cryptography

MDPI

Article

On the Effectiveness of a Common Attack to Chebyshev Chaotic Encryption Scheme

Xiaoqi Liu [1,†] and Guillermo Morales-Luna [2,*,†]

[1] Mathematics & Computer Science Program, Guangdong Technion Israel Institute of Technology, Shantou 515063, China; liu09335@gtiit.edu.cn
[2] Computer Science Department, Cinvestav, Mexico City 07300, Mexico
* Correspondence: gmorales@cinvestav.mx; Tel.: +52-555-747-3759
† These authors contributed equally to this work.

Abstract: Chebyshev polynomials define a rather canonical chaotic cryptosystem and some strong attacks have been designed to that cryptosystem. We report the numerical experiments performed with multiple precision arithmetic using conventional software as GMP and MPFR to test the Chebyshev cryptosystem and Bergamo's attack. As a conclusion, we point out the relevance in the cryptosystem robustness of the number of significant digits (length) of plaintexts and the number of correct digits (precision) of the arithmetical calculations. Furthermore, after the observed experimental results, we use techniques of Numerical Analysis to explain the occurrences of the observed results and to reinforce the importance of the above-mentioned parameters ℓ and m.

Keywords: chaotic encryption; Chebyshev polynomials; Bergamo's attack; multiple precision arithmetic

1. Introduction

The sequence $(T_r(X))_{r\in\mathbb{N}}$ of Chebyshev polynomials provides a typical example of a chaotic system and it was used to obtain a well-known and exhaustively analyzed public-key encryption scheme based on chaos [1–3]. For any index r, $T_r(X)$ has degree r and its restriction to the open real interval $I = (-1, +1)$ has as an image this interval, namely it is an onto map $I \to I$; hence, the corresponding cryptosystem has I as the space of both plaintexts and ciphertexts. Since the sequence of Chebyshev polynomials forms a semigroup, namely for any $r, s \in \mathbb{N}$, $T_r \circ T_s(X) = T_{rs}(X)$, the sequence has served as the basis of several chaotic cryptosystems:

- For encryption, a public key has the form $(x, y) = (x, T_s(x))$ for $x \in I$ and $s \in \mathbb{Z}^+$ is the private key. Then, for a message $\mu \in I$ the ciphertext is $(c_0, c_2) = (T_r(x), \mu \cdot T_r(y))$, where $r \in \mathbb{Z}^+$ is a random index selected by the sender of the message, and for decryption, the owner of the private key calculates $\frac{c_2}{T_s(c_0)}$,
- For key agreement (KA), the classic Diffie–Hellman KA scheme can be translated quite directly by the semigroup property [4],
- For authentication, a general scheme is introduced in [5] in which *servers* should authenticate *client* using a *central registry* (RC). Each server has a key s_j and each client an index r_i and the corresponding Chebyshev polynomials are evaluated on secret numbers owned by the RC; hence, the servers and the clients just know the values of their polynomials at the secret points. Also, there is an interesting application of the sequence of Chebyshev polynomials for Radio Frequency IDentification (RFID), where

check for updates

Academic Editor: Josef Pieprzyk

Received: 7 December 2024
Revised: 18 January 2025
Accepted: 27 January 2025
Published: 29 January 2025

Citation: Liu, X.; Morales-Luna, G. On the Effectiveness of a Common Attack to Chebyshev Chaotic Encryption Scheme. *Cryptography* **2025**, *9*, 10. https://doi.org/10.3390/cryptography9010010

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

the index s is broadcasted by a transceiver and each transponder selects a particular index r and codifies it in order to form an identification label [6],

- For image encryption the chaotic methods have been extensively used as well as some refinements of chaotic maps [7,8].

In [9] the behaviour Chebyshev sequence in modular arithmetic, as well as its impact on the security of cryptosystems have been analyzed within the context of Number Theory.

In 2005 the most common attack to this cryptosystem was introduced [10] (see [4,11] as well). The attack strategy is based on solving equations of the form $T_r(x) = c$ for $x, c \in I$ with respect to index r. To this end, they are considered just real numbers in I with a finite-length decimal representation. For any $m \in \mathbb{Z}^+$, let I_m be the collection of numbers in I whose decimal representation consists of m digits. Thus, by multiplying the involved real numbers by the power 10^m the stated problem gives rise to congruence relations on the integers solvable by Number Theory techniques.

We performed several experiments in order to test Bergamo's attack [10] using multiple precision arithmetic provided by the already conventional software tools GMP [12] and MPFR [13], we report here the results. We consider a *length* $\ell \in \mathbb{Z}^+$ for the specification of plaintexts and a *precision* $m \in \mathbb{Z}^+$, with $\ell \leq m$, for the number of significant digits in decimal representation. The space of plaintexts is coded as the set of words of length ℓ with symbols in $\{0, 1, \ldots, 9\}$, with cardinality 10^ℓ, and the space of ciphertexts is the set of words of length m with symbols in $\{0, 1, \ldots, 9\}$.

Bergamo's attack [10] is effective whenever ℓ is known by the attacker. This attack technique would produce several possible solutions for the recovering secret key depending on the assumed value of the length ℓ. An exhaustive search for the right secret key may still be rather costly. The length ℓ may be part of the private key in the chaotic crypto-scheme.

The purpose of the current report is to illustrate the importance of fixing the length of the plaintexts ℓ and the precision m among the participants in the crypto scenario.

It is worth mentioning that some similar remarks related to the difficulties in making the attack in [10] effective have been pointed out in [5] and the authors consider attacks based on impersonation. In [14], the importance of limiting the numerical domains of Chebyshev polynomials is also pointed out.

In Section 2, we recall the Chebyshev polynomials and in Section 3, we recall the corresponding chaotic cryptosystem and the attack in [10]. In Section 4, we present the performed experiments. Our contribution consists in testing the effectiveness of Bergamo's attack. We have concluded that it is essential for the attacker to know the length of plaintexts and certainly the arithmetical precision should be greater than this length.

2. Chebyshev Polynomials

We use the following notation: we write for any $i, j \in \mathbb{Z}$, $i \leq j$, $[\![i, j]\!] = \{i, \ldots, j\}$, and for any $a, b \in \mathbb{R}$, with $a < b$ we may consider open, semiclosed or closed intervals:

$$\begin{array}{rclcrcl} (a,b) & = & \{x \in \mathbb{R} |\ a < x < b\} & , & (a,b] & = & \{x \in \mathbb{R} |\ a < x \leq b\}\,, \\ [a,b) & = & \{x \in \mathbb{R} |\ a \leq x < b\} & , & [a,b] & = & \{x \in \mathbb{R} |\ a \leq x \leq b\}\,. \end{array}$$

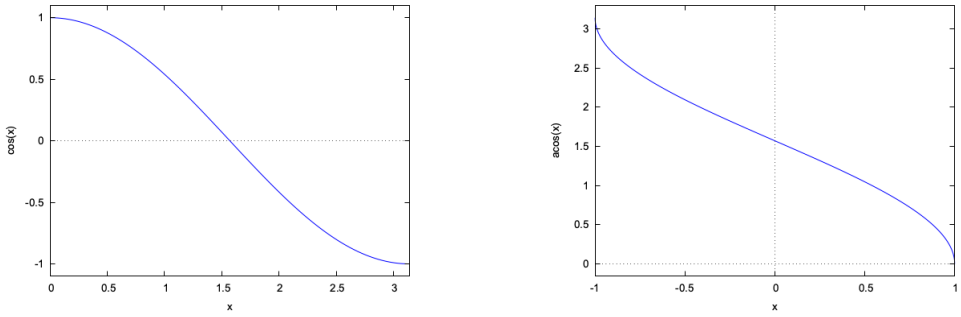
For a radix $R \in \mathbb{Z}^+$, $R > 1$, let $D_R = [\![0, R-1]\!]$ be the collection of R *digits*, then for each $m \in \mathbb{Z}^+$, $D_R[R^{-m}]$ will denote the set of real numbers in $(-1, 1)$ that can be written as polynomial expressions in terms of R^{-m} with coefficients in D_R, in other words $D_R[R^{-m}]$ is the set of fractional numbers than can be written with exactly m digits in radix R.

As a very basic trigonometric identity, we recall that for any $n \in \mathbb{Z}^+$ and any $z \in [0, \pi]$:

$$\cos((n+1)z) + \cos((n-1)z) = 2\cos(z)\ \cos(nz).$$

Furthermore, $\cos(0\,z) = 1$ and $\cos(1\,z) = \cos(z)$. With the change of variable $[0, \pi] \to [-1, +1]$, $z \mapsto x = \cos(z)$ (see Figure 1), the sequence of *Chebyshev polynomials* follows:

$$\begin{array}{rcl} T_0(X) = 1 \quad ; & T_1(X) = X & \\ \forall n \geq 1: \quad T_{n+1}(X) & = & 2XT_n(X) - T_{n-1}(X) \end{array}$$



$z \mapsto x = \cos(z)$ $\qquad$ $x \mapsto z = \arccos(z)$

Figure 1. Change of variable.

Remark 1. *The sequence* $(T_n)_{n\in\mathbb{N}}$ *is a* semigroup, *namely* $\forall n, m: \; T_n \circ T_m = T_{n\cdot m}$.

Indeed, using the above-mentioned change of variables, $\forall x \in [-1, 1]$:

$$T_n \circ T_m(x) = T_n(T_m(x)) = T_n(\cos(mz)) = \cos(n \arccos(\cos(mz))) = \cos(nm\,z) = T_{nm}(x).$$

From the definition of the Chebyshev polynomials,

$$\begin{bmatrix} T_{n-1}(X) \\ T_n(X) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2X \end{bmatrix} \cdot \begin{bmatrix} T_{n-2}(X) \\ T_{n-1}(X) \end{bmatrix}$$

hence

$$\mathbf{t}_n(X) = M(X)^{n-1}\mathbf{t}_1(X)$$

where $\mathbf{t}_n(X) = \begin{bmatrix} T_{n-1}(X) \\ T_n(X) \end{bmatrix}$ and $M(X) = \begin{bmatrix} 0 & 1 \\ -1 & 2X \end{bmatrix}$.

Let us write the powers of matrix $M(X)$ as

$$M(X)^n = \begin{bmatrix} p_{00n}(X) & p_{01n}(X) \\ p_{10n}(X) & p_{11n}(X) \end{bmatrix}.$$

Then

$$\begin{array}{rcl} \begin{bmatrix} p_{00,n+1}(X) & p_{01,n+1}(X) \\ p_{10,n+1}(X) & p_{11,n+1}(X) \end{bmatrix} & = & M(X)^{n+1} \\ & = & M(X) \cdot M(X)^n \\ & = & \begin{bmatrix} 0 & 1 \\ -1 & 2X \end{bmatrix} \cdot \begin{bmatrix} p_{00n}(X) & p_{01n}(X) \\ p_{10n}(X) & p_{11n}(X) \end{bmatrix} \\ & = & \begin{bmatrix} p_{10n}(X) & p_{11n}(X) \\ -p_{00n}(X) + 2X\,p_{10n}(X) & -p_{01n}(X) + 2X\,p_{11n}(X) \end{bmatrix} \end{array}$$

hence

$$\begin{bmatrix} p_{001}(X) & p_{011}(X) \\ p_{101}(X) & p_{111}(X) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2X \end{bmatrix}, \tag{1}$$

$$\begin{bmatrix} p_{00,n+1}(X) & p_{01,n+1}(X) \\ p_{10,n+1}(X) & p_{11,n+1}(X) \end{bmatrix} = \begin{bmatrix} p_{10n}(X) & p_{11n}(X) \\ -p_{00n}(X) + 2X\, p_{10n}(X) & -p_{01n}(X) + 2X\, p_{11n}(X) \end{bmatrix}. \tag{2}$$

From here, it follows that $\forall n \in \mathbb{Z}^+$: $\deg(p_{11n}(X)) = n$.

The powers of the matrix $M(X)$ can be calculated via *square-and-product*:

$$r = \sum_{i=0}^{k} r_i 2^i \implies M(X)^r = \prod_{i=0}^{k} \left(M(X)^{2^i} \right)^{r_i},$$

namely, by the procedure sketched in Algorithm 1.

Algorithm 1 Square-and-product procedure

Input: The matrix $M(X)$ and the integer $r \in \mathbb{Z}^+$

Output: The power $M(X)^r$

1. Express r in binary, $r = (r_k \cdots r_1 r_0)_2$;
2. $P := I_2$ (the identity matrix of order 2×2);
3. $S := M(X)$ (this matrix will be squared at each step);
4. for $i = 0$ to k do
   (a) if $r_i == 1$ then $P := S\,P$;
   (b) $S := SS$;
5. output P

The Chebyshev polynomials are defined in the interval $[-1, +1]$, and the absolute values of the points in this domain are in the unit interval $[0, 1]$.

Consider the conventional decimal radix, $R = 10$. Suppose that $x \in [-1, +1]$ is such that $x \in D_{10}[10^{-m}]$ for some $m \in \mathbb{Z}^+$, then there is an integer $x_m \in [\![-10^m, +10^m]\!]$ such that $x = \frac{x_m}{10^m}$, or $x_m = 10^m x$. Consequently, from relations (1) and (2) it follows that

$$\forall n \in \mathbb{Z}^+ : \; p_{11n}(x) \in D_{10}[10^{-nm}].$$

Furthermore,

$$10^m M(x) = 10^m \begin{bmatrix} 0 & 1 \\ -1 & 2x \end{bmatrix} = \begin{bmatrix} 0 & 10^m \\ -10^m & 2x_m \end{bmatrix} =: N_{x_m} \in \mathbb{Z}^{2\times 2}$$

and

$$\forall n \in \mathbb{Z}^+ : \; M(x)^n = 10^{-nm} N_{x_m}^n.$$

On the other hand, from step 4(a) of the procedure in Algorithm 1

$$SP = \frac{1}{10^{2m}} (10^m S)(10^m P)$$

as well as, from step 4(b),

$$SS = \frac{1}{10^{2m}} (10^m S)(10^m S).$$

Through these relations, the square-and-product algorithm can be performed within multiple precision arithmetic in $[-1, 1]$ using a multiple precision arithmetic on the integers.

Remark 2. *In the current experiments, we calculate the Chebyshev polynomials through the above "square-and-product" method in order to maintain the multiple precision of the calculations through additions and products and to avoid the dependency of implementations of the maps* cos *and* arccos.

3. The Cryptosystem Based on Chebyshev Polynomials

3.1. The General Scheme

The following is a public-key encryption scheme where the plaintext space and the ciphertext space are both the real interval $[-1, 1]$:

Key generation Choose $x \in [-1, 1]$ and $s \in \mathbb{Z}^+$ large enough. The *public* key is $(x, y) = (x, T_s(x))$ and the *private* key is s. See Table 1 below to obtain an estimation of the order of s with respect to the number of exact digits of x and the precision of the used real arithmetic.

Encryption For a plaintext $\mu \in [-1, 1]$, choose a random index $r \in \mathbb{Z}^+$, and calculate $c_0 = T_r(x)$, $c_1 = T_r(y)$, $c_2 = \mu c_1$. The ciphertext is $c = (c_0, c_2)$.

Decryption Given the ciphertext $c = (c_0, c_2)$ recover the message as $\mu = \frac{c_2}{T_s(c_0)}$.

Hence, the time complexity of the encryption procedure is

$$\text{time}_{Ch}(m, r) = O(M(m \cdot r)\, r \log r) \tag{3}$$

where $M(k)$ is the time complexity of the chosen algorithm for multiplication in $D_R[R^{-m}]$, m is the precision of calculations and r is the chosen random number.

Table 1. Relations between length ℓ, precision m and secret key s.

ℓ	m	s	NO	AvDisc	AvTime (s)
100	120	10^8	1815	$2.969 \cdot 10^{-108}$	0.00090
	140	10^{16}	6306	$4.067 \cdot 10^{-112}$	0.00335
	180	10^{32}	24582	$1.167 \cdot 10^{-117}$	0.01414
	240	10^{64}	103427	$4.709 \cdot 10^{-112}$	0.07522
	380	10^{128}	380561	$5.433 \cdot 10^{-119}$	0.46595
	640	10^{256}	1589338	$1.865 \cdot 10^{-113}$	3.92343
	1180	10^{512}	6091120	$2.019 \cdot 10^{-102}$	39.40116
200	220	10^8	1815	$3.132 \cdot 10^{-205}$	0.00113
	240	10^{16}	6306	$8.434 \cdot 10^{-208}$	0.00457
	280	10^{32}	24582	$8.226 \cdot 10^{-215}$	0.02075
	340	10^{64}	103427	$1.024 \cdot 10^{-208}$	0.11672
	480	10^{128}	380561	$1.174 \cdot 10^{-213}$	0.64108
	740	10^{256}	1589338	$1.579 \cdot 10^{-210}$	5.00373
	1280	10^{512}	6091120	$3.132 \cdot 10^{-218}$	43.47840
300	320	10^4	418	$2.051 \cdot 10^{-307}$	0.00045
	340	10^8	1815	$2.314 \cdot 10^{-320}$	0.00202
	340	10^{16}	6306	$2.197 \cdot 10^{-304}$	0.00696
	380	10^{32}	24582	$3.078 \cdot 10^{-311}$	0.03044
	440	10^{64}	103427	$1.150 \cdot 10^{-304}$	0.15059
	580	10^{128}	380561	$4.789 \cdot 10^{-311}$	0.78394
	840	10^{256}	1589338	$8.638 \cdot 10^{-305}$	6.22619
	1380	10^{512}	6091120	$3.156 \cdot 10^{-314}$	46.61281

Table 1. *Cont.*

ℓ	m	s	NO	AvDisc	AvTime (s)
400	420	10^4	418	$1.079 \cdot 10^{-403}$	0.00049
	440	10^8	1815	$1.755 \cdot 10^{-416}$	0.00259
	460	10^{16}	6306	$2.497 \cdot 10^{-420}$	0.00933
	480	10^{32}	24582	$1.720 \cdot 10^{-407}$	0.04019
	560	10^{64}	103427	$7.364 \cdot 10^{-421}$	0.20285
	680	10^{128}	380561	$9.281 \cdot 10^{-407}$	1.03725
	940	10^{256}	1589338	$1.974 \cdot 10^{-402}$	6.52374
	1480	10^{512}	6091120	$6.663 \cdot 10^{-411}$	50.79987

NO: Number of operations (multiplications); Disc: Discrepancy between plaintext and recovered text: Each row displays averaged statistics corresponding to 20 iterations for random selections of the index r in the interval $[\![\frac{s}{2}, s]\!]$.

With respect to the Decryption Procedure let us make a remark:

Suppose that a number x in the open interval $(0,1)$ is written in radix R as $x = \sum_{i=1}^{m_1} x_i R^{-i}$, with digits $x_i \in [\![0, R-1]\!]$, namely $x \in D_R[R^{-m_1}]$. Then, for any $m_0 < m_1$:

$$x = \sum_{i=1}^{m_0} x_i R^{-i} + \sum_{i=m_0+1}^{m_1} x_i R^{-i} =: \overline{x_{m_0}} + \overline{x_{m_1-m_0}},$$

where $\overline{x_{m_0}}$ is the decimal number expressed by the first m_0 digits of x and $\overline{x_{m_1-m_0}} = x - \overline{x_{m_0}}$. Thus

$$\begin{aligned} \overline{x_{m_1-m_0}} &= \sum_{i=m_0+1}^{m_1} x_i R^{-i} \\ &= R^{-(m_0+1)} \sum_{j=0}^{m_1-(m_0+1)} x_{m_0+1+j} R^{-j} \\ &< R^{-(m_0+1)} \sum_{j=0}^{m_1-(m_0+1)} R\, R^{-j} \\ &= R^{-m_0} \frac{1}{1-R^{-1}} \\ &= \frac{1}{R-1} R^{-(m_0-1)} \end{aligned} \tag{4}$$

besides

$$R^{m_1} x = R^{m_1}(\overline{x_{m_0}} + \overline{x_{m_1-m_0}}) = R^{m_1-m_0}(R^{m_0}\overline{x_{m_0}}) + R^{m_1}\overline{x_{m_1-m_0}} \in \mathbb{Z}. \tag{5}$$

Assume now that there are given two numbers $a, b \in D_R[R^{-m_1}]$, $b \neq 0$. By expressing them as in (5),

$$\begin{aligned} R^{m_1} a &= R^{m_1-m_0}(R^{m_0}\overline{a_{m_0}}) + R^{m_1}\overline{a_{m_1-m_0}} \\ R^{m_1} b &= R^{m_1-m_0}\left(R^{m_0}\overline{b_{m_0}}\right) + R^{m_1}\overline{b_{m_1-m_0}} \end{aligned}$$

thus by division there are quotients $q, q_{m_0} \in \mathbb{Z}$ and remainders $r \in [\![0, R^{m_1} b - 1]\!]$, $r_{m_0} \in [\![0, R^{m_0}\overline{b_{m_0}} - 1]\!]$ such that

$$R^{m_1} a = q\, R^{m_1} b + r \tag{6}$$

$$R^{m_0}\overline{a_{m_0}} = q_{m_0}\, R^{m_0}\overline{b_{m_0}} + r_{m_0} \tag{7}$$

From Equation (7),

$$\begin{aligned} R^{m_1}\overline{a_{m_0}} &= R^{m_1-m_0}\, R^{m_0}\overline{a_{m_0}} \\ &= R^{m_1-m_0}\, q_{m_0}\, R^{m_0}\overline{b_{m_0}} + R^{m_1-m_0} r_{m_0} \\ &= q_{m_0}\, R^{m_1}\overline{b_{m_0}} + R^{m_1-m_0} r_{m_0} \end{aligned} \tag{8}$$

by substracting (8) from (6)

$$R^{m_1}\overline{a_{m_1-m_0}} = R^{m_1} a - R^{m_1}\overline{a_{m_0}} = q\, R^{m_1} b + r - (q_{m_0}\, R^{m_1}\overline{b_{m_0}} + R^{m_1-m_0} r_{m_0})$$

hence from (4)

$$R^{m_1}\left(q\, b - q_{m_0}\,\overline{b_{m_0}}\right) + \left(r - R^{m_1-m_0} r_{m_0}\right) < R^{m_1}\frac{1}{R-1}R^{-(m_0-1)} = \frac{R}{R-1}R^{m_1-m_0}$$

and this last inequality entails $q - R^{m_1-m_0} q_{m_0} < R^{m_0}$, consequently the radix-R expression of q and q_{m_0} coincide up to the (m_0-1)-digit.

In summary:

Lemma 1. *If $a, b \in D_R[R^{-m_1}]$, $b \neq 0$, for any $m_0 \in [\![1, m_1 - 1]\!]$ the quotients $\frac{a}{b}$ and $\frac{a_{m_0}}{b_{m_0}}$ coincide up to the (m_0-1)-digit.*

Thus, in the Chebyshev encryption scheme the precision m, which is the number of significant digits in the arithmetical calculations within the interval $[-1, 1]$, shall be greater than the length ℓ of plaintexts since in the decryption process a division is involved. However, due to Lemma 1, for any $m_0 \in [\![\ell + 1, m]\!]$ the recovered plaintext is plausible up to the (m_0-1)-th digit. In order to determine the original plaintext the decipherer should know ℓ.

In an attack on the cryptosystem, given a public key (x, y) and the ciphertext (c_0, c_1), the random index $r \in \mathbb{Z}^+$ such that $T_r(x) = c_0$ should be recovered and the plaintext should be $\mu_m = \frac{c_1}{T_r(y)}$ consisting of m digits where m is the precision of calculations. The original plaintext shall be the ℓ-length prefix μ_ℓ of μ_m. Thus, a brute force procedure to recover ℓ is to look for the first length ℓ such that Encryption$(\mu_\ell, r) = (c_0, c_1)$. According to (3), the cost in time of this procedure is

$$O(m \cdot \text{time}_{Ch}(m, r)) = O(m\, M(m \cdot r)\, r \log r).$$

It is worth mentioning that in this consecutive search, for some tested lengths Bergamo's attack, presented below, fails because the conditions stated in Section 3.2.1 are not satisfied.

3.2. Bergamo's Attack

Let us recall first some basic facts of number theory.

3.2.1. Solving Linear Equations in Remainder Rings

Let $M \in \mathbb{Z}^+$ be a modulus greater than 1 and consider the equation $b\,x = a \bmod M$, with $a, b \in \mathbb{Z}_M$.

- If $b \in \mathbb{Z}_M^*$ then the solution $x = ab^{-1} \bmod M$ is unique.
- If $b \in \mathbb{Z}_M - \{0\}$ and $d = \gcd(b, M) > 1$, the equation has a solution if and only if $d|a$. In this case, express $d = c_0 b + c_1 M$, then for $x_0 = c_0 \frac{a}{d}$ we have

$$b x_0 = b c_0 \frac{a}{d} = (d - c_1 M)\frac{a}{d} = \left(a - \left(c_1 \frac{a}{d}\right) M\right) = a \bmod M,$$

hence x_0 is a solution and all residues of the form $x = x_0 + j\frac{M}{d}$, $j \in [\![0, d-1]\!]$ are solutions as well.

3.2.2. A Number Theory Problem

Consider the following

$$\textbf{Main Problem}: \text{Given } a, b \in (0,1[\text{ find } k \in \mathbb{Z} : \ a + b\,k \in \mathbb{Z} \ \vee \ -a + b\,k \in \mathbb{Z}. \tag{9}$$

We consider in particular the case in which the following condition holds:

$$\begin{array}{l} a, b \text{ have a finite-length representation in a radix, say } R, \text{ namely,} \\ \text{for an } m \in \mathbb{Z}^+,\ a, b \in D_R[R^{-m}]. \end{array} \tag{10}$$

Assume that $a + b\,k = z$ with $z \in \mathbb{Z}$. Then, $(aR^m) + (bR^m)\,k = zR^m$, and this last equation is stated over $\mathbb{Z}$. By writing $a_m = aR^m$ and $b_m = bR^m$ the equation is equivalent to

$$b_m\,k = -a_m \bmod R^m. \tag{11}$$

In fact:

$$b_m\,k = -a_m \bmod R^m \iff b_m\,(R^m - k) = a_m \bmod R^m.$$

Hence, the sign in the right side of (11) is not relevant and just an equation in (9) may be considered.

As mentioned in Section 3.2.1, if $\gcd(b_m, R^m) = 1$ then the unique solution of (11) is $k = -a_m b_m^{-1}$ in $\mathbb{Z}^*_{R^m}$. If $d = \gcd(b_m, R^m) > 1$ and $d = c_0 b_m + c_1 R^m$, only when $d | a_m$ the equation (11) can be solved and its d solutions are given as $-c_0\frac{a_m}{d} + j\frac{R_m}{d}$, $j \in [\![0, d-1]\!]$.

In summary:

Remark 3. *With the above notation, Equation (11) has solutions if $d|a_m$ where $d = \gcd(b_m, R^m)$, and d is the number of solutions.*

If we deal just with rational numbers in $[-1, +1]$ then for any radix $R \in \mathbb{Z}^+$ any such rational number has a periodic R-representation. But for any pair of rational numbers, there is a radix such that those numbers have finite representations for that radix.

Remark 4. *Whenever $a, b \in \mathbb{Q} \cap (0, 1]$, a radix R may be found such that for the main problem, $a, b \in D_R[R^{-m}]$ for some length $m \in \mathbb{Z}^+$.*

Proof. For a rational number $\frac{a}{b} \in (0, 1]$, with $a, b \in \mathbb{Z}^+$, $a < b$, and a radix $R \in \mathbb{Z}$, let $\alpha_0 = a$ and $m_{-1} = 0$. For any current index i, let $k_i \in \mathbb{N}$ be the minimum power such that $\alpha_i R^{k_i} \geq b$ and $m_i = m_{i-1} + k_i$. Then, $\alpha_i R^{k_i} = a_i b + \alpha_{i+1}$ with $a_i \in [\![0, R-1]\!]$ and $\alpha_{i+1} \in [\![0, b-1]\!]$. Thus, the radix-$R$ expression of the rational number $\frac{a}{b}$ consists of 0's except that at each entry m_i there appears the digit a_i. Since the remainder sequence $(\alpha_i)_i$ takes values on the finite set $[\![0, b-1]\!]$, it is eventually periodic and so is the digit sequence $(a_i)_i$. Clearly, the length of the period will be bounded by the denominator b; and if there is an index i such that $\alpha_{i+1} = 0$ then $\frac{a}{b} \in \mathbb{Z}[R^{m_i}]$. Obviously, for $R = b$ such an index exists. □

3.2.3. The Attack

Each Chebyshev polynomial may be written as $T_n(x) = \cos(n \arccos(x))$.

Given the public key (x, y) of Alice, the attacker, Eve, looks for $r \in \mathbb{Z}^+$ such that $T_r(x) = c_0$, then she evaluates $c_1' := T_r(y)$ and she recovers the plaintext as $\mu = \frac{c_2}{c_1'}$.

Consider the sequence

$$C_{xc_0} = \left(\frac{\arccos(c_0) + 2k\pi}{\arccos(x)}\right)_{k\in\mathbb{Z}} \cup \left(\frac{-\arccos(c_0) + 2k\pi}{\arccos(x)}\right)_{k\in\mathbb{Z}} \subset \mathbb{R}.$$

Remark 5. $\forall r \in \mathbb{Z}^+ : \ [T_r(x) = c_0 \iff r \in C_{xc_0}]$.

Proof. Indeed, let $r \in \mathbb{Z}^+$.
$\Rightarrow$) Assume $r \in C_{xc_0}$ and that for some $k \in \mathbb{Z}$, $r = \frac{\arccos(c_0)+2k\pi}{\arccos(x)}$ (the case in which $r = \frac{-\arccos(c_0)+2k\pi}{\arccos(x)}$ is similar because cos is an even function). Then

$$\begin{aligned}\cos(\arccos(x)\, r) &= \cos\left(\arccos(x)\,\frac{\arccos(c_0) + 2k\pi}{\arccos(x)}\right)\\ &= \cos(\arccos(c_0) + 2k\pi)\\ &= \cos(\arccos(c_0))\\ &= c_0\end{aligned}$$

$\Leftarrow$) Assume $T_r(x) = c_0$, then $r\arccos(x) = \arccos(c_0)$ and necessarily $r \in C_{xc_0}$. □

Let $a = \frac{\arccos(c_0)}{\arccos(x)}$ and $b = \frac{2k\pi}{\arccos(x)}$. We may calculate arccos by its Taylor series [15]:

$$\text{For } |x| < 1 : \ \arccos(x) = \frac{\pi}{2} - \sum_{k=0}^{+\infty} \frac{x}{1+2k} \prod_{\kappa=0}^{k-1}\left(x^2\frac{2\kappa+1}{2\kappa+2}\right).$$

Eve's goal consists in recovering $r \in \mathbb{Z}^+$ such that

$$\exists k \in \mathbb{Z} : \ a + bk = r \ \vee \ -a + bk = r, \tag{12}$$

or in an equivalent form, we have:

$$\begin{aligned}\text{Find } k \in \mathbb{Z} : \ &\text{either } (a \bmod 1) + k(b \bmod 1) \in \mathbb{Z}^+\\ &\text{or } \quad -(a \bmod 1) + k(b \bmod 1) \in \mathbb{Z}^+.\end{aligned} \tag{13}$$

Thus, Eve's goal (13) is an instance of the main problem (9) and it can be solved, by the method seen in Section 3.2.2.

As a final remark in this section we point out that if k, r are solutions of (12), then for a sign $\varepsilon \in \{-1, +1\}$,

$$\varepsilon \arccos(c_0) = r\arccos(x) - 2k\pi. \tag{14}$$

4. Experiments

We have used the Multiple Precision Arithmetic provided by GMP: the GNU Multiple Precision Arithmetic Library, operating on signed integers, rational numbers, and floating-point numbers. The main used function categories and data structures are MPZ, the high-level signed integer arithmetic structure, and MPF, the high-level floating-point arithmetic structure. The used version is GMP 6.3.0.

When dealing with Bergamo's attack, we used the High Performed Trigonometric Function provided by MPFR: the GNU Multiple Precision Floating-Point Reliable Library. The used version is MPFR 4.2.1.

The platform was a computer with a processor INTEL(R) CORE(TM) I7-10750H CPU @ 2.60 GHZ and 64-bit operating system UBUNTU 22.04.3 LTS.

The precision of any MPF structure from GMP, namely the number of digits in the decimal part of floating numbers, can be fixed in advance. Since $\log_2(10) \approx 3.2$, N digits preci-

sion shall be required by the built-in function `mpf_set_default_prec(`$\lfloor 3.2 \cdot N \rfloor$`)`, where the function only supports argument with N a multiple of 20. Alternatively, N digits precision may be required by the built-in function `mpfr::mpreal::set_default_prec(`$\lfloor 3.33 \cdot N \rfloor$`)`, for any integer N. Nevertheless, the arithmetic still loses digits beyond the specified precision.

In our experiments, we use two precision parameters:

- ℓ: number of digits to codify plaintexts, we will refer to this parameter as *length*,
- m: precision of arithmetical calculations in GMP and MPFR, we will refer to this parameter as *precision* by itself.

In order to obtain correct crypto operations, for any given length ℓ, the secret keys s and the random indices r shall be bounded from above and the precision m shall be bounded from below (see Table 1 below).

4.1. A Numerical Example for Symmetric Block Ciphering

Let μ be the ASCII simple message:

```
Hi!  I'm Xiaoqi.
Nice to meet you!  ^_^
```

consisting of 38 characters (there is a *Line Feed* at the end of each row). We take $\ell = 100$, and $m = 120$ as we discussed above in this example. Since $\lfloor \frac{100}{8} \rfloor = 12$, we can split the message into 4 sections, 3 consisting of 12 characters and the last of 2 characters. Each ASCII symbol is a byte and we write it in bits. By concatenating these bit strings we codify the message by the following real numbers expressed in radix 10 with $\ell' = 96 = 12 \cdot 8$ "significant digits":

$$\begin{aligned}
x_0 &= \texttt{0.010010000110100100100001001000000100100100100111}\backslash \\
&\quad \texttt{011011010010000001011000011010010110000101101111}, \\
x_1 &= \texttt{0.011100010110100100101110000010100100111001101001}\backslash \\
&\quad \texttt{011000110110010100100000011101000110111001}, \\
x_2 &= \texttt{0.011011010110010101100101011101000010000001111001}\backslash \\
&\quad \texttt{011011110111010100100001001000000101111001011111}, \\
x_3 &= \texttt{0.010111100000101}.
\end{aligned}$$

With secret key:

$$s = \texttt{100000000} = 10^8$$

we calculate the public key $(x, y) = (x, T_s(x))$ where

$$\begin{aligned}
x &= \texttt{0.111111111111111111111111111111111111111111111111111111}\backslash \\
&\quad \texttt{111111111111111111111111111111111111111111111111111111}\backslash \\
&\quad \texttt{111111111111111}, \\
y &= -\texttt{0.188140407999507433650798536496909045920371498910156002573357225608787298742255217531256391471820697}\backslash \\
&\quad \texttt{01208500438767478}.
\end{aligned}$$

In order to cipher, we consider the index

$$r = 50000000 = \frac{10^8}{2}$$

then

$$c_0 = T_r(x) = -0.63712620099964989838379022273202966608309142004703\backslash 95526936628991032039023673808752686692047258891779\backslash 52974562960402237$$

and by making $c_{1j} = x_j \cdot T_r(y)$, for $j \in [\![0,3]\!]$, we obtain the following values for the second entries of the ciphertexts:

$$\begin{aligned} c_{10} &= 0.00944505338390684509033375752569084328870992892825\backslash 01800250586160808163588618939290696190069021278695\backslash 80364061056324859, \\ c_{11} &= 0.01047354514472155582056985297243227574374752323550 1\backslash 68388586972875746923277080577905161530296641668282\backslash 3844352563913556, \\ c_{12} &= 0.01038956814744164785099417750277715028969691768365 0\backslash 71968973418687747841165698228908959059526383578938\backslash 21486076187951678, \\ c_{13} &= 0.00954044737468234174159147508818018407067478580783 3\backslash 26051227973824647541225502961370205012402245410258\backslash 799535033258606012. \end{aligned}$$

In order to decipher, we calculate the values $z_j = \frac{c_{1j}}{c_0}$, for $j \in [\![0,3]\!]$, and we obtain the following values:

$$\begin{aligned} z_0 &= 0.01001000011010010010000100100000010010010010011101101 1\backslash 01001000000101100001101001011000010110111100000000000\backslash 02488501815, \\ z_1 &= 0.01110001011010010010111000001010010011100110100101100 0\backslash 11011001010010000001110100011011110010000000000000000\backslash 02759480021, \\ z_2 &= 0.01101101011001010110010101110100001000000111100101101 1\backslash 11011010100100001001000000101111001011111000000000\backslash 0273735448, \\ z_3 &= 0.01011110000010100000000000000000000000000000000000000 0\backslash 00000000000000000000000000000000000000000000000000\backslash 02513635408. \end{aligned}$$

The original plaintexts are obtained by cutting at length $\ell = 96$.

Remark 6. *In the stated Cryptographic Scheme based on Chebyshev polynomials, two parameters are essential: ℓ which is the* length*, or the number of significant digits of plaintexts, and m which is the* precision*, or the number of correct digits in arithmetical calculations.*

A rough estimation from Table 1 is that for integer k and secret key $s \leq 10^k$, the precision m must be greater or equal than $\ell + 5 \cdot k$.

4.2. Using an Enveloping Technique for Large Plaintexts

Evidently, block ciphering of long plaintexts by this method is excessively inefficient. We may compose symmetric block ciphering with this method through the `OpenSSL` library `EVP` in order to envelop large plaintexts. We take $\ell = 100$, and $m = 120$ in this example. For the private key:

$$s = \texttt{100000000} = 10^8$$

we calculate the public key $(x, y) = (x, T_s(x))$ where

$$\begin{aligned} x &= \texttt{0.11111111111111111111111111111111111111111111111111}\backslash \\ &\quad \texttt{11111111111111111111111111111111111111111111111111}\backslash \\ &\quad \texttt{1111111111111111} \\ y &= -\texttt{0.18814040799950743365079853649690904592037149891015}\backslash \\ &\quad \texttt{60025733572256087872987422552175312563914718206977}\backslash \\ &\quad \texttt{01208500438767478} \end{aligned}$$

and consider an `envelope_key`

$$\begin{aligned} \texttt{ek} &= \texttt{0.01010101010101010101010101010101010101010101010101}\backslash \\ &\quad \texttt{10101010101010101010101010101010101010101010101010} \end{aligned}$$

Then by selecting

$$r = 50000000 = \frac{10^8}{2}$$

we obtain the cipher of `ek` as (c_0, c_1) with

$$\begin{aligned} c_0 &= \texttt{-0.63712620099964989838379022273202966608309142004703}\backslash \\ &\quad \texttt{95526936628991032039023673808752686692047258891779}\backslash \\ &\quad \texttt{52974562960402237} \\ c_1 &= \texttt{0.00953092693167499185383885721016641410637946938121 8}\backslash \\ &\quad \texttt{67286561147343567726511202587408257389424454157364}\backslash \\ &\quad \texttt{058055848041956983} \end{aligned}$$

and effectively

$$\begin{aligned} \frac{c_1}{T_s(c_0)} &= \texttt{0.01010101010101010101010101010101010101010101010101}\backslash \\ &\quad \texttt{01010101010101010101010101010101010101010101010000}\backslash \\ &\quad \texttt{000002511127043} \end{aligned}$$

The envelope key `ek` can be obtained by cutting the decimal representations up to the ℓ-th decimal digit.

4.3. A Numerical Example for Bergamo's Attack

We try here the same example that illustrates the attack in [1]. It is proposed to take:

- *Secret key.* $s = 106000$

- *Public key.* For $\theta_\infty = \frac{5}{18}\pi$, take

$$\begin{aligned} x_\infty &= \cos(\theta_\infty) = \cos\left(\frac{5}{18}\pi\right) \text{ and} \\ y_\infty &= T_s(x_\infty) = \cos\left(s\,\frac{5}{18}\pi\right) = \cos\left(\left(530000+\frac{4}{9}\right)\pi\right) = \cos\left(\frac{4}{9}\pi\right), \end{aligned}$$

  hence (x_∞, y_∞) is the public key.
- *Random exponent for ciphering.* $r = 81500$. Hence

$$\begin{aligned} T_r(x_\infty) &= \cos(r\,\arccos(x_\infty)) = \cos\left(r\,\frac{5}{18}\pi\right) = \cos\left(\left(22638+\frac{8}{9}\right)\pi\right) = \cos\left(\frac{8}{9}\pi\right) \\ T_r(y_\infty) &= \cos(r\,\arccos(y_\infty)) = \cos\left(r\,\frac{4}{9}\pi\right) = \cos\left(\left(36222+\frac{2}{9}\right)\pi\right) = \cos\left(\frac{2}{9}\pi\right) \end{aligned}$$

- *Ciphertext.* For any plaintext $\mu \in (-1, +1]$ the ciphertext is

$$(c_0, c_1) = \left(\cos\left(\frac{8}{9}\pi\right), \mu\,\cos\left(\frac{2}{9}\pi\right)\right).$$

However, these calculations are purely symbolic. All angles involved in the above calculations are irrational numbers; hence, when dealing with them with a computer we just have approximations to their values. For instance, up to 8 digits, we have

$$\begin{aligned} \frac{5}{18}\pi &\approx 0.87266462 =: \theta \\ \cos(\theta) &\approx 0.64278761 =: x \end{aligned}$$

then $|rs\frac{5}{18}\pi - rs\theta| \approx 3.53502559$ consequently $|rs\cos(\frac{5}{18}\pi) - rsx|$ is comparable with $2|x|$. Indeed, for the current example

$$\begin{aligned} T_{rs}(x_\infty) &= \cos\left(rs\cos\left(\frac{5}{18}\pi\right)\right) \approx 0.76604444 \quad \text{while} \\ T_{rs}(x) &= \cos(rsx) \approx -0.95393723 \end{aligned}$$

obviously the calculation of Chebyshev polynomials is highly sensitive to small errors, they are ill-conditioned. In this situation using numerical approximations we have

- *Secret key.* $s = 106000$
- *Public key.* $(x, y) = (x, T_r(x)) = (0.64278760\ldots, 0.17364817\ldots)$
- *Ciphering.* Take $r = 81500$. Then, for any plaintext $\mu \in (-1, +1]$, the ciphertext is

$$(c_0, c_1) = (T_r(x), \mu T_r(y)) = (-0.93969262\ldots, \mu \cdot 0.76604444\ldots).$$

Then, the attacker Eve should solve the problem (13) with

$$a = 3.2000000000\ldots \text{ and} \tag{15}$$
$$b = 7.1999999999\ldots \tag{16}$$

in decimal representation, namely with radix $R = 10$, where those coefficients are obtained according to the relations stated before problem (13).

Suppose that the values (15) and (16) are cut up to the m-th digit. According to the method described in Section 3.2.2, condition (10) does hold and consequently the problem is reduced to solve Equation (11), which, by Remark 3, has solutions only when

$$d|a \quad \text{where } d = \gcd(b, 10^m). \tag{17}$$

We proposed two cases of different precision covering the two situations listed in Bergamo's algorithm that result in the correct r', such that $T_{r'}(x) = T_r(x)'$

Case $m = 20$. In this case, the coefficients in Equation (11) are

$$\begin{aligned} a_{20} &= 0.20000000000000070759, \\ b_{20} &= 0.19999999999999999999 \end{aligned}$$

and $\gcd(10^{20}b_{20}, 10^{20}) = 1$ with

$$(10^{20}b_{20})^{-1} = 79999999999999999999$$

The unique solution of the corresponding instance of Equation (11) is

$$k_{20} = (10^{20}a_{20})\,(10^{20}b_{20})^{-1} \bmod 10^{20} = 70759.$$

Hence, the corresponding iteration r such that $T_r(x) = c_0$ is

$$r_{20} = a + k_{20} \cdot b = 509468.$$

Comparing with the original index $r = 81500$, we obtain

$$\begin{aligned} T_{r_{20}}(x) &= -0.939692620785909704047, \\ T_r(x) &= -0.939692620785908595211, \\ error &= -1.1088359707509228942 \cdot 10^{-15}. \end{aligned}$$

Case $m = 97$. In this case, the coefficients in Equation (11) are

$$\begin{aligned} a_{97} &= 0.19999999999999999999999999999999999999999999999999\backslash \\ &\quad 99999999999999999999999999999999999999999884368, \\ b_{97} &= 0.19999999999999999999999999999999999999999999999999\backslash \\ &\quad 99999999999999999999999999999999999999999999999996, \end{aligned}$$

and $(10^{97}b_{97})x_{97} + 10^{97}y_{97} = 4 = \gcd(10^{97}b_{97}, 10^{97})$ with

$$\begin{aligned} x_{97} &= 94999999999999999999999999999999999999999999999999\backslash \\ &\quad 99999999999999999999999999999999999999999999999999, \\ y_{97} &= -18999999999999999999999999999999999999999999999999\backslash \\ &\quad 99999999999999999999999999999999999999999999999996, \end{aligned}$$

and the solutions of the corresponding instance of Equation (11) are

$$\begin{aligned} k_{97,0} &= x_{97} \cdot \frac{10^{97}a_{97}}{4} \bmod 10^{97} \\ &= 39999999999999999999999999999999999999999999999999\backslash \\ &\quad 99999999999999999999999999999999999999999971092 \end{aligned}$$

and $k_{97,i} = k_{97,0} + i \cdot \frac{10^{97}}{4}$.

Hence, there are four corresponding iterations $r_{97,i}$ such that $T_{r_{97,i}}(x) = c_0$ and they are $r_{97,i} = a + k_{97,i} \cdot b, \ \ i \in [\![0,3]\!]$. Take

$$\begin{aligned} r_{97,0} &= a + k_{97,0} \cdot b \\ &= 879999999999999999999999999999999999999999999999\backslash \\ &\quad 999999999999999999999999999999999999999999791864. \end{aligned}$$

Comparing with the original index $r = 81500$, we obtain

$$\begin{aligned} T_{r_{97,0}}(x) &= -0.798375679639044035429574245118989681891634222656\backslash \\ &\quad 377142598927189655757477360445921762739820205098 1, \\ T_r(x) &= -0.939692620785908384054109277324731469936208134264\backslash \\ &\quad 464633090286662774221210995889458949745889838705 8, \\ error &= 0.40738331968914697735. \end{aligned}$$

While applying the symbolic computation to $r_{97,0}$, we get indeed a correct solution:

$$\begin{aligned} T_r(x) &= \cos(r \arccos(x)) = \cos\left(r \frac{5}{18}\pi\right) \\ &= \cos\left(\left(22638 + \frac{8}{9}\right)\pi\right) = \cos\left(\frac{8}{9}\pi\right), \\ T_{r_{97,0}}(x) &= \cos(r_{97,0} \arccos(x)) = \cos\left(r_{97,0} \frac{5}{18}\pi\right) \\ &= \cos((24444444444444444444444444444444444444 \\ &\quad 44444444444444444444444444444444444444 \\ &\quad 44444444444444386628 + \frac{8}{9}\pi \\ &= \cos(\frac{8}{9}\pi). \end{aligned}$$

Thus, for different values of m it may happen that either there are no solutions of Equation (11) ($d \not| (10^m a_m)$) or there are several solutions ($d > 1$ and $d | (10^m a_m)$). And as m increases, the solution r' increases as well. However, when r' is much larger than r, the calculations of Chebyshev polynomials may differ from that of the original index due to precision error propagation.

However, as reported in [10], Bergamo's attack succeeds by considering symbolic, not numerical, calculations:

Suppose that for some rational number $q \in \mathbb{Q}$, $x = \cos(q\pi)$. Then, according to the selection of coefficients a, b, as in Equation (12), $a = \frac{r\,\varepsilon q\pi}{q\pi} = \varepsilon r$, for a sign $\varepsilon \in \{-1, +1\}$, and $b = \frac{2k\pi}{q\pi} = 2\frac{k}{q}$. Then, by selecting a radix R such that both a and b have finite R-radix representation, as stated in Remark 4, Bergamo's attack would succeed in recovering the index r.

Besides the above example in which Bargamo's attack may fail to recover the original index r, which is the random index when ciphering in Chebyshev's chaotic cryptosystem, we also remark on the following:

By considering the relation (14), we have that the left side lies in the real interval $[0, \pi]$ while the right side is the difference of two big numbers $r \arccos(x), 2k\pi$, of the same order of magnitude. Depending on the chosen precision m of the computing platform this difference can be great when the true values of the operands are approximated. Hence,

this difference may differ too much from the right side and that may provoke that for the alleged recovered index r, $c_0 \neq \cos(r \arccos(x) - 2k\pi)$ and consequently $c_0 \neq T_r(x)$.

5. Conclusions

Due to the chaotic behaviour of Chebyshev polynomials, the breaking attack in [10] may fail if the length ℓ, the number of significant digits, in plaintexts, is unknown. Hence, ℓ must be a part of the secret key. The precision m, the number of significant digits in arithmetical calculations, may be part of the public key.

In order to evaluate the effectiveness of Bergamo's attack, we implemented numerical experiments utilizing GMP and MPFR. The square-and-product procedure is suitable for performing efficient calculations within the Chebyshev cryptographic scheme and the above-mentioned software packages manage efficiently the required multiple precision arithmetic. In Lemma 1 of the current paper, we emphasize the importance of knowing the length of plaintext in advance. The parameters ℓ (length of plaintexts) and m (precision in numerical calculations), influence greatly the effectiveness of both decryption and attack procedures. Due to the chaotic behaviour of Chebyshev polynomials, the breaking attack in [10] may fail if the length ℓ, the number of significant digits, in plaintexts is unknown. Hence, ℓ must be a part of the secret key. But also, the precision m, the number of significant digits in arithmetical calculations, should be part of the public key for effective decryption.

Author Contributions: G.M.-L. proposed the experiments, X.L. made the implementations and performed the experiments. Writing the report was a joint task. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All C/C++ programs developed for the current experimentation are available https://github.com/ChillingLiu/Chebyshev-Polynomial-based-Cryptosystem, accessed on 1 January 2025.

Acknowledgments: These experiments were realized during the stay at the Guangdong Technion Israel Institute of Technology (GTIIT) of the second author Morales-Luna kindly acknowledges the academic and administrative authorities of GTIIT for their hospitality and the great teaching and research environment in GTIIT.

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

References

1. Kocarev, L. Chaos-based cryptography: A brief overview. *IEEE Circuits Syst. Mag.* **2001**, *1*, 6–21. [CrossRef]
2. Kocarev, L.; Makraduli, J.; Amato, P. Public-Key Encryption Based on Chebyshev Polynomials. *Circuits Syst. Signal Process.* **2005**, *24*, 497–517. [CrossRef]
3. Mishkovski, I.; Kocarev, L. Chaos-Based Public-Key Cryptography. In *Chaos-Based Cryptography: Theory, Algorithms and Applications*; Kocarev, L., Lian, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 27–65. [CrossRef]
4. Xiao, D.; Liao, X.; Deng, S. A novel key agreement protocol based on chaotic maps. *Inf. Sci.* **2007**, *177*, 1136–1142. [CrossRef]
5. Ryu, J.; Kang, D.; Won, D. Improved Secure and Efficient Chebyshev Chaotic Map-Based User Authentication Scheme. *IEEE Access* **2022**, *10*, 15891–15910. [CrossRef]
6. Kardaş, S.; Genç, Z.A. Security Attacks and Enhancements to Chaotic Map-Based RFID Authentication Protocols. *Wirel. Pers. Commun.* **2018**, *98*, 1135–1154. [CrossRef]
7. Jiang, M.; Yang, H. Image Encryption Algorithm Using Multi-Level Permutation and Improved Logisticc-Chebyshev Coupled Map. *Information* **2023**, *14*, 456. [CrossRef]
8. Jiang, M.; Yang, H. Image Encryption Using a New Hybrid Chaotic Map and Spiral Transformation. *Entropy* **2023**, *25*, 1516. [CrossRef] [PubMed]
9. Chen, F.; Liao, X.; Xiang, T.; Zheng, H. Security analysis of the public key algorithm based on Chebyshev polynomials over the integer ring ZN. *Inf. Sci.* **2011**, *181*, 5110–5118. [CrossRef]

10. 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]
11. Yoshioka, D. Security of Public-Key Cryptosystems Based on Chebyshev Polynomials over Z/pkZ. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**, *67*, 2204–2208. [CrossRef]
12. Free Software Foundation. GMP: The GNU Multiple Precision Arithmetic Library. Available online: https://gmplib.org/ (accessed on 1 December 2024).
13. Fousse, L.; Hanrot, G.; Lefèvre, V.; Pélissier, P.; Zimmermann, P. MPFR: A multiple-precision binary floating-point library with correct rounding. *ACM Trans. Math. Softw.* **2007**, *33*, 13-es. [CrossRef]
14. Cheong, K.Y. One-Way Functions from Chebyshev Polynomials. Cryptology ePrint Archive, Paper 2012/263. 2012. Available online: https://eprint.iacr.org/2012/263 (accessed on 1 December 2024).
15. WolframAlpha. ArcCos Taylor Series. Available online: https://www.wolframalpha.com/input/?i=taylor+series+arccos&lk=3 (accessed on 1 December 2024).

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.