless, the principal attack channel against Multi-Prime RSA continues to be factorization. Moreover, applying RSA with ElGamal (RSA-ElGamal) was proposed to increase the security level (Banu et al., 2025; Awad et al., 2025). In fact, it is one of the double encryption algorithms to secure the information. Furthermore, RSA-ElGamal is based on both the Integer Factorization Problem (IFP) (Zhang et al., 2025; Shatnawi et al., 2023) and the Discrete Logarithm Problem (DLP) (Hashim, 2025; Han & Zhuang, 2022). Therefore, it is very difficult to break when compared with RSA. However, RSA-ElGamal requires substantial computational expenses because of ElGamal's requirement (Adeniyi et al., 2022) of selecting a new primitive root for each digital signature. Then, a high processing time is required to finish the process. Therefore, this algorithm is unsuitable for implementation because of its high computation cost.

This paper aims to present FE-RSA, modified from RSA, to increase the security level. The main concept is that FE-RSA must reduce the computation time when compared with RSA-ElGamal. However, the security level is still based on IFP and DLP. In fact, a secret key and a fake key are included in the main process to prevent digital signature forgery and retain robust security against integer factorization.

The remainder is organized as follows. Section 2, the principles of RSA and ElGamal, together with modified algorithms to increase the security and efficiency of RSA-based digital signature schemes, are reviewed. The proposed method is presented in Section 3. In addition, Section 4 reports the experimental

results and the evaluation results in terms of signing and verification time. Section 5 provides an analysis of the security properties and computational efficiency of the proposed method in comparison with existing schemes. Finally, the paper outlines directions for future work.

2 Related Works

In this section, RSA, ElGamal, and methodologies designed to enhance both the security and efficiency of RSA are reviewed. In fact, RSA and ElGamal are famous algorithms to be applied for securing the secret information and signing and verifying digital signatures. However, this study focuses only on the digital signature. Furthermore, the adaptation and integration of these algorithms to reduce the weaknesses of RSA is also reviewed to provide the essential context and motivation for the proposed method.

RSA

RSA is one of the most popular public key cryptosystems. It was established in 1978 by R. Rivest, A. Shamir, and L. Adleman. This technique is based on the mathematical challenge of factoring the product of large prime integers. The implementation of RSA using at least two prime integers to generate the modulus and Euler's function could be categorized into three primary procedures. The first is the key generation process to generate all parameters for the signing and verifying processes. In this paper, two prime numbers are required to generate the modulus. After completion of the signing process, which is the second process, the plaintext and the signature are sent to the recipient to verify the digital signature. On the recipient's side, it is the third process to verify the digital signature to confirm that this signature is valid. In fact, the verifier will compare the result from the verifying process with the original plaintext. If these values are equivalent, this digital signature is validated. On the other hand, the signature must be rejected when the result is not equal to the plaintext. In addition, the modulus must be generated with a minimum size of 1024 bits (Gulen & Baktir, 2023; Xiao et al., 2022) and should be derived from prime integers of equivalent length to avoid attacks from several factoring algorithms. Therefore, each prime number must be produced with a minimum length of 512 bits. Furthermore, a hash function (Al-Gailani, 2025; Chen & Ye, 2022) may be applied to calculate the hash value of the message, which is then signed rather than the original content directly. The benefit of employing a hash function in conjunction with a cryptographic technique is that it generates a fixed-size output irrespective of the input size. Nonetheless, hash collisions may emerge as the primary issue with the execution of this strategy. In fact, different messages may yield the same hash values. To alleviate this issue, collision-resistant hash algorithms such as SHA-256 or SHA-3 are selected to be used with cryptography algorithms. Therefore, it increases the overall dependability of digital signatures. However, a primary limitation of RSA is that its security is exclusively dependent on the difficulty of IFP. Thus, if an effective factorization technique is identified, RSA becomes vulnerable.

Nevertheless, the problem of employing RSA for digital signature applications is its exclusive dependence on only IFP. If the modulus is factored, RSA is compromised. This research will provide an improved approach that is more efficient than RSA in terms of security efficiency.

ElGamal and Other Public Key Cryptography Algorithms

ElGamal is a public key cryptography technique developed by T. ElGamal in 1985. This approach is applicable for data encryption and digital signatures. However, the security differs from RSA since it is predicated on the challenge of resolving the DLP inside modular arithmetic. It contrasts with the

proposed method, which relies on the security of both DLP and IFP. Furthermore, the proposed technique attains superior computational efficiency compared to the double-encryption technique derived from the combination of RSA and ElGamal.

Moreover, alternative public key encryption techniques include Elliptic Curve cryptography (ECC) (Ullah et al., 2023; Kumar & Sharma, 2024), where the security level is based on Elliptic Curve Discrete Logarithm Problem (EDLP) (Chattopadhyay et al., 2022; Somsuk, 2021) and the Digital Signature Algorithm (DSA) (Flores-Carapia et al., 2025). Generally, ECC may be used for both data security and digital signatures. DSA is specifically designed for digital signatures and is frequently employed in standard protocols. However, this study only examines RSA and ElGamal, because the proposed method is directly pertinent to these two cryptographic systems. The study's limited scope for a more precise comparison analysis supports the demand for using the proposed method.

Modified Algorithms

This section reviews many algorithms that enhance RSA's security. Multi-Prime RSA is a variant of RSA designed to reduce the risk of factoring huge integers. This technique's benefit lies in its higher computational cost for factoring the modulus. Thus, the duration required to complete this operation is significantly extended. However, Multi-Prime RSA remains based on IFP. Therefore, it may be rapidly attacked with the development of an effective factorization method. In fact, more computing resources are required to finish the process, when Multi-Prime RSA is applied for the digital signature. It is from the increased size of n. However, to speed up Multi-Prime RSA, the process of modular exponentiation may be enhanced by using techniques such as the Chinese Remainder Theorem (CRT) (Pathirage et al., 2021), binary exponentiation, and other efficient methods. These methodologies are equally relevant to RSA and other algorithms that significantly depend on modular exponentiation. Although the security level is increased, it is still based on IFP. A hybrid method combining RSA and ElGamal (RSA-ElGamal) has been proposed to improve security levels. Furthermore, RSA-ElGamal requires a dual-layer encryption structure to complete the procedure. The procedure for implementing RSA-ElGamal is as follows: the creator establishes parameters for both RSA and ElGamal, and thereafter shares the public parameters; the digital signature is first formed using RSA. The output is then signed using ElGamal. The result is sent to the receiver, who first authenticates the signature through ElGamal and then applies RSA for verification. The sequence of the algorithms may be reversed: the signature may first be generated using ElGamal and then signed using RSA. The verification occurs oppositely the security of this system depends on both IFP and DLP. Thus, a dual defense against adversaries is required. Although providing a dual-layer security architecture, RSA-ElGamal is constrained by certain limitations, particularly in terms of computing efficiency and algorithm combination. The probabilistic attributes of key generation for ElGamal and the sequential execution of the algorithms result in considerable computing overhead. This approach is unsuitable for real-time digital signature applications. Moreover, the hybrid structure has additional costs for communication, since each message requires two distinct signing and verification processes. The proposed approach reduces these limitations by eliminating probabilistic dependence with a deterministic key transformation that generates a counterfeit public key for RSA. Thus, the proposed method achieves a balance between computing efficiency and cryptographic robustness. It effectively rectifies the shortcomings of existing RSA-based hybrid encryption solutions.

Furthermore, other public key cryptography algorithms are selected for implementation with RSA to enhance the security level. The RSA-ECC combination was suggested to reduce attacks by addressing only IFP (Anusuya Devi & Sampradeepraj, 2024). Thus, this technique employs a dual-layer encryption

approach using both IFP and EDLP. Moreover, a combination of RSA and Chebyshev maps, termed RSA-Chebyshev, based on the Chaotic Maps Discrete Logarithm Problem (CMDL), was also introduced (Tahat et al., 2020). However, both methodologies exhibit several challenges concerning intricacy and practical implementation. The RSA-ECC system requires a high computational cost because of elliptic curve arithmetic. Hence, it transforms into an ineffective method for implementation. Therefore, RSA-ECC is unsuitable for lightweight or real-time applications, such as IoT and cloud authentication systems. Furthermore, a significant processing duration is required when RSA-Chebyshev is applied to complete the procedure. Both modular multiplication and modular subtraction are essential for solving Chebyshev polynomials. Thus, the completion of RSA-Chebyshev is significantly inefficient on both the transmitter and recipient sides. The primary objective of the proposed method is to expedite the procedure. Furthermore, the security level is still equal to RSA-ElGamal, because it is also based on both IFP and DLP. The proposed method improves the balance between cryptographic robustness and operational efficiency. Therefore, this technique offers a supplementary logical and efficient alternative to existing hybrid cryptosystems reliant on RSA, such as RSA-ECC and RSA-Chebyshev.

However, RSA-ElGamal is selected as a sample instance of a two-layer encryption algorithm to compare with the proposed method. The selection is based on the alignment of their fundamental security principles with the proposed strategy. By focusing on these two systems, the researchers provide a more direct and substantial comparison that clearly demonstrates the advantages of the proposed method.

As discussed in Section 2, although many algorithms improved from RSA were developed, they continue to face a trade-off between computational efficiency and robustness. Hybrid schemes such as RSA-ElGamal and RSA-ECC significantly strengthen security. However, computational overhead is increased. This unresolved research gap motivates the development of the proposed FE-RSA scheme, which preserves a high security standard based on both IFP and DLP, the same as RSA-ElGamal, while substantially reducing processing time.

3 The Proposed Method

From the problem of RSA and the improvement algorithms based on only IFP, if n is factored by using some factoring algorithms, the attackers can compute Euler's totient function, Φ (n), and exploit the mathematical relationship between the public key, e, and Φ (n) to recover the private key, d. Thus, RSA becomes entirely vulnerable when efficient algorithms for factoring high values of n in polynomial time are established. Furthermore, RSA-ElGamal attempts to improve security by using the mathematical complexities of IFP and DLP. However, this approach requires substantial computing expenses to complete the operation. Subsequently, the processing duration is increased.

This study enhances RSA by increasing the security level in digital signature generation and verification. The aim is to prevent digital signature fraud, regardless of an attacker's ability to factor n. The main idea of the proposed technique, termed FE-RSA, is the application of a fraudulent public key for the validation of digital signatures. This method ensures that the security level is determined by both IFP and DLP. Therefore, this indicates that FE-RSA has an equivalent security level to RSA-ElGamal. Moreover, FE-RSA substantially reduces the processing time required for both signing and verification operations, hence improving its applicability in practical situations. However, the technique is relevant only when the recipients possess the secret key. The fake public key may be derived using equation (1).

$$f = e + x \tag{1}$$

Where, f is the fake public key

x is the secret key

Furthermore, the multiplier, g, calculated using equation (2), is an essential component in the verification process to determine the integrity and authenticity of the digital signature.

$$g = t^{xd} \bmod n \quad (2)$$

Where, n is the modulus

t is the modular inverse of m mod n

Therefore, the original plaintext can be verified by using equation (3). In fact, Theorem 1 shows the reason that this plaintext is always recovered by using this equation.

$$m = s^f g \bmod n \quad (3)$$

Theorem 1: The equation (3) can be selected to recover the original plaintext.

Proof:

From,
$$\begin{aligned} s^f g \bmod n &= (m^d)^f t^{xd} \bmod n \\ &= (m^d)^f (m^{-1})^{xd} \bmod n \\ &= (m^d)^{e+x} m^{-xd} \bmod n \\ &= m^{ed} m^{xd} m^{-xd} \bmod n \end{aligned}$$

Because, $m^{xd} m^{-xd} \bmod n = 1$,

$$\begin{aligned} s^f g \bmod n &= m^{ed} \bmod n \\ &= m \end{aligned}$$

In general, the signer must calculate both s and g before transmitting s, g, and m to the recipient. After the recipient receives these parameters, equation (3) can be chosen to verify the digital signature. If the result equals m, it verifies that s is a valid digital signature. On the other hand, the verifier will invalidate the signature's validity when the result is not equal to m.

From equation (2),

$$\begin{aligned} g &= t^{xd} \bmod n \\ &= (m^{-1})^{xd} \bmod n \\ &= (m^d)^{-x} \bmod n \\ &= s^{-x} \bmod n \end{aligned}$$

Then,

$$g = s^{-x} \bmod n \quad (4)$$

In fact, g must be calculated by using equation (4). However, it must be computed before calculating the multiplier.

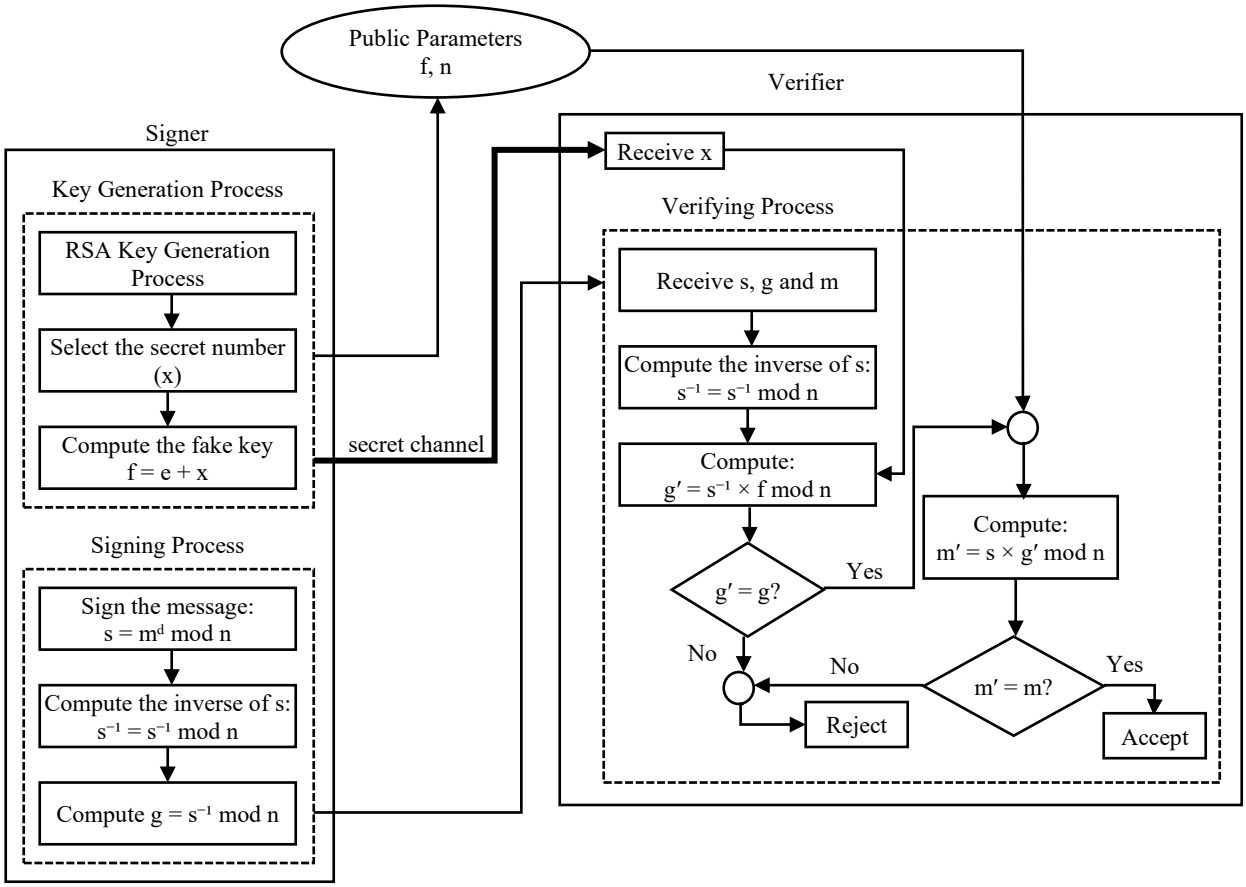

Figure 1: Flowchart of FE-RSA

Figure 1 shows the workflow for FE-RSA. After completing the traditional RSA key generation process, FE-RSA includes an additional procedure to calculate f. The publicly released system parameters therefore, consist of f and n, but the original exponent e remains undisclosed. The signer delivers x to the verifier throughout a secure channel. In the signature generation step, an additional modular exponentiation operation is performed with x as the exponent to find g. In addition, the signer will transmit m, s, and g to the verifier for the verification procedure. In the verification step, the verifier reconstructs the anticipated values using f and authenticates the signature by confirming that the retrieved message m' is congruent to the original message m. The signature is allowed when m' equals m.

Algorithm 1 defines the three phases of FE-RSA: key generation, signing, and verification processes.

Algorithm 1: Pseudo code for FE-RSA

Input: Plaintext (m)

Condition: condition of f should be different from e (gcd (f, Φ (n)) $\neq$ 1)

Key Generation Process

1 Select two prime numbers randomly, p and q

2 n ← compute modulus from p*q

3 Φ (n) ← calculate Euler's totient function from (p – 1) (q – 1)

Choose the public key e from the following conditions: $1 < e < \Phi(n)$, gcd (e,

4 $\Phi(n)) = 1$

5 d ← calculate the private key from $e^{-1} \bmod \Phi(n)$

6 x ← select the secret number

7 f ← calculate the fake key from e + x

Signing Process

8 s ← $m^d \bmod n$

9 s_inv ← $s^{-1} \bmod n$

10 g ← $s\_inv^x \bmod n$

Verifying Process

11 s_inv ← $s^{-1} \bmod n$

12 g' ← $s\_inv^x \bmod n$

13 IF g' equals g

14 m' ← $s^f g' \bmod n$

15 IF m' equals m

16 Confirm the accuracy of the signature

17 Else

18 Reject the signature

19 End IF

20 Else

21 Reject the signature

22 End IF

Algorithm 1 demonstrates the execution of FE-RSA for the generation and verification of digital signatures. In fact, f and n will be revealed to all members of the group, but the remaining parameters must remain confidential. After finishing the signing process, s, g, and m will be sent to the recipient. In the verification process, g must be selected to validate s. Furthermore, the digital signature is generated by using RSA. However, f with an additional multiplier is chosen to verify the digital signature. Therefore, FE-RSA is still a secure algorithm, although n is factored. The reason is that f does not directly relate to d. Therefore, to break FE-RSA, attackers must compute x to recover d. Nevertheless, the standard to generate f must be different from that of e to guarantee security. Specifically, gcd (e, Φ (n)) must not equal 1 to prevent the calculation of d by using basic computations.

Example 1 Implementation of FE-RSA

Key Generation Process

Assume that the key generator or the digital signature signer selects p = 881 and q = 577. This results in n = 508337 and $\Phi(n) = 506880$. If e = 13 is chosen (where gcd (13,506880) = 1), then d = 350917 is derived. Furthermore, if x = 8 is selected, then f = 21.

Public Parameters: $f = 21$ and $n = 508337$

Private Parameters: $p = 881$, $q=577$, $\Phi(n) = 506880$, $e = 13$, and $d=350917$

Signing Process:

Suppose the original message to be signed is $m = 17$. Using RSA's signing equation, s is calculated as follows:

$s = 17^{350917} \bmod 508337 = 10452$

Because $10452^{-1} \bmod 508337 = 98973$

$g = 98973^{8} \bmod 508337 = 220360$

The signer then sends $s = 10452$, $g = 220360$, and $m = 17$ to the recipient.

Verifying Process

After receiving the three parameters (s, g, and m) from the signer, the verifier follows these steps to verify the signature:

Because $10452^{-1} \bmod 508337 = 98973$, $g' = 98973^{8} \bmod 508337 = 220360 = g$

Then, $m' = (10452^{21})\ (220360) \bmod 508337 = 17$

After completing the verification procedure, it is shown that m′ is equal to m, this digital signature is approved.

Moreover, CRT may be chosen to integrate with FE-RSA to reduce the processing time required for signing the digital signature. However, all techniques evaluated in the experimental findings section might equivalently be derived from the application of CRT. Therefore, CRT was not explicitly examined in the research. In actual applications, the integration of CRT should be applied to all RSA-based algorithms to enhance performance. This illustrates that FE-RSA, combined with other RSA variants, may improve efficiency through mathematical optimisations while maintaining its additional security advantage. FE-RSA outperforms hybrid algorithms, which are derived from the combination of two public key cryptography methods, such as RSA and ElGamal. FE-RSA requires only a single encryption and decryption procedure, although it is based on both IFP and DLP. Therefore, the computational complexity is decreased without compromising the security level.

4 Experimental Results

This section discusses the processing time from the experiment. Four algorithms have been chosen for comparison: RSA, RSA-ElGamal, Multi-Prime RSA, and FE-RSA. Three large prime numbers are required to generate n for the implementation of Multi-Prime RSA. Each algorithm in every experiment requires parameters supplied with equal length to ensure fairness in evaluation. The key lengths for each experiment were 512 bits, 1024 bits, and 2048 bits. In addition, each algorithm must be executed 10 times to ensure the reliability of the outcomes. The average values and their corresponding error deviations were recorded. The experiment was split into two parts. The first part involves the signature procedure to evaluate the duration of processing. On the other hand, the processing time to verify the digital signature was analyzed in the second part. Moreover, all experiments were performed under uniform evaluation settings on a system including a 2.53 GHz Intel® Core i5 CPU and 8 GB of RAM. The implementation was executed using the Java language and the BigInteger class (Al Gailani, 2025). In fact, the benefit of using the BigInteger class is about managing large integers without limitation. All

cryptographic parameters were initialized before running each experimental to ensure reproducibility and fairness. For RSA-based algorithms, two prime numbers, p and q, of equal bit-length were randomly generated using the Java BigInteger class. In fact, e was selected such that $1 < e < \Phi$ (n) and gcd (e, Φ (n)) = 1. Furthermore, it is always smaller than the smallest prime factor of n. In addition, d was computed as the modular inverse of e modulo Φ (n). For Multi-Prime RSA, three primes of equal bit-length were generated. In FE-RSA, an additional secret parameter x was randomly selected from the range $1 < x < \Phi$ (n), and the fake public key f was derived according to the proposed scheme. The same key sizes were applied consistently across all algorithms. All random values were regenerated for each independent run.

To ensure consistency of the experimental outcomes, the average signing time, average verification time, and performance enhancement ratio are calculated using equations (5) – (7). For the signing process, the average signing time is defined as follows:

$$T_{sign}^{avg} = \frac{1}{N}\sum_{i=1}^{N} T_{sign}^{(i)} \quad (5)$$

Where, $T_{sign}^{(i)}$ denotes the processing time required to generate the digital signature in round i

In addition, the average verification time is computed as follows:

$$T_{verify}^{avg} = \frac{1}{N}\sum_{i=1}^{N} T_{verify}^{(i)} \quad (6)$$

Where, $T_{verify}^{(i)}$ denotes the processing time required to validate the digital signature in the round i

Furthermore, the percentage improvement is calculated by using the equation (7) to quantify the performance improvement of FE-RSA compared with baseline methods.

$$T = \frac{T_{baseline}^{avg} - T_{FE-RSA}^{avg}}{T_{baseline}^{avg}} x100 \quad (7)$$

Where, T is the percentage improvement $T_{baseline}^{avg}$ and T_{FE-RSA}^{avg} are the average processing times of the baseline algorithm and FE-RSA, respectively.

The dataset selected for the implementation consists of execution time values collected from cryptographic experiments. For each algorithm and key size, the signing and verification processes were executed 10 times under the same system conditions. In addition, n was generated by using the Java BigInteger class by randomly selecting two prime numbers of equal bit-length, and Φ (n) was computed for each key size. These execution time values, measured in milliseconds, constitute the dataset used for the performance analysis.

Performance Evaluation

Figure 2 shows a comparison of the processing time for executing the digital signature. The picture below displays the error deviations represented by small error bars. It clearly exhibits little variation over several tests and confirms the dependability of the experimental findings. Experimental results indicate that RSA is the most rapid algorithm. However, it is the least secure algorithm due to its reliance only on IFP. With the three alternative algorithms enhancing security, FE-RSA has the highest processing speed for digital signature generation. Although FE-RSA provides a similar level of security to RSA-ElGamal, it regularly demonstrates superior performance relative to the compared technique. For key sizes of 512 bits, 1024 bits, and 2048 bits, FE-RSA generates digital signatures approximately 90%, 35%, and 5% faster than RSA-ElGamal, respectively, on average. The decreasing performance is mostly attributed to the fixed size of parameter b, reducing the variability in ElGamal's processing time

across various key sizes. On the other hand, the parameters e and x in RSA are based on the number of prime factors of n. Therefore, processing time may be increased when the bit length is expanded. This pattern indicates that while FE-RSA's advantage decreases with increased key sizes, it continues to have a reliable superiority over other RSA-based variations regarding efficiency.

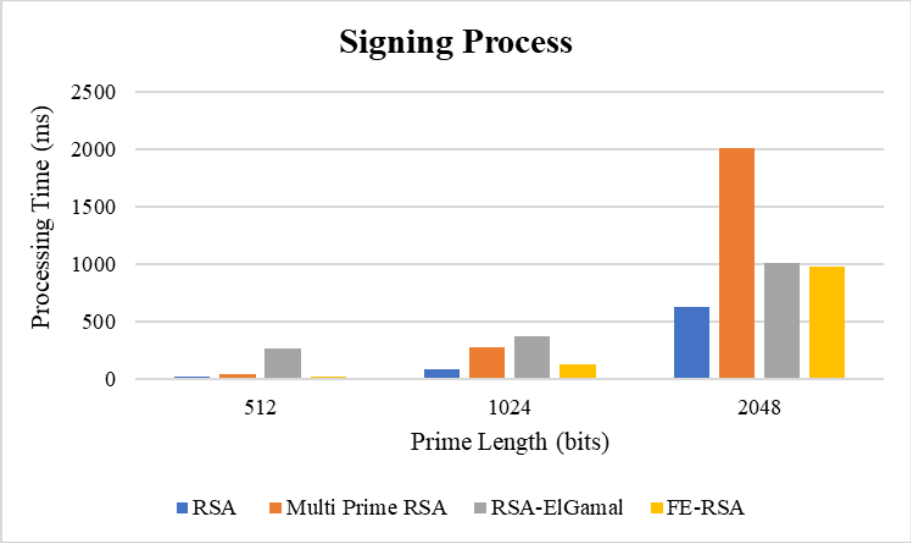

Figure 2: Comparison of processing times for digital signature generation

In addition to the visual trends illustrated in Figure 2, Table 1 reports the corresponding results expressed as mean ± standard deviation over 10 repeated trials. In fact, these values provide quantitative evidence to support the observed performance differences among the compared algorithms and confirm the reliability of the experimental results.

Table 1: Average execution time in signing process (milli seconds) for each algorithm under different key sizes (mean ± sd, n = 10)

Algorithm	512-bit	1024-bit	2048-bit
RSA	14.36±2.575	86.05±2.794	625.79±3.878
Multi Prime RSA	36.16±2.351	274.90±3.027	2017.93±3.569
RSA-ElGamal	265.65±2.795	366.69±2.294	1015.11±2.894
FE-RSA	23.76±2.250	125.79±3.036	976.08±2.244

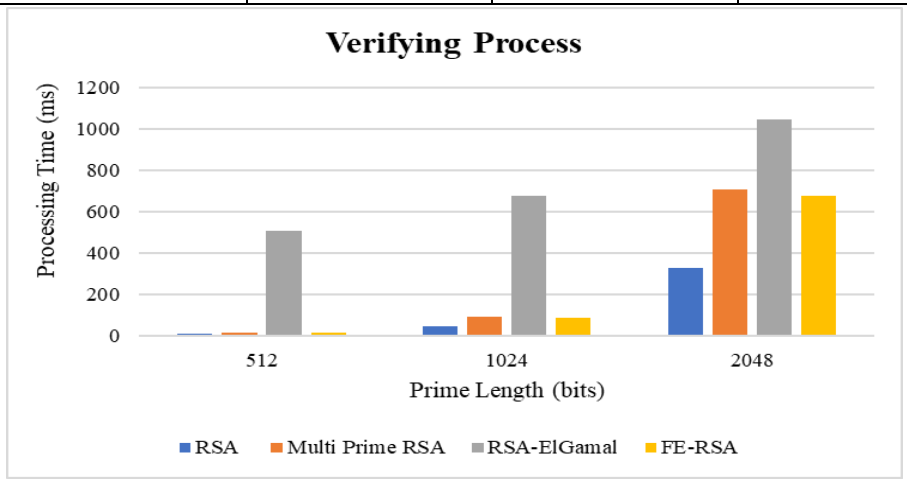

Figure 3: Comparison of processing times for digital signature verification

Table 2: Average execution time in verifying process (milli seconds) for each algorithm under different key sizes (mean ± sd, n = 10)

Algorithm	512-bit	1024-bit	2048-bit
RSA	7.14±0.84	44.56±2.84	326.16±2.10
Multi Prime RSA	14.93±2.55	94.13±3.13	706.43±2.64
RSA-ElGamal	507.52±2.45	675.33±1.88	1046.39±64.66
FE-RSA	16.00±2.28	84.80±2.29	677.37±4.70

Figure 3 demonstrates a comparison of the processing time required for the verification. The results show that RSA achieves the highest verification speed. Among the three methods developed from RSA, FE-RSA has the most efficiency in this process. Significantly, FE-RSA requires less time in the verification procedure than for signature generation. The gap exists because, in many cases, d is larger than e, resulting in a greater computational expense for modular exponentiation during the signing process compared to verification. The experimental results show that the processing time in verifying the process is reduced by around 90%, 80%, and 35% for key sizes of 512 bits, 1024 bits, and 2048 bits, respectively, when FE-RSA is compared with RSA-ElGamal. Furthermore, the proposed method is still faster than RSA-ElGamal in the signing process. However, the most significant time savings occur during the verification phase. Although the security level of FS-RSA is equal to RSA-ElGamal, there are some limitations for implementing FE-RSA. In fact, FE-RSA is especially appropriate for the condition that only the specified verifiers who have the secret key can validate the signature's validity. In situations necessitating public verification (such as any person holding the public key may authenticate the signature), the security of FE-RSA is exclusively based on the challenge of IFP. This occurs because verifiers in certain contexts are unable to authenticate the parameter g, hence diminishing the overall security guarantees of the procedure.

In addition, Table 2 confirms the trend shown in Figure 3 by reporting stable mean values with relatively small standard deviations. The results indicate that FE-RSA achieves lower verification time than RSA-ElGamal across all key sizes. Therefore, the observed performance gains are reliable.

Furthermore, FE-RSA might be efficiently combined with CRT to speed up the process. In fact, FE-RSA uses d to be the exponent to compute the modular exponentiation, md mod n. Therefore, CRT can be applied to this method for signing. The integration of FE-RSA with CRT allows the partitioning of modular calculations based on prime factors. Then, it enhances scalability and substantially decreases process time without affecting security. Moreover, while FE-RSA maintains the basic modular arithmetic framework of RSA, other optimisation methods, such as Montgomery Multiplication, may be chosen to improve the performance of modular multiplication in similar ways. Thus, the advantage of FE-RSA is that the processing time is decreased, but the security level is still high. In depth, the security level of FE-RSA is comparable to that of RSA-ElGamal, because both are based on IFP and DLP.

Security Analysis

This section provides a comparative study of RSA, RSA-ElGamal, and FE-RSA, regarding security issues. Moreover, an evaluation of the contribution of each fundamental component of the proposed approach, including the counterfeit public key, the multiplier, and the secret parameter, is also analyzed. The ablation research involves systematically deleting specific components to evaluate their impact on the accuracy of the signature verification procedure. The findings provide conceptual evidence that the security attributes of FE-RSA are attained through the collective functioning of all components, rather than being sourced from any individual parameter.

Table 3: Comparison of security properties among RSA, RSA-ElGamal, and FE-RSA

Algorithm	Based on IFP	Based on DLP	Forgery Resistant when n is Factored	Requires Verifier Secret	Public Verifiability
RSA	Yes	No	No	No	Yes
RSA-ElGamal	Yes	Yes	Yes	No	Yes
FE-RSA	Yes	Yes	Yes	Yes	No

As shown in Table 3, RSA is based on only IFP. Therefore, this algorithm cannot resist signature forgery when the modulus is factored. In contrast, both RSA-ElGamal and FE-RSA are based on the combined hardness of the IFP and DLP. However, FE-RSA differs from the other schemes in that its verification process requires a secret parameter held by the verifier, and thus does not support public verification. This limitation is compensated for by its higher structural security compared with RSA, while still maintaining better efficiency than hybrid approaches such as RSA-ElGamal.

Table 4: Structural impact analysis of core components in FE-RSA

Variant Description	Fake Public Key (f)	Multiplier (g)	Secret Parameter (x)	Signature Verification Result
Full Parameter	Yes	Yes	Yes	Pass
Removing f	No	Yes	Yes	Fail
Removing g	Yes	No	Yes	Fail
Removing x	Yes	Yes	No	Fail

Table 4 presents the structural analysis of FE-RSA. When all parameters are activated, the proposed algorithm can achieve signature verification. In contrast, disabling any of the fake public key, the multiplier, or the secret parameter immediately causes the verification process to fail. These results indicate that the security properties of FE-RSA arise from the joint operation of all components rather than from any single parameter alone.

Security and Speed Analyzing

In this section, the security and time to implement FE-RSA are evaluated. In fact, the security study is divided into two cases.

Case 1: If f is selected and gcd (f, Φ (n)) = 1, it is simple to calculate df by using the equation df = f-1 mod Φ (n). In addition, if an opponent can effectively factor n, they can certainly discover df.

For example, assuming the adversary aims to generate a signature for a message mf that the key holder refuses to support, they can proceed as follows: The first step is the selection of a value y randomly. Next, it is the process to calculate sf = $m_f^{d_f+y}$ mod n and gf = $(m_f^{-1})^{y_f}$ mod n. Then, the key holder will transmit sf, gf, and mf to the verifier. After completing the verification procedure, mf is always retrieved as follows:

$$s_f^f g_f \bmod n = (m_f^{d_f+y})^f (m_f^{-1})^{y_f} \bmod n$$

$$= m_f^{fd_f} m_f^{yf} m_f^{-yf} \bmod n$$

$$= m_f^{fd_f} \bmod n$$

$$= m_f$$

The findings indicate that the result corresponds to the original message before signing. However, if the signature verifier evaluates the multiplier, it will be seen that g = (s_f^{-1})x mod n is not equivalent to (m_f^{-1})yf mod n. This inconsistency verifies that the signature has been faked by an attacker.

Case 2: assume f is selected with the result that it fails to meet the condition gcd (f, Φ (n)) = 1. This implies that it is infeasible to calculate f-1 mod Φ (n), therefore preventing an adversary from deriving df. If the adversary can factor n, they must randomly choose two integers, a and b, such that f = a + b and gcd(a, Φ (n)) = 1, to compute da = a-1 mod Φ (n).

Suppose the adversary plans to generate a signature for m_f, a message that the key holder does not intend to sign and transmit it to the verifier. The adversary might proceed as follows: the first step is the computation of $s_f = m_f^{d_a - b}$ mod n, and then compute $g_1 = m_f^{-bd_a}$ mod n and $g_2 = m_f^{bf}$ mod n to calculate $g_f = {}^{g_1} g_2$ mod n. Finally, the adversary submits s_f, g_f, and m_f to the verifier. Based on the signature verification method, the resultant output may be obtained as follows:

$$
\begin{aligned}
s_f^f g_f \, mod \, n &= (m_f^{d_a - b})^f g_1 g_2 \, mod \, n \\
&= m_f^{fd_a} m_f^{-bf} m_f^{-bd_a} m_f^{bf} \, mod \, n \\
&= m_f^{fd_a} m_f^{-bd_a} \, mod \, n \\
&= m_f^{(a+b)d_a} m_f^{-bd_a} \, mod \, n \\
&= m_f^{ad_a} m_f^{bd_a} m_f^{-bd_a} \, mod \, n \\
&= m_f
\end{aligned}
$$

The findings demonstrate that the result corresponds with the original message before being signed. However, if the verifier checks the multipliers, it will be observed that g = (s_f^{-1})x mod n does not equal g1g2 mod n. This verifies that the signature was falsified by an opponent. Therefore, in both situations, even if the adversary can generate s, they cannot produce g. To determine the value of g, it is essential to calculate x, which is difficult. Thus, it can be believed that FE-RSA offers double levels of security: the challenge of factoring large numbers and the complexity of solving the DLP.

Table 5 presents a comparison of processing time and system security among RSA, Multi-Prime RSA, RSA-ElGamal, and FE-RSA. Although FE-RSA requires more processing time than RSA, it provides enhanced security. This enhanced security is caused by a requirement to address two difficult problems: IFP and DLP.

Table 5: Comparison of processing time and system security

Algorithm	Signing Process	Verifying Process	Security	Memory Usage	Energy Consumption
RSA	Calculate a singular modular exponentiation using *d* as the exponent.	Calculate a singular modular exponentiation using *e* as the exponent.	It is based on only the factoring problem.	Low (the storage is required for two prime factors of *n*)	Low (The process is based on a single modular computation
Multi-Prime RSA	Calculate a singular modular exponentiation using *d* as the exponent. Nevertheless, if *e* is a constant value, it is highly possible that *d* in Multi-Prime RSA is larger than the comparable value used in traditional RSA. Therefore, the processing time in Multi-Prime RSA is slower than that of RSA.	Calculate a single modular exponentiation using *e* as the exponent. However, the larger value of *n* in Multi-Prime RSA, compared to the corresponding *n* in RSA, requires more processing time for Multi-Prime RSA.	Two steps of prime factoring are required, because there are three prime numbers.	Moderate, additional primes slightly increase key storage	High, due to additional modular exponentiation steps
RSA - ElGamal	The procedure involves calculating two modular exponentiations: the first equation employs *d* as the exponent, while the next one exploits *b* as the exponent. Nonetheless, each execution of ElGamal requires the generation of a new value for *b*, leading to increased processing time relative to FE-RSA.	The procedure requires the calculation of three modular exponentiations, using significantly greater processing resources than the other methods in the comparison.	It is based on the computing challenge of prime factorization and the complexity of resolving the discrete logarithm problem.	High, must handle large temporary values during ElGamal operations	High (Double encryption algorithm increases computational energy
FE-RSA	The calculation requires two modular exponentiations, where the first equation uses *d* as the exponent and the second equation uses *x* as the exponent. Therefore, FE-RSA requires superior computing resources in comparison to RSA. However, it becomes the fastest algorithm when compared with the other modified algorithms chosen in this paper.	The calculation involves two modular exponentiations, with the first equation using f as the exponent and the next one employing *x* as the exponent. Therefore, FE-RSA requires more computing resources than RSA. However, it becomes the fastest algorithm when compared with the other modified algorithms chosen in this paper.	It is based on the computing challenge of prime factorization and the complexity of resolving the discrete logarithm problem.	Moderate, slightly higher than RSA, lower than RSA-ElGamal	Moderate, more efficient than hybrid methods due to deterministic key mapping

5 Conclusion

In this study, FE-RSA, an improvement of RSA, was proposed to enhance the security and efficiency of digital signatures by addressing the Integer Factorization Problem (IFP) and the Discrete Logarithm Problem (DLP). The bit lengths for the experiment are 512-bit, 1024-bit, and 2048-bit. In addition, RSA, Multi-Prime RSA, and RSA-ElGamal are selected to compare with FE-RSA. In fact, the proposed method combines a secret key with a fake public key to prevent digital signature forgery. Then, FE-RSA is not broken, although the modulus is factored. Therefore, FE-RSA represents a significant advancement in public key cryptography by integrating computational efficiency between RSA and ElGamal, but the process time is not increased.

According to the results shown in Figures 2 and 3, FE-RSA presents efficiency improvements over RSA-ElGamal, which is selected to be the main baseline algorithm in both signing and verification operations. In the signature process, FE-RSA provides speed improvements about 92.47%, 64.50%, and 3.82% for key sizes of 512-bit, 1024-bit, and 2048-bit, respectively. In the verification phase, FE-RSA decreases processing time by roughly 97.12%, 87.53%, and 34.33% for identical key sizes. The analytical results demonstrate that the most substantial performance improvement of FE-RSA appears in the verification phase, which is essential for real-time authentication contexts.

In addition, the experimental results show that FE-RSA has high performance for both security levels and processing time. Therefore, it is suitable for applications where both speed and security are critical. Future work may extend this approach by evaluating its applicability in large-scale distributed systems, resource-constrained devices, or post-quantum settings to further establish its practical value. In addition, FE-RSA is suitable to secure the secret information in Internet-based situations, including cloud authentication systems and e-government platforms. This integration improves the dependability and integrity of data in digital services and distributed systems.

Acknowledgement

This Work was Supported by Udon Thani Rajabhat University Research Fund.

References

[1] Adeniyi, E. A., Falola, P. B., Maashi, M. S., Aljebreen, M., & Bharany, S. (2022). Secure sensitive data sharing using RSA and ElGamal cryptographic algorithms with hash functions. *Information*, *13*(10), 442. https://doi.org/10.3390/info13100442

[2] Al Gailani, M. F. (2025). Hardware design and implementation of secure hash algorithm based on FPGA. *Journal of Internet Services and Information Security*, *15*(2), 209226.

[3] Anusuya Devi, V., & Sampradeepraj, T. (2024). End-to-end self-organizing intelligent security model for wireless sensor network based on a hybrid (AES–RSA) cryptography. *Wireless Personal Communications*, *136*(3), 1675-1703. https://doi.org/10.1007/s11277-024-11353-3

[4] Awad, Y., Jomaa, D., Alkhezi, Y., & Hindi, R. (2025). A New Approach Combining Rsa and Elgamal Algorithms: Advancements in Encryption and Digital Signatures Using Gaussian Integers. *Jordanian Journal of Computers & Information Technology*, *11*(1).

[5] Banu, Y., Rath, B. K., & Gountia, D. (2025). Analyzing cryptographic algorithm efficiency with in graph-based encryption models. *Frontiers in Computer Science*, *7*, 1630222. https://doi.org/10.3389/fcomp.2025.1630222

[6] Chattopadhyay, A. K., Nag, A., & Singh, J. P. (2022). An efficient verifiable (t, n)-threshold secret image sharing scheme with ultralight shares. *Multimedia Tools and Applications*, *81*(24), 34969-34999. https://doi.org/10.1007/s11042-021-10523-w

[7] Chen, Z., & Ye, G. (2022). An asymmetric image encryption scheme based on hash SHA-3, RSA and compressive sensing. *Optik*, *267*, 169676. https://doi.org/10.1016/j.ijleo.2022.169676

[8] Dam, D. T., Tran, T. H., Hoang, V. P., Pham, C. K., & Hoang, T. T. (2023). A survey of post-quantum cryptography: Start of a new race. *Cryptography*, *7*(3), 40. https://doi.org/10.3390/cryptography7030040

[9] Flores-Carapia, R., Silva-García, V. M., Cardona-López, M. A., & Villarreal-Cervantes, M. G. (2025). A chaotic digital signature algorithm based on a dynamic substitution box. *Scientific Reports*, *15*(1), 2435. https://doi.org/10.1038/s41598-024-83943-x

[10] Gulen, U., & Baktir, S. (2023). Side-channel resistant 2048-bit RSA implementation for wireless sensor networks and internet of things. *IEEE Access*, *11*, 39531-39543. https://doi.org/10.1109/ACCESS.2023.3268642

[11] Halak, B., Yilmaz, Y., & Shiu, D. (2022). Comparative analysis of energy costs of asymmetric vs symmetric encryption-based security applications. *Ieee Access*, *10*, 76707-76719. https://doi.org/10.1109/ACCESS.2022.3192970

[12] Han, J., & Zhuang, J. (2022). DLP in semigroups: algorithms and lower bounds. *Journal of Mathematical Cryptology*, *16*(1), 278-288.

[13] Hashim, H. R. (2025). An Efficient ElGamal Cryptosystem Based on Prime Power Moduli and Modular Key Exchange. *Journal of Prime Research in Mathematics*, *21*(2), 37-50.

[14] Joseph, D., Misoczki, R., Manzano, M., Tricot, J., Pinuaga, F. D., Lacombe, O., ... & Hansen, R. (2022). Transitioning organizations to post-quantum cryptography. *Nature*, *605*(7909), 237-243. https://doi.org/10.1038/s41586-022-04623-2

[15] Kaliyamoorthy, P., & Ramalingam, A. C. (2022). QMLFD based RSA cryptosystem for enhancing data security in public cloud storage system. *Wireless Personal Communications*, *122*(1), 755-782. https://doi.org/10.1007/s11277-021-08924-z

[16] Khudhair, A. A. T., Maolood, A. T., & Gbashi, E. K. (2024, September). Symmetric keys for lightweight encryption algorithms using a pre–trained vgg16 model. In *Telecom* (Vol. 5, No. 3, pp. 892-906). MDPI. https://doi.org/10.3390/telecom5030044

[17] Kumar, S., & Sharma, D. (2024). A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm. *Artificial Intelligence Review*, *57*(4), 87. https://doi.org/10.1007/s10462-024-10719-0

[18] Ochoa-Jiménez, E., Rivera-Zamarripa, L., Cruz-Cortés, N., & Rodríguez-Henríquez, F. (2020). Implementation of RSA signatures on GPU and CPU architectures. *IEEE Access*, *8*, 9928-9941. https://doi.org/10.1109/ACCESS.2019.2963826

[19] Pathirage, T. D., Wijewardana, H. P. D. K., Lakshan, L. A. S., Hydher, H., & Yasakethu, L. (2021, August). Multi-prime RSA verilog implementation using 4-primes. In *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)* (pp. 60-65). IEEE. https://doi.org/10.1109/ICIAfS52090.2021.9605975

[20] Qiang, L. (2025). Research on performance optimization and resource allocation strategy of network node encryption based on RSA algorithm. *Journal of Cyber Security and Mobility*, *14*(1), 101-125. https://doi.org/10.13052/jcsm2245-1439.1415

[21] Sasikumar, K., & Nagarajan, S. (2024). Comprehensive review and analysis of cryptography techniques in cloud computing. *IEEE Access*, *12*, 52325-52351. https://doi.org/10.1109/ACCESS.2024.3385449

[22] Sethukarasi, T., Hemavathi, D., Swetha, S., & Samundeswari, S. (2025). RSO-MRSA: rat swarm optimization based modified Rivest–Shamir–Adleman for secure and efficient healthcare monitoring system. *Wireless Networks*, *31*(2), 1129-1143. https://doi.org/10.1007/s11276-024-03807-0

[23] Shatnawi, A. S., Almazari, M. M., AlShara, Z., Taqieddin, E., & Mustafa, D. (2023). RSA cryptanalysis—Fermat factorization exact bound and the role of integer sequences in factorization problem. *Journal of Information Security and Applications*, *78*, 103614. https://doi.org/10.1016/j.jisa.2023.103614

[24] Shivaramakrishna, D., & Nagaratna, M. (2023). A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management and Time-Limited access control. *Alexandria Engineering Journal*, *84*, 275-284. https://doi.org/10.1016/j.aej.2023.10.054

[25] Somsuk, K. (2021). The improvement of elliptic curve factorization method to recover rsa's prime factors. *Symmetry*, *13*(8), 1314. https://doi.org/10.3390/sym13081314

[26] Somsuk, K. (2022). An efficient variant of Pollard'sp− 1 for the case that all prime factors of the p− 1 in B-Smooth. *Symmetry*, *14*(2), 312. https://doi.org/10.3390/sym14020312

[27] Srivastava, A., & Gupta, J. (2024). Attack resistant blockchain-based healthcare record system using modified RSA Algorithm. *International Journal of Information Technology*, *16*(1), 417-424. https://doi.org/10.1007/s41870-023-01588-x

[28] Tahat, N., Tahat, A. A., Abu-Dalu, M., Albadarneh, R. B., Abdallah, A. E., & Al-Hazaimeh, O. M. (2020). A new RSA public key encryption scheme with chaotic maps. *International Journal of electrical and computer engineering*, *10*(2), 1430-1437. https://doi.org/10.11591/ijece.v10i2.pp1430-1437

[29] Ullah, S., Zheng, J., Din, N., Hussain, M. T., Ullah, F., & Yousaf, M. (2023). Elliptic Curve Cryptography; Applications, challenges, recent advances, and future trends: A comprehensive survey. *Computer Science Review*, *47*, 100530. https://doi.org/10.1016/j.cosrev.2022.100530

[30] Xiao, H., Yu, S., Cheng, B., & Liu, G. (2022). FPGA-based high-throughput Montgomery modular multipliers for RSA cryptosystems. *IEICE Electronics Express*, *19*(9), 20220101-20220101.

[31] Zhang, R., Bi, J., Li, L., & Peng, H. (2025). An optimal bound for factoring unbalanced RSA moduli by solving Generalized Implicit Factorization Problem: R. Zhang et al. *The Journal of Supercomputing*, *81*(1), 102. https://doi.org/10.1007/s11227-024-06478-y

[32] Zhao, C., Cao, J., Zhang, J., & Cheng, Q. (2025). Fault attacks on multi-prime RSA signatures. *Designs, Codes and Cryptography*, *93*(5), 1357-1374. https://doi.org/10.1007/s10623-024-01554-z

Authors Biography

Thanapat Chiawchanwattana was born in Udorn Thani, Thailand. He received the B.Eng. (Computer Engineering) from Khon Kaen University in 1999, M.Eng. in Computer Engineering from Khon Kaen University (KKU) in 2006, Thailand. He is currently pursuing a Ph.D. in Electrical and Computer Engineering at the Faculty of Engineering, Mahasarakham University, Thailand. His research interests include IoT/Automation systems, Cryptography, and RF/Microwave Circuits design. Currently, He is a lecturer at the Department of Computer and Communication Engineering at Udon Thani Rajabhat University. He can be contacted at email: thanapat.ch@udru.ac.th

Kritsanapong Somsuk is an associate professor at the Department of Computer and Communication Engineering, Faculty of Technology and Engineering, Udon Thani Rajabhat University, Udon Thani, Thailand. He obtained his M.Eng. (Computer Engineering) from the Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, M.Sc. (Computer Science) from the Department of Computer Science, Faculty of Science, Khon Kaen University, and his Ph.D. (Computer Engineering) from the Department of Computer Engineering, Faculty of Engineering, Khon Kaen University. The area of research interests includes computer security, cryptography, and integer factorization algorithms. He can be contacted at email: kritsanapong@udru.ac.th.