МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА — Российский технологический университет»

РТУ МИРЭА

Институт искусственного интеллекта
Базовая кафедра № 252 «Информационная безопасность»

КУРСОВАЯ РАБОТА

по дисциплине «Вероятностные методы в криптографии»

Студент группы ККСО-03-19 Никишина А.А. _____

(учебная группа, фамилия, имя, отчество студента) *(подпись студента)*

Руководитель курсовой работы Катышев С.Ю. _____

должность, звание, ученая степень (подпись руководителя)

Работа представлена к защите «__» июня 2024 г.

Оценка_____

Оглавление

Основные определения

Вероятностное пространство

Произвольное Ω – пространство элементарных событий

Элементы $\omega \in \Omega$ – элементарное событие (взаимно исключающие)

Множество всех подмножеств $\Omega - 2^{\Omega} = \mathbb{a}$ – алгебра событий

Алгебра событий $\mathbb{a}$ называется σ-алгеброй, если $\bigcup_{n=1}^{\infty} A_n \in \mathbb{a}, \bigcap A_n \in \mathbb{a}, A_n \in \mathbb{a}, n \in N$

Случайным элементарным событием называется любой элемент $\mathbb{a}$

$\Omega \in \mathbb{a}$ – достоверное событие

$\emptyset \in \mathbb{a}$ – невозможное событие

Числовая функция $P: \mathbb{a} \to [0,1]$ – называется вероятностной мерой (вероятностью), если:

1) $P(A) > 0$, для любого A из $\mathbb{a}$
2) P(Ω)=1
3) $A, B \in \mathbb{a}: \mathrm{A} \cap \mathrm{B} = \emptyset$, то $P(A \cup B) = P(A) + P(B)$
4) $A_1 > A_2 \ldots > A_n, \bigcap A_n = \emptyset, \mathrm{n} \to \infty, \lim \mathrm{P}(\mathrm{A_n}) = 0$

(Ω, $\mathbb{a}$, P) – вероятностное пространство.

Независимость

События $A_1, .., A_n$ называются независимыми в совокупности, если $P\left(A_{i_1}, .., A_{i_k}\right) = P\left(A_{i_1}\right) * \ldots * P(A_{i_k}), k = 1, .., n, i = 1, .., n$

События $A_1, .., A_n$ называются независимыми попарно, если $P\left(A_i A_j\right) = P(A_i) P\left(A_j\right)$, где $i, j = 1, \ldots, n$

Случайная величина – произвольная функция на множестве элементарных событий: $\xi: \Omega \to R, \xi = \xi(\omega)$.

Индикатором события А называется двоичная случайная величина $I_A(\omega)$:

$$I_A(\omega): \begin{cases} 1, \omega \in A \\ 0, \omega \notin A \end{cases}$$

Независимость слсучайных величин

$\xi_1, .., \xi_n$ называют независимыми в совокупности, если $\forall B_1, .., B_n$

$$P\{\xi_1 \in B_1, .., \xi_n \in B_n\} = P\{\xi_1 \in B_1\} \ldots P\{\xi_n \in B_n\}$$

Лабораторная работа №1

Задание:

1. Генерация случайной подстановки: $2^6 \to 2^6$ от 0 до 63;

2. Построить координаты функции подстановки (вектора значений)

$$f(x_1, .., x_6) \to y_1$$

...

$$f(x_1, .., x_6) \to y_6$$

3. Найти вес функции;

4. Построить многочлен Жегалкина для каждой f;

5. Найти фиктивные переменные.

Определения:

Подстановка элементов данного множества – замена каждого из его элементов a каким-либо другим элементом $\varphi(a)$ из этого же множества, при этом должны получаться все элементы исходного множества и каждый только один раз. (то есть – взаимно однозначное соответствие/биективное отображение).

Вес функции f – называется мощность её множества истинности (количество единиц, в данном случае)

Многочлен Жегалкина – полином, коэффициентами которого являются числа 0 и 1, в качестве операций умножения и сложения выступают соответственно конъюнкция и сумма по модулю 2 (XOR).

Например, для функции от 3 переменных полином Жегалкина выглядит следующим образом:

$$f(x_1\ x_2, x_3) = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus a_{23}x_2x_3 \oplus a_{123}x_1x_2x_3$$

$a_0, .., a_{123} \in \{0, 1\}$, могут быть равны либо 0 либо 1 в зависимости от того, какие значения принимает булева функция $f(x_1, x_2, x_3)$ на том или ином наборе значений переменных.

С помощью полинома Жегалкина можно представить любую булеву функцию, причем единственным образом. Поэтому можно сказать, что многочлен Жегалкина является одним способом представления булевых функции.

Таблица истинности операции $\oplus$ (исключающее ИЛИ, сложение по модулю 2)

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Говорят, что булева функция $f(x_1, .., x_6)$ *существенно* зависит от переменной x_i, если выполняется условие:

$$f(x_1, .., x_{i-1}, 0, x_{i+1}, \dots x_n) \neq f(x_1, .., x_{i-1}, 1, x_{i+1}, \dots x_n)$$

В этом случае также говорят, что переменная x_i *существенная*, в противном случае её называют *фиктивной* переменной.

Алгоритмы, используемые для выполнения задания:

1. Для генерации случайной подстановки генерируем список от 0 до 63 и перемешиваем его случайным образом. Вывод представлен на Рис.1. (Обозначение в коде: s)

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

Рисунок 1 – Случайная подстановка (биективное отображение)

Для дальнейших вычислений переводим данную подстановку к двоичному виду.

2. Построение координат функции подстановки (вектора значений)

Приведем алгоритм построения координат функции подстановки на примере первых 6 значений подстановки, пример приведен в Таблице:

$x_1x_2x_3x_4x_5x_6$	f_1	f_2	f_3	f_4	f_5	f_6
000000	1	0	1	0	1	0
000001	1	0	1	0	0	0
000010	0	0	1	0	1	1
000011	1	1	1	1	1	1
000100	1	1	1	0	1	0
000101	1	1	0	0	1	0
…	…	…	…	…	…	…
111111	0	1	1	0	0	1

Получается, что $f_1(x_1, x_2, x_3, x_4, x_5, x_6) = (1\ 1\ 0\ 1\ 1\ 1 \dots 0)$

$$f_2(x_1, x_2, x_3, x_4, x_5, x_6) = (0\ 0\ 0\ 1\ 1\ 1 \dots 1)$$

$$f_3(x_1, x_2, x_3, x_4, x_5, x_6) = (1\ 1\ 1\ 1\ 1\ 0 \dots 1)$$

$$f_4(x_1, x_2, x_3, x_4, x_5, x_6) = (0\ 0\ 0\ 1\ 0\ 0 \dots 0)$$

$$f_5(x_1, x_2, x_3, x_4, x_5, x_6) = (1\ 0\ 1\ 1\ 1\ 1 \dots 0)$$

$$f_6(x_1, x_2, x_3, x_4, x_5, x_6) = (0\ 0\ 1\ 1\ 0\ 0 \dots 1)$$

3. Вес функции вычисляется, как количество единиц в векторе значений, приведем пример расчета веса функции f_1:

f_1 =*(***1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0***)*

Количество единиц = 32 => Вес функции $f_1 = 32$

4. Построение многочлена Жегалкина производится по таблице истинности.

Первым блоком является непосредственно функция f. Каждый блок делим пополам в левую часть нового блока переписываем левую часть старого, а в правую часть – побитовое сложение по модулю 2 (XOR) правого блока. Затем каждый блок снова делим пополам и продолжаем так до тех пор, пока длина блоков не достигнет единицы.

Единицы в последней строке такого преобразования будут говорить нам о том линейное представление каких слагаемых будет содержать многочлен Жегалкина.

Приведем пример вычисления многочлена Жегалкина для некоторой абстрактной функции f

x_1	x_2	x_3	$f(x_1, x_2, x_3)$				Слагаемые
0	0	0	1	1	1	**1**	1
0	0	1	1	1	1	0	x_3
0	1	0	1	1	0	0	x_2
0	1	1	0	0	1	**1**	x_2x_3
1	0	0	1	0	0	0	x_1
1	0	1	1	0	0	0	x_1x_3
1	1	0	0	1	1	**1**	x_1x_2
1	1	1	0	0	0	**1**	$x_1x_2x_3$

Таким образом, получается многочлен Жегалкина:

$$f(x_1, x_2, x_3) = 1 \oplus x_2x_3 \oplus x_1x_2 \oplus x_1x_2x_3$$

5. Нахождение фиктивных переменных происходит следующим образом:

- Функция принимает полином Жегалкина в виде списка (состоящего из 0 и 1) Если на какой-либо позиции стоит 1, то соответствующее произведение есть в многочлене Жегалкина.
- Возвращает список вида [x, x, x, x, x, x], где x = False соответствует переменным, которые не используются в полиноме Жегалкина, т.е. фиктивные.

Для того, чтобы выписать многочлен Жегалкина в виде полинома, функция принимает полином Жегалкина в виде строки

Фиктивная переменная отличается тем, что от неё не зависит значение функции, иными словами она представляет собой то, чего нет в многочлене Жегалкина. Если таковых нет, то все переменные считаются существенными.

Если на какой-либо позиции стоит единица => соответствующее произведение есть в многочлене Жегалкина. Выводит на экран полином Жегалкина в привычном виде.

Результат работы программы представлен ниже: вывод вектора значения функции, веса функции, полученного многочлена Жегалкина и информация о фиктивных и существенных переменных.

Вывод работы программы:

1. Генерация случайной подстановки V6->V6

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

2. Вектора значений координатных функций

f1: 1 1 0 1 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 1 0 0 1 1 1 0 1 1 0 1 0 1 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0

f2: 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 1 0 0 0 1 1 0 1 1 1 0 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 1 1 1 0 0 0 1 0 0 1 1 1 0 1 0 0 1

f3: 1 1 1 1 1 0 0 0 1 0 1 1 0 1 0 0 1 1 1 1 0 0 1 1 1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 0 1 0 0 0 0 1 1 0 0 0 1

f4: 0 0 0 1 0 0 1 1 0 1 0 0 0 1 0 0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0 1 1 0 0 1 0 0 1 0 0 0

f5: 1 0 1 1 1 1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 0 1 1 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 0

f6: 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 1 1 1 0 1 1 0 1 0 0 1 1

3. Вес координатных функций

||f1|| = 32

||f2|| = 32

||f3|| = 32

||f4|| = 32

||f5|| = 32

||f6|| = 32

4. Многочлен Жегалкина для координатных функций

f1: 1 0 1 1 0 0 0 1 1 1 1 1 1 0 1 0 0 1 1 1 0 0 0 1 0 1 1 0 1 0 1 0 0 0 0 1 0 0 1 0 1 1 1 0 0 1 0 1 1 1 0 0 1 1 1 1 1 1 0 1 0 1 1 0

*f1: 1 + x2 + x1*x2 + x1*x2*x3 + x4 + x1*x4 + x2*x4 + x1*x2*x4 + x3*x4 + x2*x3*x4 + x1*x5 + x2*x5 + x1*x2*x5 + x1*x2*x3*x5 + x1*x4*x5 + x2*x4*x5 + x3*x4*x5 + x2*x3*x4*x5 + x1*x2*x6 + x2*x3*x6 + x4*x6 + x1*x4*x6 + x2*x4*x6 + x1*x3*x4*x6 + x1*x2*x3*x4*x6 + x5*x6 + x1*x5*x6 + x3*x5*x6 + x1*x3*x5*x6 + x2*x3*x5*x6 + x1*x2*x3*x5*x6 + x4*x5*x6 + x1*x4*x5*x6 + x1*x2*x4*x5*x6 + x1*x3*x4*x5*x6 + x2*x3*x4*x5*x6*

f2: 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 0 1 1 0 1 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 1 0 1 1 0 0 0 0 0 0 1 1 0 1 1 1 0 1 1 0 1 0 1 0

*f2: x1*x2 + x3 + x2*x3 + x1*x2*x3 + x2*x4 + x1*x2*x4 + x1*x3*x4 + x2*x3*x4 + x1*x2*x3*x4 + x5 + x1*x2*x5 + x1*x3*x5 + x2*x3*x5 + x4*x5 + x1*x4*x5 + x1*x2*x4*x5 + x1*x3*x4*x5 + x1*x2*x3*x4*x5 + x1*x2*x6 + x1*x3*x6 + x1*x2*x3*x6 + x4*x6 + x2*x4*x6 + x1*x2*x4*x6 + x2*x5*x6 + x1*x2*x5*x6 + x1*x3*x5*x6 + x2*x3*x5*x6 + x1*x2*x3*x5*x6 + x1*x4*x5*x6 + x2*x4*x5*x6 + x3*x4*x5*x6 + x2*x3*x4*x5*x6*

f3: 1 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 0 0 0 0 1 1 0 1 0 0 1 1 1 0 0 1 1 1 0 1 1 1 1 0 0 1 0 0 1 1 1 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0

*f3: 1 + x1*x3 + x2*x3 + x1*x2*x3 + x1*x4 + x1*x2*x4 + x3*x4 + x1*x3*x4 + x2*x3*x4 + x1*x2*x3*x4 + x3*x5 + x1*x3*x5 + x1*x2*x3*x5 + x2*x4*x5 + x1*x2*x4*x5 + x3*x4*x5 + x1*x2*x3*x4*x5 + x6 + x1*x6 + x1*x2*x6 + x3*x6 + x1*x3*x6 + x2*x3*x6 + x1*x4*x6 + x3*x4*x6 +*

*x1*x3*x4*x6 + x2*x3*x4*x6 + x1*x2*x5*x6 + x1*x3*x5*x6 + x1*x2*x3*x5*x6 + x1*x4*x5*x6 + x3*x4*x5*x6 + x2*x3*x4*x5*x6*

f4: 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 1 1 0 0

*f4: x1*x2 + x2*x3 + x1*x2*x3 + x1*x4 + x2*x3*x4 + x1*x2*x3*x4 + x1*x5 + x2*x5 + x1*x3*x5 + x1*x2*x3*x5 + x4*x5 + x2*x4*x5 + x3*x4*x5 + x1*x3*x4*x5 + x1*x2*x3*x4*x5 + x6 + x2*x6 + x3*x6 + x1*x3*x6 + x2*x3*x6 + x1*x2*x3*x6 + x1*x4*x6 + x1*x3*x4*x6 + x2*x3*x4*x6 + x1*x5*x6 + x1*x2*x5*x6 + x1*x3*x5*x6 + x2*x3*x5*x6 + x1*x2*x3*x5*x6 + x1*x4*x5*x6 + x2*x4*x5*x6 + x1*x2*x4*x5*x6 + x3*x4*x5*x6 + x1*x3*x4*x5*x6*

f5: 1 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 1 0 1 1 1 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 1 0

*f5: 1 + x1 + x1*x2 + x1*x3 + x1*x2*x4 + x3*x4 + x1*x3*x4 + x2*x3*x4 + x5 + x1*x5 + x1*x2*x5 + x3*x5 + x1*x3*x5 + x2*x4*x5 + x2*x3*x4*x5 + x1*x2*x3*x4*x5 + x1*x6 + x2*x6 + x1*x2*x6 + x1*x3*x6 + x2*x3*x6 + x1*x4*x6 + x2*x4*x6 + x1*x2*x5*x6 + x3*x5*x6 + x1*x3*x5*x6 + x2*x4*x5*x6 + x1*x2*x4*x5*x6 + x3*x4*x5*x6 + x2*x3*x4*x5*x6*

f6: 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 1 1 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 0 1 1 0 1 0 1 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0

*f6: x2 + x2*x3 + x1*x2*x4 + x2*x3*x4 + x5 + x1*x5 + x2*x5 + x1*x2*x5 + x1*x3*x5 + x2*x3*x5 + x1*x2*x3*x5 + x1*x2*x4*x5 + x6 + x1*x2*x6 + x1*x3*x6 + x2*x3*x6 + x1*x4*x6 + x1*x2*x4*x6 + x3*x4*x6 + x2*x3*x4*x6 + x5*x6 + x1*x3*x5*x6 + x2*x3*x5*x6 + x2*x4*x5*x6 + x1*x3*x4*x5*x6*

5. Фиктивные переменные для координатных функций

f1: нет фиктивных переменных

f2: нет фиктивных переменных

f3: нет фиктивных переменных

f4: нет фиктивных переменных

f5: нет фиктивных переменных

f6: нет фиктивных переменных

Лабораторная работа №2

Задание:

1. Для f_k найти преобладание (нулей от единиц)

2. Проверить является ли f_k сильноравновероятностной

3. Выяснить, существуют ли запреты для f_k. Построить запрет

Определения:

$P_2(n)$ множества всех булевых функций от n переменных

Говорят, что булева функция *имеет запрет* $y_1, \dots, y_m \in \{0,1\}^m$, если система булевых уравнений: $f(x_1, .., x_{n+i-1}) = y_i, i = 1, .., m$ несовместна.

Несовместная система – система, которая не имеет решений.

Минимальный элемент – элемент, который встречается реже всего в функции.

Алгоритм, используемый для выполнения задания:

Буду рассматривать работу алгоритма на примере функции f_1.

Вектор значений функции f[1]: [1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0]

А для оставшихся функций результат будет представлен в выводе программы.

1) Чтобы найти преобладание, подсчитываем количество 1 в f[1].
   pr = sum(f[1]) # pr = 32
   И вычисляем по формуле $d(f) = 1 - \frac{pr}{2^{6-1}}$
   Преобладание функции f[1] = 0

Т. к. функция f генерируется на s-боксе являющемся подстановкой с одним циклом – функция f по определению будет равновероятной. Это

доказывается в пункте 1, т.к. преобладание для всех функций f будет = 0 => частота встречаемости 0 и 1 одинакова. Если все элементы встречаются одинаково, то последовательность равновероятностная.

2) Найдём запреты длинной меньше или равной 20.

   Согласно критерию Сумарокова: функция не имеет запретов, если функция является сильно равновероятной. Поэтому построив запрет или не найдя его – мы автоматически узнаем сильно равновероятная функция или нет.

   Для построения запрета последовательно изменяем таблицу истинности добавляя ветки 0 и 1 для каждой строчки и каждого элемента функции.

   Приведём пример для построения запрета для функции длинны 3

   $f = x_1x_2x_3$

x_1	x_2	x_3			f		
0	0	0	0	0	0	0	0
				1			0
			1	0		0	0
				1			0
0	0	1	0	0	0	0	0
				1			0
			1	0		0	0
				1			1
0	1	0	0	0	0	0	0
				1			0
			1	0		0	0
				1			0
0	1	1	0	0	0	0	0
				1			0
			1	0		1	0
				1			1
1	0	0	0	0	0	0	0
				1			0
			1	0		0	0
				1			0
1	0	1	0	0	0	0	0
				1			0
			1	0		0	0
				1			1
1	1	0	0	0	0	0	0
				1			0
			1	0		0	0
				1			0
1	1	1	0	0	1	0	0
				1			0
			1	0		1	0
				1			1

Заметим, что выходную последовательность 101 нам не удастся встретить ни разу, следовательно, это и есть запрет функции f.

3) На каждой итерации происходит поиск последовательности среди значений f, которая встречается 0 раз. Если таковая последовательность нашлась, то она и будет запретом.
   Если такой последовательности не найдено – добавляем новое ветвление таблицы. Делаем так до тех пор пока не будет найдена не встречающаяся последовательность или длинна последовательности превысит 20 – в этом случае будем считать, что у функции нет запрета.

Вывод полученный при выполнении лабораторной работы 2, основанный на результатах лабораторной работы 1.

Вывод работы программы представлен ниже, для экономии места представлен вывод лишь преобладания и запрета.

Вывод программы:

1. Преобладание нулей над единицами для координатных функций

f1: 0.000000
f2: 0.000000
f3: 0.000000
f4: 0.000000
f5: 0.000000
f6: 0.000000

2. Сильно равновероятность для координатных функций

И

3. Запреты для координатных функций

f1: запрет = 010010011001
функция НЕ сильно равновероятная

f2: запрет = 0011010010

функция НЕ сильно равновероятная

f3: запрет = 11100000000

функция НЕ сильно равновероятная

f4: запрет = 01010010

функция НЕ сильно равновероятная

f5: запрет = 100111101001

функция НЕ сильно равновероятная

f6: запрет = 0101100010

функция НЕ сильно равновероятная

Лабораторная работа №3

Задание:

1. Для $f_i, i = \{1, \dots, 6\}$ найти все k, при которых f_i является корреляционно иммунной порядка k, эластичной порядка k;
2. Построить спектр булевой функции для f_i
3. Найти наилучшее линейное приближение для f_i
4. Выяснить, является ли f_i бент-функцией

Определения

Корреляционная иммунность булевых функций означает отсутствие какой-либо статистической зависимости значения данной функции от значений некоторого подможества её аргументов или от значений определенных функций её аргументов.

С геометрической точки зрения *корреляционно-иммунная функция* порядка k, зависящая от n переменных, удалена от всех аффинных функций, отличных от констант и зависящих от k и менее переменных, на расстояние 2^{n-1} , то есть совпадает с ними ровно на половине входных наборов.

Порядком корреляционной иммунности *cor(F)* вектора функции F называется минимальный порядок корреляционной иммунности. Для вычисления cor(F) нужно найти ненулевой вектор и минимального веса, на котором преобразование Уолша - Адамара хотя бы одной компоненты принимает ненулевое значение.

Функция от одной или нескольких переменных называется *аффинной*, если она может быть представлена как линейная комбинация этих переменных плюс константа.

Пусть $a = (a_1, .., a_6), b = (b_1, .., b_6)$, тогда $(a, b) = a_1 b_1 + .. + a_6 b_b$ – *скалярное произведение.* Оно будет использоваться для нахождения линейной характеристики S-бокса.

Линейное приближение – приближение произвольной функции линейной функцией.

Статистическая структура двоичной функции

Определим для функции f(x)и линейной функции $(a, x) =$

$a_1 x_1 + .. + a_n x_n$ параметр исходя из равенства $P\big(f(x) = (a, x)\big) = \frac{1}{2} + \frac{\Delta_a^f}{2^n}$

Набор чисел $\{\Delta_a^f\}, a$ из F_2^n называется *статистической структурой* двоичной функции f(x).

Статистическая структура показывает насколько отличаются от ½ вероятности совпадения значений функции f со значениями линейных функций(a,x). Способ нахождения коэффициентов описан далее по тексту.

Корреляция (от лат. correlatio), *корреляционная зависимость* — взаимозависимость двух или нескольких случайных величин. Суть ее заключается в том, что при изменении значения одной переменной происходит закономерное изменение (уменьшению или увеличению) другой(-их) переменной(-ых).

Алгоритм, используемый для выполнения задания:

1) В начале программой считаются коэффициенты Фурье по следующей формуле:

   $\overline{w_f}(\vec{u}) = \frac{1}{2^n}\sum_{\vec{x}\in V_n} f(\vec{x})\chi_{\vec{u}}(\vec{x})$;

   $\chi_{\vec{u}}(\vec{x}) = (-1)^{\langle \vec{u},\vec{x}\rangle}$.

На последнем шаге получены коэффициенты Фурье для исследуемой функции f.

2) Далее считаются коэффициенты Уолша-Адамара:

$$\mathrm{w_f}(\bar{u}) = \begin{cases} 1 - 2\overline{w_f}(\bar{u}); \bar{u} = \bar{0} \\ -2\overline{w_f}(\bar{u}); \bar{u} \neq \bar{0} \end{cases}$$

Первый (нулевой) элемент считается по формуле: $1 - 2\overline{W}_f(\bar{u}), \bar{u} = \bar{0}$. В рассматриваемой функции получается, что 1 элемент коэффициентов Уолша-Адамара равен $1 - 2 * 0.5 = 0$

Остальные коэффициенты Уолша-Адамара – коэффициенты Фурье, умноженные на -2.

3) Считаем коэффициенты статистической структуры:

   Коэффициенты вычисляются по формуле $\Delta_u^f = W_f(\bar{u})2^{n-1}$.

$x_1 \dots x_6$	$\Delta_{\bar{u}}^{f_1}$	$\Delta_{\bar{u}}^{f_2}$	$\Delta_{\bar{u}}^{f_3}$	$\Delta_{\bar{u}}^{f_4}$	$\Delta_{\bar{u}}^{f_5}$	$\Delta_{\bar{u}}^{f_6}$
000000	0	0	0	0	0	0
000001	2	2	2	8	-2	-8
000010	-10	-4	4	2	0	2

000011	0	-2	-2	2	6	-2
000100	-2	-4	-8	-6	6	-4
000101	8	-2	6	2	-8	0
000110	0	4	0	-8	2	2
000111	-6	6	-2	0	-4	-6
001000	-2	6	-2	-6	-8	-4
001001	-12	4	0	2	2	0
001010	0	-6	-6	8	-4	-2
001011	-2	0	4	0	-2	-2
001100	4	2	-2	0	6	0
001101	2	0	4	0	4	0
001110	2	2	-2	2	-2	6
001111	0	-8	4	-6	4	2
010000	-6	4	0	0	-8	12
010001	4	2	-6	4	-6	0
010010	-4	8	-8	2	0	2
010011	-2	-10	2	-2	-6	2
010100	0	4	0	6	-6	-4
010101	2	2	6	2	0	4
010110	-2	4	4	-4	-2	-2
010111	0	2	2	-8	-4	2
011000	0	2	2	2	0	0
011001	6	-4	-4	-2	-2	8
011010	-10	-2	2	0	-4	-2
011011	4	0	-4	-4	2	-6
011100	-2	10	2	4	2	0
011101	-4	4	0	0	4	4
011110	0	2	-2	-2	2	2
011111	-2	4	4	2	-4	2
100000	4	-2	-6	6	-2	0
100001	-2	0	-4	2	-8	0
100010	-2	2	2	4	2	6
100011	0	4	-4	0	4	2
100100	-2	2	-6	0	4	0
100101	0	4	-8	-4	2	-4
100110	4	10	-2	2	4	10
100111	6	-4	-4	6	-6	2
101000	-6	-4	-4	4	2	0
101001	0	2	-2	0	0	4
101010	0	0	4	6	2	6
101011	-2	-2	-2	-6	-8	-2
101100	-4	-8	-4	2	0	0

101101	-6	-2	2	6	2	0
101110	-2	0	0	0	4	-6
101111	-4	-2	6	4	-2	-2
110000	2	2	2	6	2	4
110001	4	0	4	-2	0	0
110010	0	-2	-2	-4	-2	6
110011	-6	-4	0	4	4	6
110100	-4	2	-6	4	-4	-8
110101	6	0	-8	4	-2	0
110110	6	2	10	6	4	-2
110111	0	0	0	-2	-2	2
111000	0	0	0	-4	-2	4
111001	6	2	2	-4	0	-4
111010	2	-4	-4	6	6	-2
111011	0	6	-2	-2	0	2
111100	2	0	0	-2	-8	0
111101	0	2	6	-2	-2	-4
111110	0	0	0	-4	4	6
111111	-2	-6	-2	-4	2	-2

4) Проверяем корреляционную иммунность и эластичность функции:

Функция корреляционно иммунна порядка k, если все коэффициенты Уолша-Адамара для векторов веса k равны нулю, поэтому для начала проверяем коэффициенты Уолша-Адамара при векторах, содержащих 1 единицу. Если все коэффициенты при них равны 0, то функция корреляционно иммунна порядка 1. Если хоть один не равен 0, то функция корреляционно иммунна порядка 0 (т.е. не корреляционно иммунна).

Для $f_i, i \in \{1, \dots, 6\}$ проверяем вектора 000001, 000010, 000100, 001000, 010000, 100000:

$W_f(000001) \neq 0$ (иначе $\Delta^f_{(000001)} = 0) \Rightarrow$ функция не корреляционно иммунна

Эластичная порядка k функция — это сбалансированная корреляционно иммунная порядка k функция, иными словами она корреляционно имунна порядка k и преобладание равно 0.

Получается, порядок эластичности не может быть больше порядка корреляционной иммунности, но добавляется требование сбалансированности (т.е. преобладание равно нулю).

Функции $f_i, i \in \{1, \dots, 6\}$ не является корреляционно иммунными, а значит, не являются эластичными.

5) Проверим, является ли функция бент-функцией:

Функция должна быть от чётного количества переменных и все коэффициенты Уолша-Адамара должны быть равны по модулю.

Функции $f_i, i \in \{1, \dots, 6\}$ является функцией от 6 переменных, но ее коэффициенты Уолша-Адамара не равны по модулю (иначе коэффициенты статистической структуры были бы равны по модулю) ⇒ не является бент-функцией.

6) Считаем наилучшее линейное приближение

Для нахождения наилучшего линейного приближения функции от наименьшего числа переменных, в столбце коэффициентов статистической структуры выбирается наибольшее по модулю, и на основе соответствующей строки таблицы истинности строится функция (если коэффициент отрицательный, то добавляется слагаемое $\oplus 1$).

Пример:

u = 001 -> $g = x_3$

u = 010 -> $g = x_2$

u = 110 -> $g = x_1 \oplus x_2$

Из всех этих значений находится максимальный.

Далее считается соответствующий линейный многочлен.

Перевод числа в двоичный вид (001001) и с помощью цикла перевод в вид многочлена: $f_1 = x_3 \oplus x_6 \oplus 1$

Данный полином является наилучшим линейным приближением от наименьшего числа переменных.

Узнаем, с какой вероятностью результаты исходной функции и данного многочлена совпадут по формуле:

$$\Delta_u^f = 2^6 \left(P(f(\bar{x}) = <\bar{u}, \bar{x}>) - \frac{1}{2} \right), \text{где нужно найти } P$$

$$P(f_i(\bar{x}) = <\bar{u}, \bar{x}>) = \frac{1}{2^6} \max\left(\left|\Delta_u^{f_i}\right|\right) + \frac{1}{2}$$

Для f_1:

$$P(f(\bar{x}) = <\bar{u}, \bar{x}>) = 0.6875$$

Результат работы программы:

1. Корреляционная иммунность и эластичность для координатных функций

f1: функция НЕ корреляционно иммунна порядка 1
функция НЕ эластична

f2: функция НЕ корреляционно иммунна порядка 1
функция НЕ эластична

f3: функция НЕ корреляционно иммунна порядка 1
функция НЕ эластична

f4: функция НЕ корреляционно иммунна порядка 1
функция НЕ эластична

f5: функция НЕ корреляционно иммунна порядка 1
функция НЕ эластична

f6: функция НЕ корреляционно иммунна порядка 1

функция НЕ эластична

2. Построение спектра булевых функций для координатных функций

x1...x6	del(f1)	del(f2)	del(f3)	del(f4)	del(f5)	del(f6)
000000	0	0	0	0	0	0
000001	2	2	2	8	-2	-8
000010	-10	-4	4	2	0	2
000011	0	-2	-2	2	6	-2
000100	-2	-4	-8	-6	6	-4
000101	8	-2	6	2	-8	0
000110	0	4	0	-8	2	2
000111	-6	6	-2	0	-4	-6
001000	-2	6	-2	-6	-8	-4
001001	-12	4	0	2	2	0
001010	0	-6	-6	8	-4	-2
001011	-2	0	4	0	-2	-2
001100	4	2	-2	0	6	0
001101	2	0	4	0	4	0
001110	2	2	-2	2	-2	6
001111	0	-8	4	-6	4	2
010000	-6	4	0	0	-8	12
010001	4	2	-6	4	-6	0
010010	-4	8	-8	2	0	2
010011	-2	-10	2	-2	-6	2
010100	0	4	0	6	-6	-4
010101	2	2	6	2	0	4
010110	-2	4	4	-4	-2	-2
010111	0	2	2	-8	-4	2
011000	0	2	2	2	0	0
011001	6	-4	-4	-2	-2	8
011010	-10	-2	2	0	-4	-2
011011	4	0	-4	-4	2	-6

011100	*-2*	*10*	*2*	*4*	*2*	*-8*
011101	*-4*	*4*	*0*	*0*	*4*	*4*
011110	*0*	*2*	*-2*	*-2*	*2*	*2*
011111	*-2*	*4*	*4*	*2*	*-4*	*2*
100000	*4*	*-2*	*-6*	*6*	*-2*	*0*
100001	*-2*	*0*	*-4*	*2*	*-8*	*0*
100010	*-2*	*2*	*2*	*4*	*2*	*6*
100011	*0*	*4*	*-4*	*0*	*4*	*2*
100100	*-2*	*2*	*-6*	*0*	*4*	*0*
100101	*0*	*4*	*-8*	*-4*	*2*	*-4*
100110	*4*	*10*	*-2*	*2*	*4*	*10*
100111	*6*	*-4*	*-4*	*6*	*-6*	*2*
101000	*-6*	*-4*	*-4*	*4*	*2*	*0*
101001	*0*	*2*	*-2*	*0*	*0*	*4*
101010	*0*	*0*	*4*	*6*	*2*	*6*
101011	*-2*	*-2*	*-2*	*-6*	*-8*	*-2*
101100	*-4*	*-8*	*-4*	*2*	*0*	*0*
101101	*-6*	*-2*	*2*	*6*	*2*	*0*
101110	*-2*	*0*	*0*	*0*	*4*	*-6*
101111	*-4*	*-2*	*6*	*4*	*-2*	*-2*
110000	*2*	*2*	*2*	*6*	*2*	*4*
110001	*4*	*0*	*4*	*-2*	*0*	*0*
110010	*0*	*-2*	*-2*	*-4*	*-2*	*6*
110011	*-6*	*-4*	*0*	*4*	*4*	*6*
110100	*-4*	*2*	*-6*	*4*	*-4*	*0*
110101	*6*	*0*	*-8*	*4*	*-2*	*0*
110110	*6*	*2*	*10*	*6*	*4*	*-2*
110111	*0*	*0*	*0*	*-2*	*-2*	*2*
111000	*0*	*0*	*0*	*-4*	*-2*	*4*
111001	*6*	*2*	*2*	*-4*	*0*	*-4*
111010	*2*	*-4*	*-4*	*6*	*6*	*-2*
111011	*0*	*6*	*-2*	*-2*	*0*	*2*
111100	*2*	*0*	*0*	*-2*	*-8*	*0*

111101 0 2 6 -2 -2 -4

111110 0 0 0 -4 4 6

111111 -2 -6 -2 -4 2 -2

3. Нахождение наилучшего линейного приближения для координатных булеых функций

f1: g_1(x) = x3 + x6 + 1

f2: g_1(x) = x2 + x5 + x6 + 1

g_2(x) = x2 + x3 + x4

g_3(x) = x1 + x4 + x5

f3: g_1(x) = x1 + x2 + x4 + x5

f4: g_1(x) = x6

g_2(x) = x4 + x5 + 1

g_3(x) = x3 + x5

g_4(x) = x2 + x4 + x5 + x6 + 1

f5: g_1(x) = x4 + x6 + 1

g_2(x) = x3 + 1

g_3(x) = x2 + 1

g_4(x) = x1 + x6 + 1

g_5(x) = x1 + x3 + x5 + x6 + 1

g_6(x) = x1 + x2 + x3 + x4 + 1

f6: g_1(x) = x2

4. Является ли координатная функция бент-функцией

f1: данная координатная функция НЕ является бент-функцией

f2: данная координатная функция НЕ является бент-функцией

f3: данная координатная функция НЕ является бент-функцией

f4: данная координатная функция НЕ является бент-функцией

f5: данная координатная функция НЕ является бент-функцией

f6: данная координатная функция НЕ является бент-функцией

Лабораторная работа №4

Задание

Сгенерировать гамму в генераторе из 6 регистров, в котором функция усложнения берется из прошлых работ. Потом корреляционным методом по гамме восстановить начальное состояние регистров

Определения

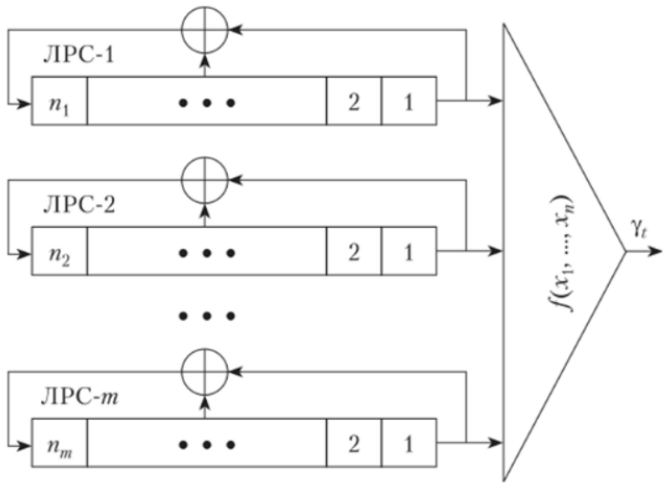
Рассмотрим преобразование $f(X): GF(2)^n \to GF(2)^n$

n – четное, $GF(2)^n -$ множество n-мерных двоичных векторов

Преобразование является биекцией и полученный вектор формируется некоторым генератор псевдослучайных последовательностей со свойством равновероятной последовательности. Функция f – функция усложнения.

Комбинирующий генератор – усложнение фильтрующего генератора и построен на ЛРС-1,..,ЛРС-m над P, m>0 и комбинирующей функции $f(x_1, .., x_T)$.

Комбинирующий генератор представлен на Рисунке ниже.



Линейная сложность последовательности - минимальная длина ЛРС (линейного регистра сдвига), который способен сгенерировать данную последовательность.

Фильтрующие и комбинирующие генераторы относятся к схемам с равномерным движением регистра (все регистры сдвигаются ровно на один знак каждый такт работы).

Алгоритм, используемый для выполнения задания

В качестве булевой функции f используем: *[0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1]*

Регистры сдвига линейной обратной связи (РСЛОС) имеют следующие многочлены обратной связи:

$$h_1(x) = x^7 \oplus x \oplus 1$$
$$h_2(x) = x^{11} \oplus x^2 \oplus 1$$
$$h_3(x) = x^{13} \oplus x \oplus 1$$
$$h_4(x) = x^{17} \oplus x^2 \oplus 1$$
$$h_5(x) = x^{19} \oplus x \oplus 1$$
$$h_6(x) = x^{23} \oplus x^2 \oplus 1$$

А их начальное состояние имеет вид:

Регистр 1: 1 1 1 0 0 0 1

Регистр 2: 0 1 0 1 1 0 0 0 1 1 0

Регистр 3: 0 0 0 0 0 1 0 0 0 0 0 0 0Ъ

Регистр 4: 0 0 0 1 1 1 0 0 1 1 1 1 0 1 0 1 1

Регистр 5: 0 0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 1 1 1

Регистр 6: 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0

Согласно корреляционному анализу, найдём вероятность линейных приближений по формуле:

$$Pf(x) = g(x) = 1/2 + \Delta_{f_g}/2^n$$

где Δ_{f_g} – коэффициент стат структуры линейной функции g.

Линейные приближения функции будут иметь вид:

$$g_1(x_1, \dots, x_6) = x_1$$

$$g_2(x_1, \dots, x_6) = x_2$$

...

$$g_6(x_1, \dots, x_6) = x_6$$

Поскольку функция f вырабатывает ключевую гамму, тогда

$$f\left(x_1^{(t)}, \dots, x_6^{(t)}\right) = \gamma_t, t \in N$$

В этом обозначении $x_1^{(t)}, \dots, x_6^{(t)}$- t-ое значение гаммы из РСЛОС с многочленами обратной связи $h_1(x), \dots, h_6(x)$. Соответственно:

$$P\{\gamma_t = x_1^{(t)}\} = \frac{1}{2} + \frac{2}{64} = 0.531250$$

...

$$P\{\gamma_t = x_6^{(t)}\} = \frac{1}{2} + \frac{6}{64} = 0.593750$$

Построим статистический критерий для различения двух гипотез о распределении случайной величины:

$$\xi_t = x_t \oplus \gamma$$

H_0: $\xi \sim \beta(q_0)$ – начальное состояние ложное.

H_1: $\xi \sim \beta(q_1)$ – начальное состояние соответствует истинному ключу, где $q_0 = \frac{1}{2}$.

Тогда гипотеза H_1 для случайных величин будет иметь вид следующие обозначения:

$$\xi_1 = x_1 \oplus \gamma \sim \beta(q_1)$$

...

$$\xi_6 = x_6 \oplus \gamma \sim \beta(q_6)$$

Для различия двух гипотез выберем вероятность ошибок первого и второго рода. В нашем случае $\alpha = \beta = 0.0000001$. Также воспользуемся квантиль $\chi_{(1-\alpha)}$ и $\chi_{(1-\beta)}$ для стандартного нормального распределения.

Так как значения $(1-\alpha)$ и $(1-\beta)$ выходят за пределы таблицы стандартного нормального распределения – будем досчитывать таблицу программно вычисляя интеграл $\int_0^x e^{-\frac{y^2}{2}} dy$.

Объем материала (меньше или равен длине γ), необходимый для того, чтобы различить теории, считаем по формуле:

$$T = \frac{\left(\chi_{(1-\alpha)}\sqrt{q(1-q)} + \chi_{(1-\beta)}\sqrt{p(1-p)}\right)^2}{(q-p)^2}$$

$$T = 28925$$

А для вычисления границы критерия C формулой:

$$\mathrm{C} = \chi_{(1-\beta)}\sqrt{\left(Tp(1-p)\right)} + T(1-p)$$

$$\mathrm{C} = 14010.1$$

При вычислении в качестве p используем $1 - q_1$, а в качестве q: $q_0 = 0.5$

С этой константой будет сравниваться количество несовпадений.

Далее перебираем начальное заполнение генератора. Начальное заполнение неизвестно, известно только строение генератора. Вырабатываем T значений.

Сравниваем полученные значения с выработанными ранее значениями γ и считаем количество несовпадений. Если оно меньше константы C, то записываем число в возможное заполнение.
При подборе кандидатов использовались следующие параметры:

№	α	β	$\xi(1-\alpha)$	$\xi(1-\beta)$	p	q	T	C
1	0.0000001	0.0000001	5.320000	5.320000	0.468750	0.5	28925.02	14010.1
…	…	…	…	…	…	…	…	…
6	0.0000001	0.0000001	5.320000	5.320000	0.406250	0.5	3163.33	1432.06

В регистре 1 было принято положительное решение по 1 кандидату:
1 1 1 0 0 0 1

...

В регистре 6 было принято положительное решение по 1 кандидату:
0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0

Так как в каждом регистре было одобрено по одному кандидату имеем единственное возможное заполнение ключа:
1 0 0 0 1 1 1 0 1 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 1 1 1 0 0 1 1
1 0 0 0 1 1 1 0 1 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0
0 0
С программной реализацией можно ознакомиться в приложении А.

Результат работы программы:

Подбираем функцию f удовлетворяющую условиям...

Функция генерации гаммы: 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 0 0 1 1 0 1 1 1 0 1
0 1 1 0 0 1 1 1 1 1 0 0 0 0 0 1 1 1

Регистр 1: 1 1 1 0 0 0 1

Регистр 2: 0 1 0 1 1 0 0 0 1 1 0

Регистр 3: 0 0 0 0 0 1 0 0 0 0 0 0 0

Регистр 4: 0 0 0 1 1 1 0 0 1 1 1 1 0 1 0 1 1

Регистр 5: 0 0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 1 1 1

Регистр 6: 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0

Истинный ключ: 1 0 0 0 1 1 1 0 1 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 1 1 1 0 0 1 1 1 0 0 0 1 1 1 0
1 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0

q1 = 0.531250 p1 = 0.421875

q2 = 0.625000 p2 = 0.421875

q3 = 0.531250 p3 = 0.421875

q4 = 0.625000 p4 = 0.421875

q5 = 0.593750 p5 = 0.421875

q6 = 0.593750 p6 = 0.421875

Подибираем кандидатов в регистр 1

Fa(0.999999900)=5.320000, Fb(0.999999900)=5.320000 alp=0.000000100, bet=0.000000100, X(1-a)=5.320000, X(1-b)=5.320000, p=0.468750, q=0.5, T=28925.025107

T = 28925

C = 14010.1

прогресс: 128/128 положительное решение по: 1 кандитату(-ам)

candidate 1: 1 1 1 0 0 0 1

Успешно!

Подибираем кандидатов в регистр 2

Fa(0.999999900)=5.320000, Fb(0.999999900)=5.320000 alp=0.000000100, bet=0.000000100, X(1-a)=5.320000, X(1-b)=5.320000, p=0.375000, q=0.5, T=1754.292191

T = 1754.29

C = 765.734

прогресс: 2048/2048 положительное решение по: 1 кандитату(-ам)

candidate 1: 0 1 0 1 1 0 0 0 1 1 0

Успешно!

Подибираем кандидатов в регистр 3

Fa(0.999999900)=5.320000, Fb(0.999999900)=5.320000 alp=0.000000100, bet=0.000000100, X(1-a)=5.320000, X(1-b)=5.320000, p=0.468750, q=0.5, T=28925.025107

T = 28925

C = 14010.1

прогресс: 8192/8192 положительное решение по: 1 кандитату(-ам)

candidate 1: 0 0 0 0 0 1 0 0 0 0 0 0 0

Успешно!

Подибираем кандидатов в регистр 4

Fa(0.999999900)=5.320000, Fb(0.999999900)=5.320000 alp=0.000000100, bet=0.000000100, X(1-a)=5.320000, X(1-b)=5.320000, p=0.375000, q=0.5, T=1754.292191

T = 1754.29

C = 765.734

прогресс: 131072/131072 положительное решение по: 1 кандитату(-ам)

candidate 1: 0 0 0 1 1 1 0 0 1 1 1 1 0 1 0 1 1

Успешно!

Подибираем кандидатов в регистр 5

Fa(0.999999900)=5.320000, Fb(0.999999900)=5.320000 alp=0.000000100, bet=0.000000100, X(1-a)=5.320000, X(1-b)=5.320000, p=0.406250, q=0.5, T=3163.326155

T = 3163.33

C = 1432.06

прогресс: 524288/524288 положительное решение по: 1 кандитату(-ам)

candidate 1: 0 0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 1 1 1

Успешно!

Подибираем кандидатов в регистр 6

Fa(0.999999900)=5.320000, Fb(0.999999900)=5.320000 alp=0.000000100, bet=0.000000100, X(1-a)=5.320000, X(1-b)=5.320000, p=0.406250, q=0.5, T=3163.326155

T = 3163.33

C = 1432.06

прогресс: 8388608/8388608 положительное решение по: 1 кандитату(-ам)

candidate 1: 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0

Успешно!

Выбранный ключ: 1 0 0 0 1 1 1 0 1 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 1 1 1 0 0 1 1 1 0 0 0 1 1 1 0 1 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0

Алгоритм ошибся на 0 бит

Приложение А.
Программная реализация доступна по ссылке:
https://disk.yandex.ru/d/3xldk_jpKwOHfQ