Research Article

Symmetric Encryption Algorithms in a Polynomial Residue Number System

I. Yakymenko, M. Karpinski, R. Shevchuk, and M. Kasianchuk

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

Copyright © 2024 I. Yakymenko et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 $\mathbb{Z}[x]$, 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 $\mathbb{Z}[x]$ 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 cryptalogarithms. 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 of 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 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 irredicible 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 \( SAC = 0.5 \), 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 paralleling 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]:

$$b_i(x) = N(x) \mod p_i(x) \quad (1)$$

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

$$N(x) = \left[\sum_{i=1}^{s} b_i(x)M_i(x)m_i(x)\right] \mod P(x) \quad (2)$$

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 $m_i(x) = M_i^{-1}(x) \mod p_i(x)$, and $s$ is the number of modules. For polynomial powers, the inequality deg $N(x) < \text{deg } P(x)$ 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 $m_i(x) = M_i^{-1}(x) \mod p_i(x)$, 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: $1 < \text{deg } k_i(x) < \text{deg } p_i(x)$ and $\text{GCD}(k_i(x), p_i(x)) = 1$, 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 $N(x) = a_0x^n + a_{n-1}x^{n-1} + \cdots + a_nx^n$, where $a_i$ is the sequence of digital representation of letters and $i = 0, \ldots, n$, $(n + 1)$ 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:

$$N'(x) = \left[\sum_{i=1}^{s} b_i(x)M_i(x)k_i(x)\right] \mod P(x) \quad (3)$$

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:

$$q_i(x) = (m_i(x) \cdot (k_i^{-1}(x) \mod p_i(x))) \mod p_i(x)$$
$$b'_i(x) = N'(x) \mod p_i(x) \quad (4)$$

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

$$b_i(x) = \left(b'_i(x)q_i(x)\right) \mod p_i(x)$$
$$= \left(b'_i(x)m_i(x)(k_i^{-1}(x) \mod p_i(x))\right) \mod p_i(x) \quad (5)$$

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:

$$N(x) = \left[\sum_{i=1}^{s} M_i(x)m_i(x)(b'_i(x)m_i(x)(k_i^{-1}(x) \mod p_i(x))\right) \mod p_i(x) \right] \mod P(x)$$

$$= \left[\sum_{i=1}^{s} M_i(x)m_i(x)(b'_i(x)q_i(x)) \mod p_i(x) \right] \mod P(x) \quad (6)$$

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

\[
b_i(x) = \left( b_i(x)q_i(x) \right) \mod p_i(x) = \left( \left( N_i'(x) \mod p_i(x) \right) \cdot (m_i(x)(k_i^{-1}(x) \mod p_i(x))) \mod p_i(x) \right) \mod p_i(x)
\]

\[
= \left( \left( \left( \left( b_i(x)k_i(x)M_i(x) \right) \mod P(x) \right) (m_i(x)(k_i^{-1}(x) \mod p_i(x))) \mod p_i(x) \right) \mod p_i(x) \right) \mod p_i(x)
\]

\[
= \left( \left( b_i(x)M_i(x) \right) \mod P(x) \right) (m_i(x)(k_i^{-1}(x) \mod p_i(x))) \mod p_i(x)
\]

\[
= \left( b_i(x)m_i(x)M_i(x) \right) \mod p_i(x) = b_i(x)
\]

(7)

2.3. An Example of Symmetric Polynomial Encryption in RNS.
Let us consider the plaintext \( PSFSDR = (15181205181703) \), 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 \( (s = 3) \), \( p_i(x) = x^2 + x + 1 \), and \( p_2(x) = x^3 + x + 1 \), and the chosen coefficients \( k_1(x) = x^2 + 2x + 3 \), \( k_2(x) = x^2 + 3x + 1 \), and \( k_3(x) = x^2 + 2x + 3 \), all the parameters are calculated as follows: \( P(x) = p_1(x)p_2(x) \equiv p_3(x) = x^2 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1 \), \( M_1(x) = P(x)/p_1(x) = (x^2 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1)/(x^2 + x + 1) = x^4 + x^2 + 2x^2 + x + 1 \), and \( M_3(x) = P(x)/p_3(x) = (x^2 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1)/(x^2 + 1) = x^4 + x^2 + 2x^2 + 2x^2 + 1 \).

The search for \( m_i(x) = M_i^{-1}(x) \mod p_i(x) \) is performed using the method of undetermined coefficients. Firstly, we look for \( m_i(x) = M_i^{-1}(x) \mod p_i(x) \) and \( m_i(x) = (x^2 + 2x^2 + x + 1)^{-1} \mod (x^2 + x + 1)^{-1} \mod (x^2 + 1)^{-1} \). To do this, we write the equation \( (x + 2)(Ax + B) \) and, after transformations, we obtain \( (Ax^2 + (2A + B)x + 2B) \) and \( (x^2 + x + 1) = (A + B)x + 2B - A = 1 \). 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 \( m_1(x) = (x^2 + 1)^{-1} \mod (x^2 + x + 1) = (1/3)x + 2/3 \). Similarly, the search for \( m_2(x) = M_2^{-1}(x) \) mod \( p_2(x) = (x^4 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1)^{-1} \mod (x^2 + x + 1)^{-1} \mod (x^2 + 1)^{-1} \) is carried out. We write \( (x^2 - x)(Ax^2 + Bx + C) \mod (x^2 + 1)^{-1} = 1 \), 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 \( (2A x^4 + 2Bx^3 + (2A + 2C)x^2 + Bx + C) \mod (x^2 + x + 1) = (C - A - B)x^2 + (-2C - B)x + A - B = 1 \), we obtain \( 2C - A = 0 \), \( C - B = 0 \), \( B - A = 0 \), and \( A - B = 1 \). From here, \( A = 2/3 \), \( B = -1/3 \), and \( C = 1/3 \). So, the inverse polynomial takes the following form: \( m_2(x) = (x^2 - x)^{-1} \mod (x^2 + x + 1) = (2/3)x^2 - (1/3)x + 1 \). Similarly, the value \( m_3(x) = M_3^{-1}(x) \) mod \( p_3(x) = (x^2 + x^6 + 3x^5 + 3x^4 + 4x^3 + 3x^2 + 2x + 1)^{-1} \mod (x^2 + 1)^{-1} \) is computed. By applying the method of undetermined coefficients, the following transformations can be performed: \( (x)(Ax + B) \) mod \( (x^2 + 1) = (Ax^2 + Bx) \) mod \( (x^2 + 1) \).
\[ N'(x) = \sum_{i=1}^{\nu} b_i(x)M_i(x) \mod P(x) \] (8)

It should be noted that the operation of finding the inverse polynomial modulus 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:

\[ b_i(x) = (b'_i(x)m_i(x)) \mod p_i(x) \]

\[ N(x) = \sum_{i=1}^{\nu} M_i(x)m_i(x) \left( (b'_i(x)m_i(x)) \mod p_i(x) \right) \mod P(x) \] (9)

For the same input parameters as in the previous example, according to Formulas (8) and (9), the following ciphertext is obtained:

\[ N'(x) = \sum_{i=1}^{\nu} b_i(x)M_i(x) \mod P(x) = ((x^5 + 2x^3 + x^2 + x + 1)^{-1}(-7x - 13) + (x^4 + x^4 + x^2 + x + 1)(2x + 2x + 2x + 2) + (x^6 + x^6 + 3x^4 + 4x^3 + 3x^2 + 2x + 1) = 16x^6 + 18x^5 + 12x^4 + 5x^3 + 18x^2 + 17x + 3. \]

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:

\[ b_i(x) = (b'_i(x)m_i(x)) \mod p_i(x) \]

\[ N(x) = \sum_{i=1}^{\nu} M_i(x)m_i(x) \left( (b'_i(x)m_i(x)) \mod p_i(x) \right) \mod P(x) \] (9)

For the same input parameters as in the previous example, according to Formulas (8) and (9), the following ciphertext is obtained:

\[ N'(x) = \sum_{i=1}^{\nu} b_i(x)M_i(x) \mod P(x) = ((x^5 + 2x^3 + x^2 + x + 1)^{-1}(-7x - 13) + (x^4 + x^4 + x^2 + x + 1)(2x + 2x + 2x + 2) + (x^6 + x^6 + 3x^4 + 4x^3 + 3x^2 + 2x + 1) = 16x^6 + 18x^5 + 12x^4 + 5x^3 + 18x^2 + 17x + 3. \]

Then, the original message is recovered as the plaintext: \( N'(x) = \sum_{i=1}^{\nu} M_i(x)m_i(x) \left( (b'_i(x)m_i(x)) \mod p_i(x) \right) \mod P(x) \)
This simplification reduces computational complexity by avoiding the operation of finding the parameters $q_i(x)$ and $k_i^{-1}(x) \mod p_i$.

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 $\deg N_i(x) \leq n - 1$; 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 $q_i(x), b_i(x) = N_i(x) \mod p_i(x) = N'_i(x)$ 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 PSMFRSN = 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 $p_i(x) = x^2 + x + 1, p_2(x) = x^2 + x + 1$, and $p_3(x) = x^2 + 1$, and the corresponding coefficients $k_1(x) = x^2 + 2x + 3, k_2(x) = x^2 + x + 1, k_3(x) = x^2 + 3x + 2$. The ciphertext for the first block is computed based on Formula (3): $N'_1(x) = (\sum_{i=1}^{3} b_i(x)M_i(x)k_i(x)) \mod P(x) = ((15x + 18)(x^2 + 2x^3 + x^3 + x + 1)(x^2 + 2x + 3) + (x^4 + x^3 + 2x^3 + x + 1)(15x + 18)(x^2 + x^2 + 1) + (x^4 + 2x^4 + 2x^2 + 2x^2 + 2x + 1)(15x + 18)(x^2 + 3x + 2)) \mod (x^2 + x^2 + x^3 + 3x^4 + 4x^2 + 2x + 1) = 105x^6 + 33x^5 + 135x^4 + 81x^3 + 78x^2 - 21x - 21$.

Upon decryption using Formulas (4) and (5), the following results are obtained: $b'_1(x) = 45x + 54, b'_2(x) = 15x^2 + 48x + 36$, and $b'_3(x) = -27x - 69$. The message is reconstructed based on the relation (5): $b'_i(x) = ((45x + 54)(1/3)x + 2/3)(-1/3)x + 1/3)) \mod (x^2 + x + 1) = ((45x + 54)(-1/9)x^2 - (1/9)x + 2/9)) \mod (x^2 + x + 1) = (-5x^3 - 11x^2 + 4x + 12) \mod (x^2 + x + 1) = 15x + 18$.

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

According to Formula (4), the remainders obtained are $b'_1(x) = 36x + 15, b'_2(x) = 12x^2 + 29x + 10$, and $b'_3(x) = -31x - 27$. The restoration of the second block of the input message is done using relation (5): $b'_i(x) = ((36x + 15)(1/3)x + 2/3)(-1/3)x + 1/3)) \mod (x^2 + x + 1) = ((36x + 15)(-1/9)x^2 - (1/9)x + 2/9)) \mod (x^2 + x + 1) = (-4x^3 - 17/3)x^2 + (19/3)x + 10/3) \mod (x^2 + x + 1) = 12x + 5$.

The ciphertext for the third block of the input message $N_3(x) = 18x + 17$ will have the following form: $N'_3(x) = (\sum_{i=1}^{3} b_i(x)M_i(x)k_i(x)) \mod P(x) = ((18x + 17)(x^2 + 2x^3 + x^3 + x + 1)(x^2 + 2x + 3) + (x^4 + x^3 + 2x^2 + x + 1)(18x + 17)(x^2 + x + 1)(18x + 17)(x^2 + x + 1)(18x + 17)(x^2 + x + 1)$.
\[ x^2 + 1 \] + \((x^2 + x^4 + 2x + 2x^2 + 2x + 1)(18x + 17)(x^2 + 3x + 2)\) \mod \((x^2 + x^6 + 3x^5 + 4x^4 + 3x^3 + 2x + 1) = 103 \]
\[ x^2 + 12x^2 + 116x^3 + 42x^4 + 43x^2 - 7x - 39. \]

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

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^3 + 2x^2 + x^2 + x + 1)(x^2 + 2x + 3) + (x^4 + x^3 + 2x^2 + x + 1)(13x + 3)(x^2 + x^2 + 2x + 1)(x^2 + x^2 + 2x + 1) \mod \((13x + 3)(x^2 + 3x + 2) + 3x^2 + 4x^2 + 3x^2 + 2x + 1) = 28x^6 - 47x^5 - 9x^4 - 81x^3 - 71x^2 - 70x - 56.\)

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

The concatenation of the coefficients of the remainders \(b_i(x)\) corresponds to the input text PSMSFRSDN = 15181205171817303. According to the agreements between the participants, the ciphertext can be either the parameter \(N_4(x)\), or the remainders \(b_i(x)\), 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]:

\[ S_p(n) = \frac{1}{n} \sum_{d|n} \mu(d)p^{n/d} \]  \hspace{1cm} (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, \(\mu(d) = 0\) if \(d\) is a divisor of degree \(n\) with an odd number of prime factors, and \(\mu(d) = -1\) if \(d\) is a divisor of degree \(n\) with an odd number of prime factors, and \(\mu(d) = 0\) if \(d\) contains a square of a prime factor. Accordingly, the number of modules \(p\) cannot exceed \(S_p(n)\). Table 1 shows the Möbius functions for the first 64 positive integers [29].

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

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: \(S_p(n) = (1/3)\sum_{d|n}\mu(d)p^{n/d} = (1/3)(\mu(1)p^{3/1} + \mu(3)p^{3/3}) = (p^3 - p)/3\).

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: \(S_2(n) = (1/32)\sum_{d|n}\mu(d)p^{32/d} = (1/32)(\mu(1)p^{32/1} + \mu(2)p^{32/2} + \mu(4)p^{32/4} + \mu(8)p^{32/8} + \mu(16)p^{32/16} + \mu(32)p^{32/32}) = (1/32)(2^{32} - 2^{16}) = 2^{16} - 2^{16} = 134215680\).

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

\[ S_3(32) = \frac{1}{32} \sum_{d|32}\mu(d)3^{32/d} = (\mu(1)3^{32/1} + \mu(2)3^{32/2} + \mu(4)3^{32/4} + \mu(8)3^{32/8} + \mu(16)3^{32/16} + \mu(32)3^{32/32}) = \frac{1}{32}(3^{32} - 3^{16}) = 57906879556410\]
\[ S_{32}(32) = \frac{1}{32} \sum_{d \mid 32} \mu(d)32^d = \left( \mu(1)32^{1/2} + \mu(2)32^{1/2} + \mu(4)32^{1/2} + \mu(8)32^{1/8} + \mu(16)32^{16/16} + \mu(32)32^{32/32} \right) \]
\[ = \frac{1}{32} \left( 32^{1/2} - 5^{16} \right) = 72759576143574 \times 10^6 \]

\[ S_{7}(32) = \frac{1}{32} \sum_{d \mid 32} \mu(d)7^{32/d} = \left( \mu(1)7^{32/1} + \mu(2)7^{32/2} + \mu(4)7^{32/4} + \mu(8)7^{32/8} + \mu(16)7^{32/16} + \mu(32)7^{32/32} \right) \]
\[ = \frac{1}{32} \left( 7^{32/2} - 7^{16} \right) = 34513364820121 \times 10^{11} \]

\[ S_{11}(32) = \frac{1}{32} \sum_{d \mid 32} \mu(d)11^{32/d} = \left( \mu(1)11^{32/1} + \mu(2)11^{32/2} + \mu(4)11^{32/4} + \mu(8)11^{32/8} + \mu(16)11^{32/16} + \mu(32)11^{32/32} \right) \]
\[ = \frac{1}{32} \left( 11^{32/2} - 11^{16} \right) = 65980552392226 \times 10^{17} \]

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

\[ S_{17}(32) = \frac{1}{32} \sum_{d \mid 32} \mu(d)17^{32/d} = \left( \mu(1)17^{32/1} + \mu(2)17^{32/2} + \mu(4)17^{32/4} + \mu(8)17^{32/8} + \mu(16)17^{32/16} + \mu(32)17^{32/32} \right) \]
\[ = \frac{1}{32} \left( 17^{32/2} - 17^{16} \right) = 73997237362646 \times 10^{23} \]

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

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

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

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

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

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

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: \( 30 \leq S_p(8) \leq 1.46^{12}, \) \( 4048 \leq S_p(16) \leq 8.54 \times 10^4, \) \( 1.34 \times 10^8 \leq S_p(32) \leq 5.832 \times 10^{19}, \) \( 2.8823 \times 10^{17} \leq S_p(64) \leq 5.4425 \times 10^{104}, \) \( 8.25293 \times 10^{46} \leq S_p(96) \leq 6.7717 \times 10^{154}, \) and \( 2.65846 \times 10^{46} \leq S_p(192) \leq 9.4788 \times 10^{46}. \) 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, \( S_{13}(8) = S_{16}(16), S_{13}(16) = S_{16}(32), S_{13}(32) = S_{16}(64), S_{13}(64) = S_{16}(96), \) and \( S_{13}(96) = S_{16}(128). \)

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
performing calculations with each one according to the following formula:

\[ O(n, l) = C_{S_p(n)}^l \cdot n^2 \log l = \frac{(S_p(n))! \cdot n^2 \log l}{(S_p(n) - l)! \cdot l!} \]  

(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: \( O(32, 5) = C_{57906879556410}^5 \cdot 32^2 \log 5 = \frac{(57906879556410)! \cdot 5!}{(57906879556405)! \cdot 5!} \cdot 32^2 \log 5 \approx 1.29 \times 10^{70} \).

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:

- \( S_3(4) \), \( l = 7 \), and \( n = 4 \);
- \( S_1(8) \), \( l = 5 \), and \( n = 8 \);
- \( S_3(16) \), \( l = 6 \), and \( n = 16 \);
- \( S_2(32) \), \( l = 3 \), and \( n = 32 \);
- \( S_2(64) \), \( l = 2 \), and \( n = 64 \).

Table 3 shows that adding one module for parameters \( n = 4 \) and \( S_3(4) \) increases the strength by about 5 orders of magnitude, for \( n = 8 \) and \( S_1(8) \) by 7 orders of magnitude, for \( n = 16 \) and \( S_3(16) \) by 6 orders of magnitude, for \( n = 32 \)
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 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).

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. 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 parameter values:

- $S_2(4)$, $l = 2$, and $n = 4$;
- $S_2(8)$, $l = 6$, and $n = 16$;
- $S_2(32)$, $l = 3$, and $n = 32$;
- $S_2(64)$, $l = 2$, 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), 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


