Symmetric Cryptosystem Based on Ring Images

Serhii Kryvyi[1,†], Kyrylo Riabov[1,†]

[1]Taras Shevchenko National University of Kyiv, Academician Glushkov Avenue 4d, Kyiv, 03680, Ukraine

Abstracts

Algorithms for the exchange of information between subscribers are proposed, based on surjective mappings of finite associative-commutative rings with unity and systems of linear equations over such rings. The paper presents algorithms for constructing finite rings, generating surjective mappings of these rings, as well as the information exchange protocol and computational features of the protocol's implementation. The main motivation for the development of such a cryptosystem is that almost all established cryptosystems require computations involving either large prime numbers or the construction of finite fields of large order. These constructions also necessitate the use of rather complex algorithms. In contrast, the proposed system does not require complex calculations, nor the construction of operation tables for rings. Its security relies on the combinatorial complexity of the set of surjective mappings and isomorphisms between finite rings of relatively small order. The algorithms for solving systems of linear equations, which are integral to the information exchange protocols over such rings, exhibit polynomial-time complexity. The operation of the cryptosystem is demonstrated through examples.

Keywords

Cryptography, Symmetric Cryptosystem, Finite Rings, Surjective Mappings, Systems of Linear Equations, Ring Isomorphism

1. Introduction

In cryptographic applications, finite fields and Diophantine equations, as well as systems of such equations, are frequently employed [1, 2]. This is primarily due to the fact that a finite field possesses a cyclic multiplicative group, which enables the efficient use of the discrete logarithm function, while algorithms for solving Diophantine equations and systems of such equations over the set of natural numbers exhibit high computational complexity [3]. Cryptosystems constructed on the basis of these structures require the generation of large prime numbers, the construction of finite fields of high order, or significant memory resources and computational time for preparatory operations [4].

The motivation for this work is to develop a cryptosystem based on objects of relatively small size that still provides the necessary level of cryptographic strength. A system of this type was proposed in [5], and the present work represents a further development of that approach. The foundation of the cryptosystem is the use of surjective mappings of finite rings and their isomorphisms, utilizing systems of linear equations over residue rings. The security of such a system is based on the combinatorial complexity of the set of mappings between rings of relatively small order.

Workshop "Intelligent information technologies" UkrProg-IIT`2025 co-located with 15th International Scientific and Practical Programming Conference UkrPROG'2025, , May 13-14, 2025, Kyiv, Ukraine

† These authors contributed equally.

sl.krivoi@gmail.com (S.Kryvyi); kyryl.ryabov@gmail.com (K.Riabov)

0000-0003-4231-0691 (S.Kryvyi); 0009-0003-4118-8492 (K.Riabov)

2. Necessary definitions and concepts

Let Z_k denote the finite residue ring modulo k, that is, Z_k – is an associative-commutative ring (AC-ring) with unity. Elements $a, b \in Z_k \setminus \{0\}$ are called additive inverses if $a + b \equiv 0 \pmod{k}$, and are called zero divisors if $a \cdot b \equiv 0 \pmod{k}$. Since the ring Z_k has a multiplicative identity, elements $c, d \in Z_k$ such, that $c \cdot d \equiv 1 \pmod{k}$ are called units. The set of units in Z_k forms an abelian group [6].

Let G_k denote a finite AC-ring with unity, isomorphic to the ring Z_k, constructed according to a given defining sequence for addition with unity. This sequence is referred to as the defining sequence and based on the laws satisfied by the addition and multiplication operations of the ring, the operation tables for G_k are constructed (algorithms for constructing the operation tables of G_k can be found in [5]). This sequence also specifies the isomorphism between the rings Z_k and G_k, which allows one to avoid constructing the operation tables for G_k, since operations can be performed in Z_k and, via the isomorphism, the results can be mapped to G_k, where operations in Z_k are more efficient.

In the general case, the defining sequence of the ring G_k $a = (1, a_1, a_2, \dots, a_{k-2}, 0)$ is specified by the mapping $f(0) = 0 + 1 = 1$, $f(1) = 1 + 1 = a_1$, $f(a_i) = a_i + 1 = a_i + 1$, $f(a_{k-2}) = a_{k-2} + 1 = a_{k-1} = 0$, where $i = 0, 1, \dots, k - 1$.

The defining sequence of the ring G_k is generated by the following algorithm.

GEN-G(a, c, l, k)
Input: Order k and coefficients of the expression $f(i) = a \cdot i + c$, where $k = lm$, $\gcd(a, m) = \gcd(a, k) = 1$.
Output: The addition table row with unity as a one-dimensional array $b = (b_1, b_2, \dots, b_k)$ of length k.
Method:

1. for $i = 0$ to $k - 1$ do $b[i + 1] := a \cdot i + c \pmod{k}$ od
2. According to common rules, transform the array bb and fix its values (creation of the common defining sequence).
3. for $i = 1$ to k do
   if $(b_i = 0 \wedge i \neq k)$ then change b_i and b_k;
   if $(b_i = 1 \wedge i \neq 1)$ then change b_i and b_1;
   od
   (* This defines the isomorphism $g(i) = b_i$, where $i = 1, 2, \dots, k$ *)
4. Using the array $b = (b_1, b_2, \dots, b_k)$ construct the array $P[1 \times k]$ (from which, if necessary, the operation tables of the ring can be constructed):
   $P[0] := b_1$;
   for $i = 1$ to $k - 2$ do $P[b_i] := b_{i+1}$ od
   $P[b_{k-1}] := 0$.

The correctness of the algorithm follows from the fact that if $\gcd(a, k)=1$ and i runs through a complete residue system, then $a \cdot i + c$ also runs through a complete residue system [6].

The time complexity of the GEN-G algorithm is $O(k \log^2 k)$, since integer multiplication has complexity $O(\log^2 k)$, and there are at most k such multiplications.

It should be noted that operator 1) of the GEN-G algorithm can generate no more than $(k - 2)\varphi(k)$ initial sequences, where φ – is Euler's totient function. For cryptographic applications, this number is insufficient. Therefore, by agreement between the parties, the initial sequence generated by the algorithm is transformed in the same way by operator 2), which defines the cryptosystem as symmetric.

Example1. Generate the defining sequence for $k = 6$ and $f(i) = i + 4$.
The first loop of the algorithm (operator 1) generates the following initial sequence:

1. $b_1 = 4$; $b_2 = 5$; $b_3 = 0$; $b_4 = 1$; $b_5 = 2$; $b_6 = 3$.
2. The second operator performs a transformation: it swaps pairs of adjacent elements and performs a single cyclic permutation of all elements. The resulting sequence is 2, 5, 4, 1, 0, 3.
3. The second loop (operator 3) places 0 and 1 in their correct positions and produces the defining sequence and isomorphism:
   $g(i) = b_i$, $i = 1, 2, \dots, 6$, where $b_1 = 1$; $b_2 = 5$; $b_3 = 4$; $b_4 = 2$; $b_5 = 3$; $b_6 = 0$.

4. The third loop (operator 4) generates, from the array $b_1 = 1$, $b_2 = 5$, $b_3 = 3$, $b_4 = 0$, $b_5 = 2$, $b_6 = 4$ the sequence $P = (1, 5, 3, 0, 2, 4$ from which the operation tables of the ring G_6 are constructed. ♠[1]

Thus, the isomorphism between the rings G_k and Z_k is determined by the defining sequence generated by the GEN-G algorithm. Specifically, we have the following correspondence:

1	2	3	4	...	$k-1$	k
b_1	b_2	b_3	b_4	...	b_{k-1}	0

where the isomorphic mapping g is defined as: $g(k) = 0$, $g(1) = b_1 = 1$, $g(i) = b_i$, $i = 2, \dots, k-1$.

3. Message Exchange Protocol

The design of the cryptosystem is based on the following scheme:

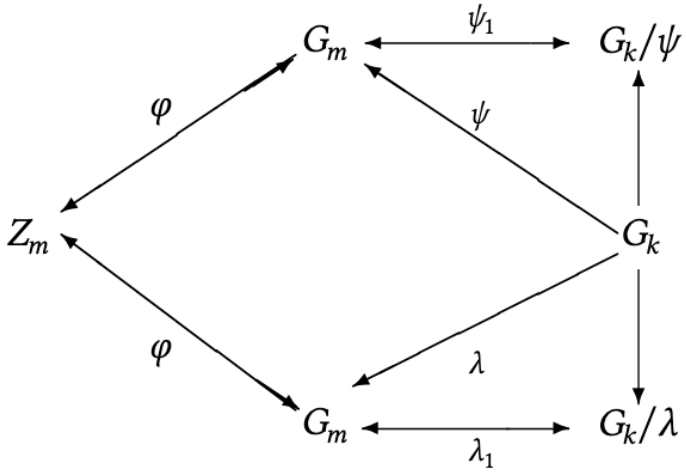

Figure. 1: System architecture

In this scheme, the mappings (see Fig. 1) are defined as follows:

- φ is an isomorphism between the rings Zm and Gm,
- ψ is a surjective mapping from the ring G_k onto the ring G_m,
- λ is a surjective mapping from the ringG_k onto the ring G_m,
- ψ_1 is a bijection between the factor set G_k/ψ and the ring G_m,
- λ_1 is a bijection between the factor set G_k/λ and the ring G_m.

The message exchange between Alice and Bob is performed according to the following protocol.

Initially, Alice and Bob exchange, via a secure channel, a quadruple (a, c, l, m), whose elements are parameters of the algorithm GEN-G(a, c, l, m). Using the expression $f(i) = a \cdot i + c$, where $\gcd(a, k) = \gcd(a, m) = 1$, they generate the initial sequences of the rings G_k and G_m and, by agreement, construct the defining sequences $b = (b_1 = 1, b_2, \dots, b_{m-1}, b_m = 0)$ and $c = (c_1 = 1, c_2, \dots, c_k = 0)$ for the rings G_m and G_k respectively, in the same manner.

After this, Alice and Bob proceed as follows:

Step 1. a) Alice constructs a system of expressions in the ring G_m:

[1] The symbols ♠ and ■ denote the end of the example and the end of the proof, respectively.

$$l(x) = Ax = \begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1q}x_q, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2q}x_q, \\ \quad \vdots \\ a_{p1}x_1 + a_{p2}x_q + \ldots + a_{pq}x_q. \end{cases}$$

b) She transforms $l(x)$ in the ring G_m as follows:

$$L(x) = Bx + a = B_r(B_{r-1}(\ldots B_2(B_1(l(x) + a) + a_1) \ldots + a_{r-1}) + a_r) + a_{r+1},$$

where B_i are non-singular matrices of dimension $p \times p$, a, a_j – are vectors of dimension $1 \times p$, $i = 1, 2, \ldots, r$, $j = 1, 2, \ldots, r + 1$. The result of this transformation is a system

$$L(x) = Bx + a = \begin{cases} b_{11}x_1 + b_{12}x_2 + \ldots + b_{1q}x_q + a_1, \\ b_{21}x_1 + b_{22}x_2 + \ldots + b_{2q}x_q + a_2, \\ \quad \vdots \\ b_{p1}x_1 + b_{p2}x_q + \ldots + b_{pq}x_q + a_q. \end{cases}$$

c) She replaces the coefficients in $l(x)$ and $L(x)$ with their counterparts from the factor set G_k/λ_1:

$$\bar{l}(x) = \bar{A}x = \begin{cases} b_{11}x_1 + b_{12}x_2 + \ldots + b_{1q}x_q, \\ b_{21}x_1 + b_{22}x_2 + \ldots + b_{2q}x_q, \\ \quad \vdots \\ b_{p1}x_1 + b_{p2}x_q + \ldots + b_{pq}x_q. \end{cases}$$

and

$$\bar{L}(x) = \bar{B}x + b = \begin{cases} c_{11}x_1 + c_{12}x_2 + \ldots + c_{1q}x_q + b_1, \\ c_{21}x_1 + c_{22}x_2 + \ldots + c_{2q}x_q + b_2, \\ \quad \vdots \\ c_{p1}x_1 + c_{p2}x_q + \ldots + c_{pq}x_q + b_q. \end{cases}$$

Alice then transmits the expressions $\bar{l}(x)$ and $\bar{L}(x)$ via a public channel or publishes them on a website.

Step 2.

a) Bob, using the expressions $\bar{l}(x)$ and $\bar{L}(x)$ and the mappings ${\lambda_1}^{-1}$ and φ^{-1}, finds the expressions $\hat{l}(x)$ and $\hat{L}(x)$ in the ring Z_m, and selects an arbitrary vector $\bar{a}$ of dimension $1 \times q$.

b) Bob wishes to send Alice a message v. To do this, he solves the system $\hat{l}(x) = v$, finds the solution $\bar{x}$ and computes the vectors $\hat{l}(\bar{a}) = d$ and $\hat{L}(\bar{x} + \bar{a}) = d_1$ in the ring Z_m.

c) Bob keeps the vector v secret, replaces the values d and d_1 with their counterparts from one of the factors sets G_k/ψ or G_k/λ and sends Alice the pair of vectors $(\bar{d}, \overline{d_1})$ via a public channel.

Step 3.

a) Alice computes the inverse matrices to the matrices B_i in the ring G_m (these computations are performed in the ring Z_m via the isomorphism φ).

b) She recovers the value v, since she possesses all the necessary data.

Proposition 1. *The message exchange according to the protocol is performed correctly.*

Proof. This follows directly from the properties of linear operators, the bijection λ_1, and the isomorphism φ. Indeed, let us denote the product of matrices $BrBr{-}1 \ldots B1 = D$, then

$$d_1 = L((\bar{x} + \bar{a}) + a_1) = D(l(\bar{x} + \bar{a}) + a_1) + b + a_{r+1} = D(l(\bar{x} + \bar{a}) + a_1) + c,$$

where $c = b + a_{r+1}$, and b is the vector of values obtained by multiplying the matrices $B_1, B_2, \dots, B_r$ by the vectors $a_1, a_2, \dots, a_r$. Then

$$D^{-1}(D(d_1 - a_{r+1})) - D^{-1}b = D^{-1}(D(l(\bar{x} + \bar{a}) + a_1) + b) - D^{-1}b = l(\bar{x} + \bar{a}) + a_1.$$

Thus, $l(\bar{x} + \bar{a}) + a_1 - [a_1 + d] = l(\bar{x})$. ■

From Figure 1, it follows that there are at least three paths for ciphertext generation in the system:

1) $G_k/\psi \longrightarrow G_m \longrightarrow Z_m \longrightarrow G_m$. Here, the expressions $l(x)$ and $L(x)$ are explicitly represented in the factor set G_k/ψ, which, via the bijections φ and ψ_1, are mapped to the ring Z_m, where computations are performed and the ciphertext is constructed in the ring G_m.

2) $G_k/\lambda \longrightarrow G_m \longrightarrow Z_m \longrightarrow G_k/\psi$. Here, the expressions $l(x)$ and $L(x)$ are explicitly represented in the factor set G_k/λ, and via the bijections λ_1 and φ, these expressions are mapped to the ring Z_m, where computations are performed and the ciphertext is constructed via the bijections φ and ψ_1 in the factor set G_k/ψ. This path corresponds to the protocol described above.

3) $G_m/\psi \longrightarrow G_m \longrightarrow Z_m \longrightarrow G_k/\lambda$. Here, the expressions $l(x)$ and $L(x)$ are explicitly represented in the factor set G_k/ψ, and via the bijections ψ_1 and φ, these expressions are mapped to the ring Z_m, where computations are performed and the ciphertext is constructed in the factor set G_k/λ.

Example 2. Consider the preparatory steps of the protocol.

Suppose Alice and Bob choose the first path for ciphertext generation, exchange the tuple (7, 5, 2, 25), and fix the following defining sequence for the ring G_{25} (after executing operators 1) and 2)):

$$b = (1, 6, 8, 10, 2, 4, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 12, 14, 16, 18, 20, 24, 22, 23, 0).$$

Operator 3) of the GEN-G(7, 5, 2, 25) algorithm defines the isomorphic mapping $\varphi : Z_{25} \longrightarrow G_{25}$, which in this case is:

$$\begin{array}{lllll}
\varphi(0) = 0, & \varphi(5) = 2, & \varphi(10) = 9, & \varphi(15) = 19, & \varphi(20) = 18, \\
\varphi(1) = 1, & \varphi(6) = 4, & \varphi(11) = 11, & \varphi(16) = 21, & \varphi(21) = 20, \\
\varphi(2) = 6, & \varphi(7) = 3, & \varphi(12) = 13, & \varphi(17) = 12, & \varphi(22) = 24, \\
\varphi(3) = 8, & \varphi(8) = 5, & \varphi(13) = 15, & \varphi(18) = 14, & \varphi(23) = 22, \\
\varphi(4) = 10, & \varphi(9) = 7, & \varphi(14) = 17, & \varphi(19) = 16, & \varphi(24) = 23,
\end{array}$$

where $\varphi(25) = \varphi(0) = 0$, $\varphi(1) = 1$, $\varphi(2) = \varphi(1 + 1) = 6$, $\varphi(3) = 6 + 1 = 8$, $\varphi(4) = 8 + 1 = 10, \dots, \varphi(24) = 23$.

Using this isomorphism, operator 4) of the GEN-G algorithm constructs the array $P[1 \times 25]$ (for convenience, it is presented as a substitution row).

$$P = \begin{pmatrix}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 \\
20 & 21 & 22 & 23 & 24 & & & & & \\
1 & 6 & 4 & 5 & 3 & 7 & 8 & 9 & 10 & 11 \\
2 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\
24 & 12 & 23 & 0 & 22 & & & & &
\end{pmatrix}.$$

Let the letters of the English alphabet be naturally enumerated, and the defining sequence of the ring G_{50}, generated by the GEN-G algorithm is as follows

1,5,49,7,10,17,2,34,11,20,39,33,48,3,45,4,37,6,41,13,43,15,36,8,38,9,
35,12,40,14,44,19,46,16,47,21,31,24,27,42,29,22,32,23,30,25,28,18,26,0.

Table 1 Numeric equivalents of alphabet symbols

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

a	b	c	d	e	f	g	h	i/j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

By defining the bijection ψ_1 from the ring G_{50}[2], to the ring G_{25} as

$$\psi_1 = \begin{cases} \psi(0) = 7 = m_0, & i = 0 \\ \psi(i \pmod{25}) = m_{i-1} + 1, & i \geq 1, \end{cases}$$

for i = 1, 2, … , 50, we obtain the ordinal number j of the class of the element m_i.

Table 2 Mapping of classes G_{50} to G_{25}

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
7	10	17	2	34	11	20	39	33	48	3	45	4	37	6	41	13	43	15	36	8	38	9	35	12
40	14	44	19	46	16	47	21	31	24	27	42	29	22	32	23	30	25	28	18	26	0	1	5	49
a	b	c	d	e	f	g	h	i/j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

This concludes the preparatory steps. ♠

3.1. Cryptanalysis of the Protocol

Let us consider possible cryptanalysis scenarios for the protocol. The cryptanalyst has access to the following data:

a) The system $\overline{L}(x)$, from which the block length of the message can be determined by the number of congruences in the system;
b) The ciphertext length, which can be determined by the number of unknowns in the congruences;
c) Possibly, the orders of the rings G_m and G_k.

Unknowns include the isomorphism φ, the bijections ψ_1, λ_1 and the surjections ψ and λ.

Suppose in case (a), the cryptanalyst has no further information. Then, the only feasible method to recover the plaintext is exhaustive search. The complexity of such a search is determined by the number of possible ways to encrypt a message, which is composed of:

1. The number of possible isomorphisms (bijections) φ – $O((m - 2)!)$, where $m$is the order of the ring G_m,
2. The number of bijections ψ_1 and λ_1: $O(m!)$ each,
3. The number of surjections ψ and λ: $O\left(\frac{m!(l!)^k}{m}\right)$, where $k = lm$.

The total complexity, even for such a simple cryptosystem as in the example above, is

$$23! \cdot 25! \cdot \frac{50!}{25!2^{25}} > \frac{23!50!}{2^{25}} > 2^{94} > 10^{31}.$$

If we assume that one combination is generated in $10^{-}14$ seconds, then to enumerate all combinations would require

$$10^{31} \cdot 10^{-14} = 10^{17}$$

seconds, which is more than 10^7 years.

Clearly, if the orders k and m are chosen to be larger, the brute-force method becomes infeasible.

b) Suppose in case (b), the cryptanalyst has access to several encrypted messages, i.e., the texts

[2] As G_{50}, you can take any set of power $25 \cdot l$, and Alice and Bob must equally order the elements of this set and construct a bijection λ_1.

$$\overline{m}_1 = \varphi(\hat{d}_{11}, \hat{d}_{12}), \quad \overline{m}_2 = \varphi(\hat{d}_{21}, \hat{d}_{22}), \quad \overline{m}_3 = \varphi(\hat{d}_{31}, \hat{d}_{32}),$$

Since there are $(m-2)!$ bijections of type φ and the vectors $m_1, m_2, \ldots$ belong to different sets, this information requires knowledge of the bijection φ, i.e., the defining sequence of G_m. However, these objects are not available to the cryptanalyst, and searching for them by brute force requires generating $(m-2)!$ combinations. Moreover, for these combinations, one must also find the symbolic equivalents (which is also $m!$ combinations), so this information does not allow the plaintext to be found in a reasonable time.

(c) Suppose in case (c), the cryptanalyst has access to both encrypted and decrypted messages, i.e., the data

$$m_1, m_2, m_3, \ldots \text{ i } m_1 = \xi^{-1}(\varphi^{-1}(\overline{m}_1)), m_2 = \xi^{-1}(\varphi^{-1}(\overline{m}_2)), m_3 = \xi^{-1}(\varphi^{-1}(\overline{m}_3)), \ldots$$

where ξ^{-1} maps the numeric text to the symbolic text.

Since the mappings ξ, φ, and the system $\hat{l}(x)$ are unknown to the cryptanalyst, it is not possible to recover the defining sequence from this data.

From the above example, it is evident that, in computational terms, the most complex step is the construction of inverse matrices in the ring G_m. To simplify these computations, it is preferable to use the isomorphism $\varphi : G_m \longrightarrow Z_m$ and perform calculations in the residue ring Z_m. Once the inverse matrices are found, the reverse substitutions can be performed to obtain the corresponding matrices in the ring G_m.

It is known that the multiplicative group of units of the ring G_k is an abelian group [6]. In order to apply the discrete logarithm function in this group, it must be cyclic, i.e., possess a generator. Thus, the question arises: under what conditions is the group of units of the ring G_k cyclic? The answer is provided by

Theorem 1. *The multiplicative group of the ring Z_k is cyclic if and only if k is equal to 2, 4, p^m or $2p^m$, where $m \geq 1$ and p is an odd prime [6].*

4. Message Formation

From the above, it follows that Alice and Bob must exchange, via a secure channel, the tuple (a, c, l, k), where a, c, l, k are the parameters of the GEN-G(a, c, l, k) algorithm. If the order of the ring k is chosen to be a multiple of the order of the ring Z_m, i.e., $k = l \cdot m$, then, based on the coprimality of k, m, and a, the methods for constructing the rings will be known, and the transformations for constructing the defining sequences of the rings can be taken identically for Gk and Gm.

Furthermore, from the protocol and the example provided, it follows that, in order to transmit the desired message $b = (b_1, b_2, \ldots, b_p)$ it is necessary that the system of equations

$$l(x) = Ax = \begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1q}x_q \equiv b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2q}x_q \equiv b_2, \\ \quad \vdots \\ a_{p1}x_1 + a_{p2}x_q + \ldots + a_{pq}x_q \equiv b_p \end{cases} \pmod{m} \qquad (1)$$

has a solution for arbitrary values of $b_1, b_2, \ldots, b_p$. The isomorphism between the rings G_m and Z_m allows us to consider only the residue ring Z_m. The compatibility criterion for a system of linear congruences $Ax \equiv b \pmod{m}$ of size $p \times q$, $(p < q)$ over the ring Z_m requires the existence of a solution to the congruence

$$d_1y_1 + d_2y_2 + \ldots + d_sy_s \equiv 1 \pmod{m},$$

where $d_1, d_2, \ldots, d_s$ are the values of the last coordinates in the solutions of the homogeneous system $Ax - bx_0 = 0$ [7]. This condition is satisfied for any b if the equations of the system are linearly

independent and the determinant of the subsystem matrix $A_1u \equiv b \pmod m$ of size $p \times p$, formed by the linearly independent columns $b_{i1}, b_{i2}, \dots, b_{ip}$ of the system $Ax \equiv b \pmod m$, is coprime with the modulus m. Then, for the subsystem matrix, there exists an inverse matrix, i.e., from $A_1u = b \pmod m$ it follows that ${A^{-1}}_1 A_1 u = u \equiv {A^{-1}}_1 b \pmod m$ for any b. A vector $a = (a_1, a_2, \dots, a_q)$, whose coordinate indices $i_1, i_2, \dots, i_p$ coincide with those of the vector $u \equiv {A^{-1}}_1 b \pmod m$, and the remaining coordinates are zero, will be a solution to the system.

Thus, Alice needs to construct a system of linear equations in which the equations are linearly independent and contain a subsystem whose matrix determinant is coprime with the modulus mm. To verify the linear independence of the expressions, Alice must solve the system $ATy \equiv 0 \pmod m$ and ensure that this system has only the trivial solution. She then constructs a subsystem with the described determinant properties.

Example 3. Let the letters of the English alphabet be naturally enumerated (see Table 1).

Step 1.

a) Suppose Alice constructs the following expressions in the ring G_{25} (expressions with negative coefficients are shown in parentheses, where negative coefficients are replaced by their additive inverses):

$$l(x) = \begin{cases} 2x_1 - 16x_2 + 7x_3 + 20x_4, \\ 0x_1 + 1x_2 - 17x_3 - 11x_4 \end{cases} \qquad \left(\begin{cases} 2x_1 + 4x_2 + 7x_3 + 20x_4, \\ 0x_1 + 1x_2 + 11x_3 + 17x_4 \end{cases} \right)$$

and transforms them into the form

$$L(x) = B_1(l(x) + (1,2)^t) = \begin{cases} 9x_1 - 13x_2 + 10x_3 - 16x_4 + 3, \\ 18x_1 + 14x_2 - 18x_3 + 19x_4 + 16, \end{cases} \quad \left(\begin{cases} 9x_1 + 15x_2 + 10x_3 + 4x_4 + 3, \\ 18x_1 + 14x_2 + 2x_3 + 19x_4 + 16, \end{cases} \right)$$

where the matrix

$$B_1 = \begin{pmatrix} 6 & 1 \\ 23 & 23 \end{pmatrix} \quad \text{and its counterpart in the ring } Z_{25} \quad \bar{B}_1 = \begin{pmatrix} 2 & 1 \\ -1 & -1 \end{pmatrix}.$$

b) Alice replaces the coefficients in the constructed expressions $l(x)$, $L(x)$ and the matrix B_1 with their counterparts from the factor set G_{50}/ψ and obtaining the expressions

$$\bar{l}(x) = \begin{cases} 17x_1 + 34x_2 + 21x_3 + 26x_4, \\ 7x_1 + 14x_2 + 42x_3 + 43x_4, \end{cases}$$

$$\bar{L}(x) = B_1(l(x) + (1,2)^t) = \begin{cases} 48x_1 + 41x_2 + 3x_3 + 46x_4 + 19, \\ 15x_1 + 32x_2 + 44x_3 + 36x_4 + 30, \end{cases}$$

which she sends to Bob via a public channel or publishes on her website.

Step 2.

a) Bob, using the bijections φ and ψ_1 finds the corresponding expressions $\hat{l}(x)$ and $\hat{L}(x)$ in the ring Z_{25}:

$$\hat{l}(x) = \begin{cases} 5x_1 + 6x_2 + 9x_3 + 21x_4, \\ 0x_1 + 1x_2 + 11x_3 + 14x_4. \end{cases}$$

$$\hat{L}(x) = \begin{cases} 10x_1 + 13x_2 + 4x_3 + 6x_4 + 7, \\ 20x_1 + 18x_2 + 5x_3 + 15x_4 + 19. \end{cases}$$

It is easy to verify that in the system of expressions $\hat{l}(x)$, the second and third columns form a subsystem whose determinant is 7, and 7 is coprime with the modulus 25 in the ring Z_{25} (the compatibility conditions for the system $\hat{l}(x)$ are satisfied).

Bob wishes to send Alice the message

tara tara tarara.

b) Bob divides the message into blocks of two symbols per block (spaces between message symbols, corresponding to element 49, are omitted for simplicity in this example), and replaces them with their numeric equivalents from Table 1:

ta	ra	ta	ra	ta	ra	ra
18,0	16,0	18,0	16,0	18,0	16,0	16,0

.

c) He solves the system of equations in the ring Z_{25}

$$\hat{l}(x) = \begin{cases} 5x_1 + 6x_2 + 9x_3 + 21x_4 = 18, \\ 0x_1 + 1x_2 + 11x_3 + 14x_4 = 0. \end{cases} \quad (\text{mod } 25)$$

and finds the solution $\bar{x} = (0, 14, 1, 0)$.

d) The value $v_1 = (18, 0)$ is kept secret. He selects the vector $\bar{a} = (0, 1, 0, 1)$ and computes $d = l(\bar{a}) = (2, 15)$. He adds the vector $\bar{a} = (0, 1, 0, 1)$ to the solutions $\bar{x} = (0, 14, 1, 0)$, obtaining $\bar{x} + \bar{a} = (0, 15, 1, 1)$ and substitutes this sum into $L(x)$, thus finding $d_1 = (12, 9)$. Bob sends Alice the counterparts of the values d and d_1 in the ring G_m.

Alice, using the received counterparts, finds the values d and d_1, and performs the following computations:

a) She computes the inverse matrix to $\overline{B}_1^{-1}$ in the ring Z_{25}:

$$B_1^{-1} = \begin{pmatrix} 1 & 1 \\ -1 & -2 \end{pmatrix}.$$

b) She computes $\overline{B}_1^{-1}(d_1{}^t - (7, 19)^t) = \overline{B}_1^{-1}((12, 9) - (7, 19))^t = \overline{B}_1^{-1}(5, 15)^t$ and finds

$$\overline{B}_1^{-1}(5, 15)^t - (2, 15)^t = (20, 15) - (2, 15) = (18, 0) = v_1.$$

a) Bob solves the system of equations

$$\hat{l}(x) = \begin{cases} 5x_1 + 6x_2 + 9x_3 + 21x_4 = 16, \\ 0x_1 + 1x_2 + 11x_3 + 14x_4 = 0. \end{cases} \quad (\text{mod } 25)$$

and finds the solution $\bar{x} = (0, 18, 12, 0)$.

b) The value $v_2 = (16, 0)$ is kept secret. He selects the vector $\bar{a} = (1, 0, 1, 0)$ and computes $d = l(\bar{a}) = (14, 11)$. He adds the vector $\bar{a} = (1, 0, 1, 0)$ to the solution $\bar{x} = (0, 18, 12, 0)$, obtaining $\bar{x} + \bar{a} = (1, 18, 13, 0)$, and substitutes this sum into $L(x)$, thus finding $d_1 = (3, 3)$.

c) Bob sends Alice the counterparts of the values d i d_1 in the ring G_m.

Step 3. Alice, using the received counterparts, finds the values d, d_1 and performs the following computations:

a) She finds the inverse matrix to $\overline{B}_1^{-1}$ in the ring Z_{25}:

b) She computes $\overline{B}_1^{-1}(d_1{}^t - (7, 19)^t) = B_1^{-1}(21, 9)^t$ and finds

$$\overline{B}_1^{-1}(21, 9)^t - (14, 11)^t = (5, 11) + (11, 14) = (16, 0) = v_2.$$

a) Bob solves the system of equations in the ring Z_{25}

$$l(x) = \begin{cases} 5x_1 + 6x_2 + 9x_3 + 21x_4 = 18, \\ 0x_1 + 1x_2 + 11x_3 + 14x_4 = 0. \end{cases} \quad (\text{mod } 25)$$

and finds the solution $\bar{x} = (0, 14, 1, 0)$. Since the next block is the same as the first, i.e., $v_3 = (18, 0)$. Bob selects a new vector $\bar{a} = (0, 0, 1, 1)$, for which he computes

$$d = l(\bar{a}) = (5, 0), \quad \bar{x} + \bar{a} = (0, 14, 2, 1), \quad d_1 = L(\bar{x} + \bar{a}) = (3, 21).$$

and sends Alice the counterparts $d = (5, 0)$, $d_1 = (3, 21)$ in the ring G_{25}.

Alice computes

$$B_1^{-1}(d_1{}^t - (7, 19)^t) = B_1^{-1}(21, 2)^t = (23, 0) = l(\bar{x} + \bar{a}).$$

From which she finds

$$(23, 0) - (5, 0) = (18, 0) = v_3.$$

Since the next block is the same as the second, i.e $v_4 = (16, 0)$, Bob selects a new vector $\bar{a} = (0, 0, 0, 1)$, for which he computes

$$d = l(\bar{a}) = (21, 14), \quad \bar{x} + \bar{a} = (0, 18, 12, 1), \quad d_1 = L(\bar{x} + \bar{a}) = (20, 18).$$

and sends Alice the counterparts $d = (21, 14)$, $d_1 = (20, 18)$ in the ring G_{25}.

Alice computes

$$B_1^{-1}(d_1{}^t - (7, 19)^t) = B_1^{-1}(13, 24)^t = (12, 14) = l(\bar{x} + \bar{a}) .$$

From which she finds

$$(12, 14) - (21, 14) = (12, 14) + (4, 11) = (16, 0) = v_4.$$

Bob and Alice repeat this procedure as many times as there are blocks in the message (in this case, three more times). Thus,

Alice obtains the ciphertext

(2,15,12,9) (14,11,3,3) (5,0,3,21) (21,14,20,18) ... ...

After decryption, Alice recovers the message

18,0	16,0	18,0	16,0	18,0	16,0	16,0
ta	ra	ta	ra	ta	ra	ra

♠.

In the given example, the same ring was used throughout, but it is possible to change the ring for each transmission session or at certain intervals between transmissions. The vectors a and $\bar{a}$, which affect the values $l(\bar{a})$ and $L(\bar{x} + \bar{a})$, can also be varied

The presented protocol can be made more complex by using different rings or different parameter values—matrices and vectors—for each encrypted block. Furthermore, if the ciphertext is represented by its counterparts in the ring G_{50}

(44,23,4,48) (6,45,2,19) (11,40,19,0) (38,22,8,15) ... ...,

then the cryptanalyst has no access to the systems of expressions, the rings G_{25}, Z_{25} or the mappings λ, λ_1, ψ, ψ_1 i φ.

5. Computational Features

Given that computations in the ring G_m are not standard in practice, the efficiency of encryption and decryption can be improved by utilizing the isomorphism between the rings G_m and Z_m. Indeed, finding the additive inverse of an element a in the ring Z_m reduces to computing $m - a$, and finding the multiplicative inverse of aa is performed using the extended Euclidean algorithm to solve the equation $ax + my = 1$ (the extended Euclidean algorithm computes the decomposition $ax + by = d$, where d = gcd(a, b)). The result of this algorithm is the value $x = a^{-1}$.

An obvious drawback of the described protocol is that the ciphertext is twice as long as the plaintext message.

Declaration on Generative AI

During the preparation of this work, the authors used AI program Chat GPT 4.1 for correction of text grammar. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] W. Mao, Modern Cryptography, Pearson Education, Prentice Hall Professional Technical Reference, Upper Saddle River, New Jersey, 2004, 768 p.

[2] P. A. Kameswari, S. S. Sriniasarao, A. Belay, An application of linear Diophantine equations to cryptography, Advanced in Mathematics: Scientific Journal 10 (2021) 2799–2806.

[3] M. Hermann, L. Juban, P. G. Kolaitis, On the complexity of counting the Hilbert basis of a linear Diophantine system, in: Lecture Notes in Computer Science, vol. 1705, Springer Verlag, 1999, pp. 13–32.

[4] A. Berczes, H. Lajos, N. Hirete-Kohno, T. Kovacs, A key exchange protocol based on Diophantine equations and S-integers, JSIAM Letters (2014) 85–88.

[5] S. Kryvyi, V. Opanasenko, O. Grinenko, Yu. Nortman, Symmetric system for exchange information on the base of surjective isomorphism of rings, in: Proceedings of the 12th International IEEE Conference on Dependable Systems, Services and Technologies (DESSERT 2022), December 9–11, 2022, pp. 1–7.

[6] V. Shoup, An Computational Introduction to Number Theory and Algebra, Cambridge University Press, 2008, 580 p.

[7] S. L. Kryvyi, Linear Diophantine constraints and their application, Interservice, Kyiv, 2021, 257 p.