Hindawi
Journal of Applied Mathematics
Volume 2024, Article ID 4894415, 12 pages
https://doi.org/10.1155/2024/4894415

Research Article

Symmetric Encryption Algorithms in a Polynomial Residue Number System

I. Yakymenko,[1] **M. Karpinski**,[2,3] **R. Shevchuk**,[4,5] **and M. Kasianchuk**[1]

[1]*Department of Cyber Security, West Ukrainian National University, Ternopil 46009, Ukraine*
[2]*Department of Cyber Security, Ternopil Ivan Puluj National Technical University, Ternopil 46001, Ukraine*
[3]*Institute of Security and Computer Science, University of the National Education Commission, 30-084 Krakow, Poland*
[4]*Department of Computer Science, West Ukrainian National University, Ternopil 46009, Ukraine*
[5]*Department of Computer Science and Automatics, University of Bielsko-Biala, Bielsko-Biala 43-309, Poland*

Correspondence should be addressed to R. Shevchuk; rsh@wunu.edu.ua

Received 21 October 2023; Revised 12 April 2024; Accepted 24 April 2024; Published 20 May 2024

Academic Editor: Saeid Abbasbandy

In this paper, we develop the theoretical provisions of symmetric cryptographic algorithms based on the polynomial residue number system for the first time. The main feature of the proposed approach is that when reconstructing the polynomial based on the method of undetermined coefficients, multiplication is performed not on the found base numbers but on arbitrarily selected polynomials. The latter, together with pairwise coprime residues of the residue class system, serve as the keys of the cryptographic algorithm. Schemes and examples of the implementation of the developed polynomial symmetric encryption algorithm are presented. The analytical expressions of the cryptographic strength estimation are constructed, and their graphical dependence on the number of modules and polynomial powers is presented. Our studies show that the cryptanalysis of the proposed algorithm requires combinatorial complexity, which leads to an NP-complete problem.

Keywords: ciphertext; cryptanalysis; cryptoalgorithm; cryptographic strength; residue number system; symmetric cryptosystem

1. Introduction

Recently, with the growth of confidential information and the spread of computer systems, the task of ensuring information security has become increasingly important [1–3]. To minimize the risks of unauthorized access, cryptographic methods of information protection are widely used [4, 5], which are divided into symmetric and asymmetric [6, 7]. In practice, symmetric cryptographic transformations are more common for encrypting large amounts of information, as asymmetric ones are quite laborious [8, 9]. The requirements for symmetric methods have become more stringent in terms of ensuring their cryptographic strength due to the rapid development of computing tools and their increased speed. Polynomial algorithms are an alternative to modern numerical cryptoalgorithms [10–12]. In the ring $Z[x]$, as in any other ring of polynomials, basic cryptographic operations are performed: addition, multiplication, and division with remainder [13–15]. The main idea of using polynomials in cryptography is that they can be used as plaintext, keys for encrypting and decrypting messages, building electronic digital signatures, and other cryptographic protocols [16–18].

The use of the residue number system (RNS) [19–21] in the implementation of cryptographic algorithms for information security based on polynomial arithmetic in the $Z[x]$ ring [22, 23], by analogy with the integer RNS [24, 25], leads to parallelization of the computation process [26, 27] and reduction of the amount of data that must be processed during cryptographic operations [28–30]. In turn, it reduces the implementation time and improves the efficiency of the encryption method.

Therefore, our work is aimed at developing the concept of polynomial symmetric cryptographic algorithms based on the RNS and their practical application.

1.1. Our Contribution. In this article, our contributions are as follows:

1. A theoretical provision for symmetric cryptographic algorithms based on the polynomial RNS was developed.
2. Mathematical frameworks and schemes for the proposed polynomial symmetric encryption within the RNS were devised. To ascertain its resilience, a deep dive was made into constructing analytical expressions, revealing that the process of cryptanalyzing the proposed algorithm required dealing with combinatorial complexity, ultimately leading to an NP-complete problem.
3. It was established that cryptographic strength notably improved with increasing degrees and dimensions of the Galois field p. The peak of cryptographic strength was reached when the number of modules equaled half of the potential count of irreducible polynomials with the given polynomial degrees and Galois field orders.

1.2. Related Work. Most modern symmetric cryptographic algorithms are block-based, and this feature limits the functionality of their implementation. In particular, the key size must be equal to or larger than the block size, leading to the encryption algorithm's multiple uses for a large message. This procedure reduces the cryptographic strength of the algorithm, increases the time complexity, and, at the same time, complicates the implementation. Many authors have studied symmetric encryption algorithms in the polynomial number system. For example, Lemaire [31] proposes an 8-bit encryption algorithm based on the ideas of well-known symmetric cryptoalgorithms. The authors use divergent polynomials with variable coefficients, bitwise data operations, and two-password identification when generating pseudorandom keys. The hardware implementation of the proposed approach and comparison of the time characteristics with the AES algorithm of the 8-bit architecture based on the Arduino Uno microcontroller (ATmega328) were carried out.

A work [32] is devoted to developing and studying hardware-implemented methods of fast polynomial arithmetic for some homomorphic encryption operations based on the Karatsuba algorithm. In addition, Jayet-Griffon et al. [33] consider the possibilities of speeding up the polynomial multiplication operation for homomorphic encryption when implemented on an FPGA. In [34], a characterization of polynomial multiplication implementations for GPU-based homomorphic encryption is presented.

In [35], a highly efficient image encryption method based on permutation polynomials in finite fields was developed that is resistant to various types of attacks. In addition, the proposed encryption algorithm has no rounding errors, so encryption is lossless.

In the work [36], our effort was dedicated to developing a multifunctional architecture for the polynomial RNS within the context of cryptography. Detailed comparisons with contemporary implementations have indicated the potential utility of polynomial residue arithmetic in modular multiplication. Article [37] presents a schematic diagram of a modular pipeline multiplier, which allows for high-speed data encryption and decryption based on nonpositional polynomial RNS. The authors substantiate the efficiency of the proposed hardware design through a timing diagram. The developed pipeline device can find application in digital computing devices, particularly for high-speed data encryption based on nonpositional polynomial RNS.

In [38], a new method for constructing S-blocks of the AES algorithm is proposed based on replacing the irreducible polynomial and affine mapping. The cryptographic strength of the created S-block is evaluated by several standard tests (bijectivity, nonlinearity, strict avalanche criterion (SAC), and bit-independence criterion). It surpasses the cryptographic strength of the known S-boxes.

An article [39] proposes a method for constructing the S-block of the AES algorithm based on the smallest number of selected irreducible polynomials that meet specific criteria. There are 17 such polynomials, and their use simplifies the hardware implementation of the S-block. The SAC is studied, and it is noted that the polynomial $p(x) = x^8 + x^7 + x^6 + x + 1$ is the best, with an outstanding value of $\mathrm{SAC} = 0.5$, which indicates the cryptographic strength and reliability of the constructed S-block.

A paper [40] proposes improving the symmetric AES encryption algorithm using dynamic S-blocks whose parameters depend on the key, dynamic irreducible polynomials, and affine constants.

A paper [41] presents the most commonly used symmetric cryptosystem AES in the ring of polynomials today. The main idea is to choose an irreducible polynomial on the basis of which the encryption algorithm is built. The proposed approach was implemented in MATLAB for 30 different irreducible polynomials. As a result of the numerical experiments, it was possible to establish a negligible effect of changing irreducible polynomials on the level of the avalanche.

The authors in [42] proposed a novel method to enhance AES security against fault attacks using the polynomial RNS. The authors parallelize byte-level AES operations over GF (2^8) by utilizing residues over smaller fields, introducing extended functionalities into AES for side-channel vulnerability analysis.

Polynomial arithmetic has also been used for asymmetric cryptosystems. In particular, in [43], modified arithmetic was developed for the RSA cryptosystem with Gauss integers and polynomials over finite fields. The analysis of the described computational procedures made it possible to determine their advantages over the classical ones. In [44], algorithmic support for the Rabin cryptosystem in the polynomial number system was proposed.

The analysis of the literary sources shows the relevance and importance of polynomial algorithms for protecting

information flows. Accordingly, the development of new methods that are resistant to attacks of various types is an important direction in the development of modern cryptosystems. In particular, the combination of polynomial arithmetic and RNS in a ring of polynomials will allow parallelizing the process of performing basic operations in a ring of polynomials, which, in turn, will increase the speed of software implementation and reduce the time complexity of the algorithm, providing the required level of security.

1.3. Organization. Section 2 of this article discusses in detail the theoretical foundations for constructing symmetric cryptographic algorithms based on a polynomial RNS. Subsequently, in Section 3, the cryptographic strength of a polynomial symmetric encryption algorithm in the system of residual classes was evaluated. Finally, in Section 4, the content of this article is summarized.

2. Materials and Methods

In Subsection 2.1, the theoretical foundations for constructing symmetric cryptographic algorithms based on a polynomial RNS are proposed. Subsection 2.2 described the features of developing polynomial symmetric encryption methods in the RNS. An example of symmetric polynomial encryption in RNS presents in Subsection 2.3. In Subsection 2.4, the polynomial symmetric encryption method based on the Chinese remainder theorem (CRT) is proposed.

2.1. Theoretical Foundations of Polynomial RNS. An arbitrary polynomial $N(x)$ in the RNS is represented as the residuals $b_i(x)$ from dividing $N(x)$ by each of the systems of pairwise mutually simple modulo-polynomials $p_i(x)$ [45–47]:

$$b_i(x) = N(x) \bmod p_i(x) \tag{1}$$

The recovery of the polynomial $N(x)$ is usually based on the CRT [48–50] in the ring of polynomials $Z[x]$:

$$N(x) = \left(\sum_{i=1}^{s} b_i(x) M_i(x) m_i(x) \right) \bmod P(x) \tag{2}$$

where $P(x) = \prod_{i=1}^{s} p_i(x)$, $M_i(x) = P(x)/p_i(x)$, $m_i(x)$ is sought from the expression $m_i(x) = M_i^{-1}(x) \bmod p_i(x)$, and s is the number of modules. For polynomial powers, the inequality $\deg N(x) < \deg P(x)$ must be satisfied.

2.2. Development of Polynomial Symmetric Encryption Methods in the RNS. The essence of one of the methods of polynomial symmetric encryption in RNS is that when recovering a polynomial from its residuals in the sum (2), the multiplication is not by the parameters $m_i(x) = M_i^{-1} \bmod p_i(x)$, but by arbitrarily chosen polynomials $k_i(x)$, mutually prime with $p_i(x)$.

Therefore, to generate keys, both subscribers must choose module systems known only to them $p_i(x)$ and the corresponding polynomials $k_i(x)$, for which the following conditions are met: $1 < \deg k_i(x) < \deg p_i(x)$ and $\mathrm{GCD}(k_i(x), p_i(x)) = 1$, where GCD denotes greatest common divisor. If $p_i(x)$ is an irreducible polynomial, then the second condition is always met. Accordingly, both the sender and the receiver know the parameters $M_i(x)$ and $m_i(x)$.

For encryption, alphabetic information must be written in numerical form. The most common classical method is to replace the letter with its number in the alphabet, with the numbering starting from 0. After that, it must be represented as a polynomial with coefficients that reflect the alphabetic information, so the plaintext $N(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0 x^o$, where a_i is the sequence of digital representation of letters and $i = 0 \cdots n$, $(n+1)$ is the length of the message. Next, the plaintext block $N(x)$ is written to the RNS according to expression (1). Encryption occurs when the number is restored to the positional number system according to the following expression:

$$N'(x) = \left(\sum_{i=1}^{s} b_i(x) M_i(x) k_i(x) \right) \bmod P(x) \tag{3}$$

The found polynomial is a ciphertext that is transmitted over an open communication channel from one subscriber to another.

When decrypting, the following values are first calculated:

$$\begin{aligned} q_i(x) &= \left(m_i(x) \cdot \left(k_i^{-1}(x) \bmod p_i(x) \right) \right) \bmod p_i(x) \\ b_i'(x) &= N'(x) \bmod p_i(x) \end{aligned} \tag{4}$$

To obtain the true residuals $b_i(x)$, you need to perform the conversion according to the following ratio:

$$\begin{aligned} b_i(x) &= \left(b_i'(x) q_i(x) \right) \bmod p_i(x) \\ &= \left(b_i'(x) m_i(x) \left(k_i^{-1}(x) \bmod p_i(x) \right) \right) \bmod p_i(x) \end{aligned} \tag{5}$$

Accordingly, the recovery of the plaintext polynomial $N(x)$ is carried out according to Formula (2) or the expression that follows from it can be used:

$$\begin{aligned} N(x) &= \left(\sum_{i=1}^{s} M_i(x) m_i(x) \left(b_i'(x) m_i(x) \left(k_i^{-1}(x) \bmod p_i(x) \right) \right) \right. \\ &\quad \left. \cdot \bmod p_i(x) \right) \bmod P(x) \\ &= \left(\sum_{i=1}^{s} M_i(x) m_i(x) \left(b_i'(x) q_i(x) \right) \bmod p_i(x) \right) \bmod P(x) \end{aligned} \tag{6}$$

Figure 1 shows a schematic of the proposed polynomial encryption method based on the RNS.

The correctness of the proposed cryptosystem is established by a formal proof from the properties of congruences,

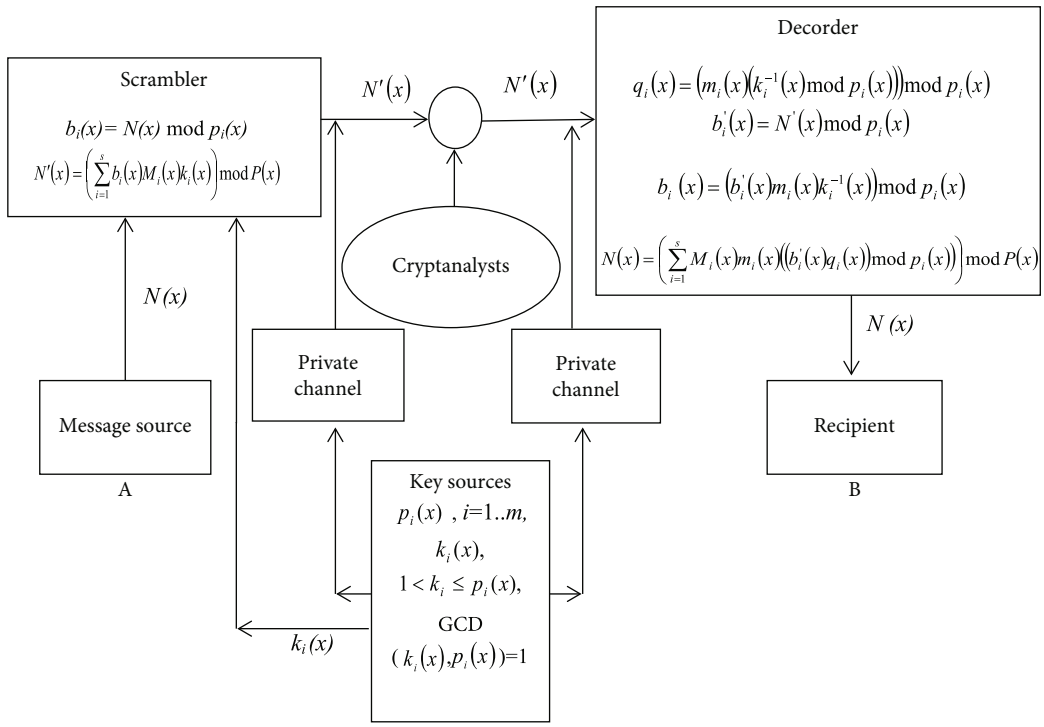

Figure 1: Scheme of the proposed polynomial symmetric encryption in RNS.

taking into account the divisibility $P(x)$ by $p_i(x)$ and the equality $m_i(x) = M_i^{-1}(x) \bmod p_i(x)$. Then, we get

$$\begin{aligned} b_i(x) &= \left(b_i'(x) q_i(x)\right) \bmod p_i(x) = \left(\left(N'(x) \bmod p_i(x)\right)\right. \\ &\quad \left. \cdot \left(m_i(x)\left(k_i^{-1}(x) \bmod p_i(x)\right)\right) \bmod p_i(x)\right) \bmod p_i(x) \\ &= \left(\left(\left(\left(\sum_{j=1}^{s} b_j(x) k_j(x) M_j(x)\right) \bmod P(x)\right) \bmod p_i(x)\right.\right. \\ &\quad \left.\left. \cdot \left(m_i(x)\left(k_i^{-1}(x) \bmod p_i(x)\right)\right) \bmod p_i(x)\right) \bmod p_i(x)\right) \\ &= \left(\left(b_i(x) k_i(x) M_i(x)\right) \bmod P(x)\left(m_i(x)\left(k_i^{-1}(x)\right.\right.\right. \\ &\quad \left.\left.\left. \cdot \bmod p_i(x)\right)\right) \bmod p_i(x)\right) \bmod p_i(x) \\ &= \left(b_i(x) m_i(x) M_i(x)\right) \bmod p_i(x) = b_i(x) \end{aligned} \tag{7}$$

2.3. An Example of Symmetric Polynomial Encryption in RNS. Let us consider the plaintext PSMFSRD = (15181205181703), which corresponds to the polynomial $N(x) = 15x^6 + 18x^5 + 12x^4 + 5x^3 + 18x^2 + 17x + 3$. According to the developed polynomial symmetric cryptosystem for three modules ($s = 3$), $p_1(x) = x^2 + x + 1$, $p_2(x) = x^3 + x + 1$, and $p_3(x) = x^2 + 1$ and the chosen coefficients $k_1(x) = x^2 + 2x + 3$, $k_2(x) = x^3 + x^2 + 1$, and $k_3(x) = x^2 + 3x + 2$, all the parameters are calculated as follows: $P(x) = p_1(x)p_2(x)p_3(x) = x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1$, $M_1(x) = P(x)/p_1(x) = (x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1)/(x^2 + x + 1) = x^5 + 2x^3 + x^2 + x + 1$, $M_2(x) = P(x)/p_2(x) = (x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1)/(x^3 + x + 1) = x^4 + x^3 + 2x^2 + x + 1$, and $M_3(x) = P(x)/p_3(x) = (x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1)/(x^2 + 1) = x^5 + x^4 + 2x^3 + 2x^2 + 2x + 1$.

The search for $m_i(x) = M_i^{-1} \bmod p_i(x)$ is performed using the method of undetermined coefficients. Firstly, we look for $m_i(x) = M_1^{-1} \bmod p_1(x) = (x^5 + 2x^3 + x^2 + x + 1)^{-1} \bmod (x^2 + x + 1) = (x + 2)^{-1} \bmod (x^2 + x + 1)$. To do this, we write the equation $(x + 2)(Ax + B) \bmod (x^2 + x + 1) = 1$, and after transformations, we obtain $(Ax^2 + (2A + B)x + 2B) \bmod (x^2 + x + 1) = (A + B)x + 2B - A = 1$. From the last equation, it follows that $2B - A = 1$ and $B + A = 0$ and takes the form $A = 1/3$ and $B = 2/3$. So, the sought-after inverse polynomial takes the form $m_1(x) = (x + 2)^{-1} \bmod (x^2 + x + 1) = (1/3)x + 2/3$. Similarly, the search for $m_2(x) = M_2^{-1} \bmod p_2(x) = (x^4 + x^3 + 2x^2 + x + 1)^{-1} \bmod (x^3 + x + 1) = (x^2 - x)^{-1} \bmod (x^3 + x + 1)$ is carried out. We write $(x^2 - x)(Ax^2 + Bx + C) \bmod (x^3 + x + 1) = 1$, where $Ax^2 + Bx + C$ is the inverse polynomial modulo. We need to find the coefficients A, B, and C that satisfies the equation. After transformations $(2A x^4 + 2Bx^3 + (A + 2C)x^2 + Bx + C) \bmod (x^3 + x + 1) = (C - A - B)x^2 + (-C - B)x + A - B = 1$, we obtain $2C - A = 0$, $C - B - A = 0$, $-C - B = 0$, and $A - B = 1$. From here, $A = 2/3$, $B = -1/3$, and $C = 1/3$. So, the inverse polynomial takes the following form: $m_2(x) = (x^2 - x)^{-1} \bmod (x^3 + x + 1) = (2/3)x^2 - (1/3)x + 1/3$. Similarly, the value $m_3(x) = M_3^{-1} \bmod p_3(x) = (x^5 + x^4 + 2x^3 + 2x^2 + 2x + 1)^{-1} \bmod (x^2 + 1) = (x)^{-1} \bmod (x^2 + 1)$ is computed. By applying the method of undetermined coefficients, the following transformations can be performed: $(x)(Ax + B) \bmod (x^2 + 1) = (Ax^2 + Bx) \bmod (x^2 + 1)$

$\Rightarrow Bx - A = 1$. The last equation leads to a system of equations that allows us to compute the coefficients' values $A = -1$ and $B = 0$ and thereby find the inverse polynomial modulo $m_3(x) = -x$. Thus, $b_1(x) = N(x) \bmod p_1(x) = (15x^6 + 18x^5 + 12x^4 + 5x^3 + 18x^2 + 17x + 3) \bmod (x^2 + x + 1) = -7x - 13$, $b_2(x) = N(x) \bmod p_2(x) = (15x^6 + 18x^5 + 12x^4 + 5x^3 + 18x^2 + 17x + 3) \bmod (x^3 + x + 1) = 3x^2 + 48x + 31$, and $b_3(x) = N(x) \bmod p_3(x) = (15x^6 + 18x^5 + 12x^4 + 5x^3 + 18x^2 + 17x + 3) \bmod (x^2 + 1) = 30x - 18$.

Therefore, according to expression (3), the ciphertext is given by $N'(x) = (\sum_{i=1}^{s} b_i(x)M_i(x)k_i(x)) \bmod P(x) = ((x^5 + 2x^3 + x^2 + x + 1)(-7x - 13)(x^2 + 2x + 3) + (x^4 + x^3 + 2x^2 + x + 1)(3x^2 + 48x + 31)(x^3 + x^2 + 1) + (x^5 + x^4 + 2x^3 + 2x^2 + 2x + 1)(30x - 18)(x^2 + 3x + 2)) \bmod (x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1) = -64x^6 - 250x^5 - 360x^4 - 545x^3 - 492x^2 - 403x - 172$.

The parameters $k_i^{-1}(x) \bmod p_i(x)$ are computed using the method of undetermined coefficients. To find the inverse polynomial $k_1^{-1}(x) \bmod p_1(x) = (x^2 + 2x + 3)^{-1} \bmod (x^2 + x + 1) = (x + 2)^{-1} \bmod (x^2 + x + 1)$, we write $(x + 2)(Ax + B) \bmod (x^2 + x + 1) = 1 \Rightarrow (Ax^2 + (2A + B)x + 2B) \bmod (x^2 + x + 1) = (A + B)x + 2B - A = 1$, where $(Ax + B)$ is the desired value. To determine the coefficients A and B, we compute the remainder and equate the corresponding values: $(A + B)x + 2B - A = 1$; from here, $A = -1/3$, and $B = 1/3$. Thus, $k_1^{-1}(x) \bmod p_1(x) = (1/3)(-x + 1)$.

Similarly, we search for $k_2^{-1}(x) \bmod p_2(x) = (x^3 + x^2 + 1)^{-1} \bmod (x^3 + x + 1) = (x^2 - x)^{-1} \bmod (x^3 + x + 1) \longrightarrow (x^2 - x)(Ax^2 + Bx + C) \bmod (x^3 + x + 1) = 1$, where $Ax^2 + Bx + C$ is the inverse polynomial modulo. After the transformation $(2Ax^4 + 2Bx^3 + (A + 2C)x^2 + Bx + C) \bmod (x^3 + x + 1) = (C - A - B)x^2 + (-C - B)x + A - B = 1$, we obtain a system of three equations with three unknowns: $C - B - A = 0$, $-C - B = 0$, and $A - B = 1$. From here, $A = 2/3$, $B = -1/3$, and $C = 1/3$. Therefore, the sought inverse polynomial in $Z[x]$ will take the following form: $k_2^{-1} \bmod p_2(x) = (x^2 - x)^{-1} \bmod (x^3 + x + 1) = (2/3)x^2 - (1/3)x + 1/3$; similarly, we can obtain $k_3^{-1}(x) \bmod p_3(x) = -(3/10)x + 1/10$.

In the next step, the following quantities are computed: $b_{1'}(x) = N'(x) \bmod p_1(x) = (-64x^6 - 250x^5 - 360x^4 - 545x^3 - 492x^2 - 403x - 172) \bmod (x^2 + x + 1) = -21x - 39$, $b_{2'}(x) = N'(x) \bmod p_2(x) = (-64x^6 - 250x^5 - 360x^4 - 545x^3 - 492x^2 - 403x - 172) \bmod (x^3 + x + 1) = 54x^2 + 124x + 59$, and $b_{3'}(x) = N'(x) \bmod p_3(x) = (-64x^6 - 250x^5 - 360x^4 - 545x^3 - 492x^2 - 403x - 172) \bmod (x^2 + 1) = -108x + 24$.

Additionally, for decryption, the following parameters need to be found: $q_1(x) = (m_1(x)(k_1^{-1}(x) \bmod p_1(x))) \bmod p_1(x) = (((1/3)x + 2/3)(-(1/3)x + 1/3)) \bmod (x^2 + x + 1) = 1/3$, $q_2(x) = (m_2(x)(k_2^{-1}(x) \bmod p_2(x))) \bmod p_2(x) = (((2/3)x^2 - (1/3)x + 1/3)((2/3)x^2 - (1/3)x + 1/3)) \bmod (x^3 + x + 1) = ((10/27)x^4 + (7/27)x^3 - (20/27)x^2 + (8/27)x - 7/27) \bmod (x^3 + x + 1) = (1/9)x^2 - (2/9)x + 5/9$, and $q_3(x) = (m_3(x)(k_3^{-1}(x) \bmod p_3(x))) \bmod p_3(x) = ((-x)(-(3/10)x + 1/10)) \bmod (x^3 + x^2 + x + 3) = ((3/10)x^2 - (1/10)x) \bmod (x^2 + 1) = -(1/10)x - 3/10$.

Then, according to Formula (6), the original message is recovered as the plaintext: $N(x) = (\sum_{i=1}^{s} M_i(x)m_i(x)(b_i'(x)m_i(x)(k_i^{-1}(x) \bmod p_i(x)) \bmod p_i(x)) \bmod P(x) = (\sum_{i=1}^{s} M_i(x)m_i(x)((b_i'(x)q_i(x)) \bmod p_i(x)))) \bmod P(x) = (((x^5 + 2x^3 + x^2 + x + 1)((1/3)x + 2/3)((1/3)(-21x - 39)) \bmod (x^2 + x + 1)) + ((x^4 + x^3 + 2x^2 + x + 1)((2/3)x^2 - (1/3)x + 1/3)(((1/9)x^2 - (2/9)x + 5/9)(54x^2 + 124x + 59)) \bmod (x^3 + x + 1)) + ((x^5 + x^4 + 2x^3 + 2x^2 + 2x + 1)(-x)((-108x + 24)(-(1/10)x - 3/10)) \bmod (x^2 + 1))) \bmod (x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1) = 15x^6 + 18x^5 + 12x^4 + 5x^3 + 18x^2 + 17x + 3$.

Accelerating the encryption and decryption process can be achieved if the participants choose parameters $k_i(x) = 1$. However, this will lead to a reduction in the cryptographic system's resilience. The encryption process is simplified by using the following formula:

$$N'(x) = \left(\sum_{i=1}^{s} b_i(x)M_i(x) \right) \bmod P(x) \tag{8}$$

It should be noted that the operation of finding the inverse polynomial modulo and multiplying it in Formula (4) disappears because $q_i(x) = m_i(x)$. The restoration of the plaintext is based on the following relationships:

$$b_i(x) = \left(b_i'(x)m_i(x) \right) \bmod p_i(x)$$
$$N(x) = \left(\sum_{i=1}^{s} M_i(x)m_i(x) \left(\left(b_i'(x)m_i(x) \right) \bmod p_i(x) \right) \right) \bmod P(x) \tag{9}$$

For the same input parameters as in the previous example, according to Formulas (8) and (9), the following ciphertext is obtained: $N'(x) = (\sum_{i=1}^{s} b_i(x)M_i(x)) \bmod P(x) = ((x^5 + 2x^3 + x^2 + x + 1)(-7x - 13) + (x^4 + x^3 + 2x^2 + x + 1)(3x^2 + 48x + 31) + (x^5 + x^4 + 2x^3 + 2x^2 + 2x + 1)(30x - 18)) \bmod (x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1) = 26x^6 + 50x^5 + 113x^4 + 121x^3 + 117x^2 + 53x$. To decrypt, it is necessary to compute additional parameters according to Formula (4): $b_{1'}(x) = N'(x) \bmod p_1(x) = (26x^6 + 50x^5 + 113x^4 + 121x^3 + 117x^2 + 53x) \bmod (x^2 + x + 1) = -x - 20$; $b_{2'}(x) = N'(x) \bmod p_2(x) = (26x^6 + 50x^5 + 113x^4 + 121x^3 + 117x^2 + 53x) \bmod (x^3 + x + 1) = -20x^2 - 79x - 45$; $b_{3'}(x) = N'(x) \bmod p_3(x) = (26x^6 + 50x^5 + 113x^4 + 121x^3 + 117x^2 + 53x) \bmod (x^2 + 1) = -18x - 30$.

The decryption process is carried out according to Formula (9): $N(x) = (\sum_{i=1}^{s} M_i(x)m_i(x)((b_i'(x)m_i(x)) \bmod p_i(x))) \bmod P(x) = (((x^5 + 2x^3 + x^2 + x + 1)((1/3)x + 2/3)(((1/3)x + 2/3)(-x - 20)) \bmod (x^2 + x + 1)) + ((x^4 + x^3 + 2x^2 + x + 1)((2/3)x^2 - (1/3)x + 1/3)(((2/3)x^2 - (1/3)x + 1/3)(-20x^2 - 79x - 45)) \bmod (x^3 + x + 1)) + ((x^5 + x^4 + 2x^3 + 2x^2 + 2x + 1)(-x)((-18x - 30)(-x)) \bmod (x^2 + 1))) \bmod (x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1) = 15x^6 + 18x^5 + 12x^4 + 5x^3 + 18x^2 + 17x + 3$.

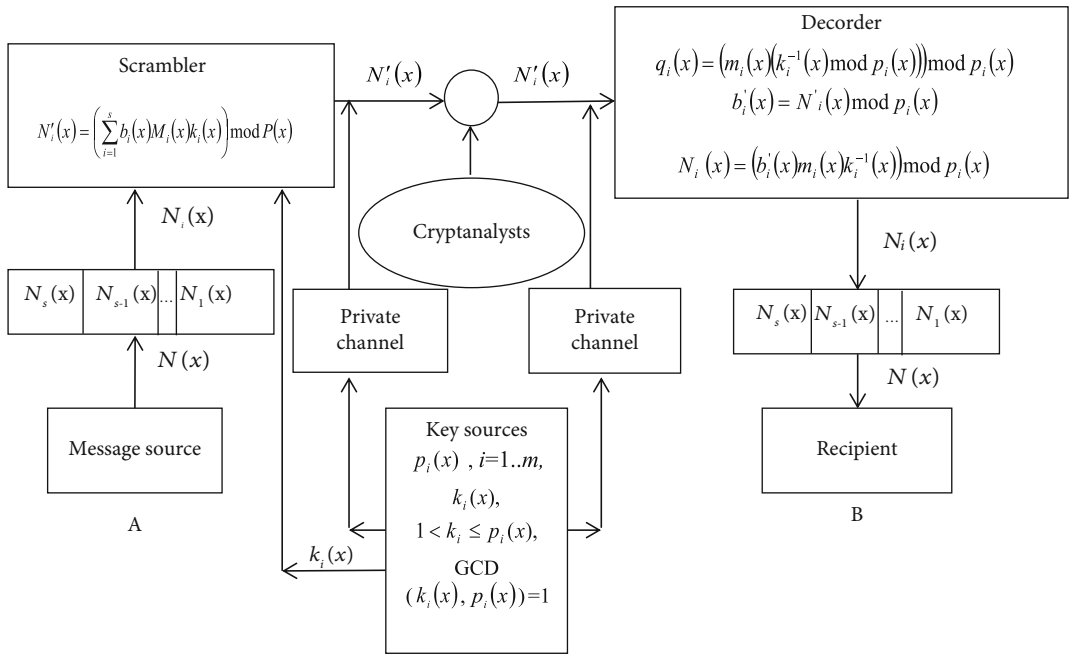

Figure 2: Polynomial symmetric encryption method in RNS.

This simplification reduces computational complexity by avoiding the operation of finding the parameters $q_i(x)$ and $k_1^{-1}(x) \bmod p_i$.

2.4. Polynomial Symmetric Encryption Method Based on CRT. Another polynomial method of symmetric encryption based on the CRT involves breaking the plaintext $N(x)$ into blocks—polynomials $N_i(x)$ of lower order than the selected polynomial modules. These blocks will act as remainders $b_i(x)$ modulo the chosen moduli, such that if $\deg p_i(x) = n$, then $\deg N_i(x) \le n-1$; that is, $N_i(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_0x^0$. After selecting the encryption parameters, the encryption is performed according to the expression (3). The ciphertext will be the value $N'(x)$.

Decryption is carried out using Formulas (4) and (5), which are used to find the parameters $q_i(x)$, $b_i(x) = N_i(x) \bmod p_i(x) = N_i'(x)$ and $b_i'(x)$. Concatenating the coefficients a_{n-1} of the polynomials $N_i(x)$ forms the plaintext. It should be noted that in the case of requiring fast decryption, the ciphertext can also be represented by the parameters $b_i(x)$.

Figure 2 depicts the scheme of the polynomial symmetric encryption method in the CRT-based encryption system.

To encrypt using the aforementioned method, the chosen plaintext PSMFSRND = 1518120518171303 is divided into four blocks of two characters each: PS = 1518, MF = 1205, SR = 1817, and ND = 1303. Then, the plaintexts are formed as polynomials, $N_1(x) = 15x + 18$, $N_2(x) = 12x + 5$, $N_3(x) = 18x + 17$, and $N_4(x) = 13x + 3$, which are remainders modulo the chosen moduli $p_1(x) = x^2 + x + 1$, $p_2(x) = x^3 + x + 1$, and $p_3(x) = x^2 + 1$, and the corresponding coefficients $k_1(x) = x^2 + 2x + 3$, $k_2(x) = x^3 + x^2 + 1$, and $k_3(x) = x^2 + 3x + 2$. The ciphertext for the first block is computed based on Formula (3): $N_1'(x) = (\sum_{i=1}^{s} b_i(x)M_i(x)k_i(x)) \bmod P(x) = ((15x + 18)(x^5 + 2x^3 + x^2 + x + 1)(x^2 + 2x + 3) + (x^4 + x^3 + 2x^2 + x + 1)(15x + 18)(x^3 + x^2 + 1) + (x^5 + x^4 + 2x^3 + 2x^2 + 2x + 1)(15x + 18)(x^2 + 3x + 2)) \bmod (x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1) = 105x^6 + 33x^5 + 135x^4 + 81x^3 + 78x^2 + 21x - 21$.

Upon decryption using Formulas (4) and (5), the following results are obtained: $b_1^{1'}(x) = 45x + 54$, $b_2^{1'}(x) = 15x^2 + 48x + 36$, and $b_3^{1'}(x) = -27x - 69$. The message is reconstructed based on the relation (5): $b_1^1(x) = ((45x + 54)((1/3)x + 2/3)(-(1/3)x + 1/3)) \bmod (x^2 + x + 1) = ((45x + 54)(-(1/9)x^2 - (1/9)x + 2/9)) \bmod (x^2 + x + 1) = (-5x^3 - 11x^2 + 4x + 12) \bmod (x^2 + x + 1) = 15x + 18$.

For the second block of the input message $N_2(x) = 12x + 5$, the following ciphertext value is obtained: $N_2(x)' = (\sum_{i=1}^{s} b_i(x)M_i(x)k_i(x)) \bmod P(x) = ((12x + 5)(x^5 + 2x^3 + x^2 + x + 1)(x^2 + 2x + 3) + (x^4 + x^3 + 2x^2 + x + 1)(12x + 5)(x^3 + x^2 + 1) + (x^5 + x^4 + 2x^3 + 2x^2 + 2x + 1)(12x + 5)(x^2 + 3x + 2)) \bmod (x^7 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1) = 37x^6 - 30x^5 + 14x^4 - 48x^3 - 41x^2 - 49x - 45$.

According to Formula (4), the remainders obtained are $b_1^{2'}(x) = 36x + 15$, $b_2^{2'}(x) = 12x^2 + 29x + 10$, and $b_3^{2'}(x) = -31x - 27$. The restoration of the second block of the input message is done using relation (5): $b_1^2(x) = ((36x + 15)((1/3)x + 2/3)(-(1/3)x + 1/3)) \bmod (x^2 + x + 1) = ((36x + 15)(-(1/9)x^2 - (1/9)x + 2/9)) \bmod (x^2 + x + 1) = (-4x^3 - (17/3)x^2 + (19/3)x + 10/3) \bmod (x^2 + x + 1) = 12x + 5$.

The ciphertext for the third block of the input message $N_3(x) = 18x + 17$ will have the following form: $N_3'(x) = (\sum_{i=1}^{s} b_i(x)M_i(x)k_i(x)) \bmod P(x) = ((18x + 17)(x^5 + 2x^3 + x^2 + x + 1)(x^2 + 2x + 3) + (x^4 + x^3 + 2x^2 + x + 1)(18x + 17)(x^3$

$+x^2+1)+(x^5+x^4+2x^3+2x^2+2x+1)(18x+17)(x^2+3x+2))$ mod $(x^7+x^6+3x^5+3x^4+4x^3+3x^2+2x+1) = 103x^6+12x^5+116x^4+42x^3+43x^2-7x-39$.

Then, according to Formula (4), the remainders obtained are $b_1^{3'}(x) = 54x+51$, $b_2^{3'}(x) = 18x^2+53x+34$, and $b_3^{3'}(x) = -37x-69$. The restoration is done using relation (5): $b_i^3(x) = (b_i^{3'}(x)q_i(x)) \bmod p_i(x) = (b_i^{3'}(x)m_i(x)(k_i^{-1}(x) \bmod p_i(x))) \bmod p_i(x)$. Hence, $b_1^3(x) = (54x+51)((1/3)x+2/3)(-(1/3)x+1/3) \bmod (x^2+x+1) = (54x+51)(-(1/9)x^2-(1/9)x+2/9) \bmod (x^2+x+1) = (-6x^3-(35/3)x^2+(19/3)x+34/3) \bmod (x^2+x+1) = 18x+17$.

Therefore, the encrypted message for the fourth block, $N_4(x) = 13x+3$ according to (3), will be the following polynomial: $N_4'(x) = ((13x+3)(x^5+2x^3+x^2+x+1)(x^2+2x+3)+(x^4+x^3+2x^2+x+1)(13x+3)(x^3+x^2+1)+(x^5+x^4+2x^3+2x^2+2x+1)(13x+3)(x^2+3x+2)) \bmod (x^7+x^6+3x^5+3x^4+4x^3+3x^2+2x+1) = 28x^6-47x^5-9x^4-81x^3-71x^2-70x-56$.

Then, according to Formula (4), the obtained remainders are $b_1^{4'}(x) = 39x+9$, $b_2^{4'}(x) = 13x^2+29x+6$, and $b_3^{4'}(x) = -36x-22$. The restoration of the fourth block is done based on expression (5): $b_i^4(x) = (b_i^{4'}(x)q_i(x)) \bmod p_i(x) = (b_i^{4'}(x)m_i(x)(k_i^{-1}(x) \bmod p_i(x))) \bmod p_i(x)$ and $b_1^4(x) = ((39x+9)((1/3)x+2/3)(-(1/3)x+1/3)) \bmod (x^2+x+1) = ((39x+9)(-(1/9)x^2-(1/9)x+2/9)) \bmod (x^2+x+1) = (-(13/3)x^3-(16/3)x^2+(23/3)x+2) \bmod (x^2+x+1) = 13x+3$.

The concatenation of the coefficients of the remainders $b_i^j(x)$ corresponds to the input text PSMFSRND = 1518120518171303. According to the agreements between the participants, the ciphertext can be either the parameter $N_i(x)$, or the remainders $b_i^{j'}(x)$, where j is the block number of the message.

3. Results

In this section, we evaluate the cryptographic strength of a polynomial symmetric encryption algorithm in the system of residual classes.

The proposed polynomial symmetric encryption method based on the CRT is cryptographically strong due to the complexity of finding all possible parameter variants and cryptotransform modules. For its cryptanalysis, it is necessary to perform a complete search of all mutually prime polynomials in the ring $Z[x]$ over a simple Galois field GF(p), where p is the prime number. The biggest challenge will be if the polynomial $f(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_0x^0$ is irreducible. Quantity $S_p(n)$ irreducible polynomials of degree n can be calculated by the following formula [51]:

$$S_p(n) = \frac{1}{n}\sum_{n/d}\mu(d)p^{n/d} \tag{10}$$

where $\mu(d)$ is the Möbius function. It is equal to 1 if d is a divisor of degree n with an even number of prime factors, -1 if d is a divisor of degree n with an odd number of prime factors, and 0 if d contains a square of a prime factor. Accordingly, the number of modules l cannot exceed $S_p(n)$. Table 1 shows the Möbius functions for the first 64 positive integers.

Table 1: The Möbius functions for the first 64 positive integers [29].

d	1	2	3	4	5	6	7	8
$\mu(d)$	1	−1	−1	0	−1	1	−1	0
d	9	10	11	12	13	14	15	16
$\mu(d)$	0	1	−1	0	−1	1	1	0
d	17	18	19	20	21	22	23	24
$\mu(d)$	−1	0	−1	0	1	1	−1	0
d	25	26	27	28	29	30	31	32
$\mu(d)$	0	1	0	0	−1	−1	−1	0
d	33	34	35	36	37	38	39	40
$\mu(d)$	−1	1	1	0	−1	1	1	0
d	41	42	43	44	45	46	47	48
$\mu(d)$	−1	−1	−1	0	0	1	−1	0
d	49	50	51	52	53	54	55	56
$\mu(d)$	0	0	1	0	−1	0	1	0
d	57	58	59	60	61	62	63	64
$\mu(d)$	1	1	−1	0	−1	1	0	0

So, to find the number of irreducible polynomials $f(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_0x^0$ over GF(p), you need to determine all divisors of degree n, calculate the Möbius function for each of them, substitute them into Formula (10), and sum them. For example, for irreducible polynomials over GF(p) of degree $n = 3$ with divisors 1 (the value of the Möbius function is 1) and 3 (the value of the Möbius function is −1), their number can be found by the following formula: $S_p(n) = (1/3)\sum_{n/d}\mu(d)p^{n/d} = (1/3)(\mu(1)p^{3/1} + \mu(3)p^{3/3}) = (p^3-p)/3$.

For a polynomial of degree 32 (the divisors are 1, 2, 4, 8, 16, and 32) over GF(2), according to Formula (10), the number of irreducible polynomials is as follows: $S_2(32) = (1/32)\sum_{32/d}\mu(d)2^{32/d} = (1/32)(\mu(1)2^{32/1} + \mu(2)2^{32/2} + \mu(4)2^{32/4} + \mu(8)2^{32/8} + \mu(16)2^{32/16} + \mu(32)2^{32/32}) = (1/32)(2^{32} - 2^{16}) = 2^{27} - 2^{11} = 134215680$.

Below is an example of calculating the number of irreducible polynomials for $n = 32$ in GF(p), where $p = 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 43$:

$$\begin{aligned} S_3(32) = \frac{1}{32}\sum_{32/d}\mu(d)3^{32/d} &= \left(\mu(1)3^{32/1} + \mu(2)3^{32/2}\right. \\ &+ \mu(4)3^{32/4} + \mu(8)3^{32/8} + \mu(16)3^{32/16} \\ &\left. + \mu(32)3^{32/32}\right) = \frac{1}{32}\left(3^{32} - 3^{16}\right) = 57906879556410 \end{aligned}$$

$$S_5(32) = \frac{1}{32}\sum_{32/d}\mu(d)5^{32/d} = \left(\mu(1)5^{32/1} + \mu(2)5^{32/2} + \mu(4)5^{32/4} + \mu(8)5^{32/8} + \mu(16)5^{32/16} + \mu(32)5^{32/32}\right) = \frac{1}{32}\left(5^{32} - 5^{16}\right) = 727595761413574 * 10^6$$

$$S_7(32) = \frac{1}{32}\sum_{32/d}\mu(d)7^{32/d} = \left(\mu(1)7^{32/1} + \mu(2)7^{32/2} + \mu(4)7^{32/4} + \mu(8)7^{32/8} + \mu(16)7^{32/16} + \mu(32)7^{32/32}\right) = \frac{1}{32}\left(7^{32} - 7^{16}\right) = 34513364820121 * 10^{11}$$

$$S_{11}(32) = \frac{1}{32}\sum_{32/d}\mu(d)11^{32/d} = \left(\mu(1)11^{32/1} + \mu(2)11^{32/2} + \mu(4)11^{32/4} + \mu(8)11^{32/8} + \mu(16)11^{32/16} + \mu(32)11^{32/32}\right) = \frac{1}{32}\left(11^{32} - 11^{16}\right) = 65980552329226 * 10^{17}$$

$$S_{13}(32) = \frac{1}{32}\sum_{32/d}\mu(d)13^{32/d} = \left(\mu(1)13^{32/1} + \mu(2)13^{32/2} + \mu(4)13^{32/4} + \mu(8)13^{32/8} + \mu(16)13^{32/16} + \mu(32)13^{32/32}\right) = \frac{1}{32}\left(13^{32} - 13^{16}\right) = 138368519930263 * 10^{20}$$

$$S_{17}(32) = \frac{1}{32}\sum_{32/d}\mu(d)17^{32/d} = \left(\mu(1)17^{32/1} + \mu(2)17^{32/2} + \mu(4)17^{32/4} + \mu(8)17^{32/8} + \mu(16)17^{32/16} + \mu(32)17^{32/32}\right) = \frac{1}{32}\left(17^{32} - 17^{16}\right) = 739972373362646 * 10^{23}$$

$$S_{19}(32) = \frac{1}{32}\sum_{32/d}\mu(d)19^{32/d} = \left(\mu(1)19^{32/1} + \mu(2)19^{32/2} + \mu(4)19^{32/4} + \mu(8)19^{32/8} + \mu(16)19^{32/16} + \mu(32)19^{32/32}\right) = \frac{1}{32}\left(19^{32} - 19^{16}\right) = 259995153315273 * 10^{25}$$

$$S_{23}(32) = \frac{1}{32}\sum_{32/d}\mu(d)23^{32/d} = \left(\mu(1)23^{32/1} + \mu(2)23^{32/2} + \mu(4)23^{32/4} + \mu(8)23^{32/8} + \mu(16)23^{32/16} + \mu(32)23^{32/32}\right) = \frac{1}{32}\left(23^{32} - 23^{16}\right) = 117527845345372 * 10^{28}$$

$$S_{29}(32) = \frac{1}{32}\sum_{32/d}\mu(d)29^{32/d} = \left(\mu(1)29^{32/1} + \mu(2)29^{32/2} + \mu(4)29^{32/4} + \mu(8)29^{32/8} + \mu(16)29^{32/16} + \mu(32)29^{32/32}\right) = \frac{1}{32}\left(29^{32} - 29^{16}\right) = 195697804967027 * 10^{31}$$

$$S_{31}(32) = \frac{1}{32}\sum_{32/d}\mu(d)31^{32/d} = \left(\mu(1)31^{32/1} + \mu(2)31^{32/2} + \mu(4)31^{32/4} + \mu(8)31^{32/8} + \mu(16)31^{32/16} + \mu(32)31^{32/32}\right) = \frac{1}{32}\left(31^{32} - 31^{16}\right) = 165357624391381 * 10^{32}$$

$$S_{37}(32) = \frac{1}{32}\sum_{32/d}\mu(d)37^{32/d} = \left(\mu(1)37^{32/1} + \mu(2)37^{32/2} + \mu(4)37^{32/4} + \mu(8)37^{32/8} + \mu(16)37^{32/16} + \mu(32)37^{32/32}\right) = \frac{1}{32}\left(37^{32} - 37^{16}\right) = 475669375729533 * 10^{34}$$

$$S_{43}(32) = \frac{1}{32}\sum_{32/d}\mu(d)43^{32/d} = \left(\mu(1)43^{32/1} + \mu(2)43^{32/2} + \mu(4)43^{32/4} + \mu(8)43^{32/8} + \mu(16)43^{32/16} + \mu(32)43^{32/32}\right) = \frac{1}{32}\left(43^{32} - 43^{16}\right) = 583231000811536 * 10^{36}$$

Table 2 shows the number of irreducible polynomials for powers of $n = 4$, 8, 16, 32, 64, 96, and 128 and parameter $p = 3$, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, and 43.

The numerical experiment shows that the number of irreducible polynomials increases with the expansion of the Galois field and the growth of the power. For the specified parameters p and n, it will lie in the following ranges: $30 \le S_p(8) \le 1.46^{12}$, $4048 \le S_p(16) \le 8.54 * 10^{24}$, $1.34 * 10^8 \le S_p(32) \le 5.832 * 10^{50}$, $2.8823 * 10^{17} \le S_p(64) \le 5.4425 * 10^{104}$, $8.25293 * 10^{26} \le S_p(96) \le 6.7717 * 10^{154}$, and $2.65846 * 10^{36} \le S_p(n) \le 9.4788 * 10^{206}$. In addition, a given number of irreducible polynomials can be obtained either by changing the degree n or by expanding the field GF(p). For instance, $S_{43}(8) \approx S_7(16)$, $S_{43}(16) \approx S_7(32)$, $S_{43}(32) \approx S_7(64)$, $S_{43}(64) \approx S_7(96)$, and $S_{43}(96) \approx S_7(128)$.

Figure 3 shows the graphical dependence of the number of irreducible polynomials on these parameters on a logarithmic scale with base 10.

In general, the security of the proposed cryptosystem with l modules will be defined as the total time of complete search of all irreducible polynomials and the complexity of

Table 2: The number of irreducible polynomials for different powers of n and values of the parameter p.

p/n	4	8	16	32	64	96	128
2	3	30	4080	134215680	2.88×10^{17}	8.25×10^{26}	2.66×10^{36}
3	18	810	2690010	5.79×10^{13}	5.37×10^{28}	6.63×10^{43}	9.21×10^{58}
5	150	48750	9.54×10^{9}	7.28×10^{20}	8.47×10^{42}	1.31×10^{65}	2.29×10^{87}
7	588	720300	2.08×10^{12}	3.45×10^{25}	1.90×10^{52}	1.40×10^{79}	1.16×10^{106}
11	3630	26793030	2.87×10^{15}	6.60×10^{31}	6.97×10^{64}	9.80×10^{97}	1.55×10^{131}
13	7098	1.02×10^{8}	4.16×10^{16}	1.38×10^{34}	3.06×10^{69}	9.04×10^{104}	3.00×10^{140}
17	20808	8.72×10^{8}	3.04×10^{18}	7.40×10^{37}	8.76×10^{76}	1.38×10^{116}	2.46×10^{155}
19	32490	2.12×10^{9}	1.8×10^{19}	2.60×10^{39}	1.08×10^{80}	5.99×10^{120}	3.74×10^{161}
23	69828	9.79×10^{9}	3.83×10^{20}	1.18×10^{42}	2.21×10^{85}	5.54×10^{128}	1.56×10^{172}
29	176610	6.25×10^{10}	1.56×10^{22}	1.96×10^{45}	6.12×10^{91}	2.55×10^{138}	1.20×10^{185}
31	230640	1.07×10^{11}	4.55×10^{22}	1.65×10^{46}	4.37×10^{93}	1.54×10^{141}	6.12×10^{188}
37	468198	4.39×10^{11}	7.71×10^{23}	4.75×10^{48}	3.62×10^{98}	3.67×10^{148}	4.19×10^{198}
41	706020	9.98×10^{11}	3.98×10^{24}	1.27×10^{50}	2.58×10^{101}	6.99×10^{152}	2.13×10^{204}
43	854238	1.46×10^{11}	8.54×10^{24}	5.83×10^{50}	5.44×10^{102}	6.77×10^{154}	9.47×10^{206}

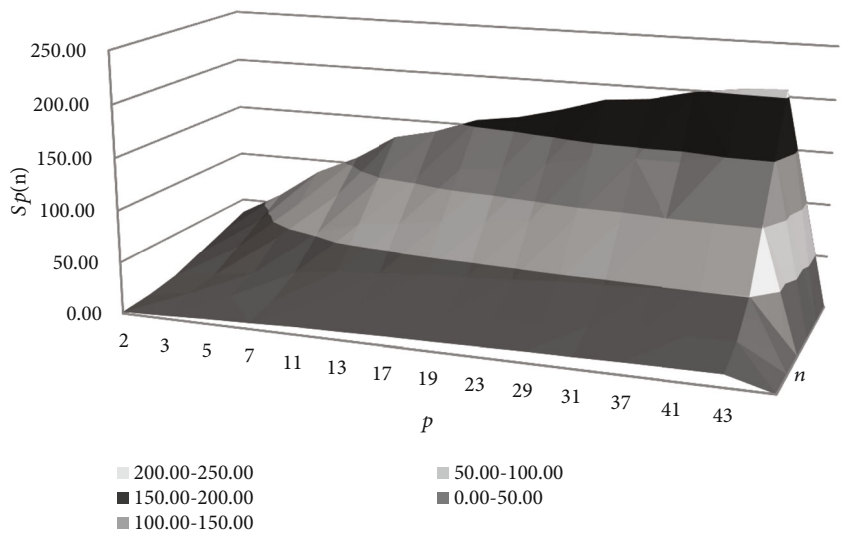

Figure 3: Number of mutually prime modules $S_p(n)$ depending on the Galois field p and degree n.

performing calculations with each one according to the following formula:

$$O(n, l) = C^{l}_{S_p(n)} n^2 \log l = \frac{(S_p(n))! * n^2 \log l}{(S_p(n) - l)! * l!} \quad (11)$$

For instance, we can calculate the time (in clock cycles) needed to cryptanalyze the proposed encryption system with $l = 5$ and $n = 32$ in the Galois field GF(3) as follows: $O(32, 5) = C^{5}_{57906879556410} * 32^2 \log 5 = ((57906879556410)!/(57906879556405)! * 5!) 32^2 \log 5 \approx 1.29 * 10^{70}$.

Table 3 estimates the cryptanalysis time in clock cycles for different parameter values of l, n, and $S_p(n)$. Notably, the modern symmetric encryption algorithm AES-128 requires around $2^{128} \approx 10^{37}$ clock cycles for resilience. Table 3 indicates that the proposed cryptosystem achieves a comparable level of security with the following parameters: $S_{31}(4)$, $l = 7$, and $n = 4$; $S_{11}(8)$, $l = 5$, and $n = 8$; $S_3(16)$, $l = 6$, and $n = 16$; $S_2(32)$, $l = 3$, and $n = 32$; and $S_2(64)$, $l = 2$, and $n = 64$.

Table 3 shows that adding one module for parameters $n = 4$ and $S_{31}(4)$ increases the strength by about 5 orders of magnitude, for $n = 8$ and $S_{11}(8)$ by 7 orders of magnitude, for $n = 16$ and $S_3(16)$ by 6 orders of magnitude, for $n = 32$

Table 3: Estimation of cryptanalysis time in clock cycles for various parameter values of l, n, and $S_p(n)$.

$l/n(S_p(n))$	$4(S_{31}(4))$	$8(S_{11}(8))$	$16(S_3(16))$	$32(S_2(32))$	$64(S_2(64))$
2	1.5×10^{11}	7.9×10^{15}	3.17×10^{14}	5.879×10^{29}	5.8×10^{37}
3	1.7×10^{16}	1.03×10^{23}	4.2×10^{20}	1.674×10^{43}	8.3×10^{54}
4	3.8×10^{21}	2.8×10^{30}	1.1×10^{27}	9.595×10^{56}	2.4×10^{72}
5	2.02×10^{26}	1.7×10^{37}	6.98×10^{32}	1.29×10^{70}	1.6×10^{89}
6	8.6×10^{31}	8.5×10^{43}	3.5×10^{38}	1.386×10^{83}	8.4×10^{105}
7	3.09×10^{35}	3.5×10^{50}	1.5×10^{44}	1.245×10^{96}	3.8×10^{122}

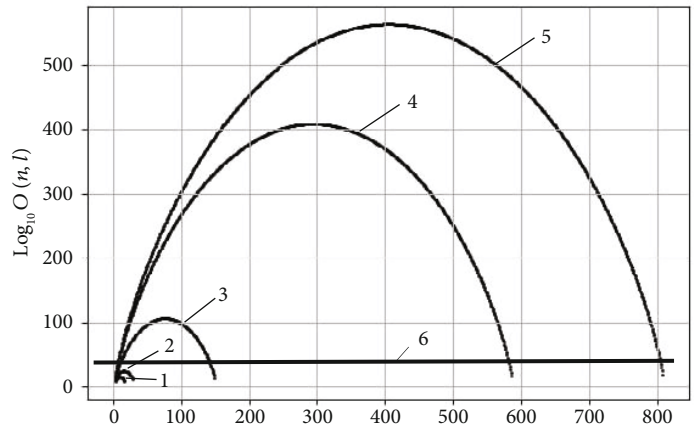

Figure 4: Graphs of cryptographic strength dependencies $O(n, l)$ in a logarithmic scale with base 10 of the proposed symmetric polynomial encryption algorithm in RNS from the number of modules l and the powers of the polynomial n (line 1, $p = 2$, $n = 4$; line 2, $p = 2$, $n = 8$; line 3, $p = 5$, $n = 4$; line 4, $p = 7$, $n = 4$; line 5, $p = 3$, $n = 8$; and line 6, algorithm AES-128).

and $S_2(32)$ by 13 orders of magnitude, and for $n = 64$ and $S_2(64)$ by 17 orders of magnitude.

Figure 4 shows the graphs of cryptographic strength dependencies $O(n, l)$ on a logarithmic scale with a base of 10 of the proposed symmetric polynomial encryption algorithm in RNS on the number of modules l for the polynomial powers $n = 4$ and 8 and the parameters $p = 2$, $p = 3$, $p = 5$, and $p = 7$. The horizontal line 6 corresponds to the strength of the modern symmetric encryption algorithm AES-128.

The figure shows that all graphs have the same bell-shaped character. The cryptographic strength increases significantly with increasing degree and dimension of the Galois field p and reaches its maximum at $l = S_p(n)/2$. This means that the cryptanalysis of the proposed algorithm requires combinatorial complexity, which leads to an NP-complete problem.

4. Conclusion

In this article, we first developed symmetric cryptographic algorithms based on the polynomial RNS. The mathematical support and schemes of the proposed polynomial symmetric encryption in the RNS are developed. To evaluate its robustness, we have studied and constructed analytical expressions that indicate that the cryptanalysis of the proposed algorithm requires combinatorial complexity, which leads to an NP-complete problem. It is established that the cryptographic strength increases significantly with the increasing degree and dimension of the Galois field p and reaches its maximum in the case when the number of modules is equal to half the possible number of irreducible polynomials with given polynomial degrees and Galois field orders. This means that finding an efficient algorithm to solve this problem requires significant computing resources and time.

We compare the strength of the proposed encryption method with the modern symmetric encryption algorithm AES-128. As a result of numerical experiments, it was found that the developed polynomial encryption methods in the RNS provide a level of resistance similar to AES-128 with the following parameters: $S_{31}(4)$, $l = 7$, and $n = 4$; $S_{11}(8)$, $l = 5$, and $n = 8$; $S_3(16)$, $l = 6$, and $n = 16$; $S_2(32)$, $l = 3$, and $n = 32$; and $S_2(64)$, $l = 2$, and $n = 64$.

Thus, the proposed cryptographic algorithm based on the polynomial RNS can be used to ensure reliable protection of confidential information in systems with limited computing resources.

Data Availability Statement

The authors confirm that the data supporting the findings of this study are available within the article.

Consent

All authors have provided their consent for the publication of this research.

Conflicts of Interest

The authors declare no conflicts of interest.

Author Contributions

Conceptualization: I.Y. and M.K. (Mykhailo Kasianchuk); methodology: I.Y. and M.K. (Mykhailo Kasianchuk); validation: M.K. (Mykhailo Kasianchuk), M.K. (Mikolaj Karpinski), and R.S.; formal analysis: M.K.(Mykhailo Kasianchuk) and M.K.(Mikolaj Karpinski); investigation: I.Y., M.K. (Mykhailo Kasianchuk), M.K.(Mikolaj Karpinski), and R.S.; data curation: I.Y. and M.K.(Mykhailo Kasianchuk); writing—original draft preparation: I.Y., M.K.(Mykhailo Kasianchuk), and R.S.; writing—review and editing: I.Y., M.K.(Mykhailo Kasianchuk), and R.S. All authors have read and agreed to the published version of the article.

Funding

The authors received no specific funding for this work.

Acknowledgments

The authors express their sincere gratitude to the Armed Forces of Ukraine for providing security, which made it possible to conduct our research.

References

[1] M. Nieles, K. Dempsey, and V. Y. Pillitteri, "An introduction to information security," *NIST Special Publication*, vol. 800, no. 12, p. 101, 2017.

[2] O. Milov, N. Kazakova, P. Milczarski, and O. Korol, "Mechanisms of cyber security: the problem of conceptualization," *Ukrainian Scientific Journal of Information Security*, vol. 25, no. 2, pp. 110–116, 2019.

[3] M. M. Alhassan and A. Adjei-Quaye, "Information security in an organization," *International Journal of Computer (IJC)*, vol. 24, no. 1, pp. 100–116, 2017.

[4] V. Adki and S. Hatkar, "A survey on cryptography techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, no. 6, pp. 469–475, 2016.

[5] M. Bufalo, D. Bufalo, and G. Orlando, "A note on the computation of the modular inverse for cryptography," *Axioms*, vol. 10, no. 2, p. 116, 2021.

[6] M. Kasianchuk, I. Yakymenko, and Y. Nykolaychuk, "Symmetric cryptoalgorithms in the residue number system," *Cybernetics and Systems Analysis*, vol. 57, no. 2, pp. 329–336, 2021.

[7] Y. Nykolaychuk, I. Yakymenko, N. Vozna, and M. Kasianchuk, "Residue number system asymmetric cryptoalgorithms," *Cybernetics and Systems Analysis*, vol. 58, no. 4, pp. 611–618, 2022.

[8] E. Ochoa-Jimenez, L. Rivera-Zamarripa, N. Cruz-Cortes, and F. Rodriguez-Henriquez, "Implementation of RSA signatures on GPU and CPU architectures," *IEEE Access*, vol. 8, pp. 9928–9941, 2020.

[9] R. Biyashev, S. Nyssanbayeva, and N. Kapalova, *Secret Keys for Nonpositional Cryptosystems. Development, Investigation and Implementation*, Lambert Academic Publishing, 2014.

[10] R. Biyashev, S. Nyssanbayeva, M. Kalimoldayev, and M. Magzom, "Development of an encryption algorithm based on nonpositional polynomial notations," in *2016 International Conference on Advanced Materials Science and Environmental Engineering*, pp. 241–243, Atlantis Press, 2016.

[11] J. H. Cheon, S. Hong, C. Lee, and Y. Son, "Polynomial functional encryption scheme with linear ciphertext size," Cryptology ePrint Archive, 2018.

[12] N. Kapalova and D. Dyusenbayev, "Security analysis of an encryption scheme based on nonpositional polynomial notations," *Open Engineering*, vol. 6, no. 1, 2016.

[13] J.-C. Bajard, L. Imbert, and T. Plantard, "Arithmetic operations in the polynomial modular number system," in *17th IEEE Symposium on Computer Arithmetic (ARITH'05)*, pp. 206–213, Cape Cod, MA, USA, 2005.

[14] L.-S. Didier, F.-Y. Dosso, and P. Véron, "Efficient modular operations using the adapted modular number system," *Journal of Cryptographic Engineering*, vol. 10, no. 2, pp. 111–133, 2020.

[15] D. Harvey and J. van Der Hoeven, "Faster polynomial multiplication over finite fields using cyclotomic coefficient rings," *Journal of Complexity*, vol. 54, article 101404, 2019.

[16] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Advances in Cryptology–EUROCRYPT 2008: 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings 27*, pp. 146–162, Springer, Berlin Heidelberg, 2008.

[17] G. Ashok, A. Kumar, and C. Kumari, "An approach of cryptosystem using polynomials and Lucas numbers," *Journal of Harbin Engineering University*, vol. 44, no. 8, pp. 25–31, 2023.

[18] G. Ashok, K. Ashok, K. Chaya, and M. Ramakrishna, "A type of public cryptosystem using polynomials and pell sequences," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 7, pp. 1951–1963, 2022.

[19] P. V. A. Mohan, "Specialized residue number systems," Springer, 2016.

[20] T. Plantard, "Efficient word size modular arithmetic," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1506–1518, 2021.

[21] A. R. Omondi and A. B. Premkumar, *Residue Number Systems: Theory and Implementation*, World Scientific, 2007.

[22] J.-C. Bajard and L. Imbert, "A full RNS implementation of RSA," *IEEE Transactions on Computers*, vol. 53, no. 6, pp. 769–774, 2004.

[23] M. Lawnik and A. Kapczy'nski, "The application of modified Chebyshev polynomials in asymmetric cryptography," *Computer Science*, vol. 20, no. 3, pp. 303–367, 2019.

[24] J. S. Milne, "Algebraic number theory (v3.08)," JS Milne, 2020. http://www.jmilne.org/math/.

[25] A. P. Fournaris, L. Papachristodoulou, L. Batina, and N. Sklavos, "Secure and efficient RNS approach for elliptic curve cryptography," in *In 6th Conference on Trustworthy Manufacturing and Utilization of Secure Devices (TRUDEVICE 2016)*, pp. 121–126, Barcelona, 2016.

[26] R. A. Patel, M. Benaissa, and S. Boussakta, "Fast parallel-prefix architectures for modulo 2n-1 addition with a single representation of zero," *IEEE Transactions on Computers*, vol. 56, no. 11, pp. 1484–1492, 2007.

[27] Z. Gao, P. Reviriego, W. Pan et al., "Fault tolerant parallel filters based on error correction codes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 2, pp. 384–387, 2015.

[28] A. Sachenko, Z. Hu, and V. Yatskiv, "Increasing the data transmission robustness in WSN using the modified error correction codes on residue number system," *Elektronika ir Elektrotechnika*, vol. 21, no. 1, pp. 76–81, 2015.

[29] K. Givaki, R. Hojabr, H. Najafi et al., "Using residue number systems to accelerate deterministic bit-stream multiplication," in *2019 IEEE 30th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, New York, NY, USA, 2019.

[30] W. Tan, S.-W. Chiu, A. Wang, Y. Lao, and K. K. Parhi, "PaR-eNTT: low-latency parallel residue number system and NTT-based long polynomial modular multiplication for homomorphic encryption," 2023, https://arxiv.org/abs/2303.02237.

[31] E. Lemaire, "Pretty modular symmetric encryption (PMSE), compact algorithm for embedded cryptography with quite low computational cost," 2019, https://arxiv.org/abs/1905.08150.

[32] V. Migliore, M. M. Real, V. Lapotre, A. Tisserand, C. Fontaine, and G. Gogniat, "Fast polynomial arithmetic for somewhat homomorphic encryption operations in hardware with Karatsuba algorithm," in *2016 International Conference on Field-Programmable Technology (FPT)*, pp. 209–212, Xi'an, China, 2016.

[33] C. Jayet-Griffon, M.-A. Cornelie, P. Maistri, P. Elbaz-Vincent, and R. Leveugle, "Polynomial multipliers for fully homomorphic encryption on FPGA," in *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, Riviera Maya, Mexico, 2015.

[34] K. Shivdikar, G. Jonatan, E. Mora et al., "Accelerating polynomial multiplication for homomorphic encryption on GPUs," in *2022 IEEE International Symposium on Secure and Private Execution Environment Design (SEED)*, pp. 61–72, Storrs, CT, USA, 2022.

[35] J. Wu, H. Liu, and X. Zhu, "Image encryption based on permutation polynomials over finite fields," *Optica Applicata*, vol. 50, no. 3, 2020.

[36] D. Schinianakis and T. Stouraitis, "Multifunction residue architectures for cryptography," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 4, pp. 1156–1169, 2014.

[37] S. Tynymbayev, M. Ibraimov, T. Namazbayev, and S. Gnatyuk, "Development of pipelined polynomial multiplier modulo irreducible polynomials for cryptosystems," *Eastern-European Journal of Enterprise Technologies*, vol. 1, no. 4, p. 115, 2022.

[38] A. B. Alamsyah and T. B. Adji, "The replacement of irreducible polynomial and affine mapping for the construction of a strong S-box," *Nonlinear Dynamics*, vol. 93, no. 4, pp. 2105–2118, 2018.

[39] A. Alamsyah, "A novel construction of perfect strict avalanche criterion S-box using simple irreducible polynomials," *Scientific Journal of Informatics*, vol. 7, no. 1, pp. 10–22, 2020.

[40] P. Agarwal, A. Singh, and A. Kilicman, "Development of key-dependent dynamic S-boxes with dynamic irreducible polynomial and affine constant," *Advances in Mechanical Engineering*, vol. 10, no. 7, Article ID 168781401878163, 2018.

[41] K. F. Alshammari, A. Mostafa, and S. Nashwan, "Avalanche analysis of variant polynomials for AES," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 14, pp. 2696–2703, 2021.

[42] J. Chu and M. Benaissa, "A novel architecture of implementing error detecting AES using PRNS," in *2011 14th Euromicro Conference on Digital System Design*, Oulu, Finland, 2011.

[43] A. N. El-Kassar, R. Haraty, Y. Awad, and N. Debnath, "Modified RSA in the domains of Gaussian integers and polynomials over finite fields," In CAINE, 2005.

[44] I. Yakymenko, M. Kasianchuk, I. Shylinska, R. Shevchuk, V. Yatskiv, and M. Karpinski, "Polynomial Rabin cryptosystem based on the operation of addition," in *2022 12th International Conference on Advanced Computer Information Technologies (ACIT)*, pp. 345–350, Ruzomberok, Slovakia, 2022.

[45] P. Turner, "Residue polynomial systems," *Theoretical Computer Science*, vol. 279, no. 1-2, pp. 29–49, 2002.

[46] F. Y. Dosso, A. Berzati, N. El Mrabet, and J. Proy, "PMNS revisited for consistent redundancy and equality test," Cryptology ePrint Archive, 2023.

[47] L.-S. Didier, F.-Y. Dosso, N. El Mrabet, J. Marrez, and P. Véron, "Randomization of arithmetic over polynomial modular number system," in *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pp. 199–206, Kyoto, Japan, 2019.

[48] M. Karpinski, S. Rajba, S. Zawislak et al., "A method for decimal number recovery from its residues based on the addition of the product modules," in *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, pp. 13–17, Metz, France, 2019.

[49] A. Aremu and A. Gbolagade, "Redundant residue number system based multiple error detection and correction using Chinese remainder theorem (CRT)," *Software Engineering*, vol. 5, no. 5, pp. 72–80, 2017.

[50] Y. Nykolaychuk, M. Kasianchuk, and I. Yakymenko, "Theoretical foundations for the analytical computation of coefficients of basic numbers of Krestenson's transformation," *Cybernetics and Systems Analysis*, vol. 50, no. 5, pp. 649–654, 2014.

[51] A. Beletsky, A. Kovalchuk, K. Novikov, and D. Poltoratskyi, "Algorithm for the synthesis of irreducible polynomials of linear complexity," *Ukrainian Information Security Research Journal*, vol. 22, no. 2, pp. 74–87, 2020.