mathematics

MDPI

Article

Algorithms for Solving Systems of Boolean Equations Based on the Transformation of Logical Expressions

Anvar Kabulov [1,2,†], Alimdzhan Babadzhanov [3,†], Abdussattar Baizhumanov [4,†], Islambek Saymanov [1,5,*,†] and Akbarjon Babadjanov [6,†]

[1] School of Mathematics and Natural Sciences, New Uzbekistan University, Movarounnahr 1, Tashkent 100007, Uzbekistan
[2] Applied Mathematics and Intelligent Technologies Faculty, National University of Uzbekistan, Tashkent 100174, Uzbekistan
[3] Department of Algorithmization, Engineering Federation of Uzbekistan, Tashkent 100003, Uzbekistan
[4] Department of Mathematics, O. Zhanibekov South Kazakhstan State Pedagogical University, A. Baitursynov Street No. 13, Shymkent 160012, Kazakhstan
[5] Department of Applied Informatics, Kimyo International University in Tashkent, Shota Rustaveli Str. 156, Tashkent 100121, Uzbekistan
[6] Department of Information Systems, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA
* Correspondence: i.saymanov@newuu.uz or i.saymanov@kiut.uz
† These authors contributed equally to this work.

Abstract

This manuscript proves specific theorems for transforming Boolean expressions of logical formulas when moving from one basis to another, simplifying the solution of complex equations, especially for cryptographic applications. The paper develops methods for solving specific nonlinear systems of Boolean equations used in cryptographic *S*-boxes using transformations to simpler forms, such as disjunctive normal forms (DNFs) and Zhegalkin polynomials. The main contributions include a mathematical basis for transforming formulas, a complexity-reducing grouping method, and the RLSY program for practical implementation. A rigorous theory, cryptographic relevance, and a detailed description of the algorithm are proposed. The grouping method reduces the system complexity by a factor of 2^{11}, as shown in a test example, improving computational efficiency. A solution to a special class of systems of nonlinear Boolean equations of the second degree, which are a logical model of algebraic cryptanalysis, is also proposed. Test examples of logical formula transformations are given.

Keywords: Boolean equation; analytical transformation; logical formulas; algebraic-logical method; Zhegalkin polynomials; elementary conjunction

MSC: 94C11

Academic Editor: Iliya Bouyukliev

Received: 12 January 2026
Revised: 29 January 2026
Accepted: 5 February 2026
Published: 8 February 2026

1. Introduction

Algebraic-logical methods are traditionally applied in solving a wide range of problems in pattern recognition theory, neural structure theory, intelligence theory, and mathematical linguistics [1–5]. These methods serve as both a means of formalization and an effective tool for solving the problems that arise in these fields [2,6,7].

We will outline some of the methods used in logical operations when solving systems of logical equations and finding their solutions. As is well known, logical problems are

primarily presented in the form of analytical formulas with arbitrary bases [8,9]. It is evident that performing any operation on these formulas is difficult, even with the help of computers. Therefore, we now examine some of the main methods commonly used in logical operations within logical propositions and explore the criteria for transforming logical formulas when transitioning from one base to another [10–13].

The solution of systems of second-degree nonlinear Boolean equations, which serve as a logical model for algebraic cryptanalysis, is investigated. Test examples of logical formula transformations are constructed [14–18]. The solution of systems of second-degree nonlinear Boolean equations of a special class is considered. The advantage of the research results compared to others lies in the ability to solve systems of special second-degree nonlinear Boolean equations based on grouping and minimizing logical propositions in the disjunctive normal form (DNF) class [19]. These equations serve as the algebraic model of S-blocks in symmetric encryption algorithms [5,20–22].

2. Criteria for Analytical Transformations of Boolean Expressions

It is easy to notice that when performing successive elementary transformations, the number of different logical operations increases significantly, requiring enormous computational resources. We will present several formulas in which logical transformations frequently occur, determining a direct transition from one expression to another, as well as the apparatus of decimal numbers, which can be easily implemented on computers [23–28].

Let us introduce the following notation: $\approx_{i=1}^{m}, \vee_{i=1}^{m}, \wedge_{i=1}^{m}, \oplus_{i=1}^{m}, /\,/_{i=1}^{m}, \Rightarrow_{i=1}^{m}$ refer to m-fold equivalence, disjunction, conjunction, modulo 2 addition (mod 2), Sheffer's operation, and implication.

Let us denote by E_n^2 the set of all binary vectors $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. It can be considered as the set of all vertices of a unit n-dimensional cube, with the vectors $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ representing the vertices of the cube [29–32].

Defination 1. *The i-th level of a cube is defined as the set of tuples in E_n^2 whose number of unit coordinates is equal to i, where $i = 0, 1, \ldots, n$.*

3. Methods of Boolean Formula Transformations for Simplifying Logical Statements

We consider analytical criteria for transformation that examine the issues of converting formulas from an arbitrary basis $(\{\neg x,\ x_1\ \&\ x_2,\ x_1 \vee x_2,\ x_1 \to x_2,\ x_1 \oplus x_2, x_1 \sim x_2, x_1/x_2\})$ into formulas over the basis $\{0, 1, x_1\ \&\ x_2,\ x_1 \oplus x_2\}$ [33–37].

In what follows, we transform logical formulas through an expression of the form $\{A_i\}_{o_1}^{m} \equiv \{A_i{}^*\}_{o_2}^{m^1}$, where $\{A_i\}_{o}^{m} = A_1 \cdot A_2 \cdot \ldots \cdot A_m$. We will denote the transformation from the sequential logical operation o_1 to the operation o_2, where o_1, o_2 are symbols for logical operations; $L_{o_2}^{o_1}(m)$ is the number of occurrences of variables in $\left\{A_j^*\right\}_{o_i}^{m^1}, i = 1, 2, \ldots, m; j = 1, 2, \ldots, m^1$; and $m^1 = K_{(o_2)}^{(o_1)}(m)$ is the number of elementary conjunctions in $\left\{A_j^*\right\}_{(o_2)}^{(m^1)}$, A_i, A_j^*—elementary conjunctions.

The criteria for transforming formulas are considered here: $\{U_i\}_{\sim}^{m} \equiv \{U_j^*\}_{\oplus}^{m^1}, \{U_i\}_{\vee}^{m} \equiv \{U_j^*\}_{\oplus}^{m^1}, \{U_i\}_{\to}^{m} \equiv \{U_j^*\}_{\oplus}^{m^1}, \{U_i\}_{I}^{m} \equiv \{U_j^*\}_{\oplus}^{m^1}$ (this allows for eliminating intermediate results in the transformation process and evaluating their complexity in $K_{o_2}^{0_1}(m), L_{0_2}^{0_1}(m)$). Additionally, a new approach to transforming formulas from an arbitrary basis to the Zhegalkin polynomial is considered and theoretically justified [38,39].

Theorem 1. *The transformation* $\{U_i\}_{/}^{m} \equiv \{U_j^*\}_{\oplus}^{m1}$ *is uniquely represented in the form of a decomposition modulo 2:*

$$\&_{i=1}^{m} U_i \oplus \sum_{i=3}^{m} (\&_{j=i}^{m} U_j) \oplus 1 \tag{1}$$

or

$$(1 \oplus U_m(1 \oplus U_{m-1}(1 \oplus U_{m-2}(1 \oplus \ldots \oplus U_3(1 \oplus U_1U_2)\ldots)))) \tag{2}$$

where (2) is obtained from (1) by factoring out common arguments and $K_{\Sigma}^{/}(m) = m$, $L_{\Sigma}^{/}(m) = \Sigma_{i=1}^{m-1} i + 1$.

Proof. We apply the method of induction:

(a) For m = 2, the following relation holds: $U_1/U_2 \equiv U_1U_2 \oplus 1$.

(b) For m = 3, we have $U_1/U_2/U_3 = U_1U_2U_3 \oplus U_3 \oplus 1$.

(c) Suppose that relation (2) holds for m = n: $^{n}//_{i=1}U_i \equiv (1 \oplus U_n(1 \oplus U_{n-1}(1 \oplus \ldots \oplus U_3(1 \oplus U_1U_2)\ldots)))$.

We will prove the validity of relation (2) for $m = n + 1$:

$$\begin{gathered} {}^{n+1}//_{i=1}U_i \equiv {}^{n}//_{i=1}U_i/U_{n+1} \equiv ({}^{n}//_{i=1}U_i)U_{n+1} \oplus 1 \equiv \\ \equiv (1 \oplus U_n(1 \oplus U_{n+1}(1 \oplus \ldots \oplus U_3(1 \oplus U_1U_2)\ldots)))U_{n+1} \oplus 1 \equiv \\ \equiv (1 \oplus U_{n+1}(1 \oplus U_n(1 \oplus U_{n-1}(1 \oplus \ldots \oplus U_3(1 \oplus U_1U_2)\ldots))) \end{gathered}$$

From this, we obtain $^{n+1}//_{i=1}U_i = 1 \oplus U_{n+1}(1 \oplus U_n(1 \oplus \ldots \oplus U_3(1 \oplus U_1U_2)\ldots))$.

The estimates $K_{\Sigma}^{/}(m)$ and $L_{\Sigma}^{/}(m)$ are easily derived from Formula (1). □

Theorem 2. *The transformation* $\{U_i\}_{\vee}{}^{m} \equiv \{U_j*\}_{\oplus}{}^{m1}$ *has the form*

$$\{U_j*\}_{\oplus}{}^{m1} = \sum (U_1^{\sigma_1} \ \& \ U_2^{\sigma_2} \ \& \ \ldots \ \& \ U_k^{\sigma_k}) \tag{3}$$

where $(\sigma_1 \vee \sigma_2 \vee \ldots \vee \sigma_m = 1), \{\sigma_1, \sigma_2, \ldots, \sigma_k\} \in \{\sigma_1, \sigma_2, \ldots, \sigma_m\}, \sigma_1 = \sigma_2 = \ldots = \sigma_k = 1$, $i_k \in \{1, 2, \ldots, m\}, k \leq m$, *and* $L_{\Sigma}{}^{V}(m) = \Sigma_{i=1}^{m}(i \cdot C_m^i)$, $K_{\Sigma}{}^{V}(m) = 2^m - 1$.

Proof. Let B_k and l_k denote the expression and the number of unit coordinates after transforming the formulas $\bigvee_{i=1}^{k} U_i$ into the Zhegalkin polynomial.

For $k = 2$ and $k = 3$, the formulas are found very easily:

$$U_1 v U_2 = U_1U_2 \oplus U_1 \oplus U_2 = B_2, l_2 = 3;$$

$$\begin{gathered} U_1 \vee U_2 \vee U_3 = B_2U_3 \oplus B_2 \oplus U_3 = U_1U_2U_3 \oplus U_1U_3 \oplus \\ \oplus U_2U_3 \oplus U_1U_2 \oplus U_1 \oplus U_2 \oplus U_3 = B_3, \ l_3 = l_2 + l_2 + 1 = 7; \end{gathered}$$

Let $k = n + 1$. Then, the following holds:

$$U_1 \vee U_2 \vee \ldots \vee U_n \vee U_{n+1} = B_nU_{n+1} \oplus B_n \oplus U_{n+1}, l_{n+1} = 2l_n + 1.$$

If for $B_k(k = 1, \ldots, n + 1)$, the number of unit coordinates is calculated sequentially,

$$\begin{gathered} k = 1, \ l_1 = 1 = 2^1 1, \\ k = 2, \ l_2 = 2l_1 + 1 = 3 = 2^2 - 1, \\ k = 3, \ l_3 = 2l_2 + 1 = 7 = 2^3 1, \\ \ldots\ldots\ldots \\ k = n + 1, \ l_{n+1} = 2l_n + 1 = 2^{n+1} - 1. \end{gathered}$$

then it is easy to observe that for m variables, we have $l_m = K_{\Sigma}^{\vee}(m) = 2^m 1$.

https://doi.org/10.3390/math14040594

If we consider that $E^2_{\mathrm{m}} = 2^{\mathrm{m}}$ and $\mathrm{K}^{\vee}_{\Sigma}(\mathrm{m}) = 2^{\mathrm{m}}\ 1$, then the decomposition consists of all possible unit elements of an m-dimensional cube, through which Formula (3) can be easily expressed in an analytical form.

It is known that at each level i of the m-dimensional cube, there are C^i_m sets, and in each set, i units participate, corresponding to m variables. Consequently, the number of variable occurrences in the polynomial $\{\mathrm{U_j}\}^m_{\Sigma}$ equals $\mathrm{L}^{\vee}_{\Sigma}(\mathrm{m}) = \Sigma^m_{i=1}(\mathrm{i}\, C^i_m)$. □

Theorem 3. *In the basis* $D_2 = \{0, 1, x_1 \ \& \ x_2, x_1 \oplus x_2\}$, *the sequence of implication operations is uniquely represented in the form of a Zhegalkin polynomial:*

$$\Rightarrow^m_{=1} U \equiv \sum_{i=1}^{\frac{m}{2}-(\frac{t}{2}-1)} \sum_{j=2i}^{m-t+2} \sum_{k=j+1}^{m-t+3} \dots \sum\nolimits_{l=n+1}^{m} (U_{2i-1} U_j U_k \dots U_l) \oplus$$
$$\oplus \sum_{i=1}^{\frac{m}{2}-\frac{(p-1)}{2}} \sum_{j=2i}^{m-p+2} \sum_{k=j+1}^{m-p+3} \dots \sum\nolimits_{l=n+1}^{m} (U_{2i-1} U_j U_k \dots U_l)$$

where

$$t = \begin{cases} 2,\ 4,\ \dots,\ m & \text{if } m \text{ is even}, \\ 1,\ 3,\ \dots,\ m & \text{otherwise}. \end{cases}$$

$$= \begin{cases} 1, 3,\ \dots,\ m-1; & \text{if } m \text{ is even}, \\ 2,\ 4,\ \dots,\ m-1; & \text{otherwise}. \end{cases}$$

$$c = \begin{cases} 1, & \text{if } m \text{ is even}, \\ 0\,, & \text{otherwise}. \end{cases}$$

and the following holds:

$$K^{\rightarrow}_{\Sigma}(m) = \begin{cases} 1 + \frac{2}{3}\,(2^m - 1), & \text{if } m \text{ is even}\,, \\ \frac{1}{3}\,(2^{m+1} - 1), & \text{otherwise}. \end{cases}$$

where p and t represent the number of arguments in the unit coordinate system.

Proof. The proof of the main formula is not provided, as it can be easily obtained using a method similar to that of the previous theorems or through a sequential transformation into the Zhegalkin polynomial. Let us consider the proof of the number of unit coordinates involved in the polynomial for arbitrary m.

Let $T_i (i = 1, 2, \dots, m)$ be the number of unit coordinates in the polynomial as the number of variables increases from 1 to m.

For $m = 2$ and $m = 3$, we have

$$\mathrm{U_1} \to \mathrm{U_2} = \mathrm{U_1 U_2} \oplus \mathrm{U_1} \oplus 1,$$

$$\mathrm{U_1} \to \mathrm{U_2} \to \mathrm{U_3} = \mathrm{U_1 U_2 U_3} \oplus \mathrm{U_1 U_2} \oplus \mathrm{U_1 U_3} \oplus \mathrm{U_1} \oplus \mathrm{U_3}$$

and, consequently, we obtain $T_2 = 3$, $T_3 = 5$.

If we sequentially construct the polynomials, it is easy to notice that the number of unit coordinates in each subsequent polynomial is either one more or one less than in the previous one. It increases by one if m is even and decreases by one otherwise.

Thus, we can construct a numerical sequence that defines the number of unit coordinates in the polynomial as m increases:

$$\mathrm{T_i} = \begin{cases} 2\mathrm{T_{i-1}} + 1, & \text{if i is even}, \\ 2\mathrm{T_{i-1}}\ 1, & \text{otherwise, where i} = 2, 3, \dots, \mathrm{m} \end{cases}$$

Let us divide the sequence into two groups based on even and odd values of m:

$$T_{2k}: \{3, 11, 43, 171, \ldots\}, k = 1, 2, \ldots, m/2;$$

$$T_{2k-1}: \{1, 5, 21, 85, 341, \ldots\}, k = 1, 2, \ldots, (m+1)/2.$$

From this, it is evident that

$T_{2k} = T_{2(k-1)} + 2^{2k-1}$ and $T_{2k-1} = T_{2(k-1)-1} + 2^{2k-2}$ or

$$T_{2k} = 1 + 2^1 + 2^3 + 2^5 + \ldots + 2^{2k-1} = 1 + 2\left(1 + 2^1 + 2^4 + \ldots + 2^{2k-2}\right) = \\ = 1 + 2\left(1 + 4^1 + 4^2 + \ldots + 4^{(k-1)}\right);$$

$$T_{2k-1} = 2^o + 2^2 + 2^4 + \ldots + 2^{2k-2} = 1 + 4^1 + 4^2 + \ldots + 4^{k-1}.$$

Using the sum of a geometric progression, we obtain

$$T_{2k} = 1 + 2\left(\frac{4^k-1}{4-1}\right) = 1 + \frac{2}{3}\left(4^k - 1\right), \\ T_{2k-1} = \left(4^k - 1\right)/3 = \frac{1}{3}\left(4^k - 1\right)$$

or, considering that $2k = m$ when m is even, and $2k - 1 = m$, $2k = m + 1$ otherwise, along with $4^k = 2^{2k}$, the following holds:

$$T_m = \begin{cases} 1 + \frac{2}{3}\left(2^m - 1\right), & \text{if } m \text{ is even} \\ \frac{1}{3}\left(2^{m+1} - 1\right), & \text{otherwise.} \end{cases}$$

□

Theorem 4. *The transformation* $\{U_i\}_{\sim}^{m} = \{U_j\}_{\Sigma}^{m}$ *can be uniquely represented in the form of a decomposition modulo 2 (mod2).*

$$\overset{m}{\underset{i=1}{\approx}} A_i = \begin{cases} \neg\left(\sum\limits_{i=1}^{m} U_i\right), & \textit{if } m \textit{ is even,} \\ \sum\limits_{i=1}^{m} U_i, & \textit{otherwise.} \end{cases}$$

and

$K_{\Sigma}^{\sim}(m) = \begin{cases} m+1, & \textit{if } m \textit{ is even,} \\ m, & \textit{otherwise.} \end{cases}$, $L_{\Sigma}^{\sim}(m) = m$.

With this transformation, the inverse problem is also solved in the same manner, i.e.,

$$\sum_{i=1}^{m} U_i = \begin{cases} \left(\overset{m}{\underset{i=1}{\approx}} U_i\right), & \textit{if } m \textit{ is even;} \\ \overset{m}{\underset{i=1}{\approx}} U_i, & \textit{otherwise.} \end{cases}$$

Theorem 5. *The transformation* $\{U_i\}_{\&}^{m} = \{U_j\}_{\Sigma}^{m1}$ *in its analytical form is uniquely represented as an expansion in terms of mod2.*

$$\{\neg U_i\}_{\&}^{m} = \overset{m}{\underset{i=1}{\&}} (\neg U_i) = \overset{m}{\underset{i=1}{\&}} U_i \oplus \sum_{i=1}^{2} \sum_{j=i+1}^{3} \ldots \sum_{k=l+1}^{m} U_i U_j \ldots U_k \oplus \\ \oplus \ldots \oplus \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} U_i U_j \oplus \sum_{i=1}^{m} U_i \oplus 1;$$

The proof of the theorem is easily provided using the method of induction.

Let $P(1, x_2, \ldots, x_n) = \sum_{i=1}^{k+1} U_i(1, 2, \ldots, n)$ be a polynomial over the basis B^*, where U_i is an elementary component, $U_l \neq U_j$, and let $U(\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k})$ denote the elementary component $U = x_{i_1}^{\sigma_{i_1}} \cdot x_{i_2}^{\sigma_{i_2}} \cdot \ldots \cdot x_{i_k}^{\sigma_{i_k}}$.

Theorem 6. *The polynomial $P(i_1, x_{i_2}, \ldots, x_{i_k})$ is identically equal to one if P can be represented in the form*

$$Uj_1(\sigma_{i_1}) \oplus Uj_2(\overline{\sigma}_{i_1}, \sigma_{i_2}) \oplus Uj_3(\overline{\sigma_{i_1}}, \overline{\sigma_{i_2}, \sigma_{i_3}}) \oplus \ldots \oplus Uj_k(\overline{\sigma_{i_1}}, \ldots, \overline{\sigma_{i_{k-1}}}, \sigma_{i_k}) \oplus Uj_{k+1}(\overline{\sigma_{i_1}}, \ldots, \overline{\sigma_{i_{k-1}}}, \overline{\sigma_{i_k}}) \quad (4)$$

where $U_{j_i} \in U_i$, $i = 1, 2, \ldots, k+1$;.

Proof. Let us express the polynomial (4) in its general form as follows:

$\sum_{i=1}^{k}(\;\&\; {}_{j=1}^{i-1} x_{i_j}^{\overline{\sigma_{i_j}}}) x_{i_j}^{\sigma_{i_j}} \oplus \;\&\; {}_{j=1}^{k} x_{i_j}^{\overline{\sigma_{i_j}}}$

or

$x_{i_1}^{\sigma_{i_1}} \oplus x_{i_1}^{\overline{\sigma_{i_1}}} x_{i_2}^{\sigma_{i_2}} \oplus x_{i_1}^{\overline{\sigma_{i_1}}} x_{i_2}^{\overline{\sigma_{i_2}}} x_{i_3}^{\sigma_{i_3}} \oplus \ldots \oplus x_{i_1}^{\overline{\sigma_{i_1}}} \ldots x_{i_{k-1}}^{\overline{\sigma_{i_{k-1}}}} x_{i_k}^{\sigma_{i_k}} \oplus x_{i_1}^{\overline{\sigma_{i_1}}} \ldots x_{i_k}^{\overline{\sigma_{i_k}}}$. □

Corollary 1. $Uj_1 \oplus \ldots \oplus Uj_{i-1} \oplus Uj_{i+1} \oplus \ldots \oplus Uj_{k+1} \equiv Uj_i \oplus 1$, holds if $U_{j_1} \oplus \ldots \oplus U_{j_{i-1}} \oplus U_{j_{i+1}} \oplus \ldots \oplus U_{j_{k+1}} = U_{j_i} \oplus 1$, satisfies the condition of formula (4), where U_{j_l}, $l = 1, 2, \ldots, k+1$.

Corollary 2. If the elementary component $U, i = 1, \ldots, m$; $G_j, j = 1, \ldots, t$ can be expressed as $U = \cdot U'$, $G_j = \times G'_j$ where U', $G_j{}'$ (elementary component) and $\sum_{i=1}^{m} U'_i = 1$, $\sum_{j=1}^{t} G'_j = 1$, then $\sum_{i=1}^{m} U \oplus \sum_{j=1}^{t} G_j = 0$.

4. Text Example

1. Initial analytical formula: $R = R_1 \tilde{R}_2 \tilde{R}_3 \tilde{R}_4$, where $\mathcal{R}_1 = X_1 \,\&\, \neg X_3 \,\&\, x_5$, $\mathcal{R}_2 = \neg X_2 \,\&\, X_3 \,\&\, \neg X_4 \,\&\, X_5$, $\mathcal{R}_3 = \neg X_3 \,\&\, \neg X_5$, $\mathcal{R}_4 = \neg X_1 \,\&\, \neg X_3 \,\&\, \neg X_5$ (See Table 1).

Table 1. Formula representation in the reduced basis.

R \ X	X_1	X_2	X_3	X_4	X_5
1	1	*[1]	0	*	1
2	*	0	1	0	1
3	*	*	0	*	0
4	0	*	0	*	0

[1] *—accepts an arbitrary value (0 or 1) and does not affect the remaining values, which makes no difference.

$\sigma_1 \sim \sigma_2 \sim \sigma_3 \sim \sigma_4 = 1$ is the set of tuples satisfying the following equality: $Q = \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\}$.

2. Corresponding disjunctive normal form (DNF)

$$R = \neg A_1 \;\&\; \neg A_2 \;\&\; \neg A_3 \;\&\; \neg A_4 \vee \neg A_1 \;\&\; \neg A_2 A_3 \;\&\; A_4 \vee \neg A_1 \;\&\; A_2 \;\&\; \;\&\; \neg A_3 \;\&\; A_4 \vee$$
$$\vee \neg A_1 \;\&\; A_2 \;\&\; A_3 \;\&\; \neg A_4 \vee A_1 \;\&\; \neg A_2 \;\&\; \neg A_3 A_4 \vee A_1 \;\&\; \neg A_2 \;\&\; A_3 \;\&\; \neg A_4 \vee$$
$$\vee A_1 \;\&\; A_2 \;\&\; \neg A_3 \;\&\; \neg A_4 \vee A_1 \;\&\; A_2 \;\&\; A_3 \;\&\; A_4;$$

3. Result of the obtained formula after minimization in the reduced basis.

4. Result in analytical form (See Table 2):

$$R2 = X_2 \;\&\; X_3 \vee X_3 \;\&\; X_4 \vee X_3 \;\&\; \overline{X}_5 \vee \overline{X}_1 \;\&\; X_2 \;\&\; X_5 \vee \overline{X}_1 \;\&\; X_4 \;\&\; X_5 \vee \overline{X}_1 \;\&\; \overline{X}_3$$

Table 2. The result of the obtained formula.

U_i \ X	X_1	X_2	X_3	X_4	X_5
1	* [1]	1	1	*	*
2	*	*	1	1	*
3	*	*	1	*	0
4	0	1	*	*	1
5	0	*	*	1	1
6	0	*	0	*	*

[1] *—accepts an arbitrary value (0 or 1) and does not affect the remaining values, which makes no difference.

5. Solution of Second-Degree Nonlinear Boolean Equation Systems of a Special Class

This study examines the solution of second-degree nonlinear Boolean equation systems, which serve as a logical model for algebraic cryptanalysis [40]. Test examples of logical formula transformations are constructed. The research also explores the solution of second-degree nonlinear Boolean equation systems of a special class. The key advantage of the findings compared to other approaches lies in the ability to solve such systems based on the grouping and minimization of logical statements in the class of disjunctive normal forms (DNFs), which serve as an algebraic model of S-boxes in symmetric encryption algorithms [41–43].

A special class of systems of nonlinear Boolean equations of the second degree is being studied:

$$R = \{f_1(x_1, x_2, \ldots, x_n) = \alpha_1,\ f_2(x_1, x_2, \ldots, x_n) = \alpha_2, \ldots, f_m(x_1, x_2, \ldots, x_n) = \alpha_m\}$$

Moreover, the statement $f(x_1, x_2, \ldots, x_n)$ from R has the form

$$f = \sum_{\substack{i,j=k \\ i<j}}^{k+3} a_{ij}x_ix_j \oplus \sum_{\substack{i,j=e \\ i<j}}^{e+3} b_{ij}x_ix_j \oplus \sum_{\substack{i,j=p \\ i<j}}^{p+3} c_{ij}x_ix_j \oplus \sum_{\substack{i,j=q \\ i<j}}^{q+3} d_{ij}x_ix_j \oplus \sum_{i=t}^{t+3} e_ix_i,$$

where $k+3 < e,\ e+3 < p,\ p+3 < q,\ q+3 < t$;

$$\sum_{i,j=k}^{k+3} a_{ij} = \sum_{i,j=e}^{e+3} b_{ij} = \sum_{i,j=p}^{p+3} c_{ij} = \sum_{i,j=q}^{q+3} d_{ij} = 4,\ \{a_{ij}, b_{ij}, c_{ij}, d_{ij}, e_i\} \in \{0,1\}$$

Here, the symbols $\oplus$, Σ are understood as summation modulo 2.

To solve nonlinear Boolean equations of the second degree, the study focuses on compactly representing statements f by grouping elements, introducing new variables, transforming f into specialized disjunctive normal forms (DNFs) using a more optimal decimal representation method, simplifying these specialized DNFs, and analyzing the key features of formula implementation for this special class of nonlinear Boolean equation systems of the second degree.

Here, sums of the form $\sum_{i,j=v}^{v+3} q_{ij}x_ix_j$ and $\sum_{i=w}^{w+1} e_ix_i$ will be referred to as groups of elements of the statement f. Moreover, the groups of different equations (statements) in the system R do not coincide pairwise.

The method for solving system R consists of compactly representing f_i by grouping elements, introducing new variables, transforming them into disjunctive normal forms (DNFs), and simplifying them. The search for solutions to system R is carried out using an algorithm for solving systems of linear Boolean equations.

The functional of an arbitrary system α, obtained from R by grouping and replacing variables in the elements of the statements, is denoted as follows:

$\psi_\alpha = \sum \varphi_y$, where $\{Y\}$ is the set of variables in system α, φ_y is the number of occurrences of $Y \in \{Y\}$ in Y, and $|Y|$ is the number of elements in Y. The algorithm for grouping system R from the given class consists of the following steps:

- Groups of elements of the statements in system R are identified;
- All possible groupings within these groups are performed, and new variables are introduced.
- From each group, such groupings of elements are selected to ensure that for the resulting system α, the functional ψ_α is maximized among all functionals ψ_β of systems β formed from the groupings of the system's groups.

Thus, for a formula like

$$f = \sum_{\substack{i,j=k \\ i<j}}^{k+3} a_{ij}x_ix_j \oplus \sum_{\substack{i,j=e \\ i<j}}^{e+3} b_{ij}x_ix_j \oplus \sum_{\substack{i,j=p \\ i<j}}^{p+3} c_{ij}x_ix_j \oplus \sum_{\substack{i,j=q \\ i<j}}^{q+3} d_{ij}x_ix_j \oplus \sum_{i=t}^{t+3} x_i,$$

the matrix A $n \times n$ is constructed, where the element $a_{ij} = 1$ if the formula f contains x_ix_j and $a_{ij} = 0$ otherwise. Here, a_{ii} corresponds to x_ix_i in formula f, i.e., x_i. Then, the maximum grouping problem in formula f is identical to the problem of finding the maximum identity submatrix of matrix A. It is easy to see that the algorithm for solving the latter problem has polynomial complexity.

It has been proven that at the initial stage, each statement $f(x_1, x_2, \ldots, x_n)$ from R involves 20 elementary conjunctions. However, after grouping and introducing new variables, the statements $F(\widetilde{x}, Y(\widetilde{x}))$ in the transformed system R^* will contain no more than nine linear conjunctions. From this data, based on $U_1 \oplus U_2 \oplus \cdots \oplus U_t = \bigvee_{\sigma_1 \oplus \sigma_2 \oplus \ldots \oplus \sigma_t = 1} U_1^{\sigma_1}$ & $U_2^{\sigma_2}$ & $\ldots$ & $U_t^{\sigma_t}$, we obtain $L_k(f) = 2^{20}$, $L_k(F) = 2^9$.

From this, it is evident that the complexity difference between statements f and F is reduced by a factor of 2^{11}, i.e., $L_k = \frac{L_k(f)}{L_k(F)} = \frac{2^{20}}{2^9} = 2^{11}$, where $L_k(Q)$ is the number of elementary conjunctions in the disjunctive normal form (DNF) Q that we obtain.

This means that the method of grouping elements and introducing new variables for second-degree equations of a special class reduces the complexity of Boolean equation statements by a factor of 2^{11}.

Solving systems of nonlinear Boolean equations is an NP-complete problem. Moreover, the SAT or Mutant solvers offer a solution to the problem based on posing all binary sets $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ of the set E_n^2 for each equation without simplifying the Boolean equation statements. This increases the complexity of the system $/E_n^2/$ times, where $/M/$ is the cardinality of the set M.

However, the article proposes grouping nonlinear polynomials. Moreover, this is not always the case; in the best case, second-order polynomials can be reduced to a linear polynomial by grouping, demonstrating the transition from the NP problem to the P problem. For example, the equation $x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 = 1$ can be transformed to the equation $(x_1 + x_2)(x_3 + x_4) = 1$, which is identical to the system $x_1 + x_2 = 1$,$x_3 + x_4 = 1$.

In addition, minimization of statements of nonlinear Boolean equations is performed, for example, in the application of the formula

$$\begin{cases} F(X) + 1 = (F(X))' \\ F(X)G(X) + F(X) = F(X)(G(X) + 1) = F(X)(G(X))' \end{cases} \tag{5}$$

Let

$$F_5\left(F_4\left(F_3\left(F_2(F_1)'\right)'\right)'\right)' = 1 \tag{6}$$

There are 13 solutions in total (See Table 3).

Table 3. The solution of the equation using Formula (6).

Number of Solutions		
$F_5 = 1$; $F_4 = 0$; F_1, F_2, F_3-optional	$F_5 = 1$; $F_4 = 1$; $F_3 = 0, F_1, F_2$-optional	$F_5 = 1$; $F_4 = 1$; $F_3 = 1, F_2 = 1, F_3 = 0$
8	4	1

Example: Solve the equation.
$F(x_1, \ldots, x_5) = x_1x_2x_3x_4x_5 + x_3x_4x_5 + x_4x_5 + x_4 = 1$.
It is easy to see that (See Table 4)

$$x_4\left(x_5\left((x_1x_2)'x_3\right)'\right)' = 1 \tag{7}$$

Table 4. The solution of the equation using the Formula (7).

Number of Solutions	
$x_4 = 1$; $x_5 = 0$ x_1, x_2, x_3-optional	$x_4 = 1$ $x_5 = 1$ $x_3 = 1; (x_1, x_2) = 0$
8	3

There are 11 solutions in total.

The following example can be used as an example of the results for S-boxes. For the GOST R34.12-2015 (Kuznechik) algorithms [44], the *S*-box ANF consists of y(0), . . . , y(7). For brevity, we present the ANF for y(0).

$y[0] = 1 + x_6x_7 + x_5x_7 + x_5x_6 + x_4x_6x_7 + x_4x_5 + x_4x_5x_6 + x_4x_5x_6x_7 + x_3x_7 + x_3x_5x_7 + x_3x_5x_6 + x_3x_4 + x_3x_4x_7 + x_3x_4x_6 + x_3x_4x_5 + x_3x_4x_5x_7 + x_3x_4x_5x_6x_7 + x_2x_7 + x_2x_6 + x_2x_5x_7 + x_2x_5x_6x_7 + x_2x_4x_6x_7 + x_2x_4x_5 + x_2x_4x_5x_6 + x_2x_4x_5x_6x_7 + x_2x_3 + x_2x_3x_7 + x_2x_3x_6 + x_2x_3x_6x_7 + x_2x_3x_5 + x_2x_3x_5x_7 + x_2x_3x_4 + x_1x_7 + x_1x_6 + x_1x_5x_6 + x_1x_4x_5 + x_1x_4x_5x_7 + x_1x_3x_7 + x_1x_3x_5 + x_1x_3x_5x_7 + x_1x_3x_5x_6x_7 + x_1x_3x_4 + x_1x_3x_4x_7 + x_1x_3x_4x_6 + x_1x_3x_4x_6x_7 + x_1x_3x_4x_5x_6x_7 + x_1x_2 + x_1x_2x_7 + x_1x_2x_6 + x_1x_2x_6x_7 + x_1x_2x_5x_7 + x_1x_2x_5x_6x_7 + x_1x_2x_4 + x_1x_2x_4x_7 + x_1x_2x_4x_5 + x_1x_2x_4x_5x_6 + x_1x_2x_3 + x_1x_2x_3x_6x_7 + x_1x_2x_3x_5 + x_1x_2x_3x_5x_7 + x_1x_2x_3x_5x_6 + x_1x_2x_3x_5x_6x_7 + x_1x_2x_3x_4x_7 + x_1x_2x_3x_4x_6 + x_1x_2x_3x_4x_6x_7 + x_1x_2x_3x_4x_5 + x_1x_2x_3x_4x_5x_7 + x_1x_2x_3x_4x_5x_6 + x_0x_6 + x_0x_5 + x_0x_5x_6x_7 + x_0x_4x_7 + x_0x_4x_6 + x_0x_4x_6x_7 + x_0x_4x_5 + x_0x_4x_5x_6 + x_0x_3x_6 + x_0x_3x_5x_6 + x_0x_3x_5x_6x_7 + x_0x_3x_4 + x_0x_3x_4x_7 + x_0x_3x_4x_6 + x_0x_3x_4x_6x_7 + x_0x_3x_4x_5 + x_0x_3x_4x_5x_7 + x_0x_2x_7 + x_0x_2x_6 + x_0x_2x_6x_7 + x_0x_2x_4 + x_0x_2x_4x_6x_7 + x_0x_2x_4x_5x_6x_7 + x_0x_2x_3 + x_0x_2x_3x_7 + x_0x_2x_3x_6x_7 + x_0x_2x_3x_5 + x_0x_2x_3x_5x_7 + x_0x_2x_3x_5x_6x_7 + x_0x_2x_3x_4x_5 + x_0x_2x_3x_4x_5x_6 + x_0x_2x_3x_4x_5x_6x_7 + x_0x_1 + x_0x_1x_7 + x_0x_1x_6 + x_0x_1x_5x_7 + x_0x_1x_5x_6x_7 + x_0x_1x_4x_7 + x_0x_1x_4x_5 + x_0x_1x_4x_5x_7 + x_0x_1x_3 + x_0x_1x_3x_7 + x_0x_1x_3x_6x_7 + x_0x_1x_3x_5 + x_0x_1x_3x_5x_7 + x_0x_1x_3x_5x_6 + x_0x_1x_3x_4 + x_0x_1x_3x_4x_7 + x_0x_1x_3x_4x_6x_7 + x_0x_1x_3x_4x_5 + x_0x_1x_3x_4x_5x_6 + x_0x_1x_3x_4x_5x_6x_7 + x_0x_1x_2 + x_0x_1x_2x_7 + x_0x_1x_2x_5 + x_0x_1x_2x_5x_6x_7 + x_0x_1x_2x_4 + x_0x_1x_2x_4x_7 + x_0x_1x_2x_4x_5x_6x_7 + x_0x_1x_2x_3x_6x_7 + x_0x_1x_2x_3x_5x_7 + x_0x_1x_2x_3x_5x_6x_7 + x_0x_1x_2x_3x_4 + x_0x_1x_2x_3x_4x_7 + x_0x_1x_2x_3x_4x_6 + x_0x_1x_2x_3x_4x_6x_7 + x_0x_1x_2x_3x_4x_5 + x_0x_1x_2x_3x_4x_5x_7 + x_0x_1x_2x_3x_4x_5x_6$

After grouping:

$$y[0] = (x_4x_5(x_0(x_3(x_2(x_6(x_7)')')')')')' + x_1x_3x_7(x_5(x_6(x_4(x_0)')')')' + x_0x_1x_7(x_5(x_6(x_2(x_4)')')')' + x_1x_2x_7(x_6(x_3(x_5(x_0)')')')' + x_0x_1x_3(x_7(x_6(x_4(x_2)')')')')' + x_2x_3(x_7(x_0(x_5(x_1(x_4)')')')')' + x_1x_3x_5(x_0(x_6(x_4(x_2)')')')' + x_3x_4x_7(x_1(x_2(x_6)')')')' + x_0x_2x_7(x_6(x_3(x_1)')')')' + x_2x_5x_7(x_1(x_3(x_4)')')')' + x_0x_4x_7(x_1(x_3(x_2)')')')' + x_3x_4x_6(x_1(x_2(x_0)')')')' + x_4x_5x_6(x_2(x_1(x_3)')')')' + x_1x_4x_5(x_0(x_3(x_2)')')')' + x_6x_7(x_4(x_5(x_2(x_0)')')')' + x_0x_5x_6x_7(x_3(x_2)')' + x_1x_4x_5x_7(x_0)' + x_3x_5x_7(x_0x_1)' + x_2x_3x_5(x_1(x_6)')' + x_0x_1x_2(x_7(x_4)')' + x_1x_2x_4(x_5(x_3)')' + x_1x_3x_4(x_6x_7)' + x_3x_4x_5(x_7(x_6)')' + x_0x_3x_4(x_1(x_2)')' + x_0x_4x_6(x_7(x_3)')' + x_0x_3x_4x_7(x_5)' + x_2x_4x_6x_7(x_0)' + x_2x_5x_6x_7(x_1)' + x_0x_4x_5x_6 + x_0x_3x_6(x_5)' + x_0x_1x_2x_5 + x_0x_2x_4(x_1)' + x_0x_3x_4x_6 + x_1x_2x_4x_7 + x_2x_3x_6(x_7)' + x_0x_2x_3(x_5)' + x_2x_3x_5x_7 + x_0x_1x_6 + x_1x_2x_6 + x_2x_3x_4 + x_0x_2x_6 + x_1x_2x_3 + x_1x_5x_6 + x_3x_5x_6 + x_2x_4x_5 + (x_6 + x_7)(x_2 + x_5) + (x_0)(x_1 + x_5 + x_6) + (x_1)(x_2 + x_6 + x_7) + (x_3)(x_4 + x_7)$$

where the formula $(F(x))' = F(x) \oplus 1$, i.e., it is the logical negation of the function $F(x)$.

To solve this system, subsystems R^* are identified using a first-order neighborhood algorithm and transformed into a system of linear logical equations,

$$x_{j_1}^{\sigma'_1} x_{j_2}^{\sigma'_2} \cdots x_{j_k}^{\sigma'_l} Y_{l_i m_i}^{\sigma'_{l+i}} Y_{l_j m_j r_j}^{\sigma'_{l+j}} Y_{l_k m_k r_k s_k}^{\sigma'_{l+k}} = 1,\ l \le n,\ i \le 8,\ j \le 16,\ k \le 32,$$

followed by the simplification of special disjunctive normal forms (DNFs) using the following null identities:

(a) For the linear form of two variables:

$x_i x_j Y_{ij} = 0,\ x_i \overline{x_j} \overline{Y_{ij}} = 0,\ \overline{x_i} x_j \overline{Y_{ij}} = 0,\ \overline{x_i x_j} Y_{ij} = 0,$

(b) For the linear form of three variables:

$$x_i x_j x_k \overline{Y_{ijk}} = 0,\ x_i x_j \overline{x_k} Y_{ijk} = 0,\ x_i \overline{x_j} x_k Y_{ijk} = 0,\ x_i x_j x_k Y_{ijk} = 0,\ x_i \overline{x_j x_k} \overline{Y_{ijk}} = 0,$$
$$\overline{x_i} x_j \overline{x_k} \overline{Y_{ijk}} = 0,\ \overline{x_i x_j} x_k \overline{Y_{ijk}} = 0,\ \overline{x_i x_j x_k} Y_{ijk} = 0.$$

(c) For the linear form of four variables:

$$x_i x_j x_k x_1 Y_{ijk1} = 0,\ x_i x_j x_k \overline{x_1} \overline{Y_{ijk1}} = 0,\ x_i x_j \overline{x_k} x_1 \overline{Y_{ijk1}} = 0,\ x_i \overline{x_j} x_k x_1 \overline{Y_{ijk1}} = 0,\ \overline{x_i} x_j x_k x_1 \overline{Y_{ijk1}} = 0,$$
$$x_i x_j \overline{x_k x_1} Y_{ijk1} = 0,\ x_i \overline{x_j} x_k \overline{x_1} Y_{ijk1} = 0,\ \overline{x_i} x_j x_k \overline{x_1} Y_{ijk1} = 0,\ x_i \overline{x_j x_k} x_1 Y_{ijk1} = 0,\ \overline{x_i} x_j \overline{x_k} x_1 Y_{ijk1} = 0,$$
$$\overline{x_i x_j} x_k x_1 Y_{ijk1} = 0,\ \overline{x_i x_j} x_k x_1 \overline{Y_{ijk1}} = 0,\ \overline{x_i x_j} x_k \overline{x_1} \overline{Y_{ijk1}} = 0,\ \overline{x_i x_j} x_k \overline{x_1} \overline{Y_{ijk1}} = 0,\ x_i \overline{x_j} x_k x_1 \overline{Y_{ijk1}} = 0,$$
$$\overline{x_i x_j} x_k x_1 Y_{ijk1} = 0.$$

Next, we simplify by applying the fundamental features of formula implementation for the special class of second-degree nonlinear Boolean equation systems:

(a) For the linear form of two variables:

$$\overline{x_i} Y_{ij} = \overline{x_i} x_j,\ x_j \overline{Y_{ij}} = x_i x_j,\ x_i Y_{ij} = \overline{x_i} x_j,\ x_i \overline{Y_{ij}} = x_i x_j,\ x_i Y_{ij} = x_i \overline{x_j},$$
$$\overline{x_i} Y_{ij} = x_i \overline{x_j},\ \overline{x_j} \overline{Y_{ij}} = \overline{x_i x_j},\ x_i x_j \overline{Y_{ij}} = x_i x_j,\ \overline{x_i} x_j Y_{ij} = \overline{x_i} x_j,$$
$$x_i \overline{x_j} Y_{ij} = x_i \overline{x_j},\ \overline{x_i x_j} \overline{Y_{ij}} = \overline{x_i x_j}.$$

(b) For the linear form of three variables:

$$x_i x_j x_k Y_{ijk} = x_i x_j x_k,\ x_i x_j \overline{x_k} \overline{Y_{ijk}} = x_i x_j x_k,\ x_i \overline{x_j} x_k \overline{Y_{ijk}} = x_i \overline{x_j} x_k,\ \overline{x_i} x_j x_k \overline{Y_{ijk}} = \overline{x_i} x_j x_k,$$
$$x_i \overline{x_j x_k} Y_{ijk} = x_i \overline{x_j x_k},\ \overline{x_i} x_j \overline{x_k} Y_{ijk} = \overline{x_i} x_j \overline{x_k},\ \overline{x_i x_j} x_k Y_{ijk} = \overline{x_i x_j} x_k,\ \overline{x_i x_j} x_k \overline{Y_{ijk}} = \overline{x_i x_j} x_k.$$

(c) For the linear form of four variables:

$$x_ix_jx_kx_1\overline{Y_{ijk1}} = x_ix_jx_kx_1,\ x_ix_jx_k\overline{x_1}Y_{ijk1} = x_ix_jx_k\overline{x_1},\ x_ix_j\overline{x_k}x_1Y_{ijk1} = x_ix_j\overline{x_k}x_1,$$
$$x_i\overline{x_j}x_kx_1Y_{ijk1} = x_i\overline{x_j}x_kx_1,\ \overline{x_i}x_jx_kx_1\overline{Y_{ijk1}} = \overline{x_i}x_jx_kx_1,\ x_ix_j\overline{x_k}x_1\overline{Y_{ijk1}} = x_ix_j\overline{x_k}x_1,$$
$$x_i\overline{x_j}x_k\overline{x_1}\overline{Y_{ijk1}} = x_i\overline{x_j}x_k\overline{x_1},\ \overline{x_i}x_jx_k\overline{x_1}\overline{Y_{ijk1}} = \overline{x_i}x_jx_k\overline{x_1},\ x_i\overline{x_j}\overline{x_k}x_1\overline{Y_{ijk1}} = x_i\overline{x_j}\overline{x_k}x_1,$$
$$\overline{x_i}x_j\overline{x_k}x_1\overline{Y_{ijk1}} = \overline{x_i}x_j\overline{x_k}x_1,\ \overline{x_i}\overline{x_j}x_kx_1\overline{Y_{ijk1}} = \overline{x_i}\overline{x_j}x_kx_1,\ \overline{x_i}x_jx_kx_1Y_{ijk1} = \overline{x_i}x_jx_kx_1.$$
$$\overline{x_i}\overline{x_j}x_k\overline{x_1}Y_{ijk1} = \overline{x_i}\overline{x_j}x_k\overline{x_1},\ \overline{x_i}x_j\overline{x_k}x_1Y_{ijk1} = \overline{x_i}x_j\overline{x_k}x_1,\ x_i\overline{x_j}x_kx_1Y_{ijk1} = x_i\overline{x_j}x_kx_1.$$

And we obtain a system of equations of the form $x_{i_1}^{\sigma_1}x_{i_2}^{\sigma_2}\ \&\ \ldots\ \&\ x_{i_k}^{\sigma_k} = 1$, for each of which we derive the solution $\overline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, where

$$\alpha_\gamma = \begin{cases} \sigma_\gamma, & if\ \gamma \in (i_1, i_2, \ldots, i_k), \\ *, & \text{otherwise.} \end{cases}$$

6. Description and Structure of the Program for Solving a Special Class System and a Test Example

RLSY Program: Solving logical equation systems.

The RLSY program is designed to solve systems of second-degree nonlinear Boolean equations of a special class. It is implemented in the Basic programming language. The block diagram of RLSY is shown in Figure 1.

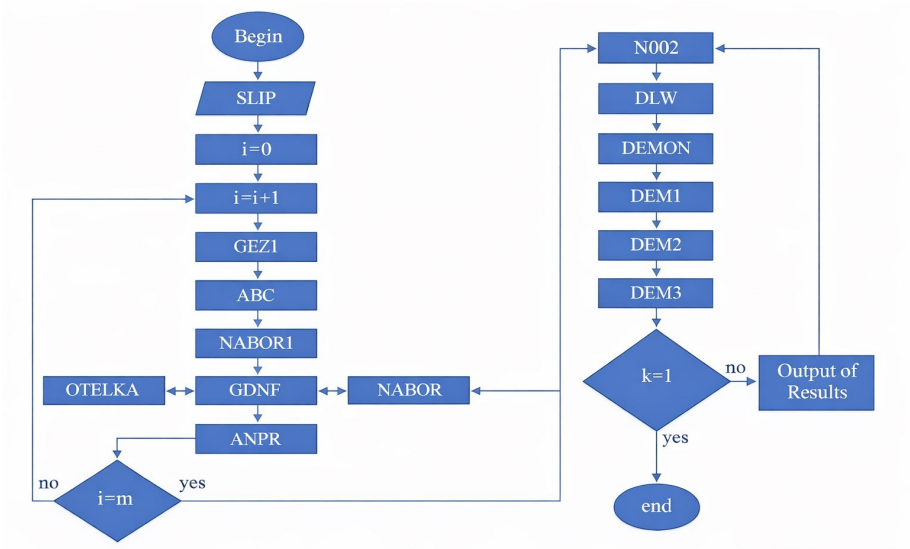

Figure 1. Algorithm for solving logical equation systems.

The program uses 14 procedures.

The SLIP procedure is a control procedure responsible for forming a dataset on a magnetic disk.

The GEZ1 (F, $Z1$, $N1$) procedure is designed for grouping elements of statements in system R and encoding logical expressions (groupings) into the $Z1$ array. Here,

- F represents a Zhegalkin polynomial of a special type (a statement of system R);
- $Z1$ is an array of encoded logical expressions;
- $N1$ is the number of structural components (s.k.).

In the $ABC(L, A)$ procedure,

- L is the number of structural components in the A array;
- A is an array of bit strings corresponding to structural components.

This procedure is used to transform structural components into bit-string arrays.

The procedure $NABOR(B2, N, K)$ forms the set of all sets $V^* = (V_1, V_2, \ldots, V_N)$, where $V_i^* \in \{0, 1\}$, $i = 1, 2, \ldots N$ such that $|V| = \sum_{i=1}^{m} V_i = K$, where

- $B2$ is the array of sets from the set FF^K;
- N is the number of bits in vectors V^* from FF^K;
- K is the number of ones in the coordinates.

This procedure generates all possible combinations of sets (or vectors), where the sum of the elements in each vector equals K, corresponding to the number of ones in the vector.

The $NABOR1(N1, BB, K1, TO)$ procedure is designed to determine the elements V^* from FF such that $V^* = (V_1, V_2, \ldots, V_n)$, where $V_1 = V_2 = \ldots = V_k = 1$; $V_{k+1} = \ldots = V_n = 0$, where

- N_1 is the number of c.c.;
- BB is a binary set;
- K_1 is the number of one-bit positions;
- TO is an auxiliary conditional parameter.

The $OTELKA(H1, BT1, N, AA, CA)$ procedure is used to perform transformations of the form: $\neg(U_1 \ \& \ U_2 \ \& \ \ldots \ \& \ U_m) = \neg U_1 \vee \neg U_2 \vee \ldots \vee \neg U_m$; $U(U_1 \vee \ldots \vee U_m) = UU_1 \vee \ldots \vee UU_m$, where $U, \ldots, U_m$ *c.c.*

Here, $H1$ represents the number of c.c.; $BT1$ is an array of bit strings for c.c.; N is the number of variables; AA and EA are arrays of bit strings corresponding to the DNF (disjunctive normal form) of the system's logical statements.

The $GDNF(B1, BN1, NN, B11, C11, N, T1, K)$ procedure transforms an array of bit strings of c.c. logical statements of the Zhegalkin polynomial into DNF:

- $B1$—a bit string of c.c. statements of the Zhegalkin polynomial;
- $BN1$—an array of elements;
- NN—the number of c.c. of the Zhegalkin polynomial at input and the number of c.c. in DNF at output;
- $B11, C11$—arrays of bit strings of c.c. in DNF;
- N—the number of variables;
- $T1$—an auxiliary conditional parameter;
- K—the number of unit coordinates.

The $ANPR(NN, B11, C11, AN, Z)$ procedure maps binary sets to an analytical representation and DNF for variable encoding:

- NN—the number of elementary conjunctions in the DNF of system FF;
- $B11, C11$—arrays of binary sets representing the form;
- AN—an elementary conjunction in analytical form;
- N—the number of variables;
- Z—an array of encoded functions.

Procedure $N002(NB, N, T, B)$ generates index vectors of complex conjunctions (c.k.) during the multiplication of the system and, using subsequent procedures, constructs sets of solutions for system R:

- NB—the number of equations in the system;
- N—the number of variables;
- T—an array containing the number of elementary conjunctions in the equations of system R;
- B—an array of complex conjunctions in system R.

Procedure DLW (NB, L, B, SCH, K7, N)

This procedure performs the multiplication of complex conjunctions according to the index vector:

- NB—an array defining the number of equations;
- L—the size of array NB;
- B—an array of elementary conjunctions in the equations;
- SCH—a conditional parameter;
- $K7$—the size of array B;
- N—the number of variables.

Procedure DEMON (N, N5, A, BB)

This procedure determines the structure of the resulting product of complex conjunctions:

- BB—a vector corresponding to the Zhegalkin polynomial in the product;
- A—a vector corresponding to the remaining part of the product BB;
- $N5$—the number of Zhegalkin polynomials in BB and the dimension of arrays A;
- N—the number of variables.

Procedure DEM1 (N, N5, A, BB)

This procedure compares the part of the product represented in the form of the Zhegalkin polynomial with the remaining part and simplifies the product in case of intersections. The parameters $N, N5, A, BB$ correspond to those in the *DEMON procedure*.

Procedure DEM2 (N, N5, A, BB)

This procedure forms sets of solutions corresponding to the remaining Zhegalkin polynomials. The parameters $N, N5, A, BB$ correspond to those in the *DEMON procedure*.

Procedure DEM3 (N5, A, B, K1)

This procedure compares the sets of solutions obtained in **DEM2** with the remaining part of the product and prints the resulting sets of solutions:

- $N5$—the number of Zhegalkin polynomials in the product;
- B—a vector corresponding to the Zhegalkin polynomial in the product;
- A—a vector corresponding to the remaining part of the product;
- $K1$—the dimension of vector A.

7. Instruction for the RLSY Program (Solution of Logical Systems of Equations)

The input data for the RLSY program (Resolution of Logical Systems of Equations) consists of the following:

- N—number of variables;
- M—number of equations;
- $FF(i)(i = 1, 2, \ldots, M)$—formulas of the system of logical equations of a special second-degree class.

Since the solution set generated during the program execution can be very large, it is recommended to publish them gradually rather than storing them in a dedicated array or to stop the program periodically, printing partial results.

The input parameters, the system of second-order Boolean equations of a special class $FF(i)(i = 1, 2, \ldots, M)$ and the sought solutions of the system are recorded as sets of size N, consisting of Z subsets. If a solution set is obtained, it corresponds to all possible solutions.

As a test example for the RLSY program (Solution of Logical Systems of Equations), we consider a system of equations where N = 10 and M = 20.

Test Example:

For simplicity and explanation, we take an analytical test example consisting of equations from a special class, divided into two groups, with 10 variables and $\alpha_i = 1$ $\left(i = \overline{1,8}\right)$.

By grouping elements on the left-hand sides of the equations, we introduce new variables:

1. Now, to illustrate the effectiveness of this method, we will demonstrate how much the complexity of the given system has been reduced using simplicity indices.

Let us introduce the following notation:

$[U \wedge \neg(\vee_{i=1}^{m} U_i)] \equiv 0$.—the number of variable letters appearing in the notation of the left-hand sides of the equations.

$[U \wedge \neg(\vee_{i=1}^{m} U_i)] \neq 0$.—the number of elementary conjunctions included in the equations.

$\tilde{\alpha} \in E_n^2$—the complexity of the system is measured using LB and LK:

From this, we obtain $\bigvee\limits_{i=1}^{m} U_i(\tilde{\alpha}) = 0$. The overall complexity difference between the systems is $R = L(R) - L(R') = 129 - 40 = 89$ (See Table 5).

Table 5. General difference in system complexity.

No.	$\boldsymbol{U(\tilde{\alpha}) = 1}$	$\boldsymbol{\neg(\vee_{i=1}^{m} U_i(\tilde{\alpha})) = 1}$	$\boldsymbol{U(\tilde{\alpha}) = 1}$	$\left.\begin{matrix} \boldsymbol{U(\tilde{x}, y(\tilde{x})) = 1} \\ \boldsymbol{U_1(\tilde{x}, y(\tilde{x})) = 0} \\ \boldsymbol{\dots \quad \dots \quad \dots} \\ \boldsymbol{U_m(\tilde{x}, y(\tilde{x})) = 0} \end{matrix}\right\}$	$\boldsymbol{\tilde{\alpha}}$	$\boldsymbol{U(\tilde{\alpha}) = 1}$
$i = 1$	11	7	18	3	2	5
$i = 2$	10	7	17	3	2	5
$i = 3$	10	7	17	3	2	5
$i = 4$	9	6	15	3	2	5
$i = 5$	9	6	15	3	2	5
$i = 6$	9	6	15	3	2	5
$i = 7$	10	7	17	3	2	5
$i = 8$	9	6	15	3	2	5
Total	77	52	129	24	16	40

Thus, using this method, the complexity of the system is reduced by more than three times.

2. Using the formula $\tilde{\alpha} \in E_n^2$, we transform the system R' into its disjunctive normal form (DNF).

Naturally, the DNF may contain opposing complex conjunctions (i.e., terms that equal zero), which are excluded from the DNF.

Based on the program's structure, the output can be a single solution, several solutions (k) as per the user's preference, or all possible solutions.

In our approach, we determine multiple solutions for the given system. To achieve this, we select one complex conjunction from each equation and apply the method of linear equation systems.

$$\begin{cases} \overline{Y_{14}} \cdot Y_{567} x_2 \cdot Y_{345} \cdot \overline{Y_{67910}} x_3 \cdot Y_{456} \cdot \overline{Y_{78910}} x_3 \cdot Y_{123} \cdot \overline{Y_{8910}} x_4 \cdot Y_{234} \cdot \overline{Y_{6810}} \\ x_3 \cdot Y_{456} \cdot \overline{Y_{789}} x_1 \cdot Y_{345} \cdot \overline{Y_{67910}} x_2 Y_{345} \overline{Y_{67910}} = 1 \end{cases}$$

From this, it is easy to understand that $x_1 = x_2 = x_3 = x_4 = 1$. They satisfy linear expressions at $x_1 = x_2 = x_3 = x_4 = 1$; $Y_{123} = Y_{234} = \overline{Y_{14}} = 1$. Therefore, we will exclude them from the complex conjunction, leaving only one from $Y345$ and one from $x_5 = 1$ and $\overline{Y_{67910}}$. From Y_{345}, it follows that $[U(\tilde{\alpha}) \to M(\tilde{\alpha})] \equiv 1$. Consequently, from Y_{456} we obtain

$x_6 = 1$, and further, we derive $Y_{567} - x_7 = 1$. These are then excluded from the system of constraints (s.c.). Thus, we obtain

$$\left.\begin{array}{c} \bar{Y}_{67910} = \bar{Y}_{910} = 1 \\ \bar{Y}_{78910} = Y_{8910} = 1 \\ \bar{Y}_{8910} = 1 \\ \bar{Y}_{6810} = Y_{810} = 1 \\ \bar{Y}_{789} = Y_{89} = 1 \\ \bar{Y}_{67910} = \bar{Y}_{910} = 1 \end{array}\right\}$$

From this, it is evident that the first and last elements are identical and equivalent, so we remove one of them from the system. The second and third elements are incompatible (as they correspond to the third and fourth equations in the original system).

Thus, by eliminating each of them, we obtain the solution to the system for the special class.

A
$$\left.\begin{array}{c} \bar{Y}_{67910} = \bar{Y}_{910} = 1 \\ \bar{Y}_{78910} = Y_{8910} = 1 \\ \bar{Y}_{6810} = Y_{810} = 1 \\ \bar{Y}_{789} = Y_{89} = 1 \end{array}\right\} \qquad \left.\begin{array}{c} x_9 \oplus x_{10} \oplus 1 = 1 \\ x_8 \oplus x_9 \oplus x_{10} = 1 \\ x_8 \oplus x_{10} = 1 \\ x_8 \oplus x_9 = 1 \end{array}\right\}$$

The solution to this system is $\tilde{\alpha} \in E_n^2$. Taking into account the previous variable roots, we obtain the general solution of the system without the fourth equation: $\{1, 1, 1, 1, 1, 1, 1, 1, 0, 0\}$.

B Thus, we obtain the general solution of the system without the third equation: $\{1, 1, 1, 1, 1, 1, 1, 0, 1, 1\}$.

8. Conclusions

This article explores various transformations of logical formulas when transitioning from one basis to another in the process of solving systems of logical equations and determining their solution roots. The necessary criteria for formula transformations during basis transitions are established in the form of theorems, which describe the statements of Boolean equation systems. A solution is proposed for systems of second-degree nonlinear Boolean equations, which serve as a logical model for algebraic cryptanalysis. Test examples of logical formula transformations are provided. Additionally, a solution is presented for systems of second-degree nonlinear Boolean equations of a special class. The practical significance of the research results lies in the ability to solve systems of special second-degree nonlinear Boolean equations based on the grouping and minimization of logical statements in the class of disjunctive normal forms (DNFs), which serve as an algebraic model of S-boxes in symmetric encryption algorithms.

Author Contributions: Conceptualization, A.K. and I.S.; methodology, I.S.; software, A.B. (Alimdzhan Babadzhanov); validation, A.K., A.B. (Abdussattar Baizhumanov) and A.B. (Akbarjon Babadjanov); formal analysis, I.S.; investigation, A.B. (Alimdzhan Babadzhanov) and I.S.; resources, A.K., A.B. (Abdussattar Baizhumanov) and A.B. (Akbarjon Babadjanov); data curation, A.B. (Alimdzhan Babadzhanov); writing—original draft preparation, I.S.; writing—review and editing, A.B. (Alimdzhan Babadzhanov), A.B. (Abdussattar Baizhumanov) and A.B. (Akbarjon Babadjanov); visualization, A.B. (Akbarjon Babadjanov); supervision, I.S.; project administration, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study.

Acknowledgments: The author of the article, Islambek Saymanov, expresses gratitude to the participants of the project "AL-9424104925-R1", implemented at the New Uzbekistan University, funded by the Agency for Innovative Development under the Ministry of Higher Education, Science and Innovation of the Republic of Uzbekistan, for their assistance in writing the article.

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

References

1. Chong, J.; Jiang, N.; Zhuo, Z; Zhang, W. Boolean Functions with Two Distinct Nega-Hadamard Coefficients. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2024**, *107*, 1603–1608. [CrossRef]
2. Carlet, C. *Boolean Functions for Cryptography and Coding Theory*; Cambridge University Press: Cambridge, UK, 2020.
3. Picek, S.; Carlet, C.; Guilley, S.; Miller, J.F.; Jakobovic, D. Evolutionary Algorithms for Boolean Functions in Diverse Domains of Cryptography. *Evol. Comput.* **2016**, *24*, 667–694. [CrossRef]
4. Garcia, F.; Simmons, H.; Kumar, S. Classification of Boolean Functions. *Discret. Appl. Math.* **2022**, *290*, 9–23.
5. Bushnell, M.L.; Agrawal, V.D. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*; Kluwer Academic Publishers: Boston, MA, USA, 2000.
6. O'Donnell, R. *Analysis of Boolean Functions*; Cambridge University Press: New York, NY, USA, 2014. [CrossRef]
7. Lee, S.-Y.; Riener, H.; Mishchenko, A.; Brayton, R.K.; De Micheli, G. A Simulation-Guided Paradigm for Logic Synthesis and Verification. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 2573–2586. [CrossRef]
8. Wang, C.; Agarwal, R.P.; Regan, D.O.; Sakthivel, R. *Theory of Translation Closedness for Time Scales*; Springer Nature: New York, NY, USA, 2020. [CrossRef]
9. Ahmed, M.; Baker, T.; Patel, S. On the Cryptographic Properties of Boolean Functions. *J. Cryptogr. Eng.* **2022**, *12*, 231–245.
10. Muhiddinov, M.; Ochilov, N.; Umurkulov, B.; Kholnazarov, U.; Sapaev, I.; Bazarova, N.; Kholova, M.; Aripova, G. Privacy-Aware Information Security for E-Learning Platforms in History Using Attribute-Based Encryption Algorithm. *J. Internet Serv. Inf. Secur.* **2025**, *15*, 305–315. [CrossRef]
11. Xia, B.; Mantegh, I.; Xie, W.-F. Hybrid Framework for UAV Motion Planning and Obstacle Avoidance: Integrating Deep Reinforcement Learning with Fuzzy Logic. In Proceedings of the 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT), Valletta, Malta, 1–4 July 2024 ; Volume 9, pp. 2662–2669. [CrossRef]
12. Bakhronova, D.; Narziyeva, M.; Yuldosheva, N.; Zayniyeva, U.; Yusupov, J.; Uralov, B.; Sapaev, I.; Khikmatov, N. Intelligent Information Security System for Language and History Education Using Machine Learning-based Intrusion Detection Algorithm. *J. Internet Serv. Inf. Secur.* **2025**, *15*, 520–529. [CrossRef]
13. Kabulov, A.; Saymanov, I.; Yarashov, I. Muxammadiev, F. Algorithmic method of security of the Internet of Things based on steganographic coding. In Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 21–24 April 2021; pp. 1–5. [CrossRef]
14. Cusick, T.W.; Stănică, P. *Cryptographic Boolean Functions and Applications*; Academic Press: San Diego, CA, USA, 2017.
15. Srinivasan, C.; Lakshmy, K.V.; Sethumadhavan, M. Complexity measures of cryptographically secure boolean functions. In *Cyber Security Cyber Crime and Cyber Forensics Applications and Perspectives*; IGI Global: Hershey, PA, USA, 2011; pp. 220–230. [CrossRef]
16. Rosen, K.H. *Discrete Mathematics and Its Applications*, 8th ed; McGraw-Hill: New York, NY, USA, 2019; p. 1120.
17. Zheng, J.; Security analysis of Boolean algebra based on Zhang-Wang digital signature scheme. *AIP Conf. Proc.* **2014**, *1618*, 507–509. [CrossRef]
18. Kim, J.; Nguyen, F.; Lee, S. Boolean Functions in Cryptography. *IEEE Trans. Inf. Theory* **2024**, *70*, 310–325.
19. Green, R.; Ahmed, L.; Qureshi, N. Circuit Realizations of Boolean Functions. *J. Log. Comput.* **2021**, *31*, 222–236.
20. Jackson, T.; Velasquez, M.; Dutta, R. Spectral Properties of Boolean Functions. *Comput. Complex.* **2022**, *29*, 1065–1079.
21. Kim, D.; Nair, V.; Zhang, X. Boolean Functions in Quantum Computing. *Quantum Comput. Quantum Inf.* **2022**, *33*, 12–29.
22. Jukna, S. *Boolean Function Complexity: Advances and Frontiers*; Springer: Berlin, Germany, 2012. [CrossRef]
23. Ballani F. The surface pair correlation function for stationary Boolean models. *Adv. Appl. Probab.* **2007** *39*, 1–15. [CrossRef]
24. Wegener, I. *The Complexity of Boolean Functions*; John Wiley & Sons: Chichester, UK, 1987.
25. López-López, I.; Sosa-Gómez, G.; Segura, C.; Oliva, D.; Rojas, O. Metaheuristics in the Optimization of Cryptographic Boolean Functions. *Entropy* **2020**, *22*, 1052. [CrossRef]
26. Bonich, T.A.; Panferov, M.A.; Tokareva, N.N. On the number of ℓ-suitable Boolean functions in constructions of filter and combining models of stream ciphers. *Prikl. Diskretn. Mat.* **2023**, *62*, 1211–1216. [CrossRef]
27. Sun, Q.; Wei, Sh.; Saymanov, I.; Lu, Y. Deng, W.; Lou, J. A Mechanical–Electrical Damage Model for Performance Analysis of Crack-based Strain Sensor. *Int. J. Appl. Mech. Source Preview* **2025**, *18*, 2550124. [CrossRef]
28. Sun, Z.; Ambrosi, E.; Bricalli, A.; Ielmini, D. Logic Computing with Stateful Neural Networks of Resistive Switches. *Adv. Mater.* **2018**, *30*, 1802554. [CrossRef]

29. Stallings, W. *Cryptography and Network Security: Principles and Practice*, 7th ed.; Pearson Prentice Hall: Boston, MA, USA, 2017.
30. Mesnager, S. *Bent Functions: Fundamentals and Results*; Springer: Cham, Switzerland, 2016. [CrossRef]
31. Brayton, R.K.; Hachtel, G.D.; McMullen, C.T.; Sangiovanni-Vincentelli, A.L. *Logic Minimization Algorithms for VLSI Synthesis*; Kluwer Academic Publishers: Boston, MA, USA, 1984. [CrossRef]
32. Yang, Y.; Zhang, L.; An, L. Fixed-Time Adaptive Fault-Tolerant Control for Uncertain Nonlinear Systems with Actuator Faults. *Appl. Math. Comput.* **2025**, *516*, 129867–129867. [CrossRef]
33. Kuzmin, A.S.; Markov, V.T.; A.A. Nechaev; A.S. Neljubin. A Generalization of the Binary Preparata Code. *Discret. Appl. Math.* **2005**, *154*, 337–345. [CrossRef]
34. Hiep, X.H.; Bao, H.L.; Hung V.C.L.; Tam, T.L.; Nghia, D.-T. Design of an IoT ultrasonic-vision based system for automatic fruit sorting utilizing size and color. *Internet Things* **2024**, *25*, 101017. [CrossRef]
35. Tran, T.C.T.; Phan, L.P.; Huynh, H.X. Approach of Item-Based Collaborative Filtering Recommendation Using Energy Distance. *J. Adv. Inf. Technol.* **2024**, *15*, 10–16. [CrossRef]
36. Shukla, S.; Hussain, Sh.; Irshad, R.R.; Alattab, A.A.; Thakur, S.; Breslin, J.G.; Hassan, M.F.; Abimannan, S.; Husain, Sh.; Jameel, S.M. Network analysis in a peer-to-peer energy trading model using blockchain and machine learning. *Comput. Stand. Interfaces* **2024**, *88*, 103799. [CrossRef]
37. Chang, Y.-S.; Huang, S.-T.; Haobijam, B.; Abimannan, S.; Kushida, T. Marine ecological information prediction by using adjacent location spatiotemporal deep learning model with ensemble learning techniques. *Ecol. Inform.* **2025**, *85*, 102964. [CrossRef]
38. Gotarane, V.; Abimannan, S.; Hussain, S.; Irshad, R.R. A Hybrid Framework Leveraging Whale Optimization and Deep Learning With Trust-Index for Attack Identification in IoT Networks. *IEEE Access* **2024**, *12*, 36296–36310. [CrossRef]
39. She, Y.; Hong, Y.; Shen, S.; Yang, B.; Zhang, L.; Wang, J. Consistency regularization for few shot multivariate time series forecasting. *Sci. Rep.* **2025**, *15*, 14195. [CrossRef]
40. Makhmudov, F.; Privalov, A.; Egorenkov, S.; Pryadkin, A.; Kutlimuratov, A.; Bekbaev, G.; Cho, Y.I. Analytical Approach to UAV Cargo Delivery Processes Under Malicious Interference Conditions. *Mathematics* **2025**, *13*, 2008. [CrossRef]
41. Kabulov, A.; Baizhumanov, A.; Saymanov, I. Synthesis of Optimal Correction Functions in the Class of Disjunctive Normal Forms. *Mathematics* **2024**, *12*, 2120. [CrossRef]
42. Saymanov, I. Logical automatic implementation of steganographic coding algorithms. *J. Math. Mech. Comput. Sci.* **2024**, *121*, 122–131. [CrossRef]
43. Kabulov, A.; Normatov, I.; Saymanov, I.; Baizhumanov, A. On the Completeness of Classes of Correcting Functions of Heuristic Algorithms. *Azerbaijan J. Math.* **2025**, *15*, 51–64. [CrossRef]
44. *GOST R 34.12-2015*; Information Technology. Cryptographic Data Security. Block Ciphers. Technical Committee for Standardization "Cryptography and Security Mechanisms" (TC 26): Moscow, Russia, 2015.