ОЦЕНКА КЛЮЧЕЙ КОМБИНАЦИОННОГО УСТРОЙСТВА ВСЕВОЗМОЖНЫХ ПЕРЕСТАНОВОК

Кохновский С. И., Иванюк А. А.
Кафедра информатики, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: stan.ver.i.esk.slav@gmail.com, ivaniuk@bsuir.by

В работе приведена методика построения комбинационного запутывающего устройства и рассмотрена зависимость Хэмминга расстояния между векторами, которые поступают на вход и выход комбинационного устройства, от свойств ключа и входного вектора.

Введение

Актуальность защиты авторских прав объектов интеллектуальной собственности обусловлена возрастающей конкуренцией на рынке продуктов сферы информационных технологий. В качестве одного из методов нарушения прав интеллектуальной собственности может быть использовано обратное проектирование интегральных схем [1]. Одним из способов повышения сложности обратного проектирования интегральной схемы может выступать внедрение в схему запутывающего устройства.

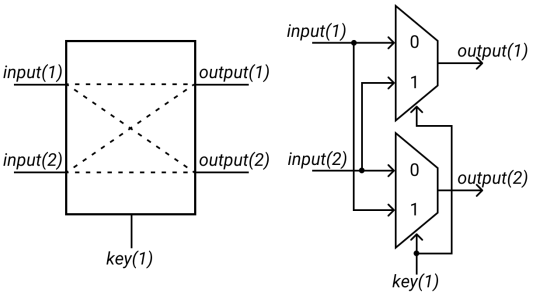

Рис. 1 – Блок

I. Устройство всевозможных перестановок

Запутывающим устройством называют цифровое устройство, задающее отображение множества входных сигналов I на множество выходных O в зависимости от подаваемого ключа из множества ключей K. Множества I, O, K – счётные и конечные, причём $|I| = |O| = n$ и $|K| = m$. Мощности множеств связаны соотношением $m = \frac{n}{2}(n-1)$.

На входные линии комбинационной схемы подаются векторы $input = (i_1, i_2, \ldots, i_n), input \in I$ и $key = (k_1, k_2, \ldots, k_m), key \in K$. На выходные линии схема возвращает вектор $output = (o_1, o_2, \ldots, o_n), output \in O$ – перестановку символов входного вектора. Устройство с простейшей конфигурацией (блок), имеющее параметры $n = 2$ и $m = 1$, принимает на вход векторы $input$ и key, а возвращает вектор $output$. Вариантом реализации и описания логики работы блока могут быть два мультиплексора (см. рис. 1).

II. Методика построения устройства-кодера

Возможно обобщить схему построения устройства на произвольную размерность (см. рис. 2). Пусть блоки размещаются на линиях $\{1,2\},\{2,3\},\ldots,\{n-1,n\}$, далее происходит размещение блоков на линиях $\{n-2,n-1\},\{n-3,n-2\},\ldots,\{1,2\}$, что позволяет обеспечить достижимость всех выходов из входов $input(1)$ и $input(n)$. Под линиями будем понимать отрезок, соединяющий $input(i)$ с $output(i)$. Далее повторяются эти же действия: размещение блоков на линиях $\{2,3\},\{3,4\},\ldots,\{n-1,n\},\{n-2,n-1\},\{2,3\}$, позволяющее передать сигнал с входов $input(2)$, $input(n-1)$ на любой из $n-2$ выходов. Устройство сконфигурировано, когда не осталось входов, для которых не обеспечена достижимость любого выхода. Назовём полученное устройство кодером, переставляющим символы входного вектора в зависимости от поступающего ключа.

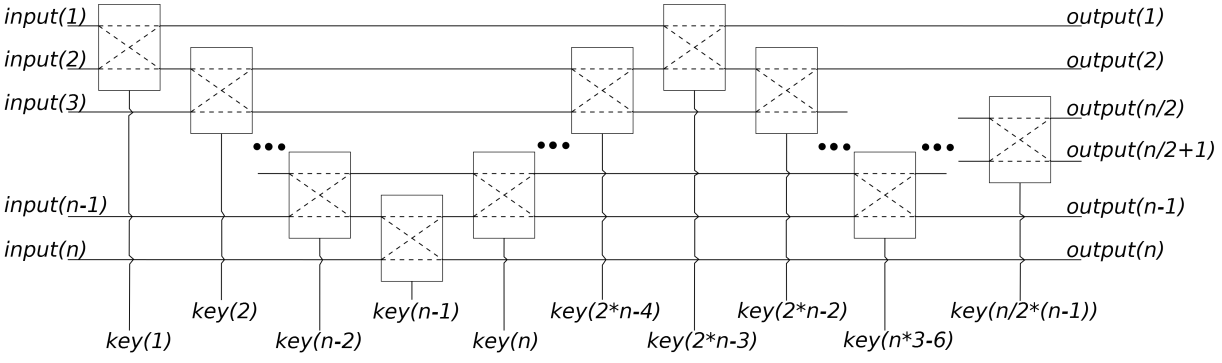

Рис. 2 – Комбинационное устройство размерности $n = k * 2, k \in N^*$

III. Методика построения устройства-декодера

Для построения декодера рассматривается устройство размерности $n = 3$ (см. рис. 3). В этом случае при передаче бит ключа в обратном порядке зашифрованная последовательность будет расшифрована устройством-кодером.

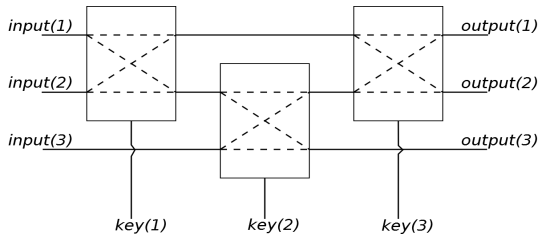

Рис. 3 – Устройство размерности $n = 3$

При изменении логики работы устройства на обратную (подаче бит закодированного вектора на выходы устройства, снятии результата с входов устройства и неизменных конфигурации устройства и ключе), закодированная последовательность будет декодирована, что справедливо для устройств произвольной размерности. Устройство-декодер может быть использовано в качестве кодера для вектора *input* с произвольным ключом, тогда как устройство-кодер, построенное по вышеописанному алгоритму, будет являться декодером для устройства-декодера на этом же ключе.

IV. Оценка Хэммингових расстояний между векторами *input* и *output*

Построим таблицу отображения вектора *input* на вектор *output* размерности $n = 3$ в зависимости от вектора *key* (таблица 1).

Таблица 1 – Всевозможные векторы *output* для устройства размерности 3

	input							
key	000	001	010	011	100	101	110	111
000	000	001	010	011	100	101	110	111
001	000	001	100	101	010	011	110	111
010	000	010	001	011	100	110	101	111
011	000	100	001	101	010	110	011	111
100	000	001	100	101	010	011	110	111
101	000	001	010	011	100	101	110	111
110	000	010	100	110	001	011	101	111
111	000	100	010	110	001	101	011	111

Введём расстояние по Хэммингу $D_H = d(input, output)$ между векторами *input* и *output*, а также среднее расстояние по Хэммингу $D_H^* = \frac{1}{n}\sum_{i=1}^{n} d(input_i, output_i)$ в зависимости от ключа *key*. Замерим D_H в зависимости от подаваемого вектора *key*, также рассчитаем D_H^* для каждой строки и вычислим среднее расстояние для каждого столбца (таблицы 2, 3).

Таблица 2 – Таблица D_H для устройства размерности 3

	input							
key	000	001	010	011	100	101	110	111
000	0	0	0	0	0	0	0	0
001	0	0	2	2	2	2	0	0
010	0	2	2	0	0	2	2	0
011	0	2	2	2	2	2	2	0
100	0	0	2	2	2	2	0	0
101	0	0	0	0	0	0	0	0
110	0	2	2	2	2	2	2	0
111	0	2	0	2	2	0	2	0

Таблица 3 – Таблица D_H^* относительно всевозможных *key* и средних расстояний для всевозможных *input*

№	000	001	010	011	100	101	110	111
input	0	1	1.25	1.25	1.25	1.25	1	0
key	0	1	1	1.5	1	0	1.5	1

Интересными также являются максимальные средние расстояния по Хэммингу для различных размерностей в зависимости от ключа (таблица 4).

Таблица 4 – Таблица максимальных D_H^* для различных размерностей

Размерность	3	4	5	6
D_H^*	1.5	1.88235	2.42424	2.95385

Выводы

В результате анализа таблиц Хэмминговых расстояний различных размерностей получены следующие выводы:

- входные векторы *input* имеют наибольшее среднее расстояние от выходных векторов *output* для всевозможных ключей, если соотношение количеств единичных и нулевых бит в векторе *input* максимально близко к 1;
- ключи, обеспечивающие наибольшее среднее расстояние в зависимости от всевозможных векторов *input*, имеют не менее $n - 1$ единичных бит;
- в отдельную группу могут быть выделены ключи, обеспечивающие работу блока на каждой паре линий нечётное количество раз, что положительно сказывается на среднем Хэмминговом расстоянии для подобных ключей.

Список литературы

1. Privacy protection of VLSI circuits through high level transformation based obfuscation [Electronic resource] / S. Bhuvaneswari, A. Hemamalini, A. Anbazhagan. – Global Journal of Pure and Applied Mathematics, 2016. – Mode of access: https://www.researchgate.net/publication/319135767_PRIVACY_PROTECTION_OF_VLSI_CIRCUITS_THROUGH_HIGH_LEVEL_TRANSFORMATION_BASED_OBFUSCATION. – Date of access: 18.10.2020.