

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук
Кафедра компьютерной безопасности

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Руководитель ООП

канд. техн. наук, доцент

 В.Н. Тренькаев
« 17 » 01 2022 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА
(ДИПЛОМНАЯ РАБОТА)

АНАЛИЗ ВЕКТОРНЫХ БУЛЕВЫХ ФУНКЦИЙ С КООРДИНАТАМИ
ОГРАНИЧЕННОЙ СЛОЖНОСТИ

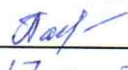
по специальности 10.05.01 Компьютерная безопасность,
специализация (профиль) «Анализ безопасности компьютерных систем»

Петров Георгий Константинович

Научный руководитель ВКР

канд. физ.-мат. наук, доцент

зав. лаб. комп. криптографии

 И.А. Панкратова
« 17 » 2022 г.

Автор работы

студент группы № 1165

 Г.К. Петров
« 17 » 2022 г.

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ
Руководитель ООП
канд. техн. наук, доцент

 В.Н. Тренькаев
подпись
« 07 » 10 2021 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы специалиста обучающегося
Петрова Георгия Константиновича
по специальности 10.05.01 Компьютерная безопасность, специализация (профиль)
«Анализ безопасности компьютерных систем»

1 Тема выпускной квалификационной работы

Анализ векторных булевых функций с координатами ограниченной сложности

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:

а) в учебный офис / деканат – 17 января 2022 г. б) в ГЭК – 28 января 2022 г.

3 Исходные данные к работе:

Объект исследования – криптосистема на векторных булевых функциях.

Предмет исследования – свойства векторных функций, их задание и вычисление значения

Цель исследования – исследование способов задания векторных булевых функций и вычисления их значения и разработка алгоритма вычисления их нелинейности

Задачи:

Разработать, реализовать и сравнить способы задания векторных булевых функций ограниченной сложности, способы шифрования сообщения с заданным параметром и алгоритм вычисления нелинейности векторной булевой функции.

Методы исследования:

теоретический и экспериментальный на базе ЭВМ

Организация или отрасль, по тематике которой выполняется работа, –

Лаборатория компьютерной криптографии ТГУ.

4 Краткое содержание работы

Рассмотреть способы задания векторных функций, сравнить их по памяти и скорости вычисления, способы шифрования сообщения с заданным параметром и их исследование, алгоритм вычисления нелинейности векторной булевой функции, проанализировать результаты экспериментов.

Научный руководитель выпускной

квалификационной работы

доцент, канд. физ.-мат. наук

зав. лаб. комп. криптографии

Задание принял к исполнению

студент группы № 1165

 / И.А. Панкратова

 / Г.К. Петров

АННОТАЦИЯ

Дипломная работа содержит 39 страниц, 1 рисунок, 9 таблиц, 6 литературных источников и 2 приложения.

ВЕКТОРНЫЕ БУЛЕВЫ ФУНКЦИИ, ОБРАТИМЫЕ ФУНКЦИИ, ФУНКЦИИ ОГРАНИЧЕННОЙ СЛОЖНОСТИ, НЕЛИНЕЙНОСТЬ.

Объект исследования: криптосистема на векторных булевых функциях.

Цель работы: исследование способов задания векторных булевых функций и вычисления их значения, и разработка алгоритма вычисления их нелинейности.

Метод исследования: теоретический и экспериментальный на базе ЭВМ.

Результат: Предложены и реализованы способы задания векторных булевых функций ограниченной сложности, проведено их сравнение по объему занимаемой памяти и скорости вычисления значения.

Предложены и реализованы способы шифрования сообщения с заданным параметром при функции ограниченной сложности в качестве порождающей, проведено их сравнение по скорости шифрования и получены некоторые рекомендации.

Разработан и реализован алгоритм вычисления нелинейности векторной булевой функции ограниченной сложности.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1. Математические определения	7
2. Способы задания функций ограниченной сложности	9
2.1. Векторный	9
2.2. Конструкция «каменщик»	11
3. Способы шифрования сообщения с заданным параметром	12
3.1. Без перестроения G	12
3.2. С перестроением G	13
4. Алгоритм вычисления нелинейности векторной булевой функции	15
4.1. Предпосылки алгоритма	15
4.2. Алгоритм вычисления нелинейности функции $F \in B_{n,k}$	16
5. Экспериментальные данные	19
5.1. Среднее значение нелинейности	19
5.1.1. Для произвольной функции	19
5.1.2. Для обратимой функции	20
5.2. Время работы программы	22
6. Реализация	23
ЗАКЛЮЧЕНИЕ	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ	30
Приложение А Программный код вычисления нелинейности векторной булевой функции	31
Приложение Б Программный код шифрования сообщения двумя способами	36

ВВЕДЕНИЕ

Большое количество современных криптосистем в качестве главных составляющих используют векторные булевы функции. Но представление таких функций в памяти – трудоемкая задача. Поэтому в работе рассматриваются функции, координаты которых существенно зависят от ограниченного числа переменных, и способы их задания.

Также, в целях обеспечения безопасности информации необходимо, чтобы эти функции обладали определенными свойствами и более того, чтобы эти свойства могли препятствовать различным методам криптоанализа.

Одним из таких свойств является нелинейность векторной булевой функции. При достаточно большом значении она дает устойчивость к быстрой корреляционной атаке на поточные шифры и линейному криптоанализу блочных шифров [1]. Не менее важной является скорость вычисления нелинейности. Для достижения ее более высоких значений необходимо наличие соответствующего алгоритма.

В данной работе рассматривается криптосистема на векторных булевых функциях ограниченной сложности. Цель работы – исследование способов задания векторных булевых функций и вычисления их значения, и разработка алгоритма вычисления их нелинейности.

Для достижения данной цели были поставлены следующие задачи:

- 1) Разработать, реализовать и сравнить способы задания векторных булевых функций.
- 2) Разработать, реализовать и сравнить способы вычисления значения ключевой функции при функции ограниченной сложности в качестве порождающей.
- 3) Разработать и реализовать алгоритм вычисления нелинейности векторной булевой функции.

Работа состоит из 6 глав. В главе 1 приведены основные определения. В главе 2 описаны основные способы задания векторной булевой функции от

ограниченного числа переменных. В главе 3 описаны способы вычисления значения векторной функции при функции ограниченной сложности в качестве порождающей. В главе 4 описан алгоритм вычисления нелинейности векторной булевой функции и предпосылки к нему. В главе 5 приведены результаты экспериментов: сравнение среднего значения нелинейности произвольных и обратимых функций и время работы программы для разных значений. В главе 6 описана программная реализация полученного в ходе работы алгоритма.

1. Математические определения

Определение 1. Булевой функцией от n переменных называется функция

$$f: \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2.$$

Определение 2. Векторной булевой функцией от n переменных называется упорядоченный набор функций $F(x) = (f_1(x), \dots, f_m(x))$, где $x = (x_1 \dots x_n)$ и $f_i(x)$ – булева функция, $i = 1, \dots, m$:

$$F: \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m.$$

Будем обозначать класс векторных булевых функций $V_{n,m}$.

Определение 3. Булева функция $f(x_1 \dots x_i \dots x_n)$ существенно зависит от переменной x_i , если существует такой набор $a_1 \dots a_{i-1} a_{i+1} \dots a_n$, что выполняется неравенство:

$$f(a_1 \dots a_{i-1} 0, a_{i+1}, \dots, a_n) \neq f(a_1 \dots a_{i-1} 1, a_{i+1}, \dots, a_n).$$

В этом случае также говорят, что переменная x_i существенная.

Определение 4. Будем называть векторную булеву функцию $F(x) = (f_1(x), \dots, f_m(x))$ функцией ограниченной сложности, если каждая её координата f_i существенно зависит не более чем от k переменных, где $k < n$. Будем рассматривать случай, когда $m=n$. Обозначим $C_{n, \leq k}$ – класс функций, все координаты которых существенно зависят не более чем от k переменных.

Определение 5. Для $x = (x_1, \dots, x_n)$ и набора индексов $i_1, \dots, i_k$, $k \leq n$, $1 \leq i_1 < \dots < i_k \leq n$, будем называть проекцией вектора x вектор $a = (x_{i_1}, \dots, x_{i_k})$.

Определение 6. $x^\sigma = (x_1^{\sigma_1}, \dots, x_n^{\sigma_n})$ будем называть инверсией вектора x , где $x_i^{\sigma_i} = x_i$, если $\sigma_i = 1$, и $x_i^{\sigma_i} = \neg x_i$, если $\sigma_i = 0$, для всех $i = 1, \dots, n$.

Определение 7. Компонентой функции F называется булева функция

$$vF = v_1 f_1 \oplus \dots \oplus v_n f_n,$$

где $v = v_1 \dots v_n \in \mathbb{Z}_2^n \setminus \{0^n\}$; 0^n – нулевой вектор длины n .

Определение 8. Скалярным произведением векторов $a, x \in \mathbb{Z}_2^n$ называется значение

$$(a, x) = a_1 x_1 \oplus \dots \oplus a_n x_n.$$

Функция степени 0 или 1 называется *аффинной*. Множество всех аффинных функций от n переменных обозначается $A(n)$:

$$A(n) = \{a_0 \oplus (a, x) : a_0 \in \mathbb{Z}_2, a \in \mathbb{Z}_2^n\}.$$

Определение 9. *Преобразование Уолша-Адамара* булевой функции f называется функция

$$\hat{f} : \mathbb{Z}_2^n \rightarrow \mathbb{Z},$$

где для каждого $a \in \mathbb{Z}_2^n$

$$\hat{f}(a) = \sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x) \oplus (a, x)}.$$

Определение 10. *Нелинейностью* функции f называется расстояние от f до класса аффинных функций. Нелинейность можно вычислить по формуле

$$N_f = 2^{n-1} - \frac{1}{2} \max_{a \in \mathbb{Z}_2^n} |\hat{f}(a)|.$$

Определение 11. Нелинейностью векторной булевой функции $F \in V_{n,m}$ называется целое число $N_F = \min_{v \neq 0^n} N_{vF}$.

Определение 12. Функция $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ называется *бент-функцией*, если для любого $a \in \mathbb{Z}_2^n$ $\hat{f}(a) = \pm 2^{n/2}$.

2. Способы задания функций ограниченной сложности

Любую векторную булеву функцию можно задать таблицей 2^n х m бит. Пример функции для $n=2$ и $m=4$ приведен в таблице 1.

Таблица 1 – Функция $F: \mathbb{Z}_2^2 \rightarrow \mathbb{Z}_2^4$

x_1x_2	$f_1f_2f_3f_4$
00	0010
01	0111
10	0110
11	1011

Вычисление заданной таким способом функции требует одного обращения к памяти. Это общий способ задания, который не учитывает ограниченную сложность. Опишем каждый из реализованных способов задания функций класса $C_{n,\leq k}$.

2.1. Векторный

Функцию $F \in C_{n,\leq k}$, $F = (f_1, \dots, f_n)$, будем задавать парой $F = (G, M)$, где $G \in V_{k,n}$, $G = (g_1, \dots, g_n)$ – ядро функции F , $M_{n \times n} = \|a_{ij}\|$, $a_{ij} = 1$, если и только если f_i существенно зависит от x_j , то есть $f_i(x_1, \dots, x_n) = g_i(x_{j_1}, \dots, x_{j_k})$, где $\{j_1, \dots, j_k\} = \{j: a_{ij} = 1\}$. В нашем случае G задана таблицей – булевой матрицей размера 2^k х n .

Пример: пусть $n=4$, $k=2$ и G хранит 4 вектора размера n (таблица 2).

Таблица 2 – Функция $F = (G, M)$

x_1x_2	$g_1g_2g_3g_4$
00	0010
01	0111
10	0110
11	1011

Зададим векторы для каждой f_i , которые и будут являться строками матрицы M :

для f_1 : 1100,

для f_2 : 1010,

для f_3 : 0101,

для f_4 : 0011.

Тогда $F = (g_1(x_1, x_2), g_2(x_1, x_3), g_3(x_2, x_4), g_4(x_3, x_4))$. Вычисление функции $F(x)$ происходит в 2 этапа. Пусть $x = 1010$.

1. Строим проекции a_i для всех f_i :

$$a_1 = (x_1, x_2) = 10,$$

$$a_2 = (x_1, x_3) = 11,$$

$$a_3 = (x_2, x_4) = 00,$$

$$a_4 = (x_3, x_4) = 10.$$

2. Вычисляем все f_i :

$$f_1(x) = g_1(a_1) = 0,$$

$$f_2(x) = g_2(a_2) = 0,$$

$$f_3(x) = g_3(a_3) = 1,$$

$$f_4(x) = g_4(a_4) = 0.$$

В итоге получаем $F(x) = (f_1(x), f_2(x), f_3(x), f_4(x)) = 0010$. Таким образом, понадобится всего лишь $(2^k + n)n$ бит памяти. Посчитаем сложность вычисления $F(x)$ при данном способе задания. Для каждой

координаты строим проекцию, перебирая k единиц в соответствующей строке матрицы M . Так как имеем n координат, получаем сложность вычисления $O(n * k)$.

2.2. Конструкция «каменщик»

Частным и широко используемым на практике случаем является конструкция «каменщик» [2].

Пусть $k|n$, $n=ks$, $F_1, \dots, F_s \in V_{k,k}$, тогда $F = F_1|F_2|\dots|F_s \in V_{n,n}$,

$$F = (f_1(x_1, \dots, x_k), \dots, f_k(x_1, \dots, x_k), \dots, f_{n-k+1}(x_{n-k+1}, \dots, x_n), \dots, f_n(x_{n-k+1}, \dots, x_n)),$$

где $(f_1, \dots, f_k)$ обозначим F_1 , ..., $(f_{n-k+1}, \dots, f_n)$ — F_s (рисунок 1).

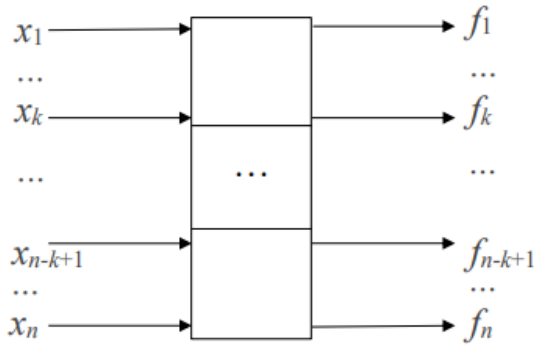


Рисунок 1 – Конструкция «каменщик»

Обозначим класс таких функций $B_{n,k}$. Для них матрицу M можно не хранить (так как она всегда одинаковая). В этом способе проекция будет строиться по номеру координатной функции.

Рассмотрим пример из п. 2.1. Единственным отличием здесь является построение матрицы M :

для f_1 : 1100,

для f_2 : 1100,

для f_3 : 0011,

для f_4 : 0011.

Построение матрицы будет одинаковым для всех функций рассматриваемого класса. Вычисление $F(x)$ будет аналогичным. По памяти данный способ занимает $2^k n$ бит. Сложность вычисления — $O(n)$.

3. Способы шифрования сообщения с заданным параметром

Векторная булева функция, заданная некоторым способом, может являться элементом ключевого пространства некоторого шифра. Поэтому требуется найти способ наиболее эффективного вычисления её значения.

Пусть у нас имеется криптосистема [3] с порождающей функцией $F: \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^n$. Пусть также задан некоторый параметр a , который включает в себя две инверсии – (σ_1, σ_2) и две перестановки – (π_1, π_2) . Шифрованием любого сообщения x будет являться функция $H = F^a(x) = \pi_2(g^{\sigma_2}(\pi_1(x^{\sigma_1})))$, где $F = (G, M)$.

Будем считать, что параметр a — фиксированный (или редко заменяющийся), а количество сообщений x достаточно велико (так как для малого количества дальнейшее разбиение на варианты не имеет смысла). Тогда возникает два возможных варианта шифрования.

3.1. Без перестроения G

В качестве примера рассмотрим векторный способ задания функции. Пусть $k=2$, $\sigma_1=1100$, $\sigma_2=0111$, $\pi_1=(2,4,3,1)$, $\pi_2=(4,3,1,2)$, G (таблица 3).

Таблица 3 – Функция G

x_1x_2	$g_1g_2g_3g_4$
00	0010
01	0111
10	0110
11	1011

Зададим вектора для каждого f_i : для f_1 : 1001, для f_2 : 1010, для f_3 : 1100, для f_4 : 0101. Пусть $x=1110$, зашифруем его:

$$a) x^{\sigma_1} = 1110^{1100} = 1110 \oplus 0011 = 1101$$

$$b) \pi_1(x^{\sigma_1}) = \pi_1(1101) = 1101$$

$$c) g(\pi_1(x^{\sigma_1})) = 1010$$

$$d) g^{\sigma_2}(\pi_1(x^{\sigma_1})) = \pi_2(0010) = 0100$$

$$e) \pi_2(g^{\sigma_2}(\pi_1(x^{\sigma_1}))) = \pi_2(0010) = 0100.$$

Таким образом, $F^a(x) = 0100$.

3.2.С перестроением G

Перед шифрованием сообщений преобразуем таблицу G в $G' = \pi_2(G^{\sigma_2})$. Далее, все поступающие на вход сообщения x будут шифроваться следующим образом:

$$H = G'(\pi_1(x^{\sigma_1})).$$

Рассмотрим пример: пусть $k=2$, $\sigma_1=1100$, $\sigma_2=0111$, $\pi_1=(2,4,3,1)$, $\pi_2=(4,3,1,2)$, G (таблица 4).

Таблица 4 – Функция G

x_1x_2	$g_1g_2g_3g_4$
00	0101
01	1001
10	0000
11	1110

Зададим вектора для каждого f_i : для f_1 1001, для f_2 1010, для f_3 1100, для f_4 0101. Преобразуем G в $\pi_2(G^{\sigma_2})$ (таблица 5).

Так как при преобразовании G в G' координаты векторной функции изменили свой порядок ($f_4f_3f_1f_2$), то соответственно вектора существенной зависимости тоже должны изменить свой порядок, чтобы остаться

закрепленными за своей координатой: 1001 для f_1 , 1010 для f_2 , 1100 для f_3 , 0101 для f_4 .

Таблица 5 – Функции G^{σ_2} и $\pi_2(G^{\sigma_2})$

x_1x_2	G^{σ_2}	$\pi_2(G^{\sigma_2})$
00	1101	1011
01	0001	1000
10	1000	0010
11	0110	0101

Пусть $x=1110$, зашифруем его:

- $x^{\sigma_1} = 1110^{1100} = 1110 \oplus 0011 = 1101$
- $\pi_1(x^{\sigma_1}) = \pi_1(1101) = 1101$
- $G'(\pi_1(x^{\sigma_1})) = 0100$.

Таким образом, $F^a(x) = 0100$.

В итоге, возникает задача нахождения эффективного способа шифрования. В ходе эксперимента для малых значений ($n \leq 31$) были получены следующие оценки:

1. при $k \leq n/2$, выгоднее преобразование таблицы G в G' ;
2. при $n/2 < k$, скорость шифрования одинаковая;
3. однако, при приближении значения k к n , перестроение перестает иметь смысл, так как размер таблицы G становится слишком велик.

4. Алгоритм вычисления нелинейности векторной булевой функции

4.1. Предпосылки алгоритма

Пусть заданы булевы функции $f(x_1, \dots, x_n)$ и $g(y_1, \dots, y_m)$, множества аргументов которых не пересекаются. В [2] приведена следующая формула вычисления нелинейности функции $f \oplus g$:

$$N_{f \oplus g} = 2^{n+m-1} - \frac{1}{2}(2^n - 2N_f)(2^m - 2N_g).$$

Преобразуем это выражение:

$$\begin{aligned} N_{f \oplus g} &= 2^{n+m-1} - \frac{1}{2}(2^n - 2N_f)(2^m - 2N_g) = 2^{n+m-1} - (2^{n+m-1} - 2^n N_g - \\ &\quad - 2^m N_f + 2N_f N_g) = 2^m N_f + 2^n N_g - 2N_f N_g. \end{aligned}$$

Получили:

$$N_{f \oplus g} = 2^m N_f + 2^n N_g - 2N_f N_g. \quad (1)$$

Утверждение. Пусть

1. $f \in \{vF: v \in \{0,1\}^n \setminus 0^n\}$, $g \in \{uG: u \in \{0,1\}^m \setminus \{0^m\}\}$.
2. $h \in \{aF \oplus bG: a \in \{0,1\}^n, b \in \{0,1\}^m, a \neq 0^n \vee b \neq 0^m\}$.

Тогда:

$$\min_{a,b} N_h = 2^m \min_v N_f + 2^n \min_u N_g - \min_v N_f \min_u N_g. \quad (2)$$

Доказательство.

Пусть выполняются условия 1) и 2). Фиксируя некоторое значение N_f , из (1) получаем:

$$N_{f \oplus g} = 2^m N_f + 2^n N_g - 2N_f N_g = \underbrace{2^m N_f}_{C_1} + N_g \underbrace{(2^n - 2N_f)}_{C_2}.$$

Здесь C_1 и C_2 – константы. Из определения 10 и того, что не все коэффициенты Уолша-Адамара равны нулю, следует, что $N_f < 2^{n-1} \Rightarrow C_2 > 0 \Rightarrow$ при уменьшении N_g уменьшается $N_{f \oplus g}$. Аналогичное доказательство при фиксации N_g . Из вышеприведенного доказательства следует (2).

Выведем формулу вычисления нелинейности для функций из класса

$B_{n,k}$. Пусть $n=2k$, $F_1 \in V_{k,k}$, $F_2 \in V_{k,k}$ и $F_1|F_2 \in V_{n,n}$. Покажем, что

$$N_{F_1|F_2} = 2^k \min\{N_{F_1}, N_{F_2}\}. \quad (3)$$

Из [1, следствие 22] получаем:

$$\begin{aligned} N_{F_1|F_2} &= 2^{k+k-1} - \frac{1}{2} \max\{2^k(2^k - 2N_{F_2}), 2^k(2^k - 2N_{F_1})\} \\ &= 2^{2k-1} - 2^{k-1}(2^k - 2 \min\{N_{F_1}, N_{F_2}\}) = 2^k \min\{N_{F_1}, N_{F_2}\} \end{aligned}$$

Нетрудно показать, что, при $n=ks$, из (3) следует

$$N_{F_1|...|F_s} = 2^{n-k} \min\{N_{F_1}, \dots, N_{F_s}\}. \quad (4)$$

4.2. Алгоритм вычисления нелинейности функции $F \in B_{n,k}$

Пусть $k|n$, $n=ks$, $F_1, \dots, F_s \in V_{k,k}$, тогда $F=F_1 | F_2 | \dots | F_s \in V_{n,n}$,

$$F=(f_1(x_1, \dots, x_k), \dots, f_k(x_1, \dots, x_k), \dots, f_{n-k+1}(x_{n-k+1}, \dots, x_n), \dots, f_n(x_{n-k+1}, \dots, x_n)),$$

где $F_1 = (f_1, \dots, f_k), \dots, F_s = (f_{n-k+1}, \dots, f_n)$.

Аналитическое описание алгоритма:

1. Пусть $F=(f_1(x_1, \dots, x_k), \dots, f_k(x_1, \dots, x_k), \dots, f_{n-k+1}(x_{n-k+1}, \dots, x_n), \dots, f_n(x_{n-k+1}, \dots, x_n))$, $n=ks$.
2. Разобьем F на блоки: $\underbrace{f_1 \dots f_k}_{1 \text{ блок}} \underbrace{f_{k+1} \dots f_{2k}}_{2 \text{ блок}} \dots \underbrace{f_{n-k+1} \dots f_n}_{s \text{ блок}}$.
3. Создадим массив D из s элементов. Найдем D_i для $i=1, \dots, s$, где D_i хранит минимальное значение нелинейности среди компонент функции F_i . Если $D_i = 0$, то $N_F = 0$ и переход на 5.
4. $N_F = 2^{n-k} \min_i D_i$.
5. Выход. Ответ: N_F .

Алгоритм вычисления нелинейности векторной булевой функции ограниченной сложности:

Вход: $F=F_1 | F_2 | \dots | F_s$, $n=ks$.

Выход: N_F – нелинейность F .

1. $N_F = 2^{n-1}$.
2. Если $s=1$, то

$$N_F = \min_{v \neq 0^k} N_{vF},$$

где $vF = \bigoplus_{i=1}^n v_i f_i$ и переход на 4.

3. Создаем массив D размера s и для $i=1, \dots, s$:

- $D_i = \min_{v \neq 0^k} N_{vG}$, где $G = (f_{i*k+1}, \dots, f_{(i+1)*k})$.
- Если $D_i = 0$, то $N_F = 0$ и переход на 4.
- Если $2^{n-k} D_i < N_F$, то $N_F = 2^{n-k} D_i$.

4. Выход. Ответ: N_F .

В шагах 2 и 3а перебор компонент производится в соответствии с кодом Грея [4].

Вычислим сложность данного алгоритма. Если $k < n$, то при подсчете количества операций участвуют все шаги кроме второго. В шаге 3 строится 2^k-1 компонент для каждого из s блоков. Для каждой компоненты вычисляется ПУА $(2^k k)$ и осуществляется поиск максимального коэффициента (2^k) . Следовательно, сложность алгоритма – $O(2^{2k} n)$.

При $k=n$, получаем сложность вычисления по определению – $O(2^{2n} n)$.

Приведем пример для функции F , заданной таблицей 6.

1. Разобьем F на блоки. Первый блок – $f_1 f_2 f_3$, второй – $f_4 f_5 f_6$, $N_F = 32$.

$$2. N_{f_1} = 1$$

$$N_{f_4} = 1$$

$$N_{f_1 \oplus f_2} = 2$$

$$N_{f_4 \oplus f_5} = 1$$

$$N_{f_2} = 1$$

$$N_{f_5} = 2$$

$$N_{f_2 \oplus f_3} = 2$$

$$N_{f_5 \oplus f_6} = 1$$

$$N_{f_1 \oplus f_2 \oplus f_3} = 1$$

$$N_{f_4 \oplus f_5 \oplus f_6} = 2$$

$$N_{f_1 \oplus f_3} = 2$$

$$N_{f_4 \oplus f_6} = 2$$

$$N_{f_3} = 1$$

$$N_{f_6} = 1$$

$$D_1=1, D_2=1 \Rightarrow N_F = 8.$$

Таблица 6 – Функция G

$x_1x_2x_3 (x_4x_5x_6)$	$g_1g_2g_3g_4g_5g_6$
000	011001
001	011101
010	100111
011	000100
100	010100
101	001011
110	100001
111	100100

5. Экспериментальные данные

5.1. Среднее значение нелинейности

Цель эксперимента – исследовать, каких значений достигает нелинейность, и как изменяется это значение относительно верхней границы с ростом параметров n и k .

В ходе эксперимента, при $n = k$, генерировалась случайная функция, при $k < n$ – функция класса $B_{n,k}$.

5.1.1. Для произвольной функции

Верхняя граница нелинейности произвольной функции вычисляется следующим образом:

- а) $2^{n-k}(2^{k-1} - 2^{\frac{k}{2}-1})$ для четных k ;
- б) $2^{n-k}(2^{k-1} - 2^{\frac{k+1}{2}-1})$ для нечетных k .

Вычисленные значения нелинейности функции для различных параметров n и k представлены в таблице 7.

Количество запусков программы – $q=500000$.

Как можно заметить, с ростом k уменьшается разница (в процентах) между средним значением и верхней границей. Это объясняется тем, что с ростом k уменьшается количество блоков векторной функции, то есть, как только k становится равным n , значение нелинейности становится максимально близким к верхней границе.

Таблица 7 – Вычисление средней нелинейности произвольной функции

Фиксированный параметр	Изменяющийся параметр	Среднее значение нелинейности	Верхняя граница нелинейности
$n=12$	$k=2$	0	0
	$k=3$	153	1024
	$k=4$	512	1536
	$k=6$	1024	1792
	$k=12$	1871	2016
$n=16$	$k=2$	0	16384
	$k=4$	7782	24576
	$k=8$	23398	30720
	$k=16$	31947	32640

5.1.2. Для обратимой функции

Чаще всего в криптосистемах используют обратимые функции. Поэтому в данном разделе проведем такой же эксперимент, как и в п. 5.1.1, только для обратимых векторных булевых функций.

В случае обратимых функций можно установить более точную верхнюю границу нелинейности. Вычислим её при некоторых фиксированных значениях n и изменяющихся значениях k .

1. $n=12$

а. $k=2$ – верхняя граница нелинейности достигается на бент-функциях, значит, она равна 1, но у обратимой функции все координаты и компоненты уравновешены (бент-функции не уравновешены [5]), следовательно, верхняя граница равна 0.

б. $k=3$ – (бент-функции существуют только при четных k [5]) при

нечетном k , получаем $N_F \leq 2^{n-k}[2^{k-1} - 2^{\frac{k}{2}-1}]$. Следовательно, в данном случае верхняя граница равна 512.

- с. $k=4$ – по аналогичным рассуждениям, при $k=2$, имеем верхнюю границу равную 6, но в силу уравниваемости и четности нелинейности обратимых функций получаем, что максимальная нелинейность равна $4 * 2^{n-k} = 1024$.

Аналогично шагу с) получаем, что при $k=6$ верхняя граница равна 1664, при $k=12$ – 2014 (таблица 8).

Таблица 8 – Вычисление средней нелинейности обратимой функции

Фиксированный параметр	Изменяющийся параметр	Среднее значение нелинейности	Верхняя граница нелинейности
$n=12$	$k=2$	0	0
	$k=3$	84	512
	$k=4$	434	1024
	$k=6$	957	1664
	$k=12$	1462	2014
$n=16$	$k=2$	0	0
	$k=4$	4727	16384
	$k=8$	10265	30208
	$k=16$	12084	32638

Количество запусков программы – $q=500000$.

Процентное соотношение между значением нелинейности и верхней границей обратимых функций такое же (с небольшой погрешностью), как и у произвольных функций.

5.2.Время работы программы

Цель эксперимента – исследовать скорость вычисления (в секундах) нелинейности векторной булевой функции при различных значениях n и k и сравнить полученные результаты с подсчитанной сложностью алгоритма.

Результаты по измерению времени работы программы представлены в таблице 9. Количество запусков программы – $q=5000000$.

Таким образом, время работы программы соответствует сложности алгоритма и подтверждает, что время растет быстрее с ростом k (при фиксированном n), чем наоборот.

Таблица 9 – Время работы программы при разных значениях параметров

n	k	t, c	k	n	t, c
12	2	40	2	2	3
	3	50		4	6
	4	150		6	10
	6	2000		8	16
	12	7000000		10	20
16	2	150	3	3	11
	4	200		6	30
	8	31000		9	40
24	2	4000		12	50
	3	300		15	60
	4	400	5	5	180
	6	3700		10	400
	8	60000		15	700

6. Реализация

В данном разделе описаны основные функции, использованные в программе для вычисления нелинейности векторной булевой функции. Программа реализована на языке программирования ЛЯПАС-Т [6]. Векторная булева функция хранится в комплексе L1 так, что $L1i = G(i)$. Полный код программы приведен в приложении А и приложении В.

Существует способ вычисления ПУА за $2^n n$ операций; он называется *быстрым преобразованием Уолша-Адамара*, или *схемой Грина* [5]. Покажем его работу на примере для $n=3$.

$$f = (1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0) \rightarrow (-1 \ -1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1) \rightarrow (-2 \ 0 \ 2 \ 0 \ 0 \ 2 \ 0 \ -2) \rightarrow (0 \ 0 \ -4 \ 0 \ 0 \ 0 \ 0 \ 4) \rightarrow (0 \ 0 \ -4 \ 4 \ 0 \ 0 \ -4 \ -4).$$

Следовательно, $\hat{f} = (0 \ 0 \ -4 \ 4 \ 0 \ 0 \ -4 \ -4)$.

1. Вычисление нелинейности булевой функции.

Вход: L2 – булев вектор длины 2^k , k — количество переменных функции.

Выход: a – нелинейность функции f .

Nf(L2,k/a)

$Ik \Rightarrow l \ @+L3(l) \ l \Rightarrow Q3$

$OL3 \ *PUA(L2,l/L3) \ *maxPUA(L3/m)$

$k-1 \Rightarrow a \ l \ a \Rightarrow a \ m > 1 \Rightarrow m \ a-m \Rightarrow a$

**

2. Преобразование Уолша-Адамара

Вход: L2 – булев вектор длины m .

Выход: L3 – коэффициенты ПУА.

$PUA(L2,m/L3)$

$\neg j \neg k$

$m/32 \Rightarrow b \text{ Oi } \uparrow(b>0)^2$

§1 $\Delta j \oplus m \sqsubset 4 \Delta k \text{ } 1 \Rightarrow L3k \text{ Ij \& L2i } \sqsubset 1 \neg L3k \rightarrow 1$

§2 $\Delta j \oplus 32 \sqsubset 3 \Delta k \text{ } 1 \Rightarrow L3k \text{ Ij \& L2i } \sqsubset 2 \neg L3k \rightarrow 2$

§3 $\neg j \Delta i \oplus b \mapsto 2 \text{ Oi}$

§4 $\text{Oi } 1 \Rightarrow j \text{ } 1 \Rightarrow k \text{ Q3} \Rightarrow b \Rightarrow z$

§5 $j-1 \Rightarrow a \text{ } a^*k+i \Rightarrow a+k \Rightarrow p \text{ L3a} \Rightarrow q \text{ q+L3p} \Rightarrow \text{L3a q-L3p} \Rightarrow \text{L3p}$

$\Delta i \oplus k \mapsto 5$

$\text{Oi } \Delta j \Delta j$

$\uparrow(j \leq b)^5 \text{ } 1 \Rightarrow j$

$k < 1 \Rightarrow k \text{ Q3/k} \Rightarrow b$

$k \oplus z \mapsto 5$

§6 **

3. Нахождение максимального по абсолютной величине коэффициента ПУА.

Вход: L1 – массив коэффициентов ПУА.

Выход: m – модуль максимального по абсолютной величине коэффициента ПУА.

$\max_{\text{PUA}}(L1/m)$

$\neg i \text{ Om}$

§1 $\Delta i \oplus Q1 \sqsubset 3 \text{ L1i} \Rightarrow q$

$q \& I31 \sqsubset 2 \text{ q} \neg +1 \Rightarrow q$

§2 $\uparrow(q < m)^1 \text{ q} \Rightarrow m \rightarrow 1$

§3 **

4. Нахождение минимальной нелинейности внутри j -ого блока.

Вход: L1 – функция G , n – общее число переменных, k – количество

существенных переменных, j – номер блока.

Выход: x – минимальная нелинейность данного блока, z – компонента, такая, что $N_{zF_j} = \min_{v \neq 0^k} N_{vF_j}$.

$\text{minNinsideblockj}(L1, n, k, j/x, z)$

$\neg x \text{ Oi } l k \Rightarrow m \text{ m}/32 \Rightarrow d \uparrow (d \leq 1) 9 @+L2(d) d \Rightarrow Q2 \text{ OL2} \rightarrow 10$

§9 $@+L2(1) 1 \Rightarrow Q2 \text{ OL2}$

§10 $Ov \text{ Oi } j * k \Rightarrow s \text{ Of}$

§1 $\Delta i \oplus m \boxed{\subset} 8 i > 1 \Rightarrow u \oplus i \Rightarrow u \oplus v \Rightarrow b \text{ u} \Rightarrow v \text{ b} < s \Rightarrow b \uparrow X8bw \neg 1 \text{ Or } Og \text{ 32} \Rightarrow e \uparrow (d > 0) 4$

§2 $\Delta l \oplus m \boxed{\subset} 3 Iw \& L1l \boxed{\subset} 2 Ilvg \Rightarrow g \rightarrow 2$

§3 $L2r \oplus g \Rightarrow L2r \rightarrow 7$

§4 $\Delta l \oplus e \boxed{\subset} 5 Iw \& L1l \boxed{\subset} 4 Ilvg \Rightarrow g \rightarrow 4$

§5 $L2r \oplus g \Rightarrow L2r \text{ Og } \forall l \text{ e} + 32 \Rightarrow e \Delta r \oplus d \mapsto 4$

§7 $*Nf(L2, k/a)$

$\uparrow (a \geq x) 1 a \Rightarrow x \text{ v} \Rightarrow z \rightarrow 1$

§8 **

5. Нахождение минимальной нелинейности среди всех блоков.

Вход: $L1$ – функция G , n – общее число переменных, k – количество существенных переменных.

Выход: b – минимальная нелинейность среди всех блоков, y – компонента такая, что $N_{yF} = N_F$.

$\text{minNblocks}(L1, n, k/b, y)$

$n/k \Rightarrow s \text{ Oj}$

$@+L3(s) \text{ OQ3}$

$n-k \Rightarrow m \text{ Im} \Rightarrow m \neg b \text{ Ot}$

§1 $*\text{minNinsideblockj}(L1, n, k, j/x, z)$

$x @ > L3 \uparrow (b \leq x) 2 x \Rightarrow b \text{ z} \Rightarrow y \text{ j} \Rightarrow c$

§2 $\Delta_j \oplus s \mapsto 1 \ b^*m \Rightarrow b \ c^*k \Rightarrow c \ y < c \Rightarrow y$

§3 **

6. Вычисление нелинейности векторной булевой функции из класса $B_{n,k}$.

Вход: L1 – функция G , n – общее число аргументов, k – количество существенных переменных.

Выход: b – нелинейность векторной функции G .

main(L1,n,k/b)

*minNbblocks(L1,n,k/b,z)

§1 **

7. Генерация произвольной векторной булевой функции.

Вход: L1 – пустой комплекс, n – общее число аргументов, k – количество существенных переменных.

Выход: L1 – векторная функция G .

init_rand_table(L1,n,k/L1)

$I_k \Rightarrow m \ O_i$

§1 $I_{n-1} \Rightarrow L1_i \ 32-n \Rightarrow l \ X > l \Rightarrow r \& L1_i \Rightarrow L1_i \ \Delta_i \oplus m \mapsto 1$

**

8. Генерация обратимой векторной булевой функции.

Вход: L1 – пустой комплекс, n – общее число аргументов, k – количество существенных переменных.

Выход: L1 – векторная функция G .

init_reverse_table(L1,n,k/L1)

$I_k \Rightarrow m \ @+L2(m) \ OQ2 \ O_i$

§1 $i@>L2 \ \Delta_i \oplus m \mapsto 1$

Ol Oh Oi

§2 *creating_permutation(L2,k/L2)

§3 $L2i < h \vee L1i \Rightarrow L1i \Delta i \oplus m \mapsto 3 \quad Oi \Delta l^*k \Rightarrow h \oplus n \mapsto 2$

§4 **

9. Перестановка чисел.

Вход: L1 – числа от 1 до k по порядку.

Выход: L1 – перестановка чисел.

creating_permutation(L1,k/L1)

$\neg i \quad l k \Rightarrow m$

§1 $\Delta i \oplus m \hookrightarrow 2 \quad 32-k \Rightarrow a \quad X > a \Rightarrow a \uparrow (i=a)1 \Leftrightarrow (L1ia) \rightarrow 1$

§2 **

10. Построение проекции для функций класса $C_{n,\leq k}$.

Вход: вектор x , y – вектор, содержащий единицы в разрядах, номера которых соответствуют существенным переменным.

Выход: a – проекция x .

projection(x,y/a)

$\neg i \quad Oa$

§1 $\uparrow X3y_j$

§2 $\Delta i \quad l_j \& x \boxed{\subseteq} 1 \quad l_i \vee a \Rightarrow a \rightarrow 1$

§3 **

11. Вычисление значения функции класса $C_{n,\leq k}$.

Вход: L1 – функция G , L2 – матрица M , входной вектор x , k – количество существенных переменных, n – общее число переменных.

Выход: $y = F(x)$.

main(L1,L2,x,k,n/y)

¬i Oy

§1 $\Delta i \oplus Q2 \boxed{\subset} 2$ *projection(x,L2i/a)

L1a&Ii $\boxed{\subset} 1$ IiVy $\Rightarrow$ y $\rightarrow 1$

§2 **

12. Построение проекции для функций класса $B_{n,k}$.

Вход: вектор x, j, k – натуральные числа, где $j \geq k$.

Выход: a – проекция $x, a = (x_{j-k}, \dots, x_{j-1})$.

projection_B(x,j,k/a)

§1 $j-k \Rightarrow i$ $I_{k-1} \langle i \& x \rangle i \Rightarrow a$

§2 **

13. Вычисление значения функции класса $B_{n,k}$.

Вход: L1 – функция G , входной вектор x, k – количество существенных переменных, n – общее число переменных.

Выход: $y = F(x)$.

main_B(L1,x,k,n/y)

¬i Oy $k \Rightarrow j$

§1 $\Delta i \oplus n \boxed{\subset} 3$ $\uparrow(i < j) 2$ $j+k \Rightarrow j$

*projection_B(x,j,k/a)

§2 L1a&Ii $\boxed{\subset} 1$ IiVy $\Rightarrow$ y $\rightarrow 1$

§3 **

ЗАКЛЮЧЕНИЕ

В ходе данной работы был исследован класс векторных булевых функций ограниченной сложности; а именно, разработаны, реализованы и исследованы:

1. способы задания векторных булевых функций ограниченной сложности. Данное исследование показало, что и «каменщик», и векторный способ выгоднее общего по памяти.
2. способы шифрования сообщения с заданным параметром при функции ограниченной сложности в качестве порождающей. В ходе их сравнения были получены следующие оценки:
 - при $k \leq n/2$, выгоднее преобразование таблицы G в G' ;
 - при $n/2 < k$, скорость шифрования одинаковая;
 - однако, при приближении значения k к n перестроение перестает иметь смысл, так как размер таблицы G становится слишком велик.
3. алгоритм вычисления нелинейности векторной функции класса $B_{n,k}$.
Вычисление по описанному алгоритму производится быстрее, чем вычисление по определению, в $2^{2(n-k)}$ раз.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Городилова А. А. От криптоанализа шифра к криптографическому свойству булевой функции // Прикл. диск. мат. – 2016. – № 3(33). – С. 16-44.
2. Álvarez-Cubero J. A. Cryptographic Criteria on Vector Boolean Functions / José Antonio Álvarez-Cubero, Pedro J. Zufiria // Cryptography and Security in Computing. – 2012. – Chapter 3. – P. 51-70.
3. Agibalov G. P. Substitution block ciphers with functional keys // Прикл. дискр. мат. – 2017. – № 38. – С. 57–65.
4. Киселева Н. М. Алгоритмы вычисления криптографических характеристик векторных булевых функций / Н. М. Киселева, Е. С. Липатова, И. А. Панкратова, Е. Е. Трифонова // Прикл. дискр. мат. – 2019. – № 46. – С. 78–87.
5. Панкратова И. А. Булевы функции в криптографии: учеб. пособие. — Томск: Издат. Дом Том. гос. ун-та, 2014 г. — 88 с.
6. Агибалов Г. П. О криптографическом расширении и его реализации для Русского языка программирования / Г. П. Агибалов, В. Б. Липский, И. А. Панкратова // Прикл. диск. мат. – 2013. – № 3(21). – С. 93–104.

Приложение А

Программный код вычисления нелинейности векторной булевой функции

*** $\Rightarrow \neg \Updownarrow \neg \% \vee \& \oplus \Delta \nabla \S \rightarrow \boxed{\mathbb{G}} \mapsto \uparrow \neq > < \geq \leq$ ***

head(/)

@+F3(1000)

/Введите n:\n'>C

/F3<C *s2n(F3,10/n) OQ3

/Введите k:\n'>C

/F3<C *s2n(F3,10/k) OQ3

T $\Rightarrow$ X Ik $\Rightarrow$ m @+L1(m) OQ1

***init_rand_table(L1,n,k/L1)

*init_reverse_table(L1,n,k/L1)

§1 *main(L1,n,k/f)

§2 **

*** перевод строки в число (q - основание системы счисления) ***

s2n(F1,q/a)

$\neg i$ Oa

§1 $\Delta i \oplus Q1 \boxed{\mathbb{G}}^2 *c2n(F1i/b) a*q+b \Rightarrow a \rightarrow 1$

§2 **

*** перевод цифры в число ***

c2n(a/b)

a-'0' $\Rightarrow$ b $\uparrow$ (b<10)1 a-'A'+10 $\Rightarrow$ b $\uparrow$ (a $\leq$ 'Z')1 a-'a'+10 $\Rightarrow$ b

§1 **

*** перевод числа в строку (q - основание системы счисления) ***

n2s(a,q/F1)

OQ1

§1 $a/q \Rightarrow a \ Z \Rightarrow b \ *n2c(b/c) \ c@>F1 \ a \mapsto 1$

$O_i \ Q1-1 \Rightarrow j$

§3 $\uparrow(i \geq j)_4 \Leftrightarrow (F1ij) \ \Delta i \ \nabla j \rightarrow 3$

§4 **

*** перевод числа в цифру ***

$n2c(a/b)$

$a+'0' \Rightarrow b \ \uparrow(a < 10)_1 \ a-10+'a' \Rightarrow b$

§1 **

*** перевод знакового числа в строку

$nsign2s(a/F1)$

$10 \Rightarrow d \ OQ1 \ O_i \ a \&I31 \ \boxed{\square}_1 \ '-@>F1 \ \Delta i \ a \neg +1 \Rightarrow a$

§1 $a/d \Rightarrow a \ Z \Rightarrow b \ *n2c(b/c) \ c@>F1 \ a \mapsto 1$

$Q1-1 \Rightarrow j$

§3 $\uparrow(i \geq j)_4 \Leftrightarrow (F1ij) \ \Delta i \ \nabla j \rightarrow 3$

§4 **

*** перевод строки в булев вектор ***

$input2(F1/a)$

$O_a \ O_i \ Q1-1 \Rightarrow j$

§1 $\uparrow(i \geq j)_2 \Leftrightarrow (F1ij) \ \Delta i \ \nabla j \rightarrow 1$

§2 $\neg i$

§3 $\Delta i \oplus Q1 \ \boxed{\square}_4 \ F1i \oplus '0' \ \boxed{\square}_3 \ Ii \vee a \Rightarrow a \rightarrow 3$

§4 **

*** вывод булевой матрицы nxm ***

$printmat(n,m,L1,k/)$

$O_i \ @+F2(m) \ O_j$

§1 $/ ' '>C \ \Delta j \oplus k \mapsto 1 \ / ' '>C \ m-1 \Rightarrow j$

§2 $*n2s(j,10/F2) /F2>C \ OQ2 \ \nabla j \mapsto 2 \ *n2s(j,10/F2) /F2>C \ OQ2 \ /\backslash n'>C$

§3 $\uparrow(i=n)_4$

$*output2(i,k/F2) /F2>C \ / ' '>C \ *output2(L1i,m/F2) /F2>C \ OQ2$

$$\wedge n' > C \Delta i \rightarrow 3$$

§4 **

*** перевод булева вектора а длины n в строку ***

output2(a,n/F1)

Oi

§1 '0' $\Rightarrow$ F1i a&Ii $\square_2 \Delta$ F1i

§2 $\Delta i \uparrow (i < n) 1 n \Rightarrow Q1$

Oi Q1-1 $\Rightarrow j$

§3 $\uparrow (i \geq j) 4 \Leftrightarrow (F1ij) \Delta i \forall j \rightarrow 3$

§4 **

*** инициализация произвольной таблицы G

init_rand_table(L1,n,k/L1)

Ik $\Rightarrow m$ Oi

§1 In-1 $\Rightarrow L1i$ 32-n $\Rightarrow l$ X>l $\Rightarrow r$ &L1i $\Rightarrow L1i \Delta i \oplus m \mapsto 1$

**

init_reverse_table(L1,n,k/L1)

Ik $\Rightarrow m$ @+L2(m) OQ2 Oi

§1 i@>L2 $\Delta i \oplus m \mapsto 1$

Oi Oh Oi

§2 *creating_permutation(L2,k/L2)

§3 L2i<hV L1i $\Rightarrow L1i \Delta i \oplus m \mapsto 3$ Oi $\Delta l^* k \Rightarrow h \oplus n \mapsto 2$

§4 **

PUA(L2,m/L3)

$\neg j \neg k$

m/32 $\Rightarrow b$ Oi $\uparrow (b > 0) 2$

§1 $\Delta j \oplus m \square_4 \Delta k 1 \Rightarrow L3k$ Ij&L2i $\square_1 \neg L3k \rightarrow 1$

§2 $\Delta j \oplus 32 \square_3 \Delta k 1 \Rightarrow L3k$ Ij&L2i $\square_2 \neg L3k \rightarrow 2$

§3 $\neg j \Delta i \oplus b \mapsto 2$ Oi

§4 Oi 1 $\Rightarrow j 1 \Rightarrow k Q3 \Rightarrow b \Rightarrow z$

§5 j-1 $\Rightarrow a a^* k + i \Rightarrow a + k \Rightarrow p$ L3a $\Rightarrow q q + L3p \Rightarrow L3a q - L3p \Rightarrow L3p$

$$\Delta i \oplus k \mapsto 5$$

$$O_i \Delta_j \Delta_j$$

$$\uparrow(j \leq b) 5 \ 1 \Rightarrow j$$

$$k < 1 \Rightarrow k \ Q3/k \Rightarrow b$$

$$k \oplus z \mapsto 5$$

§6 **

*** нелинейность функции f

*** k - число существенных переменных

$$Nf(L2, k/a)$$

$$Ik \Rightarrow m \ @+L3(m) \ m \Rightarrow Q3$$

$$OL3 \ *PUA(L2, m/L3) \ *maxPUA(L3/l)$$

$$k-1 \Rightarrow a \ I a \Rightarrow a \ l > 1 \Rightarrow l \ a-l \Rightarrow a$$

**

*** минимальная нелинейность внутри блока j ***

$$\min N_{insideblockj}(L1, n, k, j/x, z)$$

$$\neg x \ O_i \ I k \Rightarrow m \ m/32 \Rightarrow d \ \uparrow(d \leq 1) 9 \ @+L2(d) \ d \Rightarrow Q2 \ OL2 \ \rightarrow 10$$

$$\S 9 \ @+L2(1) \ 1 \Rightarrow Q2 \ OL2$$

$$\S 10 \ O_v \ O_i \ j * k \Rightarrow s \ O_f$$

$$\S 1 \ \Delta i \oplus m \ \boxed{\square} 8 \ i > 1 \Rightarrow u \oplus i \Rightarrow u \ u \oplus v \Rightarrow b \ u \Rightarrow v \ b < s \Rightarrow b \ \uparrow X8bw \neg 1 \ Or \ Og \ 32 \Rightarrow e \ \uparrow(d > 0) 4$$

$$\S 2 \ \Delta l \oplus m \ \boxed{\square} 3 \ I w \& L1 l \ \boxed{\square} 2 \ Il \vee g \Rightarrow g \ \rightarrow 2$$

$$\S 3 \ L2r \oplus g \Rightarrow L2r \ \rightarrow 7$$

$$\S 4 \ \Delta l \oplus e \ \boxed{\square} 5 \ I w \& L1 l \ \boxed{\square} 4 \ Il \vee g \Rightarrow g \ \rightarrow 4$$

$$\S 5 \ L2r \oplus g \Rightarrow L2r \ Og \ \forall l \ e + 32 \Rightarrow e \ \Delta r \oplus d \mapsto 4$$

$$\S 7 \ *Nf(L2, k/a)$$

$$\uparrow(a \geq x) 1 \ a \Rightarrow x \ v \Rightarrow z \ \rightarrow 1$$

§8 **

$$\min N_{blocks}(L1, n, k/b, y)$$

$$n/k \Rightarrow s \ O_j$$

$$@+L3(s) \ OQ3$$

$$n-k \Rightarrow m \ Im \Rightarrow m \ \neg b \ Ot$$

$$\S 1 \ * \min N_{insideblockj}(L1, n, k, j/x, z)$$

$x @>L3 \uparrow (b \leq x)2 \ x \Rightarrow b \ z \Rightarrow y \ j \Rightarrow c$
 $\S 2 \ \Delta j \oplus s \mapsto 1 \ b^* m \Rightarrow b \ c^* k \Rightarrow c \ y < c \Rightarrow y$
 $\S 3 \ **$

$\text{main}(L1, n, k/b)$
 $*\text{minNbblocks}(L1, n, k/b, z)$
 $\S 1 \ **$

$\text{beautifuloutputN}(v/)$
 $/N\{ '>C \ @+F4(n) \ \uparrow X2vj \ /'f>C \ *n2s(j, 10/F4) \ /F4>C \ OQ4$
 $\S 1 \ \uparrow X2vj \ /' \oplus \ f>C \ *n2s(j, 10/F4) \ /F4>C \ OQ4 \rightarrow 1$
 $\S 2 \ /\} = '>C$
 $\S 3 \ **$

$\text{maxPUA}(L1/m)$
 $\neg i \ O m$
 $\S 1 \ \Delta i \oplus Q1 \ \boxed{\subseteq} 3 \ L1 i \Rightarrow q$
 $q \&I31 \ \boxed{\subseteq} 2 \ q \neg +1 \Rightarrow q$
 $\S 2 \ \uparrow (q < m)1 \ q \Rightarrow m \rightarrow 1$
 $\S 3 \ **$

$\text{creating_permutation}(L1, k/L1)$
 $\neg i \ I k \Rightarrow m$
 $\S 1 \ \Delta i \oplus m \ \boxed{\subseteq} 2 \ 32-k \Rightarrow a \ X > a \Rightarrow a \ \uparrow (i=a)1 \ \Leftrightarrow (L1 i a) \rightarrow 1$
 $\S 2 \ **$

Приложение Б

Программный код шифрования сообщения двумя способами

```

head(/)
  @+F3(1000)
  /Введите x:\n'>C
  /F3<C *s2n(F3,2/x) Q3⇒n OQ3
  /Введите k:\n'>C
  /F3<C *s2n(F3,10/k) OQ3
  Ik⇒m
  In-1⇒b X&b⇒b
  @+L4(n) n⇒Q4 Oi
§1 i⇒L4i Δi⊕n→1
  *creating_permutation(L4/L4)
  Oi OQ1
  @+L1(m) @+L2(n)
§2 In-1⇒L1i X&L1i⇒L1i Δi⊕m→2
  Oi OQ2 Ik-1⇒c
§3 c@>L2 Δi⊕n→3
§4 x⊕b⋈L4⇒z
§5 *main(L1,L2,z,k,n/t)
  t⊕b⋈L4⇒z

§6 L1i⊕b⋈L4⇒L1i Δi⊕m→6
x⊕b⋈L4⇒z
§7 *main(L1,L2,z,k,n/t)
§8 **

*** перевод строки в число (q - основание системы счисления) ***
s2n(F1,q/a)
  -i Oa
§1 Δi⊕Q1  $\square_2$  *c2n(F1i/b) a*q+b⇒a →1
§2 **

```

*** перевод цифры в число ***

c2n(a/b)

$a-'0'\Rightarrow b \uparrow (b<10)1 \ a-'A'+10\Rightarrow b \uparrow (a\leq 'Z')1 \ a-'a'+10\Rightarrow b$

§1 **

*** перевод числа в строку (q - основание системы счисления) ***

n2s(a,q/F1)

OQ1

§1 $a/q\Rightarrow a \ Z\Rightarrow b \ *n2c(b/c) \ c@>F1 \ a\mapsto 1$

Oi Q1-1 $\Rightarrow$ j

§3 $\uparrow (i\geq j)4 \Leftrightarrow (F1ij) \ \Delta i \ \nabla j \rightarrow 3$

§4 **

*** перевод числа в цифру ***

n2c(a/b)

$a+'0'\Rightarrow b \uparrow (a<10)1 \ a-10+'a'\Rightarrow b$

§1 **

*** перевод строки в булев вектор ***

input2(F1/a)

Oa Oi Q1-1 $\Rightarrow$ j

§1 $\uparrow (i\geq j)2 \Leftrightarrow (F1ij) \ \Delta i \ \nabla j \rightarrow 1$

§2 $\neg i$

§3 $\Delta i \oplus Q1 \boxed{\neg} 4 \ F1i \oplus '0' \boxed{\neg} 3 \ Ii \vee a \Rightarrow a \rightarrow 3$

§4 **

*** вывод булевой матрицы pxm ***

printmat(n,m,L1,k/)

Oi @+F2(m)

§1 $\uparrow (i=n)2$

$*output2(i,k/F2) /F2>C \ /' \ '>C \ *output2(L1i,m/F2) /F2>C \ OQ2$

$/^{\wedge}n'>C \ \Delta i \rightarrow 1$

§2 **

*** перевод булева вектора a длины n в строку ***

output2(a,n/F1)

Oi

§1 '0'⇒F1i a&Ii $\boxed{\subseteq}$ 2 ΔF1i

§2 Δi ↑(i<n)1 n⇒Q1

Oi Q1-1⇒j

§3 ↑(i≥j)4 ⇔(F1ij) Δi ∇j →3

§4 **

projection(x,y/a)

¬i Oa

§1 ↑X3yj

§2 Δi Ij&x $\boxed{\subseteq}$ 1 IiVa⇒a →1

§3 **

main(L1,L2,z,k,n/y)

¬i @+F4(n) Oy

§1 Δi⊕Q2 $\boxed{\subseteq}$ 2 *projection(z,L2i/a)

L1a&Ii $\boxed{\subseteq}$ 1 IiVy⇒y →1

§2 **

projection_B(x,j,k/a)

§1 j-k⇒i Ik-1<i&x>i⇒a

§2 **

main_B(L1,x,k,n/y)

¬i Oy k⇒j

§1 Δi⊕n $\boxed{\subseteq}$ 3 ↑(i<j)2 j+k⇒j

*projection_B(x,j,k/a)

§2 L1a&Ii $\boxed{\subseteq}$ 1 IiVy⇒y →1

§3 **

*** тасование Кнута ***

creating_permutation(L1/L1)

$\neg i \text{ } Q1/2 \Rightarrow c$

$\S 1 \Delta i \oplus c \boxed{\subseteq}^2 X; Q1 \Rightarrow j \text{ } X; Q1 \Rightarrow k \uparrow (k=j) 1 \Leftrightarrow (L1kj) \rightarrow 1$

$\S 2 **$

*** x в степени сигма ***

sigma(x,b/y)

Oy

$\S 1 \text{ } b \neg \Rightarrow y \text{ } I n-1 \& y \Rightarrow y$

$\S 2 \text{ } x \oplus y \Rightarrow y$

**

Отчет о проверке на заимствования №1



Автор: Петров Георгий Константинович
Проверяющий: Петров Георгий Константинович (georgypetrov1973@gmail.com / ID: 8520887)
Отчет предоставлен сервисом «Антиплагиат» - users.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 14
Начало загрузки: 24.01.2022 15:49:45
Длительность загрузки: 00:00:01
Имя исходного файла: ВКР_Петров_1165.pdf
Название документа: ВКР_Петров_1165
Размер текста: 34 кБ
Символов в тексте: 34623
Слов в тексте: 5135
Число предложений: 236

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 24.01.2022 15:49:47
Длительность проверки: 00:00:04
Комментарии: не указано
Модули поиска: Интернет Free



ЗАИМСТВОВАНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
4,18%	0%	0%	95,82%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	1,41%	http://vital.lib.tsu.ru/vital/access/services/Download/vital:6890/SOURCE01 http://vital.lib.tsu.ru	07 Сен 2020	Интернет Free
[02]	1,06%	http://vital.lib.tsu.ru/vital/access/services/Download/vtls:000472715/SOURCE1 http://vital.lib.tsu.ru	24 Янв 2020	Интернет Free
[03]	0,78%	Почти совершенно нелинейные функции: характеристика через подфункции и дифференциальная эквивалентность http://a-server.math.nsc.ru	01 Дек 2020	Интернет Free

Еще источников: 7
Еще заимствований: 0,94%

Handwritten signature