TELKOMNIKA, Vol. 11, No. 2, February 2013, pp. 864~870
ISSN: 2302-4046

An Improved Public Key Encryption Algorithm Based on Chebyshev Polynomials

Jinhui Sun*[1], Geng Zhao[1], Xufei Li[1]
[1]School of Communication Engineering, Xidian University Xi'an 710071, China,13439925967
*Corresponding author, e-mail: sjhyunle@163.com

Abstract

This paper proposes an improved public key encryption algorithm based on Chebyshev polynomials. On the base of the semi-group property of Chebyshev polynomials, we import the alternative multiply coefficient K_i to forge the ciphertext tactfully which can make the cipher text-only aattack out of work. The chosen of K_i is decided by the value of $T_r(T_s(x)) \bmod N$, and the number of K_i can be chosen as required. Besides, The digital signature of the ciphertext not only can prevent the result from faking and tampering attack, but also can make the algorithm have the function of identity authentication. Experimental results and performance analyses show that the improved algorithm has much higher security and practical value.

Key words: *chebyshev polynomials, public key encryption, alternative multiply coefficient, semi-group property, digital signature*

1. Introduction

Chaos possesses certain intrinsic properties such as sensitive dependence on initial condition, random-like behavior, and continuous broadband power spectrum. These characteristics match the confusion, diffusion, and key sensitivity requirements of cryptography. In recent years, there have been a tremendous amount of reports in how to use chaotic systems to design cryptographic algorithms. Although most of them aim at symmetric-key schemes, such as [1-4]. There are still some for asymmetric-key or public key cryptosystems, such as [5, 6]. The focus of this paper is on the later one, i.e., public-key cryptosystems based on chaos.

In [7], Kocarev et al. suggested public key cryptography based on the commutative property of Chebyshev polynomials over real numbers. However, it was later cryptanalyzed by Bergamo et al [2], and other researches [8, 9]. The fundamental weakness of this algorithm is that Chebyshev polynomials of order n have an explicit algebraic expression $T_n(\mathrm{x}) = \cos(n \arccos x)$ over real numbers. To resist this attack, Kocarev et al. modified the algorithm by employing the Chebyshev polynomials defined over the finite field Z_N [10]. The explicit algebraic expression of Chebyshev polynomials over Z_N doesn't help to find n giving an initial value x_0 and a final iterated value x_n. Furthermore, Kocarev et al. pointed out that the problem of computing n reduces to the discrete logarithm problem [10]. But it is not always true and it depends on the choice of N as the analysis [11]. There the authors analyzed the period distribution of sequences generated by Chebyshev polynomials over finite fields when the modulus N is a prime. An attack on the public key algorithm was also proposed, followed by an improvement of the algorithm to make it for real world applications. Besides, the security of this class cryptosystems is investigated from a practical viewpoint.

In this paper, we proposed an improved public key encryption algorithm based on Chebyshev chaotic map, which overcomes the drawbacks of the previous schemes and provided a higher level of security. Analytical and experimental results show that it is robust to generic attacks.

This paper is organized as follows. In section II, we give a description of the Chebyshev chaotic map and some properties of it. In Section 3, the basic encryption algorithm over Chebyshev polynomials is presented, and an attack to it was described, and then an

Received August 30, 2012; Revised December 30, 2012; Accepted January 14, 2013

improved encryption algorithm is proposed. Section 4 contains the solutions to some software implementation issues, an example and the experimental results. Section 5 presents some performance analyses. Finally, conclusion will be drawn in the last section.

2. Preliminaries

Definition 1. Let $n \in Z^{+}$ and $x \in R$, then a Chebyshev polynomial of order n, $T_n(x): R \to R$ is recursively defined using the following recurrent relation:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), n \geq 2 \quad (1)$$

where $T_0(x) = 1$ and $T_1(x) = x$.

The first few Chebyshev polynomials are

$T_2(x) = 2x^2 - 1$

$T_3(x) = 4x^3 - 3x$

$T_4(x) = 8x^4 - 8x^2 + 1$

$\cdots$

When x is a real number, $T_n(x)$ always has the following explicit algebraic expression:

$$\begin{cases} T_n(x) = \cos(n\cos^{-1}(x)), & x \in [-1,1] \\ T_n(x) = \cosh(n\cosh^{-1}(x)), & x \in [1,+\infty) \end{cases} \quad (2)$$

Some important properties of Chebyshev polynomials are as follows.

$$T_r(T_s(x)) = T_s(T_r(x)) \quad (3)$$

$$T_n\left(\tfrac{x+x^{-1}}{2}\right) = \tfrac{x^n + x^{-n}}{2} \quad (4)$$

Proposition 2 can be easily deduced from the explicit algebraic expression (2) and it is this commutative property that is employed by Kocarev et al. to construct a novel public key algorithm [7-10].

When $x \in R$, the explicit algebraic expression of $T_n(x)$ to a security loophole in the public-key cryptosystem based on Chebyshev polynomials defined over the real number field [12]. Therefore, Kocarev et al. extended the definition of $T_n(x)$ to the finite field Z_n [10].

Definition 2. Let $n \geq 0$ be an integer, a variable $x \in Z_N$ and N be a positive integer. Chebyshev polynomial of order n is recursively defined by

$$T_n(x) \equiv (2xT_{n-1}(x) - T_{n-2}(x)) \bmod N \quad (5)$$

where $T_0(x) \equiv 1 \bmod N$ and $T_1(x) \equiv x \bmod P$.

It is easy to verify that the above propositions of $T_n(x)$ also holds over Z_N.

3. The Improved Public Key Encryption Algorithm

The public key algorithm proposed by Kocarev et al. in [10] is as follows. Suppose Alice wants to communicate with Bob. They do the followings.

1) Bob generates a large integer s, selects a random number $x \in Z_N$, and computes $T_s(x)$ mod N, then sets the public key to $(x, T_s(x))$ and the private key is s.

2) In order to send the message to Bob, Alice obtains Bob's authentic public key $(x, T_s(x))$, represents the message as a number $M \in Z_N$, then, she generates a large random integer r and

computes $C_1 = T_r(x) \bmod N, C_2 = M \times T_r(T_s(x)) \bmod N$, then sends the ciphertext $C = (C_1, C_2)$ to Bob.
3) In order to decrypt the message, Bob uses his private key s to compute $T_s(C_1) = T_s(T_r(x)) = T_{sr}(x) = T_r(T_s(x))$, thus he recovers the plaintext by computing $M = X / T_s(T_r(x))$.

In [10], N is chosen as a prime and so the decryption is always correct. But when N is a composite, this algorithm encounters a problem: the inverse of $T_s(T_r(x))$, says $T_s(T_r(x))^{-1} \bmod N$, does not always exist which is the same problem with Rabin public key cryptosystem. This problem is equivalent to that the solution for M is not unique if $T_s(T_r(x))^{-1} \bmod N$ is not invertible, i.e. $T_s(T_r(x))$ and N have common divisors. There are two simple methods to solve it.
1) Add extra information to indicate which plaintext is encrypted.
2) When the generated random number r leads to common divisors between $T_s(T_r(x))$ and N, reject it and choose another one until they are coprime.

With these measures, a composite N is also allowed in the cryptosystem. In the following discussion, we only address the situation when N is a prime. This is because composite N leads to a more complicated situation than prime N.

The above public key encryption algorithm is simple, and it could resist Bergamo et al.'s attack. However, it is just a basic idea and cannot be used directly in practice. There are still some security problems, such as vulnerability to man-in-the-middle attack, nonsupport mutual authentication and so on.

Here we introduce one kind of man-in-the-middle attack.

Suppose Malice intercepted a piece of ciphertext $C = (T_r(x), M = T_r(T_s(x)))$ before Alice and Bob communicated. Now, he can modify the ciphertext to $C = (T_r(x), M = kT_r(T_s(x)))$, where k belongs to Z_N and is known to Malice. Then, Malice sends the modified ciphertext to Bob for decryption and gets back the plaintext $M^{'} = KM$. Then, Malice can recover M by $M = M^{'} / k$, which was the message Alice and Bob once communicated. This attack is a chosen-ciphertext attack with which many public key cryptosystems based on a rigid mathematical structure may suffer. To avoid this attack, a common approach is to sign the ciphertext to make sure that it has not been altered. So, a more detailed public key encryption algorithm is needed for secure communication.

The improved public key encryption algorithm is similar to the above one, the critical factor is the adoption of the alternative multiply coefficient $K_i (K_i \in Z_N)$. K_i is secret , which is only shared with participants. Accordingly, this algorithm fortifies the complexity without increasing the calculation difficulty. The digital signature prevents the ciphertext from man-in-the-middle attack and tamper attack. So, the improved algorithm inherits the advantage of the above one, but has better security and higher reliability.

The choice of K_i is decided by the value of $T_r(T_s(x)) \bmod N$.

$$K_i = \begin{cases} K_1, & 0 \le T_r(T_s(x)) \bmod N \le \frac{N}{4} \\ K_2, & \frac{N}{n} < T_r(T_s(x)) \bmod N \le \frac{2N}{n} \\ \vdots & \\ K_i, & \frac{(i-1)N}{n} < T_r(T_s(x)) \bmod N \le \frac{iN}{n} \\ \vdots & \\ K_n, & \frac{(n-1)N}{n} < T_r(T_s(x)) \bmod N < N \end{cases} \tag{6}$$

The algorithm is described as follows: (assume Alice wants to communicate with Bob)
(1) Key pair generation

In order to generate the keys, Bob does the following: Randomly select an integer number s and $x \in Z_N$, and computes $T_s(x) \bmod N$, and His private key is s, and his public key is $(x, T_s(x) \bmod N)$.

(2) Message encryption

Assume that Alice wants to send the message $M \in Z_N (M \neq 0)$ to Bob. She does the followings: Randomly select an integer number r. Get Bob's public key $(x, T_s(x) \bmod N)$, computes $T_r(x) \bmod N$, $T_{rs}(x) \bmod N = T_r(T_s(x) \bmod N)$, and then choses K_i according to equation 6, and computes $X = K_i M T_{rs}(x) \bmod N, Y_A = E_K(Sig_A(X))$. $E_K(\cdot)$ is one of symmetric cryptography. Sends the ciphertext $(T_r(x) \bmod N, X, Y_A)$ to Bob.

(3) Message decryption

After receiving the encrypted message, in order to decrypt it, Bob does the followings:
(1) Decrypt Y_A to check $Sig_A(X)$. If it is right, continue, or stop.
(2) Uses his private key s to compute $T_s(T_r(x)) \bmod N = T_{sr}(x) \bmod N = T_r(T_s(x)) \bmod N$, and choses K_i according to equation 6.
(3) Recovers M by computing $M = X / (K_i T_{sr}(x) \bmod N)$.

4. Software Implementation
4.1. Feasibility Analysis

There are two main software implementation issues of this algorithm. One is the correctness of the algorithm when it is implemented in finite fields. The semi-group property of Chebyshev polynomials holds over Z_N. $T_s(T_r(x))^{-1} \bmod N$ exists as long as N is a prime. So we can recover M by computing $M = X / (K_i T_{sr}(x) \bmod N)$.Another issue is how to evaluate Chebyshev polynomials so that the computation time of $T_n(x)$ could be reduced. There are several kinds of measures. The first is assume the large number s (r is the same) is written as

$$s = s_1^{k_1} s_2^{k_2} \cdots s_i^{k_i} \tag{7}$$

Then

$$T_s(x) \bmod P = \underbrace{T_{s_1}(\cdots T_{s_1}}_{k_1} \cdots \underbrace{T_{s_i}(\cdots T_{s_i}}_{k_i}(x))) \bmod P \tag{8}$$

So, for computing $T_s(x)$ one needs only $k_1 + k_2 + \cdots k_i$ iterations of the Chebyshev map instead of s iterations [7]. Because choosing s and then factorizing s to get $k_i (i = 1, 2, 3...)$ may cost a lot of time, but a reverse order can be adopted easily, i.e., k_i is chosen randomly and then s is constructed by k_i.The second is the fast algorithm of Chebyshev polynomials[13]. Rewrite the Chebyshev polynomials as

$$\begin{bmatrix} T_n(x) \\ T_{n+1}(x) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix} \begin{bmatrix} T_{n-1}(x) \\ T_n(x) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix}^n \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \tag{9}$$

From equation 9, we can find out that the key point of computing $T_n(x)$ is to compute the value of matrix $\begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix}^n$. Figure 1 illustrates the detailed process in a simplified sequence flow diagram.

After getting A, we can get the result $T_n(x) \bmod N$ easily using modulo operator. In [13], author has verified the high efficiency of the fast algorithm of Chebyshev polynomials.

4.2. An Example

Here we present a simple example to illustrate the basic algorithm and main steps of the algorithm. Assume $N = 61$, the choice of K_i is decided by equation 10.

$$K_i = \begin{cases} K_1 = 3, & 0 \le T_r(T_s(x)) \bmod N \le \frac{N}{4} \\ K_2 = 5, & \frac{N}{4} < T_r(T_s(x)) \bmod N \le \frac{N}{2} \\ K_3 = 9, & \frac{N}{2} < T_r(T_s(x)) \bmod N \le \frac{3N}{4} \\ K_4 = 11, & \frac{3N}{4} < T_r(T_s(x)) \bmod N < N \end{cases} \quad (10)$$

(1) Bob randomly selects an integer number $s = 3$, and $x = 5$, computes $T_s(x) \bmod N = T_3(5) \bmod 61 = 58$. Bob's private key is $s = 3$, and his public key is $(x, T_s(x) \bmod N) = (3, 58)$.

(2) Alice randomly selects an integer number $r = 7$, she wants to send the plaintext $M = 16$, so she computes $T_r(x) \bmod N = T_7(5) \bmod 61 = 10$, $T_{rs}(x) \bmod N = T_7(T_3(5) \bmod 61) = 5$, and then choses $K_i = K_1 = 3$ by equation 10. $X = K_i M T_{rs}(x) \bmod N = K_1 M T_{7 \cdot 3}(5) \bmod 61 = 57$. So she sends the ciphertext $(T_r(x) \bmod N, X) = (10, 57)$ to Bob.

(3) Bob gets the ciphertext, uses his private key $s = 3$ to compute $T_s(T_r(x)) \bmod N = T_3(T_7(5)) \bmod 61 = 5$, and choses $K_i = K_1 = 3$ according to equation 10, then he can recovers M by computing $M = X / (K_i T_{sr}(x) \bmod N) = 57 / 3 \cdot 5 \bmod 61 = 16$.

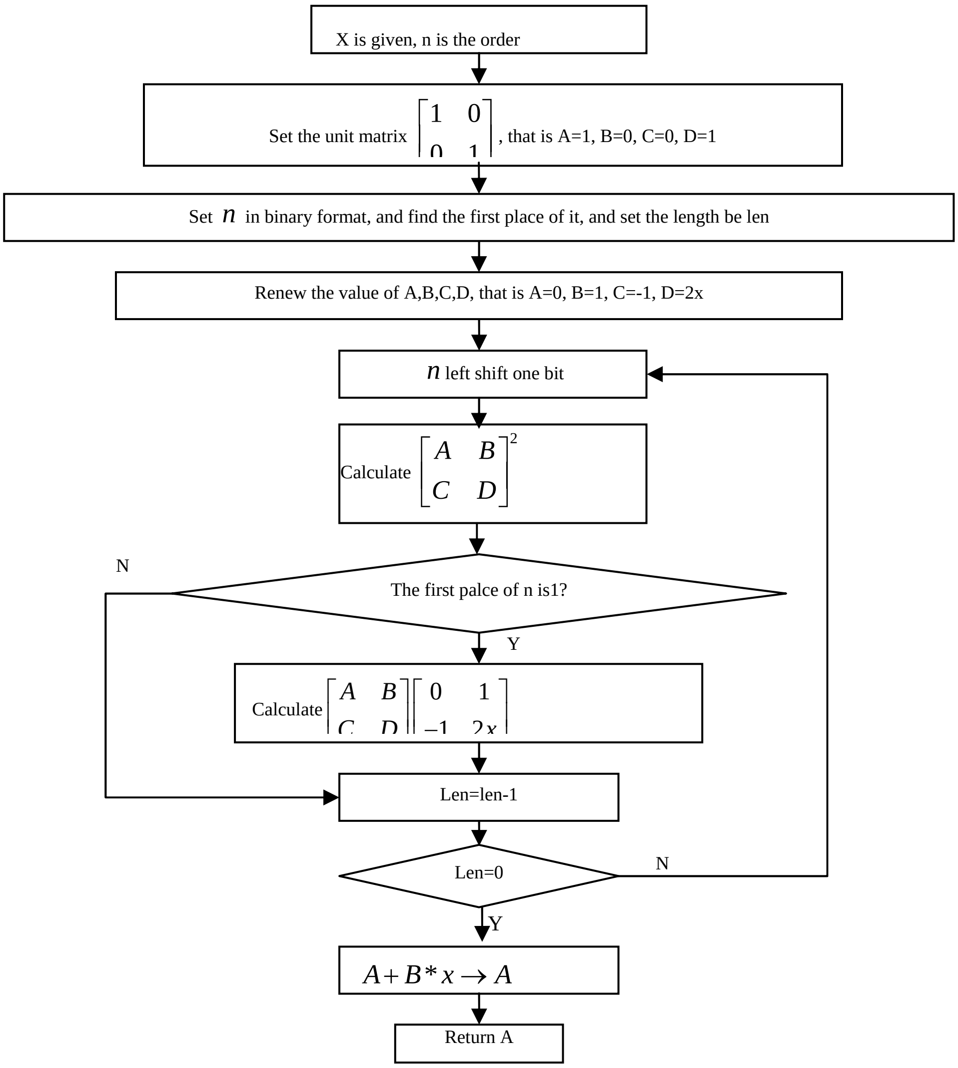

Figure 1. The Fast Algorithm of Chebyshev Polynomials

4.3. Experimental Analysis

The software environment is that basic frequency is 2.60GHz and RAM is 1.99GB. We achieve the algorithm by programming and record the time running on different bits. The result is shown in Figure 2.

The experimental results show that this algorithm has the high efficiency when the bit is small. But when the bit is relatively large, the running time is not satisfactory. This is the point where we intend to make some improvement.

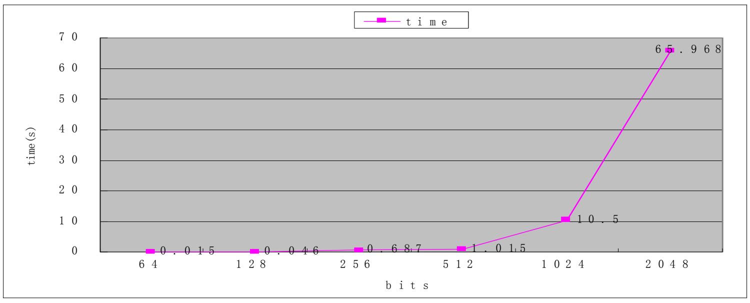

Figure 2. The Running Time of Algorithm

5. Performance Analyses

We present several security analyses of the improved public key encryption algorithm here. Theoretical analyses prove the improved algorithm could effectively resist common attacks. Besides, it is efficient and practical.

5.1. Security Resistant to Man-in-the-middle Attack which is Described Above

Malice alters the ciphertext by sneaking the alternative multiply coefficient k, and then gets the plaintext by calculating $M = M^{'} / k$. However, in the encryption process, we apply the alternative multiply coefficient K_i tactfully. K_i is not only decided by the value of $T_r(T_s(x)) \bmod N$ but also is only shared with participants, so we do not need worry about the safety of the chosen of K_i.

5.2. Security Resistant to Tamper Attack

Alice signs the ciphertext $Sig_A(X)$ in the encryption process, Bob can check whether the result is tampered, forged or not accordingly. If right, continue, or stop.

5.3. Identity Authentication

Because Alice signs the ciphertext in the encryption process, $Y_A = E_K(Sig_A(X))$, so Bob can verify the identity of Alice by Alice's public key in decryption process easily.

5.4. Practicability Analysis

Modulus N is a large prime, and $K_i \in Z_N$, so K_i has a lot of choice space and can be altered regularly. Here, i is a optional parameters, we can chose K_i slickly as needed. Besides, if the number of users increased, we can increase the number of i accordingly. In the practical cryptography application, altering key K_i regularly to enhance the security of cryptography has the features of high convenience, efficiency, maneuverability.

5.5. Comparison Between the Improved Algorithm and the Algorithm Proposed in [10]

The two algorithm both have achieved the function of encryption. But the improved algorithm has higher security and reliability. Its critical point is alternative multiply coefficient K_i ,which can prevent the result suffering cipher text-only attack . The use of digital signature can help validate identities and avoid tampering attack. The performance comparison analysis are shown in the following Table1.

Table 1. The Performance Comparisons

Attacks /functions	The algorithm proposed in [10]	the improved algorithm
Bergamo et al attack[2]	Not safe	safe
man-in-the-middle attack	Not safe	safe
tampering attack	Not safe	safe
Authentication	Not given	given
practicability	ok	Very good

From the process of public key encryption algorithm, we can see that the computation complexity is the same with the former one. The application of digital signature is pretty practised. So, in practical terms, the improved algorithm is superior to the former one.

6. Conclusions

In this paper, we introduce one kind of man-in-the-middle attack and propose an improved public key encryption algorithm based on Chebyshev polynomials. This algorithm imports alternative multiply coefficient K_i to forge the ciphertext and adopts digital signature tactfully to ensure Alice's identity. All of these measures make the system not only can resist chosen-ciphertext attack and tamper attack, but also has the function of identity authentication. Experimental results and performance analyses show that it is more secure and more practical.

References

[1] H Liu, X Wang. Color image encryption based on one-time keys and robust chaotic maps. *Computers and Mathematics with Applications*. 2010; 59(10): 3320-3327.

[2] R Schmitz. Use of chaotic dynamical systems in cryptography. *Journal of the Franklin Institute.* 2001; 338(4): 429-441.

[3] Yong Zhang, Jiali Xia, Peng Cai, Bin Chen. Plaintext Related Two-level Secret Key Image Encryption Scheme .TELKOMNIKA. October 2012; 10(6): 1254-1262.

[4] Gang Feng Yan, Xian He Huang. Chaos on Phase Noise of Van Der Pol Oscillator. TELKOMNIKA. December 2010; 8(3): 301-308.

[5] R Tenny, L Tsimring, H Abrabanel. Using distributed nonlinear dynamics for public key encryption . Physical Review Letters. 2003; 90(4): 47903.

[6] Cai Qiong, Peng Tao. One type public key encryption on chebyshev. Software Guide. 2011; 10(1): 162-164.

[7] L Kocarev, Z Tasev. *Public-key encryption based on Chebyshev maps.* ISCAS 2003. IEEE international symposium on circuits systems. Bangkok, Thailand. 2003; 3: 28-31.

[8] G Maze. Algebraic Methods for Constructing One-way Trapdoor Functions. *PhD thesis. Univ. of Notre Dame.* 2003.

[9] K Cheong, T Koshiba. More on Security of Public-Key Cryptosystems Based on Chebyshev Polynomials. *IEEE Trans Circuits and Systems II, Express Briefs*. 2007; 54(9): 795-799.

[10] L Kocarev, J Makraduli, P Amato. Public-Key Encryption Based on Chebyshev Polynomials. *Circuits, Systems and Signal Processing*. 2005; 24(5): 497-517.

[11] Xiaofeng Liao, Fei Chen, Kwok-Wo Wong, et al. On the security of public-key algorithms based on Chebyshev polynomials over the finite field Z_N. *IEEE transactions on computers.* 2010; 59(10): 1392-1400.

[12] P Bergamo, D'Arco P, De Santis A, Kocarev L. Security of public-key encryption based on Chebyshev polynomials. *IEEE Trans. Circuits and Systems I: Regular Papers*. 2005; 52(7): 1382-1393.

[13] Hao Shuxin, Zhao Geng, Xu Gang, Tao Tao. *A New Identity Authentication Scheme Based on Chebyshev Polynomials*. Asia-Pacific Conference on Information Network and Digital Content Security(APCID). Beijing. 2010; 139-133.