symmetry

MDPI

Article

An Improved Public Key Cryptographic Algorithm Based on Chebyshev Polynomials and RSA

Chunfu Zhang [1,2], Yanchun Liang [3,*], Adriano Tavares [2,*], Lidong Wang [1,*], Tiago Gomes [2] and Sandro Pinto [2]

[1] School of Statistics and Data Science, Zhuhai College of Science and Technology, Zhuhai 519041, China; cfzhang@zcst.edu.cn
[2] Department of Industrial Electronics, University of Minho, 4800-058 Guimaraes, Portugal; mr.gomes@dei.uminho.pt (T.G.); sandro.pinto@dei.uminho.pt (S.P.)
[3] School of Computer Science, Zhuhai College of Science and Technology, Zhuhai 519041, China
[*] Correspondence: ycliang@jlu.edu.cn (Y.L.); atavares@dei.uminho.pt (A.T.); wld@zcst.edu.cn (L.W.)

Abstract: Due to its very desirable properties, Chebyshev polynomials are often used in the design of public key cryptographic systems. This paper discretizes the Chebyshev mapping, generalizes the properties of Chebyshev polynomials, and proposes an improved public key encryption algorithm based on Chebyshev chaotic mapping and RSA, i.e., $CRPKC - K_i$. This algorithm introduces alternative multiplication coefficients K_i, the selection of which is determined by the size of $T_r(T_d(x)) mod\ N = T_d(T_r(x)) mod\ N$, and the specific value selection rules are shared secrets among participants, overcoming the shortcomings of previous schemes. In the key generation and encryption/decryption stages, more complex intermediate processes are used to achieve higher algorithm complexity, making the algorithm more robust against ordinary attacks. The algorithm is also compared with other RSA-based algorithms to demonstrate its effectiveness in terms of performance and security.

Keywords: public-key cryptosystem; Chebyshev polynomials; RSA; alternative multiplication coefficients; semi-group property

check for updates

Citation: Zhang, C.; Liang, Y.; Tavares, A.; Wang, L.; Gomes, T.; Pinto, S. An Improved Public Key Cryptographic Algorithm Based on Chebyshev Polynomials and RSA. *Symmetry* **2024**, *16*, 263. https://doi.org/10.3390/sym16030263

Academic Editor: Aviv Gibali

Received: 26 January 2024
Revised: 16 February 2024
Accepted: 18 February 2024
Published: 21 February 2024

1. Introduction

At its essence, cryptography design and analysis is a mathematical technique for ensuring secure communication over insecure channels, closely related to computer technology and electronic communication technology. Although traditional symmetric encryption systems are highly efficient, the security of the ciphertext is entirely dependent on the secrecy of the key in the process of key distribution and management, and once the key is leaked, the confidentiality is lost [1]. When the sending and receiving ends are far apart and the key needs to be changed frequently, it is difficult to pass the key between each other. The concept of public-key cryptography was first proposed by Diffie and Hellman in their 1976 paper "New Directions in Cryptography" [2], which addresses the confidentiality problems in symmetric encryption systems, particularly in multi-user communication networks.

In 1977, Rivest, Shamir, and Adleman proposed the first relatively complete public-key cryptographic algorithm, the RSA algorithm [3]. Since then, a large number of public-key cryptographic algorithms have been proposed based on different computational problems, such as the Mer–Hellman knapsack algorithm, McEliece algorithm, ElGamal algorithm, elliptic curve cryptography, lattice-based cryptography, and password-based public-key cryptography.

The RSA public-key cryptographic system is still recognized as a well-performing cryptographic system. It is the first algorithm that can be used for both digital encryption and digital signatures. Its security is based on integer factorization, and the construction of the system is based on Euler's theorem, which has an extremely important position in modern cryptography. With the rapid development of computer technology, many encryption methods for RSA have become relatively easy to crack [4–8]. Traditional public-key cryptographic algorithms are constantly facing various challenges, so it is necessary to study new practical public-key cryptographic algorithms as supplements or replacements for current public-key cryptographic algorithms.

According to relevant methods, the author conducted a systematic search for research articles on standard or modified RSA algorithms and their various applications, and a total of approximately 84 related articles were found [9]. These articles were then divided into several domains or categories, such as cloud security [10,11], image cryptography [12,13], wireless security, and others. In recent years, lightweight cryptographic systems using hybrid variants of RSA have been proposed, showing effective results in smart devices and IoT devices [14,15]. Based on the literature [9], we also analyzed the literature from the past two years [16,17]. Figure 1 categorizes the progress of all RSA variants based on different categories of publications each year. We can see that enhancements to the RSA algorithm have always been a popular research direction, which is also the goal pursued in this paper.

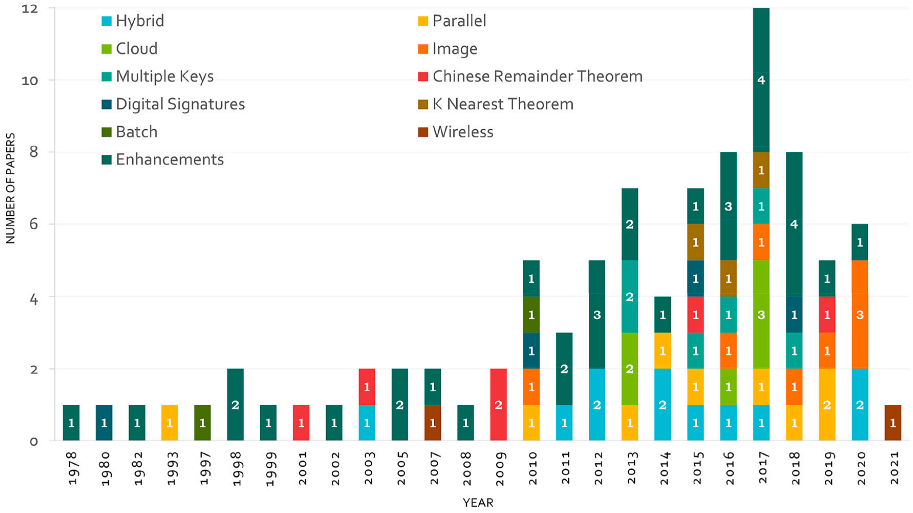


Figure 1. Yearly trend of all modified variants related to RSA algorithm [9].

In Table 1, we focus on introducing popular RSA algorithms that are relevant to this research and have improved the performance and security of RSA to varying degrees. Some of these methods are borrowed and compared with the proposed method, while others are evaluated theoretically.

Table 1. Algorithm analysis-and-evaluation of the recent related enhanced models.

Algorithm Model	Total Modulus	Intermediate Variables as of Standard RSA	Encryption Scheme	Key Generation	Decryption Scheme	Shortcomings/ Comments
Rivest, R.L. et al. [3]	1, i.e., N	Not Any	$C = M^e modN$	$ed = 1mod\varphi(N)$	$M = C^d modN$	Security is not efficient
Minni et al. [18]	1, i.e., N	$X(\leftrightarrow N)$	$C = M^e modX$	$ed = 1mod\varphi(X)$	$M = C^d modX$	Higher time Complexity [19]
Thangavel et al. [19]	3, i.e., n,m, N	$e_1, e_2, e = {e_1}^{e_2}$	$C = M^e modN$	$ed = 1mod(\varphi(N) \cdot e_1)$	$M = C^d modN$	Same security level as RSA [20]
Mathur et al. [21]	1, i.e., N, L → N	$P, Q, O, N, s.t., Q = P \cdot e$	$C = M^e modN$	$ed = 1mod\varphi(N)$	$M = C^d modN$	Slower, i.e., higher time [22]
Panda and Chattopadhyay [23]	1, i.e., N	P_1, W	$C = M^e modW$	$ed = 1mod(\varphi(N) \cdot P_1)$	$M = C^d modW$	Security is not efficient [24]
Raza Imam et al. [17]	3, i.e., N	$e_1, e_2 = e\ XOR\ N$	$C = M^{e_2} modN$	$e_1 d_1 = 1mod\varphi(N)$	$M = C^{d_2} modN$	Lack of proof

The natural relationship between the basic properties of chaos transformations, such as the mixture, sensitivity to parameters and initial values, and cryptography were mentioned in Shannon's classic paper [25]. Research on chaotic public-key cryptography is still relatively weak compared to chaotic sequences and block cipher systems, and there are few practical and efficient encryption schemes based on chaos. Paying attention to the good properties of chaotic mappings and traditional cryptology criteria is still a new research field [26]. In [27], L. Kocarov and Z. Tasev proposed a public-key encryption method based on Chebyshev polynomials. Unfortunately, P. Bergamo pointed out its insecurity and inefficiency [28]. Zhao et al. proposed a new key agreement scheme based on chaotic mappings using the traditional RSA algorithm and discrete logarithm problems over finite fields. The scheme is based on the values of Chebyshev polynomials over finite fields, thereby avoiding various active attacks in the past and ensuring key agreement security. At the same time, the key agreement scheme can also achieve identity authentication functions [29]. Benasser et al. proposed an identity-based encryption scheme based on Chebyshev polynomials, but it has security flaws. Wang et al. proposed a corresponding identity authentication scheme, and the key exchange protocol based on Chebyshev polynomials is gradually being improved and perfected [30].

In recent years, there have been some new research advances in chaotic public-key cryptography [31–34]. In [35], Hsiao proposed an RSA encryption algorithm based on chaotic synchronization. However, RSA must use at least 1024 bits of digits to be considered secure under today's computational capabilities. Ruzai, W.N.A. proposed a solution by giving examples to illustrate that near-square prime numbers could potentially lead to RSA being completely broken in [36]. Lawnik proposed two new cryptographic systems based on modified Chebyshev polynomials in the paper [37]. Gupta proposed a new lightweight image encryption algorithm based on session key using Chebyshev chaotic mapping and hybrid mix [38]. Patgiri proposed future solutions for RSA on traditional computers in the paper [39]. Ryu proposed an improved and secure Chebyshev chaotic mapping user authentication scheme in the paper [40].

This article discretizes the Chebyshev mapping, promotes the properties of Chebyshev polynomials, and designs an improved public key encryption algorithm based on the Chebyshev chaotic mapping and RSA, i.e., $CRPKC - K_i$. This algorithm overcomes the shortcomings of previous schemes and provides higher security. The analysis and experimental results show that this algorithm has strong robustness against ordinary attacks.

The article is organized as follows. Section 2 is the foundation of the rest of the article, as it includes the description of public key encryption algorithm, the extensive properties of Chebyshev mapping, and the Chebyshev-RSA public key cryptosystem. In Section 3, we propose an improved public key encryption algorithm based on Chebyshev polynomials and RSA. Section 4 describes the software implementation and provides examples. Section 5 presents some performance analyses. Finally, the conclusion and future scope is drawn in the last section.

2. Preliminaries

2.1. The General Public Key Cryptography Algorithm Can Be Described as Follows [1]

1. Each user generates a key pair $k = (k_d, k_e)$, where k_d represents the private key and k_e represents the public key. The public key cryptography algorithm requires that k_d can be derived theoretically from k_e, but in practice, it is not feasible due to the large computational complexity.
2. The information sender encrypts the plaintext p using the publicly available key k_e of the information receiver: $E(p, k_e) = C$, where p represents the plaintext and C represents the encrypted ciphertext.
3. The information receiver decrypts using their privately held secret key k_d: $D(C, k_d) = p$.

2.2. Chebyshev Polynomials and Their Properties

Let $C[-1, 1]$ be a vector space consisting of all continuous real-valued functions on $[-1, 1]$, then $\{cos(ncos^{-1}(x))\}_{n=0}^{n=\infty}$ is a set of bases on $C[-1, 1]$. Let

$$T_n = cos(ncos^{-1}(x)), \tag{1}$$

then T_n is called the n order Chebyshev polynomial of the first kind. It has the following properties:

- First,

$$T_0 = 1,\ T_1 = x,\ \text{and}\ T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x); \tag{2}$$

- Second, define the mapping $\rho_n : x \rightarrow T_n$ on $[-1,1]$, then the following semi-group property holds:

$$\rho_{mn}(x) = \rho_m(\rho_n(x)) = \rho_n(\rho_m(x)). \tag{3}$$

To clearly state the design idea of the chaotic key scheme, we need to extend its properties.

Theorem 1. *The semi-group property of Chebyshev polynomials holds on the interval* $(-\infty, +\infty)$;

Theorem 2. *If* x *is a positive integer and* p *is an odd prime number, then*

$$T_p(x) \equiv x(modp); \tag{4}$$

Theorem 3. *Let* p *and* q *be relatively prime, and* x *and* e *be positive integers. If* $T_e(x) \equiv x(modp)$ *and* $T_e(x) \equiv x(modq)$, *then*

$$T_e(x) \equiv x(modpq). \tag{5}$$

Proof. Since $(p,q) = 1$, there exist integers s and t such that $sp + tq = 1$, therefore, $T_e(x) = spT_e(x) + tqT_e(x) \equiv spx + tqx \equiv x(modpq)$. □

Corollary 1. *If* $T_\varphi(x) \equiv 1$ *for all* x *and there exist positive integers* e *and* d *satisfying* $ed \equiv 1(mod\varphi)$, *then*

$$T_{ed}(x) \equiv x(modpq), \tag{6}$$

where p *and* q *are coprime primes.*

2.3. Rivest–Shamir–Adleman

The security of RSA is based on the difficulty of factoring large numbers. Its public key and private key consist of a pair of large primes. The difficulty of recovering plaintext from a public key and ciphertext is equivalent to factoring the product of two large primes.

A detailed description of the RSA algorithm is as follows:

1. Select two random large primes p and q, each with at least 512 bits, ensuring that the difference between them is not too large or too small;
2. Calculate the modulus $n = pq$ and the Euler's totient function value $\varphi(n)$;
3. Select a number e and use the extended Euclidean algorithm to find d that satisfies $ed = 1(mod\varphi(n))$;
4. The information receiver publishes the public key (e,n), with d as the private key;
5. The information sender divides the plaintext into blocks, such that each block is a positive integer m less than n, and encrypts it using $c = m^e(modn)$;
6. The information receiver decrypts using the privately saved key d according to the equation $m = c^d(modn)$.

From the above algorithm, it can be observed that:

$$m = c^d = (m^e)^d = m^{ed} = m$$

Usually, attacks on RSA target the protocols rather than the specific algorithm itself, but that discussion is beyond the scope of this explanation.

2.4. Chebyshev-RSA Public Key Cryptosystem

Kocarev et al. [41] proposed a Chebyshev-based RSA-type public key cryptography algorithm (Chebyshev-RSA Public Key Cryptosystem, CRPKC) based on the periodicity of $T_n(x)$, as shown below.

Assuming that Alice is the sender and Bob is the receiver, the communication process is as follows.

1. Key generation. Bob randomly generates two different large prime numbers p and q, calculates $N = pq$ and $\varphi = \left(p^2 - 1\right)\left(q^2 - 1\right)$, chooses a random number e such that $1 < e < \varphi$ and $gcd(e, \varphi) = 1$, and calculates the integer d such that $1 < d < \varphi$ and $ed \equiv 1 mod\varphi$; Bob's public key is (N, e) and the private key is d;
2. Message encryption. Alice converts the message to be encrypted into an integer $m(0 \leq m < N)$, calculates the ciphertext $c = T_e(m)(modN)$ based on Bob's public key (N, e), and sends it to Bob;
3. Message decryption. Bob receives the ciphertext c and uses the private key d to obtain the plaintext $m = T_d(c)(modN)$.

Although CRPKC has its unique characteristics, due to the lack of widespread application and research, it has certain uncertainties in terms of security, such as man-in-the-middle attacks and tampering attacks. Additionally, it does not provide identity authentication.

3. Improved Public Key Encryption Algorithm

The improved public key encryption algorithm is based on the aforementioned CRPKC algorithm and uses alternative multiplication factors K_i. The selection of K_i is determined by the magnitude of $T_r(T_d(x))modN = T_d(T_r(x))modN$, where $T(x)$ represents the Chebyshev polynomial, $i = 1, 2 \cdots n$. The specific rules for selecting K_i values and the size of n are secrets shared only among the participants. Therefore, this algorithm not only enhances computational complexity but also introduces digital signatures to prevent ciphertext from being subjected to man-in-the-middle attacks and tampering. The improved algorithm, $CRPKC - K_i$, inherits the advantages of the CRPKC algorithm while also taking into account the good properties of Chebyshev polynomials, resulting in better security and reliability.

The introduction of selective factors K_i follows the following selection rules:

$$\begin{cases} K_1, 0 \leq T_r(T_d(x))modN = T_d(T_r(x))modN \leq \frac{N}{n} \\ K_2, \frac{N}{n} \leq T_r(T_d(x))modN = T_d(T_r(x))modN \leq \frac{2N}{n} \\ \vdots \\ K_i, \frac{(i-1)N}{n} \leq T_r(T_d(x))modN = T_d(T_r(x))modN \leq \frac{iN}{n} \\ \vdots \\ K_n, \frac{(n-1)N}{n} \leq T_r(T_d(x))modN = T_d(T_r(x))modN \leq N \end{cases}. \tag{7}$$

In the above equation, N is a large prime number, and n represents the number of K_i. It is easy to see from Chebyshev's semigroup property that $T_r(T_d(x))modN = T_d(T_r(x))modN$.

$(AmodN)(modp) = A(modp)$, where A represents Chebyshev polynomial, and $N = p \cdot q$, where p and q are two large prime numbers.

The algorithm is described as follows (assuming Alice wants to communicate with Bob):

3.1. Key Generation

1. Alice randomly generates two distinct large prime numbers p and q, with similar sizes, but $p - q$ is a large integer;
2. Calculate $N = pq$ and $\varphi = (p - 1)(q - 1)$;
3. Randomly select an integer e, choose a random number $x \in Z_N$ such that $1 < e < \varphi$ and $gcd(e, \varphi) = 1$;

4. Use the extended Euclidean algorithm to calculate d, such that $1 < d < \varphi$ and $ed \equiv 1 mod \varphi$, and calculate $A = T_d(x)(modN)$;

At this point, the keys have been generated. Alice's public key is (N, A, e), and the private key is d.

3.2. Encryption

1. Bob obtains Alice's public key (N, A, e);
2. Randomly choose $r < N$, calculate $T_r(A)(modN) = T_r(T_d(x))(modN)$, and select K_{iB} based on Equation (6);
3. Express the message to be encrypted as an integer m, calculate $M = K_{iB}m$, and satisfy $0 \leq M < N$;
4. Use the public key to calculate $C = R_e(M) = M^e(modN)$, and $B = T_r(x)(modN)$. $Y_A = E_k(Sig_A(C, B))$, where $E_k(\square)$ is a mature symmetric encryption system;
5. Send the ciphertext (C, B, Y_A) to Alice.

3.3. Decryption

1. Alice receives the ciphertext (C, B, Y_A), decrypts Y_A, and checks $Sig_A(C, B)$. If it is correct, continue; otherwise, stop;
2. Calculate $M = R_{ed}(C) = C^d(modN)$ using the private key d, $T_d(B)(modN) = T_d(T_r(x))(modN)$, and select K_{iA} based on Equation (7);
3. Calculate $m = M/K_{iA}$.

4. Algorithm Implementation

4.1. Feasibility Analysis

Since the semigroup property of Chebyshev polynomials holds on the interval $(-\infty, +\infty)$, the decryption calculation is feasible. It is an important problem to evaluate Chebyshev polynomials in order to reduce the calculation time of $T_n(x)$. There are two ways to implement a fast algorithm for Chebyshev polynomial $T_n(x)$.

The first method is to use the semigroup property of Chebyshev polynomials to implement a fast algorithm. The algorithm is described as follows:

Let the integer s be decomposed as:

$$s = \underbrace{s_1 \cdots s_1}_{k_1}\underbrace{s_2 \cdots s_2}_{k_2} \cdots \underbrace{s_i \cdots s_i}_{k_i} = s_1^{k_1} s_2^{k_2} \cdots s_i^{k_i} \tag{8}$$

Then, due to the semigroup property of Chebyshev polynomials, it can be obtained that:

$$T_s(x) = T_{s_1}^{k_1}(T_{s_2}^{k_2}(\cdots T_{s_i}^{k_i}(x))) \tag{9}$$

To calculate $T_s(x)$, the number of iterations required is $k_1 + k_2 + \cdots k_i << s$. And the efficiency is higher when the value of s has more factors.

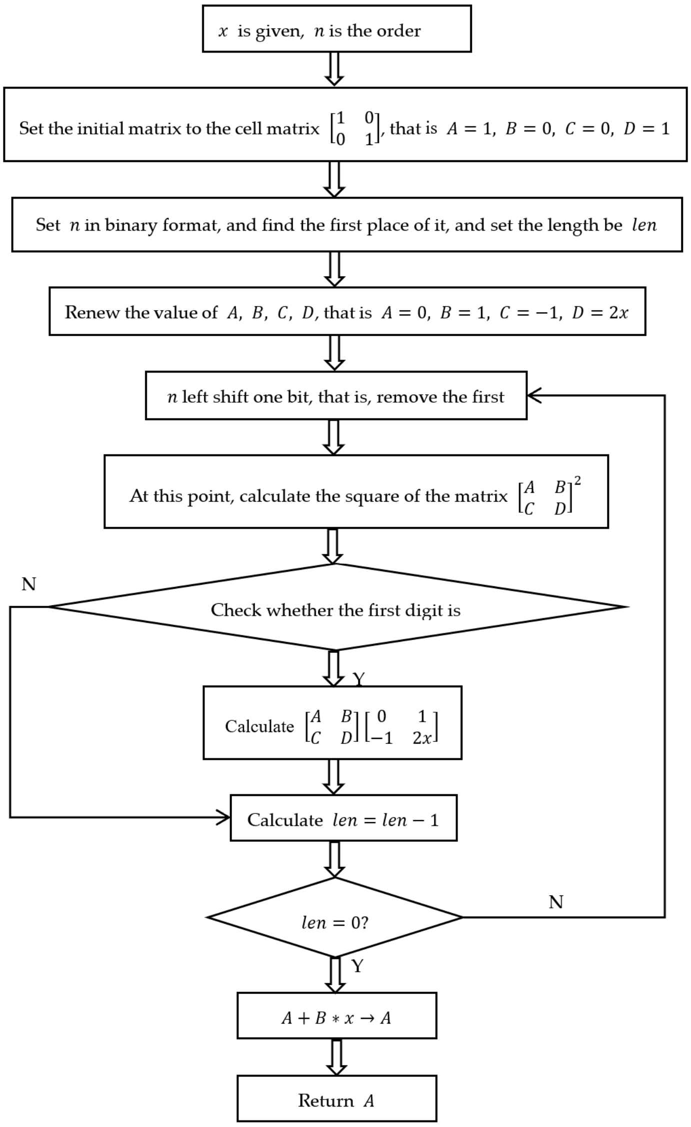
Although this algorithm is much faster than recursively calculating according to the definition, it is also composed of many loops and has high memory requirements for computers. Especially when n is a large number, the amount of computation is quite large. Therefore, the second matrix-based calculation method, which has been experimentally proven to be more efficient, is introduced below.

From the definition of Chebyshev polynomials $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$, $T_0 = 1$, $T_1 = x$, if we transform the above equation into a matrix expression, it can be represented as follows:

Let the integer s be decomposed as:

$$\begin{bmatrix} T_n \\ T_{n+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix} \begin{bmatrix} T_{n-1} \\ T_n \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix}^n \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \tag{10}$$

From the above equation, it can be seen that the key to calculating $T_n(x)$ lies in finding the value of the matrix $\begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix}^n$. The flowchart of the specific algorithm is shown in Figure 2.



x is given, n is the order
Set the initial matrix to the cell matrix $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, that is $A = 1,\ B = 0,\ C = 0,\ D = 1$
Set n in binary format, and find the first place of it, and set the length be len
Renew the value of $A,\ B,\ C,\ D$, that is $A = 0,\ B = 1,\ C = -1,\ D = 2x$
n left shift one bit, that is, remove the first
At this point, calculate the square of the matrix $\begin{bmatrix} A & B \\ C & D \end{bmatrix}^2$
N
Check whether the first digit is
Y
Calculate $\begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix}$
Calculate $len = len - 1$
$len = 0?$
N
Y
$A + B * x \rightarrow A$
Return A

Figure 2. The fast algorithm of Chebyshev polynomials.

4.2. An Example

Here we list a simple example to illustrate the algorithm and basic steps.

We now discuss the software implementation of the algorithm and provide a practical example. PYTHON is a popular and powerful programming software currently available for calculations. In our software implementation, we used an algorithm library in the Windows environment.

4.2.1. Generation of Secret Key

1. Alice randomly generates two distinct large prime numbers, $p = 127$ and $q = 113$;
2. Compute $N = pq = 14{,}351$ and $\varphi_{(n)} = (p-1)(q-1) = 14{,}112$;
3. Select a random integer $e = 5$ and a random number $x = 13 \in Z_N$, where $gcd(e, \varphi) = gcd(5,\ 14{,}112) = 1$;
4. Use the extended Euclidean algorithm to calculate $d = 5645$, satisfying $1 < d < \varphi$, and $ed \equiv 1 mod \varphi$. Calculate the Chebyshev polynomial $A = T_d(x)(modN) = 13{,}558$;

At this point, Alice's public key is $(N, A, e) = (14{,}351,\ 13{,}558,\ 5)$, and the private key is $d = 5645$.

4.2.2. Encryption

1. Bob obtains Alice's public key $(N, A, e) = (14{,}351,\ 13{,}558,\ 5)$;
2. Choose a random $r = 15 < N$ and calculate $T_r(A)(modN) = T_r(T_d(x))(modN) = 9257$. Select K_{iB} according to Equation (7). For simplicity, take $n = 5$ and $K_{iB} = 2i + 1$ as an arithmetic progression. In this case, calculate Equation (11) and select $K_{iB} = K_4 = 9$.

$$K_i = \begin{cases} 3, & 0 \le T_r(T_d(x))modN \le 2870.2 \\ 5, & 2870.2 \le T_r(T_d(x))modN \le 5740.4 \\ 7, & 5740.4 \le T_r(T_d(x))modN \le 8610.6 \\ 9, & 8610.6 \le T_r(T_d(x))modN \le 11{,}480.8 \\ 11, & 11{,}480.8 \le T_r(T_d(x))modN \le 14{,}351 \end{cases} \tag{11}$$

In this example, $8610.6 < T_r(A)(modN) = 9257 < 11{,}480.8$; therefore, choose $K_{iB} = K_4 = 9$;

3. Express the message to be encrypted as an integer $m = 1234$ and calculate $M = K_{iB}m = 11{,}106$, satisfying $0 \le M < 14{,}351$;
4. Compute the ciphertext $C = R_e(M) = M^e(modN) = 8438$;
5. $B = T_r(x)(modN) = 12{,}347$.

Send the ciphertext $(C, B) = (8438,\ 12{,}347)$ to Alice.

4.2.3. Decryption

1. Alice receives the ciphertext $(C, B) = (8438,\ 123{,}47)$;
2. Calculate $M = R_{ed}(C) = C^d(modN) = 8438^{5456}(mod14{,}351) = 11{,}106$ using the private key d. At this point, $8610.6 < T_d(B)(modN) = T_d(T_r(x))(modN) = 9257 < 11{,}480.8$. Similarly, calculate Equation (12) and select $K_{iA} = K_4 = 9$;

$$K_i = \begin{cases} 3, & 0 \le T_d(T_r(x))modN \le 2870.2 \\ 5, & 2870.2 \le T_d(T_r(x))modN \le 5740.4 \\ 7, & 5740.4 \le T_d(T_r(x))modN \le 8610.6 \\ 9, & 8610.6 \le T_d(T_r(x))modN \le 11{,}480.8 \\ 11, & 11{,}480.8 \le T_d(T_r(x))modN \le 14{,}351 \end{cases} \tag{12}$$

3. Calculate $m = M/K_{iA} = 11{,}106/9 = 1234$.

Then, send it to Alice. Finally, Alice recovers the message m.

In this example, p, q should be set as large primes in practical applications to achieve higher security, and the selection of other integers follows a similar approach.

4.3. Mathematical Proof of The Proposed $CRPKC - K_i$ Algorithm

Mathematically, the proposed algorithm $\mathrm{CRPKC} - \mathrm{K_i}$ is proven as follows.

The encrypted text C is calculated using:

$$C = R_e(M) = M^e(modN) = (m \cdot K_{iB})^e(modN) \tag{13}$$

Plaintext message m is calculated using:

$$m = M/K_{iA} = C^d(modN)/K_{iA} \tag{14}$$

Now, the objective is to fetch back the message M from $C^d(modN)$, then calculate the original message $m = M/K_{iA}$.

Proof. Since Equations (13) and (14), we can say, $C^d(modN) = M^{ed}(modN) = (K_{iB} \cdot m)^{ed}(modN)$, where, Bob calculates $K_{iB} = T_r(A)(modN) = T_r(T_d(x))(modN)$, Furthermore, $ed = 1mod\varphi(N)$ such that $C^d(modN) = (K_{iB} \cdot m)^{1+k\cdot\varphi(N)}(modN) = (K_{iB} \cdot m)^1(K_{iB} \cdot m)^{k\cdot\varphi(N)}(modN) = (K_{iB} \cdot m)^1(modN) = K_{iB} \cdot m = M$, Since $K_{iA} = T_d(T_r(x))(modN) =T_r(T_d(x))(modN) = K_{iB}$, using Equation (14), therefore, $m = M/K_{iA} = (K_{iB} \cdot m)/K_{iA} = m$, which is the original plaintext.

5. Implementation Results and Analysis

This paper conducts a security analysis on the improved public key encryption algorithm. The theoretical analysis shows that the improved algorithm is a public key cryptographic algorithm based on polynomials and RSA, which takes into account the security advantages of both and can effectively resist common attacks.

We can list the following advantages:

- Attackers cannot use the periodicity of Chebyshev polynomials to break it, because the cosine representation of Chebyshev polynomials defined on the interval $(-\infty, +\infty)$ is not valid.
- The three typical scenarios of RSA cipher attacks cannot be the same as the proposed scheme, because the following inequalities hold:

$$R_n(m_1)R_n(m_2) \neq R_n(m_1m_2) \tag{15}$$

$$R_{n_1+n_2}(m)(modn) \neq R_{n_1}(m)T_{n_2}(m)(modn) \neq R_{n_1}(R_{n_2}(m))(modn) \tag{16}$$

  where Equation (15) is an analogy to the multiplication property of RSA.
- It can resist common modular attacks.
- After encryption transformation, the linear independence between plaintexts cannot be maintained, which is immune to low exponent attacks.

The proposed CRPKC-Ki algorithm is implemented using SageMath's Jupyter Notebook. SageMath (or simply Sage) is a free and open-source mathematical software system that integrates numerous functionalities of symmetric and asymmetric key encryption, providing a unified interface and API. SageMath is developed based on the Python programming language and supports various functionalities such as mathematical computation, data analysis, graph plotting, and programming, and it can run on various operating systems. In this paper, the implementation and analysis were conducted using an 11th Gen Intel® Core™ i5-11320H @ 3.20 GHz processor, with 16.0 GB of RAM (15.8 GB available), and the operating system is Windows 10, 64-bit. The following two sections focus on the performance and security evaluation of CRPKC-Ki compared to RSA and its variants.

In reference [17], Raza Imam et al. performed a performance analysis of the XRSA algorithm, including key generation time, encryption time, and decryption time, and compared it with Rivest et al.'s standard RSA, as well as two other enhanced versions of RSA, namely ESRKGS [19] and MRSA [42]. The analysis results showed that the XRSA algorithm outperforms other similar algorithms in terms of performance, and the algorithm achieves higher algorithm complexity by introducing parameters in the key generation and encryption/decryption stages, which enhances the algorithm's security. This is similar to the motivation for improving the algorithm in this paper; hence, we introduce this algorithm as a comparative algorithm for performance analysis.

5.1. Performance Analysis

5.1.1. Key Generation

The Miller–Rabin primality testing algorithm is used for generating primes and tested and analyzed on different bit sizes of initial primes, ranging from 64 to 128, 256, 512, 1024, and 2048. Table 2 shows the comparative analysis of CRPKC-Ki with RSA, XRSA, and CRPKC. Figure 3 shows the comparison of key generation time under different key sizes, and it is evident that CRPKC has the longest key generation time, while standard RSA has the shortest and the best key generation time. For ease of comparison, the vertical axis uses a logarithmic scale with a base of 10 when plotting. The proposed CRPKC-Ki algorithm is essentially similar to the XRSA algorithm and falls in the middle. Due to the addition of the selective coefficient Ki in the proposed algorithm, which adds an initial parameter compared to RSA, the increase in time is acceptable considering the enhanced security.

Table 2. Analysis and comparison of CRPKC-Ki model with RSA, XRSA, and CRPKC models.

Model	Length of Primes (in bits)	Key Generation Time (in ms)	Encryption Time (in ms)	Decryption Time (in ms)	Total Execution Time (in ms)
RSA	64	20.27	0.26	0.22	20.75
	128	25.47	0.37	0.33	26.17
	256	40.50	0.97	0.85	42.32
	512	76.55	1.75	1.68	79.99
	1024	820.14	15.28	14.54	849.96
	2048	4575.03	52.26	37.39	4664.67
XRSA	64	32.00	0.57	0.54	33.11
	128	47.61	1.30	1.05	49.95
	256	93.54	2.02	2.01	97.56
	512	188.91	10.53	9.56	209.00
	1024	922.81	39.68	36.06	998.56
	2048	8706.22	185.98	221.41	9113.61
CRPKC	64	50.03	0.96	0.94	51.93
	128	76.33	1.46	1.42	79.21
	256	147.21	4.61	3.77	155.59
	512	228.62	11.81	12.14	252.57
	1024	1582.39	40.12	39.25	1661.76
	2048	14,203.75	198.01	232.38	14,634.14
CRPKC-Ki	64	35.02	0.65	0.64	36.32
	128	54.19	1.29	1.08	56.57
	256	105.99	3.12	3.09	112.19
	512	177.18	10.93	10.01	198.12
	1024	1250.09	39.87	39.24	1329.19
	2048	10,036.74	1896.98	235.31	12,169.03

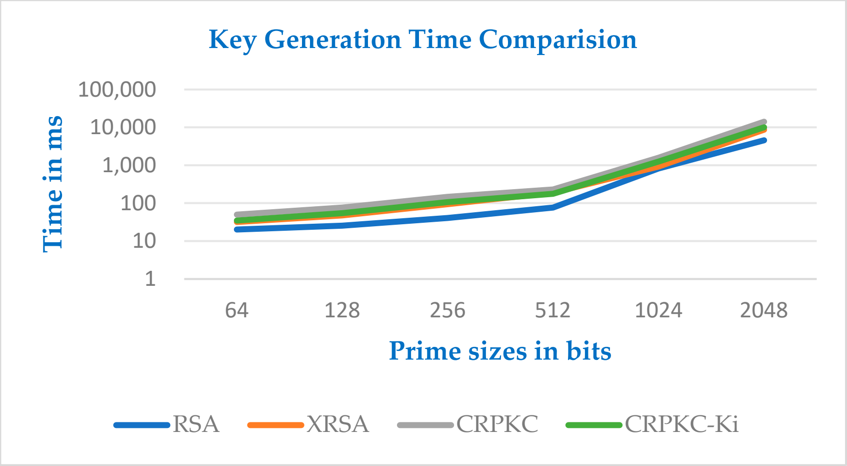


Figure 3. Key generation time comparison with different key sizes.

5.1.2. Encryption and Decryption

Considering the encryption and decryption time shown in Figures 4 and 5, it can be observed that the standard RSA has the lowest execution time in both encryption and decryption, making it the best in terms of encryption and decryption time. Due to the additional steps involved in the encryption and decryption stages of the CRPKC-Ki algorithm, the overall complexity of the proposed algorithm is increased. The introduction of a selection coefficient in the final encryption and decryption steps prevents attackers from determining the private key or plaintext using the product of the public key and the prime numbers. Furthermore, until the typical size of 1024 bits, CRPKC-Ki achieves similar performance in terms of encryption and decryption time compared to XRSA. Compared to CRPKC, CRPKC-Ki has better efficiency in terms of encryption, decryption, and key generation time per bit. For higher sizes of 2048 bits, CRPKC-Ki is slightly behind XRSA in encryption and decryption time, but the recommended prime size is usually 1024 bits, and the proposed CRPKC-Ki is more efficient and performs better among all the discussed algorithms. From the overall performance evaluation, the CRPKC-Ki algorithm is equally efficient and performs at the same level as the XRSA algorithm discussed, making it the best algorithm after the standard RSA. CRPKC is the least efficient among the discussed variations. Compared to the standard RSA, the added key generation time in CRPKC-Ki is reasonable because the introduction of Ki increases the complexity, making it more time-consuming to crack the CRPKC-Ki system and, thus, improving security.

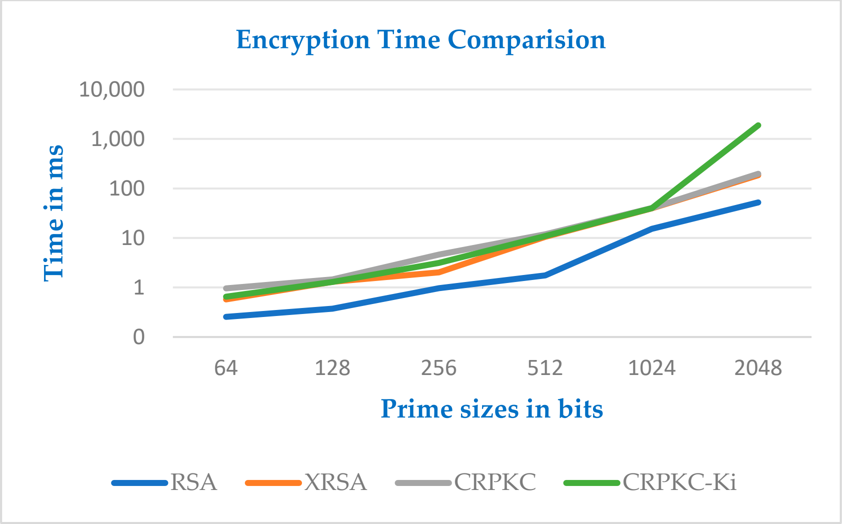


Figure 4. Encryption time comparison with different key sizes.

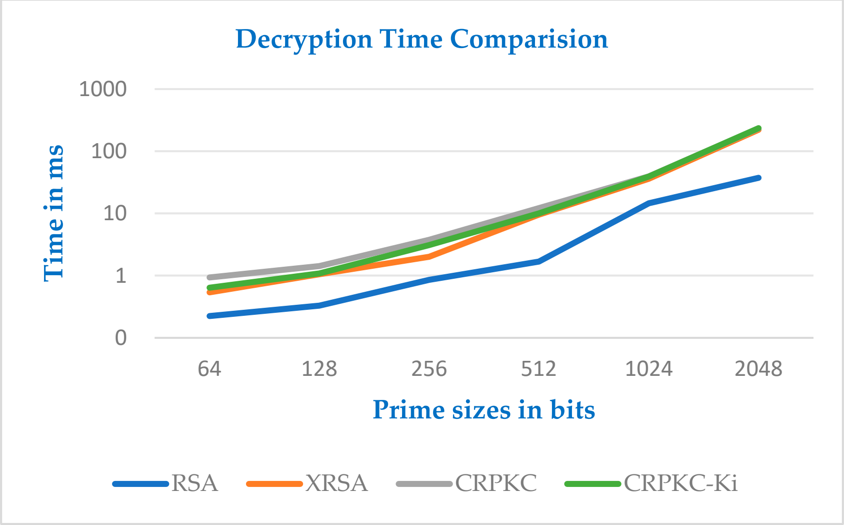


Figure 5. Decryption time comparison with different key sizes.

5.2. Security Analysis

5.2.1. Security Resistance to Man-in-the-Middle Attack Which Is Described Above

The man-in-the-middle attack refers to the ability of an attacker to intercept, replay, substitute, or modify the information exchanged between the communication parties. Malicious attackers can change ciphertext by hiding alternate multiplication coefficients k, and then calculate $m = m'/k$ to obtain plaintext. However, during the encryption process, we cleverly apply alternative multiplication coefficients K_i, which are not only determined by the values of $T_r(A)(modN) = T_r(T_d(x))(modN)$ and $T_d(B)(modN) = T_d(T_r(x))(modN)$, but also only share specific value selection strategies with participants. For example, in the example given in this paper, K_i is selected by an arithmetic sequence, which is just a simple example. In actual applications, we can adopt more complex rules, so we do not need to worry about the security of K_i selection.

5.2.2. Security against Tampering Attacks and Identity Authentication

Since Alice will sign the ciphertext during the actual encryption process, $Y_A = E_k(Sig_A(C, B))$, Bob can use this to check if the result has been tampered with or forged. Bob can easily verify Alice's identity through her public key during the decryption process.

5.2.3. Practicability Analysis

Modulus N is a large prime number, and $K_i \in Z_N$, providing a large selection space that can be systematically changed. Here, $i = 1, 2 \cdots n$ is an optional parameter, and we can choose K_i as needed. Additionally, if the number of users increases, we can increase the quantity accordingly. In practical cryptographic applications, periodically changing the values of K_i to enhance cryptographic security has the advantages of convenience, efficiency, and operability.

5.2.4. Comparison between Improved Algorithm and Original Algorithm

Both algorithms achieve encryption functionality, but the CRPKC $- K_i$ algorithm has higher security and reliability. Its innovation lies in the clever use of alternative multiplication coefficients K_i to prevent common ciphertext attacks. The use of digital signatures helps verify identity and avoid tampering attacks. The performance comparison analysis is shown in Table 3.

Table 3. Performance comparisons.

Attacks/Functions	**The Algorithm, CRPKC**	**The Improve Algorithm, CRPKC−K_i**
Man-in-the-middle attack	Not safe	Safe
Tampering attack	Not safe	Safe
Authentication	Not given	Safe
Practicability	Ordinary	Better

From the process of the public key encryption algorithm, we can see that although its computational complexity is slightly higher than the former, the order of magnitude is basically the same. The application of digital signatures will adopt more mature solutions. Therefore, in practical applications, the improved algorithm is superior to the original algorithm.

6. Conclusions and Future Scope

Compared to the private key cryptographic system, the public key cryptographic system has unique advantages in key exchange, communication between unknown entities, secure services, and authentication services. This paper introduces the CRPKC public key encryption algorithm and proposes an improved public key cryptographic algorithm based on Chebyshev polynomials and RSA, named CRPKC $- K_i$. The improved algorithm introduces an alternative multiplication coefficient K_i to forge ciphertext and uses digital signatures to ensure Alice's identity. These measures enable the system to resist chosen ciphertext attacks and tampering attacks while providing identity authentication functionality. In conclusion, after performance and security analysis, the proposed CRPKC $- K_i$ algorithm based on an RSA variant shows better results with improved security.

The proposed algorithm is equivalent to encrypting the plaintext twice. Although it requires some extra computation time compared to other methods, we believe its advantage lies in the fact that even if an attacker guesses the private key in the same amount of time, they still need to decipher the true plaintext from the forged plaintext. This enhances the algorithm's security. However, in the case of large prime bit sizes, the efficiency of the algorithm decreases significantly, especially beyond 2048 bits, and the multiplication coefficient cannot be set too large, otherwise it will have a negative impact on encryption/decryption time. This can be seen as a drawback. Therefore, as future work, we plan to extend this algorithm to parallel machines so that multiple related operations can be executed simultaneously on multi-core processors. We will also discuss the optimal values for the multiplication coefficient in detail to reduce computational costs. Additionally, combining the algorithm protocol with symmetric cryptography and ensuring its wide acceptability in IoT devices will be explored.

Author Contributions: Conceptualization, C.Z., Y.L., A.T. and L.W.; Methodology, C.Z. and L.W.; Software, C.Z., T.G. and S.P.; Validation, Y.L., A.T. and L.W.; Formal analysis, C.Z. and L.W.; Investigation, Y.L. and A.T.; Resources, Y.L., A.T., T.G. and S.P.; Writing—original draft, C.Z.; Writing—review & editing, C.Z., Y.L., A.T., L.W., T.G. and S.P.; Supervision, Y.L., A.T. and L.W.; Project administration, Y.L., A.T., L.W., T.G. and S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62372494, the Guangdong Province College Youth Innovative Talent Project, grant number 2023KQNCX132, the Guangdong Universities' Innovation Team Project, grant number 2021KCXTD015, the Guangdong Key Disciplines Project, grant number 2021ZDJS138, and the Applied Statistics Discipline Construction Project of Zhuhai College of Science and Technology, grant number, 2019XJCQ001.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors are thankful to the anonymous reviewers for their helpful remarks.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, G.; Ma, Y. Introduction to chaotic cryptographic algorithms. In *Chaotic Applied Cryptography*, 1st ed.; Cheng, J., Ed.; Science Press: Beijing, China, 2021; pp. 1–7.
2. Diffie, W.; Hellman, M.E. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
3. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
4. Arora, S. Enhancing cryptographic security using novel approach based on enhanced-RSA and Elamal: Analysis and comparison. *Int. J. Comput. Appl.* **2015**, *975*, 8887.
5. Luo, C.; Fei, Y.; Kaeli, D. Side-channel timing attack of RSA on a GPU. *ACM Trans. Archit. Code Optim. (TACO)* **2019**, *16*, 1–18. [CrossRef]
6. Zheng, M. Generalized implicit-key attacks on RSA. *J. Inf. Secur. Appl.* **2023**, *77*, 103562. [CrossRef]
7. Zhang, P. Quantum Related-Key Attack Based on Simon's Algorithm and Its Applications. *Symmetry* **2023**, *15*, 972. [CrossRef]
8. Nitaj, A.; Susilo, W.; Tonien, J. A new attack on some RSA variants. *Theor. Comput. Sci.* **2023**, *960*, 113898. [CrossRef]
9. Imam, R.; Areeb, Q.M.; Alturki, A.; Anwer, F. Systematic and critical review of rsa based public key cryptographic schemes: Past and present status. *IEEE Access* **2021**, *9*, 155949–155976. [CrossRef]
10. El Makkaoui, K.; Beni-Hssane, A.; Ezzati, A.; El-Ansari, A. Fast cloud-RSA scheme for promoting data confidentiality in the cloud computing. *Procedia Comput. Sci.* **2017**, *113*, 33–40. [CrossRef]
11. Moghaddam, F.F.; Alrashdan, M.T.; Karimi, O. A hybrid encryption algorithm based on RSA small-e and efficient-RSA for cloud computing environments. *J. Adv. Comput. Netw.* **2013**, *1*, 238–241. [CrossRef]

12. AlSabti, K.D.M.; Hashim, H.R. A new approach for image encryption in the modified RSA cryptosystem using MATLAB. *Glob. J. Pure Appl. Math.* **2016**, *12*, 3631–3640.
13. Jagadiswary, D.; Saraswady, D. Estimation of modified RSA cryptosystem with hyper image encryption algorithm. *Indian J. Sci. Technol.* **2017**, *10*, 1–5. [CrossRef]
14. Mustafa, I.; Khan, I.U.; Aslam, S.; Sajid, A.; Mohsin, S.M.; Awais, M.; Qureshi, M.B. A lightweight post-quantum lattice-based RSA for secure communications. *IEEE Access* **2020**, *8*, 99273–99285. [CrossRef]
15. Rawat, A.S.; Deshmukh, M. Computation and communication efficient secure group key exchange protocol for low configuration system. *Int. J. Inf. Technol.* **2021**, *13*, 839–843. [CrossRef]
16. Chait, K.; Laouid, A.; Kara, M.; Hammoudeh, M.; Aldabbas, O.; Al-Essa, A.T. An Enhanced RSA-Based Aggregate Signature Scheme to Reduce Blockchain Size. *IEEE Access* **2023**, *11*, 110490–110501. [CrossRef]
17. Imam, R.; Anwer, F.; Nadeem, M. An Effective and enhanced RSA based Public Key Encryption Scheme (XRSA). *Int. J. Inf. Technol.* **2022**, *14*, 2645–2656. [CrossRef]
18. Minni, R.; Sultania, K.; Mishra, S.; Vincent, D.R. An algorithm to enhance security in RSA. In Proceedings of the 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, India, 4–6 July 2013; pp. 1–4.
19. Thangavel, M.; Varalakshmi, P.; Murrali, M.; Nithya, K. An enhanced and secured RSA key generation scheme (ESRKGS). *J. Inf. Secur. Appl.* **2015**, *20*, 3–10. [CrossRef]
20. Lüy, E.; Karatas, Z.Y.; Ergin, H. Comment on "An enhanced and secured RSA key generation scheme (ESRKGS)". *J. Inf. Secur. Appl.* **2016**, *30*, 1–2. [CrossRef]
21. Mathur, S.; Gupta, D.; Goar, V.; Kuri, M. Analysis and design of enhanced RSA algorithm to improve the security. In Proceedings of the 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 9–10 February 2017; pp. 1–5.
22. Akhter, S.; Chowdhury, M.B. Bangla and English text cryptography based on modified blowfish and Lempel-Ziv-Welch algorithm to minimize execution time. In Proceedings of the 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 10–12 January 2019; pp. 96–101.
23. Panda, P.K.; Chattopadhyay, S. A hybrid security algorithm for RSA cryptosystem. In Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 January 2017; pp. 1–6.
24. Agrawal, S.; Patel, M.; Sinhal, A. An enhance security of the color image using asymmetric RSA algorithm. In *Proceedings of the International Conference on Communication and Computational Technologies: ICCCT-2019*; Springer: Singapore, 2020; pp. 279–286.
25. Shannon, C.E. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
26. Mohamed, K.S. *New Frontiers in Cryptography: Quantum, Blockchain, Lightweight, Chaotic and DNA*, 1st ed.; Springer Nature: Cham, Switzerland, 2020; pp. 13–28.
27. Kocarev, L.; Tasev, Z. Public-key encryption based on Chebyshev maps. In Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03, Bangkok, Thailand, 25–28 May 2003; p. III. [CrossRef]
28. Bergamo, P.; D'Arco, P.; De Santis, A.; Kocarev, L. Security of public-key cryptosystems based on Chebyshev polynomials. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2005**, *52*, 1382–1393. [CrossRef]
29. Zhao, G.; Sun, J.; Zhao, F. Key Agreement Scheme Based on Chebyshev Polynomial over finite field. *Appl. Res. Comput.* **2012**, *29*, 3794–3796.
30. Wang, D.; Hu, Z.; Tong, Z.; Zha, X. An identity authentication system based on Chebyshev polynomials. In Proceedings of the 2009 First International Conference on Information Science and Engineering, Nanjing, China, 26–28 December 2009; pp. 1648–1650.
31. Muhammad, A.S.; Özkaynak, F. SIEA: Secure Image Encryption Algorithm Based on Chaotic Systems Optimization Algorithms and PUFs. *Symmetry* **2021**, *13*, 824. [CrossRef]
32. Dai, W.; Xu, X.; Song, X.; Li, G. Audio Encryption Algorithm Based on Chen Memristor Chaotic System. *Symmetry* **2022**, *14*, 17. [CrossRef]
33. Lu, Q.; Yu, L.; Zhu, C. Symmetric Image Encryption Algorithm Based on a New Product Trigonometric Chaotic Map. *Symmetry* **2022**, *14*, 373. [CrossRef]
34. Alsaif, H.; Guesmi, R.; Kalghoum, A.; Alshammari, B.M.; Guesmi, T. A Novel Strong S-Box Design Using Quantum Crossover and Chaotic Boolean Functions for Symmetric Cryptosystems. *Symmetry* **2023**, *15*, 833. [CrossRef]
35. Hsiao, F.H. Chaotic synchronization cryptosystems combined with RSA encryption algorithm. *Fuzzy Sets Syst.* **2018**, *342*, 109–137. [CrossRef]
36. Ruzai, W.N.A.; Abd Ghafar, A.H.; Salim, N.R.; Ariffin, M.R.K. On (Unknowingly) Using Near-Square RSA Primes. *Symmetry* **2022**, *14*, 1898. [CrossRef]
37. Lawnik, M.; Kapczyński, A. Application of modified Chebyshev polynomials in asymmetric cryptography. *Comput. Sci.* **2019**, *20*, 289–303. [CrossRef]
38. Gupta, M.; Gupta, K.K.; Shukla, P.K. Session key based novel lightweight image encryption algorithm using a hybrid of Chebyshev chaotic map and crossover. *Multimed. Tools Appl.* **2021**, *80*, 33843–33863. [CrossRef]
39. Patgiri, R.; Singh, L.D. An Analysis on the Variants of the RSA Cryptography. In Proceedings of the 2022 International Conference on Information Networking (ICOIN), Jeju-si, Republic of Korea, 12–15 January 2022; pp. 40–45.

40. Ryu, J.; Kang, D.; Won, D. Improved secure and efficient Chebyshev chaotic map-based user authentication scheme. *IEEE Access* **2022**, *10*, 15891–15910. [CrossRef]
41. Kocarev, L.; Makraduli, J.; Amato, P. Public-key encryption based on Chebyshev polynomials. *Circuits Syst. Signal Process.* **2005**, *24*, 497–517. [CrossRef]
42. Islam, M.A.; Islam, M.A.; Islam, N.; Shabnam, B. A modified and secured RSA public key cryptosystem based on "n" prime numbers. *J. Comput. Commun.* **2018**, *6*, 78. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.