

MARCIN LAWNIK 
ADRIAN KAPCZYŃSKI 

APPLICATION OF MODIFIED CHEBYSHEV POLYNOMIALS IN ASYMMETRIC CRYPTOGRAPHY

Abstract *Based on Chebyshev polynomials, one can create an asymmetric cryptosystem that allows for secure communication. Such a cryptosystem is based on the fact that these polynomials form a semi-group due to the composition operation. This article presents two new cryptosystems based on modifications of Chebyshev's polynomials. The presented analysis shows that their security is the same as in the case of algorithms associated with the problem of discrete logarithms. The article also shows methods that allow for the faster calculation of Chebyshev polynomials.*

Keywords asymmetric encryption, Chebyshev polynomials, chaos

Citation Computer Science 20(3) 2019: 289–303

1. Introduction

In the modern world, information resources play a key role in every aspect of our everyday life. Data is downloaded and accessed by almost all electronic devices; however, certain data is only valuable if it is kept secret from everyone except those who are authorized to access it. For this reason, many methods have been created primarily for securing data, such as cryptography, steganography (concealing information in objects; e.g., [16, 34]), biometrics (using human anthropometric and behavioral features; e.g., [3, 20]), etc.

Among the above-mentioned techniques, the most common is cryptography; this is currently one of the most dynamically developing fields of science related to information security. In cryptography, a key role is played by data encryption; this can be divided into symmetric algorithms (the same encryption and decryption key; e.g., DES [10] and AES [12]) and asymmetric algorithms (various keys for encryption and decryption; e.g., DH [13], RSA [32], ElGamal [14]). Currently, one of the most noticeable trends in security engineering is cryptography based on chaos theory. This uses chaotic projections such as logistic maps [25, 26] or piecewise linear maps [26, 27] to generate a sequence of values that are similar to random ones on one hand and obtained in a deterministic way on the other. As part of chaotic cryptography, we can distinguish symmetric algorithms [4] and asymmetric algorithms [31].

This article is about chaotic asymmetric cryptography, which uses Chebyshev's polynomials (among others) as a basic mathematical tool for generating keys and carrying out the encryption process. The article is divided into main five parts: (1) introduction, with a brief overview of the subject of the article; (2) preliminaries and related work, with a detailed review of issues related to Chebyshev polynomials in cryptography; (3) Chebyshev's polynomials over \mathbb{Z}_p , where modified Chebyshev polynomials are formulated; (4) algorithms of new cryptosystems, where new cryptosystems were defined and their security and efficiency shown; and finally (5) the summary and references.

2. Preliminaries and related work

Asymmetric cryptography based on chaos theory uses Chebyshev $T_n(x)$ I-type polynomials, which can be defined as follows [33]:

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

where $n \in \mathbb{N}$, or through the same relationship:

$$T_n(x) = \cos(n\theta) \quad (2)$$

where $x = \cos \theta$. Polynomials (1) can be determined alternatively by the following recursive relationship [33]:

$$T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x) \quad (3)$$

where $T_0(x) = 1$ and $T_1(x) = x$. Polynomials (1) have many interesting properties that make them extremely useful. For instance, one can include the fact that these polynomials (considered as a dynamic system) are a chaotic map with the Lyapunov exponent equal to $\ln n$ [31].

In addition, for polynomials (1), among others, the following property occurs [5]:

$$T_n(T_m(x)) = T_{nm}(x) \quad (4)$$

Dependency relationship (4) was used in [22] to create a cryptosystem that can be used by Alice and Bob for secure transmission. In the literature, this cryptosystem is known as Kocarev's cryptosystem and consists of three procedures: the key-generation algorithm, the encryption algorithm, and the decryption algorithm (see Algorithms 1 and 2).

Algorithm 1. Alice key creation algorithm

Data: $s \in \mathbb{Z}; x \in [-1, 1]$

Result: $T_s(x) \in [-1, 1]$

1. set a large integer number s
2. choose a random value $x \in [-1, 1]$ and calculate $T_s(x)$
3. Alice's private key is s , while the public key $(x, T_s(x))$

Algorithm 2. Bob encryption algorithm

Data: $r \in \mathbb{Z}; x, T_s(x), M \in [-1, 1]$

Result: $C, T_r(x) \in [-1, 1]$

1. choose Alice's public key $(x, T_s(x))$
2. message M present as a numerical value, $M \in [-1, 1]$
3. set a large integer number r
4. calculate $T_r(x), T_r(T_s(x)) = T_{rs}(x), C = MT_{rs}(x)$
5. send to Alice $(T_r(x), C)$

After receiving the message from Bob, Alice can decrypt the message by conducting the following steps of Algorithm 3.

Algorithm 3. Alice's decryption algorithm

Data: $s \in \mathbb{Z}; T_r(x), C \in [-1, 1]$

Result: $M \in [-1, 1]$

1. using s , calculate $T_s(T_r(x)) = T_{sr}(x)$.
2. recover message M calculating $M = C/T_{sr}(x)$

This cryptosystem is a modification of the ElGamal algorithm as well as the Diffie-Hellman key exchange protocol (DH). However, it turns out that it is not secure [7]. Knowing Alice's public key $(x, T_s(x))$ and Bob's cryptogram $(T_r(x), C)$, Eve can perform the attack [7] (see Algorithm 4).

Algorithm 4. Bergamo attack

Data: $r, \bar{r} \in \mathbb{Z}; T_r(x), T_s(x), C \in [-1, 1]$

Result: $M \in [-1, 1]$

1. find such \bar{r} , that $T_{\bar{r}}(x) = T_r(x)$
2. calculate $T_{\bar{r}}(T_s(x)) = T_{\bar{r}s}(x)$
3. recover message M calculating $M = C/T_{\bar{r}s}(x)$

Keeping Bergamo’s attack in mind, Kocarev has described Chebyshev’s polynomials (1) above ring \mathbb{Z}_p [21]:

$$T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x) \pmod p \tag{5}$$

where p is strong prime, $T_0(x) = 1$, and $T_1(x) = x \pmod p$. Then, he built a new cryptosystem that uses exactly the same algorithms; however, all operations are performed in \mathbb{Z}_p [21]. This also means x and M belong to set \mathbb{Z}_p . The security of such a defined cryptosystem with polynomials (5) is the same as in the case of the discrete logarithm problem. This is due to the computationally difficult problem, which is the calculation of n from $a = T_n(x)$ (where a is known). The solution is [15]:

$$n = \log_{x+\sqrt{x^2-1}} a + \sqrt{a^2 - 1} \tag{6}$$

Dependency (6) can be called *the problem of Chebyshev’s discrete logarithm*.

However, it turns out that Kocarev’s cryptosystem with polynomials (5) is not secure in special cases. In [29], one can find the attack that uses the periodicity of polynomials (5). This is only effective in a certain case; however, it shows that the analyzed Kocarev’s cryptosystem has some weaknesses.

In [35], one can find the information about the modified Kocarev’s cryptosystem with (5) by introducing additional secret values K_i , which are used to scale the value of the ciphertext; i.e., the encryption now proceeds according to procedure $C = K_iMT_{rs}(x)$. These values are obtained by solving Equation (7).

$$K_i = \begin{cases} K_1, & 0 \leq T_{sr}(x) \pmod n \leq \frac{p}{n} \\ K_2, & \frac{p}{n} \leq T_{sr}(x) \pmod n \leq \frac{2p}{n} \\ \vdots & \vdots \\ K_i, & \frac{(i-1)p}{n} \leq T_{sr}(x) \pmod n \leq \frac{ip}{n} \\ \vdots & \vdots \\ K_n, & \frac{(n-1)p}{n} \leq T_{sr}(x) \pmod n \leq p \end{cases} \tag{7}$$

Polynomials (5) were also used to create a cipher using Alice’s e-mail (for example) as its public key (this is the so-called Identity-based encryption). Such an algorithm can be found in the literature [2]; however, it turned out that it is not secure enough. Its cryptanalysis was carried out and described in [18].

In [37], Chebyshev polynomials were used to encrypt the images. The presented results show that the cryptosystem with polynomials (5) is an interesting alternative to the traditional ElGamal algorithm. Moreover, a key exchange protocol was presented in [40], which was intended to significantly improve the security and effectiveness of using Chebyshev polynomials. Nevertheless, its cryptanalysis was carried out in [39]. A similar approach can be found in [8].

What is more, polynomials (4) were used to create a digital signature [17] as well as the authentication protocol [23, 36]. In addition, Chebyshev polynomials have also found application in creating a one-way function. Such an algorithm was described in [9, 19].

The literature also included the problem of the quick calculation of the value of Chebyshev polynomials. Due to the size of the key, this operation should be fast enough. In [1, 28], several algorithms are given that address this issue.

In [30], polynomials $T_n(x)$ were defined in a finite body $GF(p)$. In addition, the authors used the defined polynomials to determine the new version of Kocarev's cryptosystem given by use of algorithms (1)–(3).

The performed literature review indicates a considerable interest in the use of Chebyshev polynomials of the first kind in asymmetric cryptography.

In addition to polynomials (1), one can find Chebyshev polynomials of the second kind $U_n(x)$ in the literature, which can be determined by Equation (8).

$$U_n(x) = \frac{\sin((n+1)\theta)}{\sin\theta} \quad (8)$$

Furthermore, one can find the so-called modified Chebyshev polynomials in the literature [38]; that is, Vieta-Lucas polynomials $\Omega_n(x)$ and Vieta-Fibonacci polynomials $V_n(x)$ defined by Equations (9) and (10).

$$\Omega_n(x) = 2T_n\left(\frac{x}{2}\right) \quad (9)$$

$$V_n(x) = U_n\left(\frac{x}{2}\right) \quad (10)$$

Polynomials $U_n(x)$, $\Omega_n(x)$, and $V_n(x)$ can be described using the following recursive relationships:

$$U_{n+2}(x) = 2xU_{n+1}(x) - U_n(x) \quad (11)$$

where $U_0(x) = 1$ and $U_1(x) = 2x$;

$$\Omega_{n+2}(x) = x\Omega_{n+1}(x) - \Omega_n(x) \quad (12)$$

where $\Omega_0(x) = 2$ and $\Omega_1(x) = x$;

$$V_{n+2}(x) = xV_{n+1}(x) - V_n(x) \quad (13)$$

where $V_0(x) = 1$ and $V_1(x) = x$.

For the above polynomials the following properties (among others) occur [5, 38] (Equations (14)–(16)):

$$\Omega_n(\Omega_m(x)) = \Omega_{nm}(x) \quad (14)$$

$$U_{m-1}(T_n(x))U_{n-1}(x) = U_{nm-1}(x) \quad (15)$$

$$V_{n-1}(\Omega_m(x))V_{m-1}(x) = V_{nm-1}(x) \quad (16)$$

Dependence (14) served in [24] to create Kocarev's cryptosystem using polynomials (9).

Analyzing the literature concerning the use of Chebyshev polynomials in asymmetric cryptography, we can conclude that the presented algorithms can be extended using the dependence of (15), which has not been used for this purpose. In addition, the modified Chebyshev polynomials have properties that are analogous to their classic equivalents; therefore, it is reasonable to construct a cryptosystem based on (16). To achieve this goal, it is necessary to specify Polynomials (8)–(10) over the ring, as in the case of Polynomials (5).

3. Chebyshev polynomials over \mathbb{Z}_p

Polynomials $U_n(x)$, $\Omega_n(x)$, and $V_n(x)$ can be specified over ring \mathbb{Z}_p . Their definitions are as follows:

$$U_{n+2}(x) = 2xU_{n+1}(x) - U_n(x) \pmod p \quad (17)$$

where $U_0(x) = 1$ and $U_1(x) = 2x \pmod p$;

$$\Omega_{n+2}(x) = x\Omega_{n+1}(x) - \Omega_n(x) \pmod p \quad (18)$$

where $\Omega_0(x) = 2$ and $\Omega_1(x) = x \pmod p$;

$$V_{n+2}(x) = xV_{n+1}(x) - V_n(x) \pmod p \quad (19)$$

where $V_0(x) = 1$ and $V_1(x) = x \pmod p$.

For the specified Chebyshev polynomials, the dependencies still remain true (14)–(16). Based on these, the above polynomials can be used in asymmetrical cryptography. This is presented in a later part of this paper.

4. Algorithms of new cryptosystems

The presented modified Chebyshev polynomials (17)–(19) can be used in two ways in asymmetric cryptography:

- use (15) dependency for polynomials specified by (5) and (17);
- use dependencies (16) for polynomials specified by (18) and (19).

4.1. Cryptosystem I: Polynomials $T_n(x)$ and $U_n(x)$

The dependence of (15) on polynomials and defined by means of (5) and (17) allows us to create a new cryptosystem, which consists of the following three algorithms (Algorithms 5, 6, and 7).

Algorithm 5. Alice key-generation algorithm

Data: p – strong prime; $s \in \mathbb{Z}$; $x \in \mathbb{Z}_p$

Result: $T_s(x), U_{s-1}(x) \in \mathbb{Z}_p$

1. set a large integer number s
2. choose a random value $x \in \mathbb{Z}_p$ and calculate $T_s(x)$ and $U_{s-1}(x)$
3. Alice's private key is s , while the public key is $(x, T_s(x), U_{s-1}(x))$

The encryption algorithm used to communicate between Bob and Alice looks like this:

Algorithm 6. Bob encryption algorithm

Data: p – strong prime; $r \in \mathbb{Z}$; $x, T_s(x), U_{s-1}(x), M \in \mathbb{Z}_p$

Result: $C \in \mathbb{Z}$; $T_r(x), U_{r-1}(x) \in \mathbb{Z}_p$

1. choose Alice's public key $(x, T_s(x), U_{s-1}(x))$
2. message M present as a numerical value, $M \in \mathbb{Z}_p$
3. set a large integer number r
4. calculate $T_r(x), U_{r-1}(x), U_{r-1}(T_s(x))U_{s-1}(x) = U_{rs-1}(x), C = MU_{rs-1}(x)$
5. send to Alice $(T_r(x), U_{r-1}(x), C)$

After receiving the message from Bob, Alice can decrypt the message by following the steps in Algorithm 7.

Algorithm 7. Alice's decryption algorithm

Data: $s \in \mathbb{Z}$; $T_r(x), U_{r-1}(x) \in \mathbb{Z}_p$; $C \in \mathbb{Z}$

Result: $M \in \mathbb{Z}_p$

1. using s , calculate $U_{s-1}(T_r(x))U_{r-1}(x) = U_{sr-1}(x)$
2. recover message M calculating $M = C/U_{sr-1}(x)$

The above cryptosystem is correct, as Alice and Bob both share the same value:

$$U_{s-1}(T_r(x))U_{r-1}(x) = U_{sr-1}(x) = U_{rs-1}(x) = U_{r-1}(T_s(x))U_{s-1}(x) \quad (20)$$

4.1.1. Security of cryptosystem

Attempting to break the cryptosystem described with the above algorithms, Eve must find s or r and then calculate the shared value. Without a loss of generality (assuming that Eve wants to find s), she has the following dependencies.

$$a = T_s(x) \quad (21)$$

$$b = U_{s-1}(x) \quad (22)$$

Dependence (21) leads to the problem of the discrete logarithm of Chebyshev, which is described in the literature featured with (6). So, s can be found from Equation(23).

$$s = \log_{x+\sqrt{x^2-1}} a + \sqrt{a^2 - 1} \quad (23)$$

In the case of (22), the solution for s we are looking for is:

$$s = \log_{x+\sqrt{x^2-1}} \left(b\sqrt{x^2 - 1} + \sqrt{b^2(x^2 - 1) + 1} \right) \quad (24)$$

Dependence (24) results from the following transformations:

$$b = U_{s-1}(x) = \frac{\sinh(s \cosh^{-1} x)}{\sinh \cosh^{-1} x} \quad (25)$$

Then, using the following dependencies:

$$\sinh^2 x - \cosh^2 x = 1 \quad (26)$$

expression (25) takes the following form:

$$b = \frac{\sinh(s \cosh^{-1} x)}{\sqrt{x^2 - 1}} \quad (27)$$

After the algebraic transformation, we obtain a solution:

$$s = \frac{\sinh^{-1}(b\sqrt{x^2 - 1})}{\cosh^{-1} x} \quad (28)$$

which, after using the following equation:

$$\sinh^{-1} x = \ln(x + \sqrt{x^2 + 1}) \quad (29)$$

$$\cosh^{-1} x = \ln(x + \sqrt{x^2 - 1}) \text{ for } x \geq 1 \quad (30)$$

takes its final form (24).

The performed analysis shows that both (23) and (24) lead to the problem of discrete logarithms. The security of the cryptosystem described in Algorithms 5–7 is, therefore, the same as in the case of Kocarev's cryptosystem as well as classical algorithms known from the literature (such as the ElGamal cipher).

4.1.2. Efficiency

The definitions of recursive polynomials (5) and (17) differ only in their initial states. For this reason, the methods allowing for the quick calculation of polynomials $T_n(x)$ will also be appropriate for polynomials $U_n(x)$. In addition, there are dependencies that allow us to associate $T_n(x)$ and $U_n(x)$. An example of such a dependency is [6]:

$$2T_n(x) = U_n(x) - U_{n-2}(x) \quad (31)$$

As compared to Kocarev's cryptosystem, the presented cryptosystem is slightly slower in the phase of determining the shared value after taking those methods into account that allow for the quick calculation of polynomials $T_n(x)$ and $U_n(x)$. This is due to the necessity of counting the values for both polynomials as well as the greater number of operations that are necessary in calculating the value of a given relationship with Dependency (15). This means that the Kocarev's cryptosystem can be successfully replaced with Algorithms 5–7.

4.2. Cryptosystem II: Polynomials $\Omega_n(x)$ and $V_n(x)$

Polynomials $\Omega_n(x)$ and $V_n(x)$ have the following Property (16), which is analogous to Property (15) for polynomials $T_n(x)$ and $U_n(x)$. This means that the cryptosystem described in the previous subsection described with Algorithms 5–7 can be modified by replacing polynomials $T_n(x)$ and $U_n(x)$ with $\Omega_n(x)$ and $V_n(x)$. The modified cryptosystem looks like this (Algorithms 8–10):

Algorithm 8. Alice key-generation algorithm

Data: p – strong prime; $s \in \mathbb{Z}$; $x \in \mathbb{Z}_p$

Result: $\Omega_s(x), V_{s-1}(x) \in \mathbb{Z}_p$

1. set a large integer number s
2. choose a random value $x \in \mathbb{Z}_p$ and calculate $\Omega_s(x)$ and $V_{s-1}(x)$
3. Alice's private key is s , while the public key is $(x, \Omega_s(x), V_{s-1}(x))$

The encryption algorithm used to communicate between Bob and Alice is as follows:

Algorithm 9. Bob encryption algorithm

Data: p – strong prime; $r \in \mathbb{Z}$; $x, \Omega_s(x), V_{s-1}(x), M \in \mathbb{Z}_p$

Result: $C \in \mathbb{Z}$; $\Omega_r(x), V_{r-1}(x) \in \mathbb{Z}_p$

1. choose Alice's public key $(x, \Omega_s(x), V_{s-1}(x))$
2. message M present as a numerical value, $M \in \mathbb{Z}_p$
3. set a large integer number r
4. calculate $\Omega_r(x), V_{r-1}(x), V_{r-1}(\Omega_s(x))V_{s-1}(x) = V_{rs-1}(x), C = MV_{rs-1}(x)$
5. send to Alice $(\Omega_r(x), V_{r-1}(x), C)$

After receiving the message from Bob, Alice can decrypt it by following the steps of the following algorithm:

Algorithm 10. Alice's decryption algorithm

Data: $s \in \mathbb{Z}$; $\Omega_r(x), V_{r-1}(x) \in \mathbb{Z}_p$; $C \in \mathbb{Z}$

Result: $M \in \mathbb{Z}_p$

1. using s , calculate $V_{s-1}(\Omega_r(x))V_{r-1}(x) = V_{sr-1}(x)$
2. recover message M calculating $M = C/V_{sr-1}(x)$

The above cryptosystem is correct, because both Alice and Bob share the same value:

$$V_{s-1}(\Omega_r(x))V_{r-1}(x) = V_{sr-1}(x) = V_{rs-1}(x) = V_{r-1}(\Omega_s(x))V_{s-1}(x) \quad (32)$$

4.2.1. Security of cryptosystem

In order to break the cryptosystem described by Algorithms 8–10, Eve must find s or r and then calculate the shared value. Without a loss of generality (assuming that Eve wants to find s), she has the following dependencies:

$$a = \Omega_s(x) \quad (33)$$

$$b = V_{s-1}(x) \quad (34)$$

After carrying out the appropriate transformations, the solution of the above dependencies leads to the discrete logarithm of Chebyshev (similar to the cryptosystem from the previous subsection). Thus, the cryptographic power of the cryptosystem defined by Algorithms 8–10 is the same as in the case of the Kocarev cryptosystem or ElGamal cipher.

4.2.2. Efficiency

The values of polynomials $\Omega_n(x)$ and $V_n(x)$ can be calculated using recursive Relationships (18) and (19). However, this is very inefficient and unprofitable from a practical point of view. In addition, no special methods have been defined thus far that allow for the rapid calculation of the values of these polynomials (as in the case of the Chebyshev polynomials). Below are the basic methods for their faster calculation; to this end, the following relationships can be used:

$$\begin{bmatrix} \Omega_{n+2}(x) \\ \Omega_{n+1}(x) \end{bmatrix} = \begin{bmatrix} x & -1 \\ 1 & 0 \end{bmatrix}^n \cdot \begin{bmatrix} \Omega_2(x) \\ \Omega_1(x) \end{bmatrix} \quad (35)$$

$$\begin{bmatrix} V_{n+2}(x) \\ V_{n+1}(x) \end{bmatrix} = \begin{bmatrix} x & -1 \\ 1 & 0 \end{bmatrix}^n \cdot \begin{bmatrix} V_2(x) \\ V_1(x) \end{bmatrix} \quad (36)$$

Dependence (35) is correct, as:

$$\begin{aligned} \begin{bmatrix} \Omega_{n+2}(x) \\ \Omega_{n+1}(x) \end{bmatrix} &= \begin{bmatrix} x & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Omega_{n+1}(x) \\ \Omega_n(x) \end{bmatrix} = \\ &= \begin{bmatrix} x & -1 \\ 1 & 0 \end{bmatrix}^2 \cdot \begin{bmatrix} \Omega_n(x) \\ \Omega_{n-1}(x) \end{bmatrix} = \dots = \begin{bmatrix} x & -1 \\ 1 & 0 \end{bmatrix}^n \cdot \begin{bmatrix} \Omega_2(x) \\ \Omega_1(x) \end{bmatrix} \end{aligned} \quad (37)$$

By analogy, one can prove the correctness of Dependence (36). It is worth noting here that the same matrix can be found in both (35) and (36):

$$\begin{bmatrix} x & -1 \\ 1 & 0 \end{bmatrix}^n \quad (38)$$

It can be concluded that the calculation of $\Omega_n(x)$ and $V_n(x)$ can be combined, which requires multiplying matrix (38) by the appropriate vectors of the initial values of the polynomials. As a result, calculating the values of the modified Chebyshev polynomials can be significantly accelerated. In addition to calculating Matrix (38), one can use the known methods of raising the matrix to the power like it was presented in [11]. Using the above methods of calculating the values of the modified Chebyshev polynomials, the performance of the presented cryptosystem can be significantly accelerated. In practice, it is slightly slower in the phase of determining the shared value (similar to the cryptosystem from the previous subsection). This is due to the necessity of counting the values for both polynomials as well as the more operations that are necessary when calculating the value of any shared data with Dependency (16).

4.2.3. Example

Let Alice's private key be the value of $s = 53419$. Moreover, let $x = 12681$ and $p = 59063$. Then, Alice's public key are the following values: $\Omega_s(x) = 6521$ and $V_{s-1}(x) = 54661$. Willing to send the message to Alice, Bob then takes over her public key. Then, he selects the value of $r = 31269$ and converts the plain text to a numerical value; let this value be $M = 1234$. Furthermore, Bob calculates shared value $V_{sr-1}(x) = 24495$ and ciphertext $C = 30226830$. In the next step, Bob passes the value of encrypted message C as well as the value of his public key ($\Omega_r(x) = 16598$ and $V_{r-1}(x) = 6874$) to Alice.

Alice receives encrypted value C and Bob's public key: $\Omega_r(x)$ and $V_{r-1}(x)$. She calculates shared value $V_{sr-1}(x) = 24495$ and then recovers the encrypted message ($M = 1234$).

5. Conclusions

The obtained results regarding the security of the presented cryptosystems show that they are related to the problem of discrete logarithms. From this point of view, the use of any of the cryptosystems is just as good and can replace the original Kocarev's

cryptosystem. In turn, an analysis of the efficiency of the presented algorithms leads to the conclusions that Cryptosystems I and II are slower than the Kocarev cryptosystem in the keying phase. This means that the time required for this procedure is longer; however, it is acceptable due to the increasing computing power of modern-day computers. On the other hand, in case of the brute-force attack (i.e. searching pairs of polynomials to encounter the correct one), the time needed to break the system will also be extended, thus making it more secure. Further research works include a performance analysis of the cryptosystem proposed in this paper as well as exploring the application of modified Chebyshev polynomials in steganography.


References

- [1] Algehawi M.B., Samsudin A.: A new Identity Based Encryption (IBE) scheme using extended Chebyshev polynomial over finite fields \mathbb{Z}_p , *Physics Letters A*, vol. 374, pp. 4670–4674, 2010.
- [2] Algehawi M., Samsudin A., Jahani S.: Calculation Enhancement of Chebyshev Polynomial over \mathbb{Z}_p , *Malaysian Journal of Mathematical Sciences*, vol. 7, pp. 131–143, 2013.
- [3] Banasik A., Kapczyński A.: Fuzzy evaluation of biometric authentication systems. In: *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2011*, pp. 803–806, 2011.
- [4] Baptista M.S.: Cryptography with chaos, *Physics Letters A*, vol. 240(1–2), pp. 50–54, 1998.
- [5] Benjamin T.A., Ericksen L., Jayawant P., Shattuck M.: Combinatorial trigonometry with Chebyshev polynomials, *Journal of Statistical Planning and Inference*, vol. 140, pp. 2157–2160, 2010.
- [6] Benjamin T.A., Walton D.: Counting on Chebyshev Polynomials, *Mathematics Magazine*, vol. 82(2), pp. 117–126, 2009.
- [7] Bergamo P., D’Arco P., De Santis A., Kocarev L.: Security of public-key cryptosystems based on Chebyshev polynomials, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52(7), pp. 1382–1393, 2005.
- [8] Chen Y., Xushuai J., Jiang Q., Gong L.: Key agreement protocol based on Chebyshev polynomials for wireless sensor network, *Journal of Computational Information Systems*, vol. 10(2), pp. 589–594, 2014.
- [9] Cheong K.Y.: One-way Functions from Chebyshev Polynomials. In: *Cryptology ePrint Archive, Report 2012/263*, 2012. <https://eprint.iacr.org/2012/263>.
- [10] Coppersmith D.: The Data Encryption Standard (DES) and its strength against attacks, *IBM Journal of Research and Development*, vol. 38(3), pp. 243–250, 1994.
- [11] Coppersmith D., Winograd S.: Matrix multiplication via arithmetic progressions, *Journal of Symbolic Computation*, vol. 9(3), pp. 251–280, 1990.


- [12] Daemen J., Rijmen V.: *AES Proposal: Rijndael*, 1999. <https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf>.
- [13] Diffie W., Hellman M.: New Directions in Cryptography, *IEEE Transactions on Information Theory*, vol. IT-22(6), 1976.
- [14] Elgamal T.: A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, vol. IT-31(4), pp. 469–472, 1985.
- [15] Fee G.J., Monagan M.B.: Cryptography using Chebyshev polynomials, pp. 1–15, 2004. <http://www.cecm.sfu.ca/CAG/papers/Cheb.pdf>.
- [16] Ghebleh M., Kanso A.: A robust chaotic algorithm for digital image steganography, *Communications in Nonlinear Science and Numerical Simulation*, vol. 19(6), pp. 1898–1907, 2014.
- [17] Hafizul Islam S.K.: Identity-based encryption and digital signature schemes using extended chaotic maps, *IACR Cryptology ePrint Archive*, 275, pp. 1–6, 2014.
- [18] Haifeng Q., Xiangxue L., Yu Y.: Pitfalls in Identity Based Encryption Using Extended Chebyshev Polynomial, *China Communications*, vol. 1, pp. 58–63, 2012.
- [19] Jianli Y., Dahu W.: Applying Extended Chebyshev Polynomials to Construct a Trap-door One-way Function in Real Field. *2009 First International Conference on Information Science and Engineering*, pp. 1680–1682, 2009.
- [20] Kapczyński A., Banasik A.: Model of intelligent detection mechanism against false biometric data injection in fingerprint-based authentication systems. *2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp. 496–498, 2009.
- [21] Kocarev L., Makraduli J., Amato P.: Public-Key Encryption Based on Chebyshev Polynomials, *Circuits, Systems and Signal Processing*, vol. 24(5), pp. 497–517, 2005.
- [22] 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*, vol. 3, pp. 28–31, 2003.
- [23] Lai H., Xiao J., Li L., Yang Y.: Applying Semigroup Property of Enhanced Chebyshev Polynomials to Anonymous Authentication Protocol, *Mathematical Problems in Engineering*, pp. 1–17, 2012.
- [24] Lawnik M.: Wykorzystanie wielomianów Viete’a–Lucasa w kryptografii asymetrycznej, *Studia Informatica*, vol. 38(4), pp. 69–77, 2017.
- [25] Lawnik M.: Generalized logistic map and its application in chaos based cryptography, *Journal of Physics: Conference Series*, vol. 936(1742-6588), pp. 1–4, 2017.
- [26] Lawnik M.: Combined logistic and tent map, *Journal of Physics: Conference Series*, vol. 1141(012132), pp. 1–6, 2018.

- [27] Lawnik M.: The problem of the inverse Lyapunov exponent and its application, *Nonlinear Analysis-Modelling and Control*, vol. 23(6), pp. 951–960, 2018.
- [28] Li Z., Cui Y., Xu H.: Fast algorithms of public key cryptosystem based on Chebyshev polynomials over finite field, *The Journal of China Universities of Posts and Telecommunications*, vol. 18(2), pp. 86–93, 2011.
- [29] Liao X., Chen F., Wong K.W.: On the Security of Public-Key Algorithms Based on Chebyshev Polynomials over the Finite Field Z_N , *IEEE Transactions on Computers*, vol. 59(10), pp. 1392–1401, 2010.
- [30] Lima J.B., Panario D., Campello de Souza R.M.: Public-key encryption based on Chebyshev polynomials over $GF(q)$, *Information Processing Letters*, vol. 111(2), pp. 51–56, 2010.
- [31] Mishkovski I., Kocarev L.: Chaos-Based Public-Key Cryptography. In: Kocarev L., Lian S. (eds.), *Chaos-Based Cryptography. Studies in Computational Intelligence*, vol. 354, Springer, Berlin–Heidelberg, pp. 27–65, 2011.
- [32] Rivest R.L., Shamir A., Adleman L.: A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, vol. 21(2), pp. 120–126, 1978.
- [33] Rivlin T.: *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*, Wiley, 1990.
- [34] Roy R., Sarkar A., Changder S.: Chaos based Edge Adaptive Image Steganography, *Procedia Technology*, vol. 10, pp. 138–146, 2013.
- [35] Sun J., Zhao G., Li X.: An Improved Public Key Encryption Algorithm Based on Chebyshev Polynomials, *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11(2), pp. 864–870, 2013.
- [36] Toan-Thinh T., Minh-Triet T., Anh-Duc D.: Improved Chebyshev Polynomials-Based Authentication Scheme in Client-Server Environment, *Security and Communication Networks*, vol. 2019(4250743), pp. 1–11, 2019.
- [37] Vairachilai S., Kavithadevi M., Gnanajeyaraman R.: Public Key Cryptosystems Using Chebyshev Polynomials Based on Edge Information. In: *2014 World Congress on Computing and Communication Technologies*, pp. 243–245, 2014.
- [38] Wituła R., Słota D.: On modified Chebyshev polynomials, *Journal of Mathematical Analysis and Applications*, vol. 324(1), pp. 321–343, 2006.
- [39] Xiang T., Wong K.W., Liao X.: On the security of a novel key agreement protocol based on chaotic maps, *Chaos, Solitons & Fractals*, vol. 40(2), pp. 672–675, 2009.
- [40] Xiao D., Liao X., Deng S.: A novel key agreement protocol based on chaotic maps, *Information Sciences*, vol. 177(4), pp. 1136–1142, 2007.

Affiliations

Marcin Lawnik 

Silesian University of Technology, Faculty of Applied Mathematics, Gliwice, Poland,
marcin.lawnik@polsl.pl, ORCID ID: <https://orcid.org/0000-0002-0235-0878>

Adrian Kapczyński 

Silesian University of Technology, Faculty of Applied Mathematics, Gliwice, Poland,
adrian.kapczynski@polsl.pl, ORCID ID: <https://orcid.org/0000-0002-9299-1467>

Received: 28.05.2019

Revised: 8.08.2019

Accepted: 9.08.2019