МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Сибирский государственный университет науки и технологий
имени академика М.Ф. Решетнева»**

Институт Информатики и телекоммуникаций
Направление Информационная безопасность
Профиль Безопасность автоматизированных систем
Кафедра Безопасности информационных технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Вид ВКР: бакалаврская работа

ПОВЫШЕНИЕ КАЧЕСТВА КЛЮЧЕВОЙ ИНФОРМАЦИИ НА ОСНОВЕ СВОЙСТВ БУЛЕВЫХ ФУНКЦИЙ

Обучающийся _____ К.П. Спирин

Руководитель _____ О.Н. Жданов

Ответственный за нормоконтроль _____ Н.И. Чугунова

Допускается к защите

Заведующий кафедрой _____ В.В. Золотарев

«_____» _____ 20__ г.

Красноярск 2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Сибирский государственный университет науки и технологий
имени академика М.Ф. Решетнева»**

Информатики и телекоммуникаций
Безопасности информационных технологий

УТВЕРЖДАЮ
Заведующий кафедрой
_____ В.В. Золотарев
« ____ » _____ 20 ___ г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме бакалаврской работы
Обучающийся Спирин Константин Петрович
Группа БКБ16-01 Направление 10.03.01
Информационная безопасность
Тема выпускной квалификационной работы Повышение качества ключевой информации на основе свойств булевых функций

Утверждена приказом по университету от _____№_____
Руководитель ВКР – О.Н. Жданов, доцент, кандидат ф.-м. н., доцент кафедры БИТ
Исходные данные для ВКР свойства булевых функций, методика выбора ключевой информации, алгоритмы шифрования.

Перечень разделов ВКР
1 Описание булевых функций
2 Разработка методики повышения стойкости при генерации блоков замен на основе булевых функций
3 Реализация методики повышения стойкости при генерации блоков замен на основе булевых функций

Срок сдачи обучающимся первого варианта ВКР – « ___ » _____ 20__ г.
Срок сдачи обучающимся окончательного варианта ВКР – « ___ » _____ 20__ г.

Руководитель ВКР _____ О.Н. Жданов

Задание принял к исполнению _____ К.П. Спирин

« ___ » _____ 20__ г.

АННОТАЦИЯ

к бакалаврской работе

«Повышение качества ключевой информации на основе свойств булевых функций»

Спирин Константин Петрович

Ключевые слова: блок замен, ключевая информация, булевы функции, коэффициенты матрицы корреляции, расстояние нелинейности.

Целью данной работы является разработка и реализация методики повышения стойкости при генерации блоков замен на основе булевых функций.

Данная цель определила необходимость постановки и решения основных задач:

– изучение теоретических сведений о булевых функциях;

– разработка методики для повышения стойкости при генерации блоков замен на основе булевых функций;

– реализация методики повышения стойкости и исследование полученных результатов.

Предмет исследования – свойства строгого лавинного критерия, корреляционные свойства таблиц замен.

По результатам исследования найдена зависимость между двумя критериями качества блоков замен – расстоянием нелинейности и статистической зависимости выхода блока замен от его входа. Полученные результаты обеспечат повышение защищенности блоков замен при получении ключевой информации.

Работа включает: 53 страниц, 2 таблицы, 9 рисунков, 3 приложения. Использованных источников – 9.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Зачастую в открытых каналах связи появляется необходимость скрыть информацию от посторонних глаз. Для этого применяется шифрование информации – преобразование открытой информации в зашифрованную, называемую шифртекстом.

Для шифрования используется ключ. В стандарте ГОСТ 28147-89 понятие ключ определено следующим образом: «Конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования, обеспечивающее выбор одного преобразования из совокупности всевозможных для данного алгоритма преобразований». Информация, которую мы зашифруем с использованием данного ключа может быть расшифрована только с использованием этого же ключа или ключа, определенно с ним связанным.

Ключевая информация является одним из важнейших элементов шифрования, с ее помощью злоумышленник может получить зашифрованную информацию, в связи с этим является актуальным рассмотрение вопроса повышение качества ключевой информации на основе свойств булевых функций, так как именно они применяются в большинстве криптоалгоритмов.

Цель данной работы – разработка и реализация методики повышения стойкости при генерации блоков замен на основе булевых функций.

Для достижения указанной выше цели необходимо выполнение следующих задач:

– изучение теоретических сведений о булевых функциях;

– разработка методики для повышения стойкости при генерации блоков замен на основе булевых функций;

– реализация методики повышения стойкости и исследование полученных результатов.

Работа содержит 53 страниц, состоит из введения, трех глав, заключения, списка использованных источников из 9 источников; основной текст включает 9 рисунков и 2 таблицы.

1 ОПИСАНИЕ БУЛЕВЫХ ФУНКЦИЙ

1.1 Применение булевых функций в криптографии

Своим существованием двоичная логика обязана британскому математику Джорджу Булю. Джордж Буль родился в Англии 2 ноября 1815 года. Всю свою жизнь он работал учителем математики и физики в школе. Ученый стремился сделать науку о мышлении, логику, такой строгой как, например, математика. Для этого Буль стал обозначать высказывания в виде букв и составлял из них уравнения, которые были схожи с алгебраическими и с их помощью можно было определить истинность или ложность высказывания. Так появилась Булева алгебра.

Булевой функцией называется отображение $\{0,1\}^k \rightarrow \{0,1\}$, т. е. правило, однозначно сопоставляющее вектору из бит значение 0 или 1. [7]

Именно булевы функции применяются в большинстве систем шифрования. К этим функциям предъявляется ряд требований, целью которых является осложнение дешифровки сообщения лицом, не являющимся его адресатом. На рисунке 1 представлен шифр Вернама.

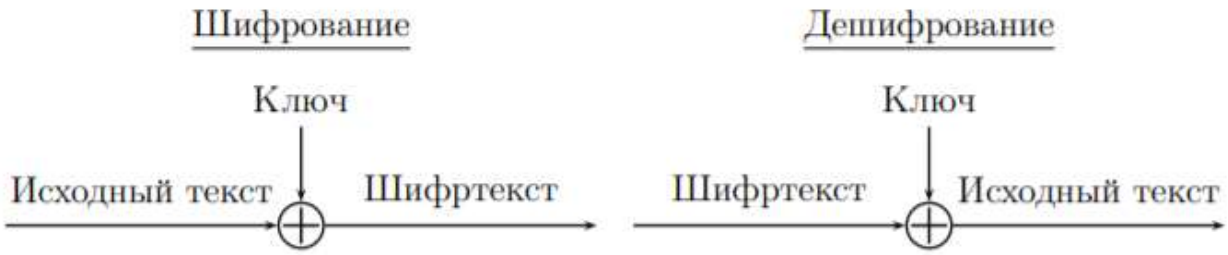

Рисунок 1 – Шифр Вернама

Исходный текст, ключ, шифртекст в данном случае это бинарные строки одинаковой длины. Операция $\oplus$ означает побитовое сложение по модулю 2. Справа изображено дешифрование – оно аналогично шифрованию, но шифртекст и исходный текст меняются местами. Именно поэтому дважды шифр текст не используют – при сложении двух шифртекстов, соответствующих одному ключу, получается сумма исходных текстов, которые зачастую можно прочесть.

Шеннон доказал, что при совершенно случайном ключе, используемом один раз, шифр Вернама является абсолютно стойкой криптосистемой, то есть

перехват шифртекста не даёт никакой информации о переданном сообщении. Это единственный в настоящее время шифр с таким свойством [2].

Однако на практике, как правило, отправитель и получатель сообщений выбирают вместо ключа в шифре Вернама псевдослучайную последовательность, которая по оговорённому алгоритму генерируется из короткого секретного ключа. Такая последовательность носит название ключевой поток или гамма.

1.2 Принципы Шеннона

Клод Шеннон превратил криптографию из искусства в науку, так как появилась возможность доказывать защищенность информации при помощи шифра. Используя вероятностную модель шифра, он впервые сформулировал понятие совершенно стойкого шифра и показал, что он существует, на примере шифра Вернама, называемого также одноразовым блокнотом.

По К. Шеннону, шифрование должно использовать следующие принципы:

– рассеивание (Diffusion) – распространение влияния одного знака открытого текста на много знаков шифртекста, а также распространение влияния одного элемента ключа на много знаков шифртекста;

– перемешивание, усложнение, запутывание (Confusion) – свойство шифрующего преобразования усложнять взаимосвязи между элементами данных, что затрудняет восстановление функциональных и статистических связей между открытым текстом, ключом и шифртекстом [3].

Эти два общих принципа конструирования криптосистемы – очень общие и неформальные. Шеннон также описал более специфичные принципы конструирования.

Первый заключается в том, чтобы свести проблему обеспечения секретности системы к одной из известных вычислительно сложных задач. Этот принцип часто используется при создании криптосистем с открытым ключом, но не используется для криптосистем с секретным ключом.

Второй принцип Шеннона заключается в том, чтобы сделать систему устойчивой ко всем известным атакам, и это до сих пор лучший из известных принципов построения криптосистем с секретным ключом.

Так, по Шеннону, противник, прослушивающий канал, по которому идет передача данных, может быть двух типов:

– пассивный (прослушивающий): криптоаналитик пытается вычислить открытый текст (а еще лучше – ключ) по шифртексту;

– активный (взламывающий): криптоаналитик пытается активно воздействовать на передаваемые данные. Например, пытается изменить передаваемый шифртекст.

Чтобы не позволить противнику понять, как легитимные участники передачи данных выработали свой общий ключ, нужно выполнять следующие условия:

– все ключи должны быть равновероятными и всегда выбираться из ключевого пространства случайным образом. Часто предполагают, что все детали используемого отправителем и получателем криптоалгоритма известны противнику, кроме конкретного значения секретного ключа;

– условие Керкхоффса. Противник знает все детали алгоритмов зашифрования и дешифрования, кроме конкретного значения секретного ключа [3].

Более полувека, минувшие с момента формулирования принципов Шеннона, подтвердили их значимость. За эти годы предпринимались различного вида атаки на криптосистемы, в связи с которыми появились основные криптографические характеристики булевых функций, некоторые из которых больше относятся к рассеиванию, другие больше к запутыванию.

Все эти характеристики надо учитывать при конструировании булевых функций. Требуется компромисс между ними, так как булева функция не может быть оптимальна сразу по всем криптографическим показателям.

1.3 Свойства булевых функций

Дадим определение булевой функции от n переменных. Пусть $Z_2 = \{0, 1\}$. Через Z^n_2 будем обозначаем множество всех двоичных векторов $v = (v_1, ..., v_n)$ длины n. Пусть все векторы упорядочены, т.е. при $n = 2$ порядок будет следующий: (00), (01), (10), (11).

Произвольная функция из множества Z^n_2 в Z_2 называется булевой функцией от n переменных [4]. Например,

$f : Z^2_2 \rightarrow Z_2$ такая, что $f(00) = 1$, $f(01) = 0$, $f(10) = 0$, $f(11) = 1$. (1)

Для булевых функций действуют следующие законы:

1. $a + b = b + a$ – коммутативность сложения.

2. $a + (b + c) = (a + b) + c$ – ассоциативность сложения.

3. $\exists 0 \in K(a + 0 = 0 + a = a)$ – существование нейтрального элемента относительно сложения.

4. $\forall a \in K\ \exists b \in K(a + b = b + a = 0)$ – существование противоположного элемента относительно сложения.

5. $(a \times b) \times c = a \times (b \times c)$ – ассоциативность умножения.

6. $a \times (b + c) = (a \times b) + (a \times c)$ и $(b + c) \times a = (b \times a) + (c \times a)$ – два закона дистрибутивности.

1.4 Нелинейность булевых функций

Каждая функция имеет единственное представление в виде алгебраической нормальной формы (АНФ), в отечественной литературе распространен также термин «полином Жегалкина».

Степенью нелинейности $deg(f)$ булевой функции f называется число переменных в самом длинном слагаемом ее АНФ. Функция называется аффинной, квадратичной, кубической и т.д., если ее степень равна соответственно 1, 2, 3 и т. д. Каждая аффинная функция от n переменных $v_1,...,v_n$ имеет вид $<u,v> \oplus a$ для подходящих вектора u и константы a. [4]

Пусть $<u,v>=u_1v_1 \oplus ... \oplus u_nv_n$ – скалярное произведение векторов. Через $dist(f, g)$ обозначим расстояние Хэмминга между функциями f и g, т. е. число позиций, в которых различаются векторы f и g.

Максимально нелинейной называется булева функция от n переменных (n любое) такая, что расстояние Хэмминга N_f от данной функции до множества всех аффинных функций является максимально возможным. Величину N_f называют нелинейностью булевой функции. В случае четного n максимально возможное значение нелинейности равно $2^{n-1}-2^{(n/2)-1}$. [4]

Термин «максимально нелинейная функция» принят в русскоязычной литературе, тогда как в англоязычной широкое распространение получил термин «бент-функция». При этом при четном числе переменных n бент-функции и

максимально нелинейные функции совпадают, а при нечетном n бент-функции (в отличие от максимально нелинейных) не существуют.

Для криптографических целей булева функция не должна быть линейной (точнее, аффинной). Вообще, чем меньше функция «похожа на аффинную», тем лучше. В это неформальное пожелание можно вложить различные смысловые оттенки.

Вот некоторые из них:

1. «Функция с хорошей нелинейностью» далека от множества аффинных функций в смысле какой-либо метрики.

2. «Функция с хорошей нелинейностью» должна выражаться полиномом как можно более высокой степени.

3. «Функция с хорошей нелинейностью» не должна линейно зависеть ни от одной из своих переменных и не должна приобретать такую зависимость после какой-либо линейной замены переменной.

Это свойство формулируют так: функция не должна иметь ненулевых линейных структур. Собственно термин «нелинейность» принят для показателя нелинейности, использующего понятие расстояния Хэмминга.

1.5 Строгий лавинный критерий

Лавинный эффект – криптографическое свойство для шифрования, из которого следует, что изменение малого количества битов во входном тексте или ключе приведет к изменению значений выходных битов шифртекста и отображает зависимость всех выходных битов от каждого входного.

Лавинный критерий, основанный на лавинном эффекте, требует изменения в среднем половины бит в выходном (зашифрованном) значении при изменении каждого отдельно взятого бита во входном (исходном) значении [5].

Математик Эмиль Пост ввел следующие замкнутые классы булевых функций:

- T_0 – сохраняющие константу 0;
- T_1 – сохраняющие константу 1;
- SS – самодвойственные функции;
- M – монотонные функции;
- L – линейные функции.

Теорема Поста (о полноте) гласит, что набор булевых функций является полным тогда и только тогда, когда он не содержится полностью ни в одном из классов *S, M, L, T_0, T_1*.

То есть набор полон, когда в нем имеется хотя бы одна функция, не сохраняющая ноль, хотя бы одна функция, не сохраняющая один, хотя бы одна несамодвойственная функция, хотя бы одна немонотонная функция и хотя бы одна нелинейная функция.

Строгий лавинный критерий (СЛК) впервые был введен Вебстером и Таваресом и комбинировал полноту и лавинный эффект булевых функций.

Функция *f* удовлетворяет строгому лавинному критерию, если для всех *i, j* $\in$ (1, 2,...,*n*) изменение входного бита *i* изменяет выходной бит *j* с вероятностью ровно в половину. Таким образом *S*-блок удовлетворяет строгому лавинному критерию если

$$\frac{1}{2^n} W\left(a_j^{e_i}\right) = \frac{1}{2}, \text{ для всех } i,j. \quad (2)$$

где

$$W\left(a_j^{e_i}\right) = \sum_{X \in \{0;1\}^n} a_j^{e_i} \quad (3)$$

это общее изменение j-й лавинной переменной $a_j^{e_i}$, вычисленное по всему входному алфавиту размера 2^n.

Данная формула может быть изменена для определения параметра СЛК $k_{СЛК}(i,j)$:

$$k_{СЛК}(i,j) = \frac{1}{2^n} W\left(a_j^{e_i}\right) = \frac{1}{2} \quad (4)$$

$k_{СЛК}(i,j)$ может принимать значения в диапазоне [0,1] и это следует интерпретировать как вероятность изменения *j*-го выходного бита при изменении *i*-го бита во входной строке. Если $k_{СЛК}(i,j)$ отличается на ½ для любой пары (*i, j*), то говорят, что *S*-блок не удовлетворяет СЛК. Однако точное выполнение данного критерия для всех значений *i* и *j* не является реалистичным ожиданием и, скорее всего, следует ожидать некую погрешность [6].

Очевидно, что лавинный критерий и СЛК похожи и соответственно *S*-блок, который удовлетворяет СЛК, должен также удовлетворять лавинному критерию, но удовлетворение этому критерию не подразумевает удовлетворение строгому лавинному критерию.

Для оценки криптографического качества булевых функций используют один из следующих критериев:

– алгебраическая степень нелинейности;

– расстояние нелинейности;

– критерий распространения ошибки порядка m, частный случай – строгий линейный критерий;

– матрица коэффициентов корреляции векторов выхода и входа *S*-блока.

1.6 Выводы

В данной главе были рассмотрены булевы функции и их свойства. Были описаны принципы Шеннона. Было рассмотрено применение булевых функций в криптографии и перечислены критерии их криптографического качества.

Данная информация будет использована при разработке и реализации методики повышения стойкости при генерации блоков замен на основе булевых функций.

2 РАЗРАБОТКА МЕТОДИКИ ПОВЫШЕНИЯ СТОЙКОСТИ ПРИ ГЕНЕРАЦИИ БЛОКОВ ЗАМЕН НА ОСНОВЕ БУЛЕВЫХ ФУНКЦИЙ

2.1 ГОСТ 28147-89

ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования» – это советский и российский стандарт симметричного шифрования. Он был введен в 1990 году. Является *DES*-подобной криптосистемой, созданной на основе классической итерационной схеме Фейстеля.

ГОСТ 28147-89 – это блочный шифр с 256-битным ключом и 32 циклами (раундами) преобразования, оперирующий 64-битными блоками.

2.2 Структура алгоритма ГОСТ 28147-89

В основе алгоритма заложена сеть Фейстеля (рисунок 2).

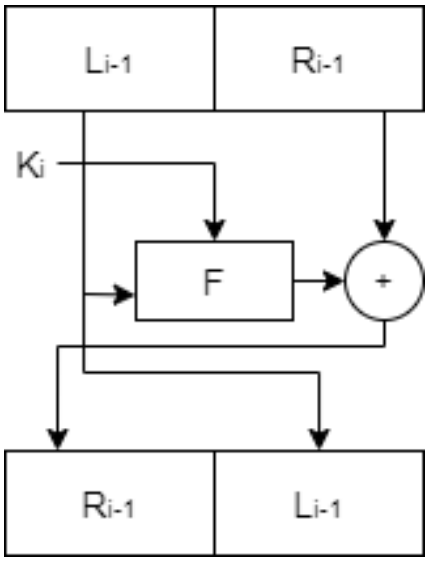

Рисунок 2 – Сеть Фейстеля

Работа данной схемы:

– каждый блок разбивается на два подблока – левый и правый;

– исходное заполнение правого блока записывается в левый блок на выходе;

– над правым блоком производится криптографическое преобразование с применением ключевых данных;

– левый (исходный) и правый (преобразованный) блоки складываются по модулю 2 в сумматоре по модулю 2;

– данная процедура повторяется несколько раз.

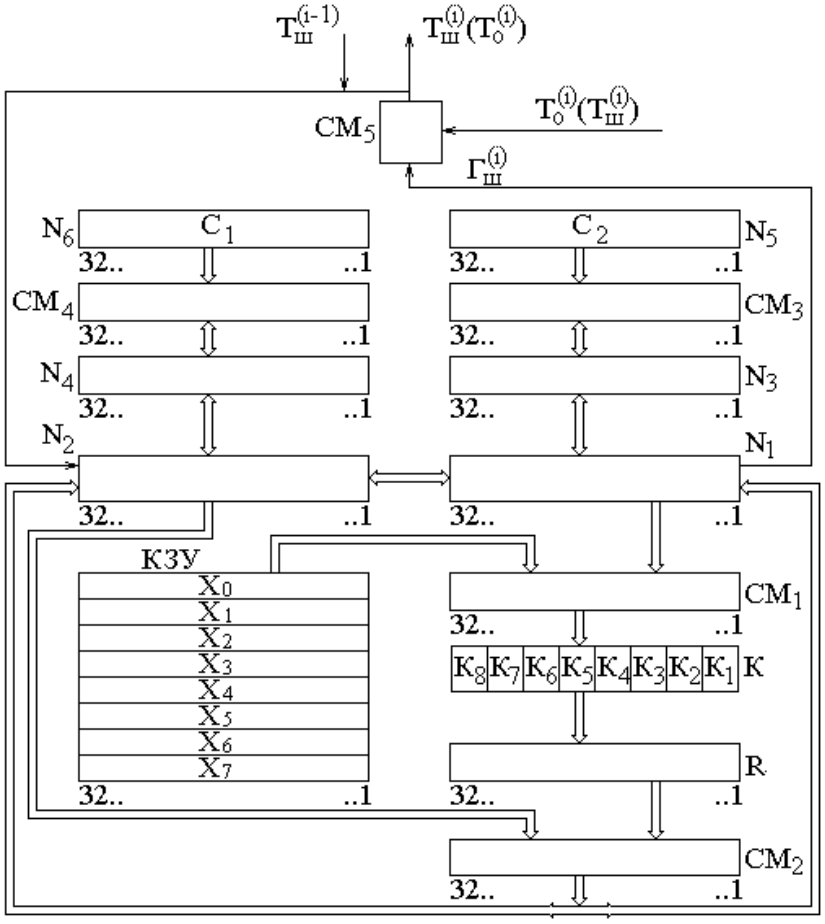

Рисунок 3 – Структурная схема алгоритма

Данная схема содержит:

– четыре накопителя по 32 бита: N_1, N_2, N_3, N_4.

– два 32-разрядных накопителя: N_5 и N_6, – с записанными в них постоянными заполнениями C_2 и C_1 соответственно.

– ключевое запоминающее устройство (КЗУ) на 256 бит. КЗУ состоит из восьми накопителей по 32 разряда каждый: X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7.

– 32-разрядный сумматор по модулю 2: CM_2.

– еще один сумматор по модулю 2, который не имеет ограничения на разрядность (но используется 64 бита): CM_5.

– два сумматора по модулю 2^{32} разрядности 32 бита: CM_1, CM_3.

– сумматор по модулю (2^{32}-1): CM_4.

– блок подстановки К: восемь узлов замены K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8, каждый с памятью на 64 бита.

– регистр циклического сдвига влево на 11 бит R. [8]

Под ключевое запоминающее устройство (КЗУ) определено 256 бит. Ключ разбивается на 8 блоков по 32 бита и каждый бит каждого блока последовательно вводится в накопитель X соответствующего порядка. Первый бит ключа вводится в первый разряд накопителя X_0, 33-й бит ключа в первый разряд накопителя X_1, 256-й бит ключа вводится в 32-й разряд накопителя X_7.

Считывается ключ в зависимости от режима работы алгоритма.

Блок подстановки K содержит в себе таблицу замены размером 16x8, она является долговременным ключом.

Строки определяют «что» нужно заменить (число от 0 до 15 в шестнадцатеричной системе счисления). Столбцы указывают «на что» заменять. Поступающий таким образом в блок 32-битовый вектор разбивается на восемь четырехбитовых блока, каждый из которых преобразуется в соответствии с таблицей замены.

Ключи как в КЗУ, так и в блоке подстановки K, являются секретными, и требуются меры по сохранению их в секретности.

Используя данные компоненты ГОСТ 28147-89 способен работать в четырех режимах работы:

– простая замена;

– гаммирование;

– гаммирование с обратной связью;

– режим выработки имитовставки.

ГОСТ не определяет способы генерации S-блоков, поэтому произведем исследование: сгенерируем блоки замен и найдем закономерность между некоторыми их криптографическими качественными характеристиками.

2.3 Примеры блоков замен

Приведем примеры хороших и плохих блоков замен. Возьмем следующий блок замен: S = [0 1 2 4 3 5 7 6]. Для того чтобы определить его качество вычислим его матрицу коэффициентов корреляции:

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & 0 \end{bmatrix}$$

В данной матрице отсутствует единица, поэтому данная матрица может считаться качественной.

Теперь приведем блок замен с плохими корреляционными характеристиками: S = [1 0 2 3 4 5 6 7].

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

Наличие в данной матрице единиц, тем более на главной диагонали, свидетельствует о наличии корреляции и плохом криптографическом качестве блока.

2.4 Генерация блоков замен

Блоком замен $k_1 \times k_2$ бит называется отображение, т.е. отображение $\{0,1\}^{k1} \rightarrow \{0,1\}^{k2}$, однозначно сопоставляющее любому входному вектору из бит выходной вектор из бит [7].

То есть блок замен – это набор булевых функций, создающих зависимость битов на выходе от битов на входе *S*-блока. Для оценки криптографических свойств *S*-блока его представляют в виде его компонентных булевых функций.

При генерации *S*-блоков важно учитывать то, что используемая булева функция, должна быть сбалансированной, то есть такой функцией, которая на всей своей области определения принимает значение 0 столько же раз сколько 1.

Помимо сбалансированности функция должна удовлетворять строгому лавинному критерию (критерий распространения m). Булева функция *f(x)*, где *x* – вектор из n переменных, удовлетворяет СЛК, если при изменении одного из n входных битов выходной бит меняется с вероятностью ровно 1/2.

Пример булевой функции трех переменных, для которой выполняется СЛК приведен в таблице 1.

Таблица 1. Пример булевой функции удовлетворяющей СЛК

Входные биты *x*	000	001	010	011	100	101	110	111
Выходной бит *f(x)*	1	1	1	0	1	0	0	0

В данной работе будут сгенерированы S-блоки, основанные на булевых функциях, и удовлетворяющие строгому лавинному критерию и сбалансированности.

2.5 Критерии криптографического качества

2.5.1 Расстояние нелинейности

Расстояние нелинейности *S*-блока принято определять как степень удаленности, в смысле некоторой метрики, его компонентных функций от множества функций, принятых за линейные.

Вычисляется расстояние нелинейности по следующей формуле:

$$N_f = 2^{k-1} - \frac{1}{2} max\{|S|\} \quad (5)$$

В данной формуле $max\{|S|\}$ – это максимально возможная длина величины линейности.

2.5.2 Матрица коэффициентов корреляции

Данный критерий определяется следующим образом: каждый бит исходного вектора *y* должен быть статистически независимым от каждого бита входного вектора *x*. Количественно степень линейной статистической (корреляционной) связи между выходными и входными битами описывается с помощью корреляционной матрицы. [7]

Матрицей коэффициентов корреляции *S*-блока $\rho = ||\rho_{v,\mu}||$, $v, \mu = \overline{1,\ k}$ длины $N = p^k$ называется матрица, элементы которой вычисляются в соответствии со следующей формулой:

$$\rho_{v,\mu} = \frac{\sum_{t=1}^{\mathbb{N}} x_{v,t} y_{\mu,t} - \frac{\sum_{t=1}^{\mathbb{N}} x_{v,t} \sum_{t=1}^{\mathbb{N}} y_{\mu,t}}{N}}{\sqrt{\left(\sum_{t=1}^{\mathbb{N}} (x_{v,t})^2 - \frac{(\sum_{t=1}^{\mathbb{N}} x_{v,t})^2}{N}\right) \times \left(\sum_{t=1}^{\mathbb{N}} (y_{\mu,t})^2 - \frac{(\sum_{t=1}^{\mathbb{N}} y_{\mu,t})^2}{N}\right)}} \quad (6)$$

где μ, v — номера компонентных функций исследуемого *S*-блока и тривиальной подстановки 0,1, ..., $N-1$; $k = log_p N$ — количество компонентных функций.

Для хорошего шифра характерно отсутствие корреляции между битами выхода и входа либо равномерное распределение коэффициентов корреляции.

2.6 Оценка качества блоков замен

Для оценки качества сгенерированных *S*-блоков будет произведено вычисление матрицы коэффициентов нужно представить их в виде компонентных функций и рассчитать коэффициент корреляции между значениями на входе и выходе *S*-блока. Как можно больше коэффициентов должно быть равно 0, а значение максимума модулей коэффициентов иметь малое значение.

Также *S*-блоки для обеспечения криптографической стойкости шифра блоки замен должны быть высоконелинейными.

2.7 Выводы

Таким образом для генерирования надежных S-блоков необходимо учитывать следующие моменты:

– использовать функции, удовлетворяющие строгому лавинному критерию и сбалансированности;

– генерация на основе этих функций *S*-блоков, имеющих наиболее качественные коэффициенты корреляционной матрицы;

– проверить данные *S*-блоки на расстояние нелинейности, постараться выявить взаимосвязь между этими качественными криптографическими характеристиками.

Далее будет написана программа, выполняющая данные задачи, и проведено исследование полученных результатов.

3 РЕАЛИЗАЦИЯ МЕТОДИКИ ПОВЫШЕНИЯ СТОЙКОСТИ ПРИ ГЕНЕРАЦИИ БЛОКОВ ЗАМЕН НА ОСНОВЕ БУЛЕВЫХ ФУНКЦИЙ

3.1 Реализация методики посредством программы

Для реализации методики, описанной во второй главе, была разработана программа код которой содержится в Приложении А. Код написан на языке C++ в среде разработки Microsoft Visual Studio 2019.

Данная программа вычисляет функции, удовлетворяющие строгому лавинному критерию и сбалансированности, и на их основании создает блоки замен для которых вычисляются матрица коэффициентов корреляции и расстояние нелинейности. Также для исследования были отдельно сгенерированы блоки замен без строгого лавинного критерия.

3.1.1 Генерация функций для блоков замен

На первом этапе своей работы программа выписывает в отдельный файл функции, которые удовлетворяют сбалансированности и строгому лавинному критерию методом перебора функций от первой сбалансированной функции до последней. Функции трех переменных, удовлетворяющие представлены в таблице ниже.

Таблица 2 – Функции, удовлетворяющие сбалансированности и СЛК

00001111	00111100	01101001	10011001	11000101
00010111	01000111	01101010	10011010	11000110
00011011	01001011	01101100	10011100	11001001
00011101	01001101	01110001	10100011	11001010
00011110	01001110	01110010	10100101	11001100
00100111	01010011	01110100	10100110	11010001
00101011	01010101	01111000	10101001	11010010
00101101	01010110	10000111	10101010	11010100
00101110	01011001	10001011	10101100	11011000
00110011	01011010	10001101	10110001	11100001
00110101	01011100	10001110	10110010	11100010
00110110	01100011	10010011	10110100	11100100
00111001	01100101	10010101	10111000	11101000
00111010	01100110	10010110	11000011	11110000

Список функций для четырех переменных представлен в Приложении Б.

3.1.2 Генерация блоков замен

S-блок называется биективным, если его кодирующая *Q*-последовательность содержит все элементы последовательности 0, 1, ..., *N*-1. [7]

На основе полученных на первом шаге СЛК-функций происходит генерация *S*-блоков, при этом учитывается биективность.

Q-последовательности прошедшие проверку выписывались во второй файл и передавались далее для вычисления их криптографического качества.

3.1.3 Матрица коэффициентов корреляции

Для исследования качества полученных *S*-блоков будем исследовать следующие критерии криптографического качества функций:

– матрица коэффициентов корреляции;

– расстояние нелинейности;

Для расчета матрицы коэффициентов корреляции использовалась ранее описанная формула (6).

В программе в первую очередь подсчитывались отдельные суммы входных и выходных векторов, которые затем подставлялись в общую формулу. Результат представлен во втором файле в виде матрицы.

3.1.4 Расстояние нелинейности

Расстояние нелинейности *S*-блока принято определять как степень удаленности, в смысле некоторой метрики, его компонентных функций от множества функций, принятых за линейные.

Для вычисления расстояния нелинейности использовалась формула:

$$N_f = 2^{k-1} - \frac{1}{2} max\{|S|\} \quad (7)$$

Из этой формулы следует, что чем меньше максимум модуля коэффициента *S*, тем выше нелинейность функции *f*.

Для вычисления нелинейности S-блока q-значной логики используется формула:

$$NL = \begin{cases} q^k - max\{|S|\}, q > 2; \\ 2^{k-1} - \frac{1}{2}\{|S|\}, q = 2. \end{cases} \quad (8)$$

Данное выражение является определением q-нелинейности S-блока.

В качестве примера рассмотрим один из S-блоков ГОСТ 34.12-2018 «Магма»:

S = [6 8 2 3 9 10 5 12 1 14 4 7 11 13 0 15]

Представим в виде компонентных булевых функций:

$$F_2 = \begin{bmatrix} 0100110101001101 \\ 1000001101110101 \\ 1011010001011001 \\ 0001101010011101 \end{bmatrix} \quad (9)$$

Получим нелинейность $NL = \min\{4,4,4,4\} = 4$

В программе расстояние нелинейности записывается во второй файл после матрицы коэффициентов корреляции.

3.1.5 Схема Кима

С помощью схемы Кима из S-блоков длины N = 8 были получены блоки длинной N = 16. Схема Кима осуществляет идею двухэтапного построения блоков, то есть позволяет из небольших S-блоков построить блоки нужной длины, при сохранении их оптимальности.

Получаем следующий алгоритм работы:

1. Переборным методом находим все малые блоки замен, они должны удовлетворять критерию нулевой корреляции между блоками входа и выхода.
2. Применяем схему Кима, получаем увеличение длины блока в 2 раза.
3. Производим перестановку столбцов в обратном порядке, сохраняя таким образом оптимальность и увеличивая количество блоков замен.
4. Если требуемая длина не достигнута, то повторно применяем схему Кима [9].

Для полученных S-блоков вычисляем матрицу коэффициентов корреляции и расстояние нелинейности.

3.1.6 Обобщенная схема работы программы

В результате получим обобщённую схему работы программы, представленную на рисунке 4.

Рисунок 4 – Схема работы программы

3.1.7 Тестирование программы

Программа была протестирована для генерации блоков замен на основе булевых функций, удовлетворяющих сбалансированности и строгому лавинному критерию для трех переменных. Программа не предполагает действий со стороны пользователя, однако в параметрах можно увеличить количество пе-

ременных, что повлечет за собой значительное увеличение времени работы программы. При количестве переменных $k = 3$, работа программы занимает около 5 минут. Для большего количества переменных время работы увеличивается в несколько раз и занимает более часа. Причина увеличения времени кроется в значительном увеличении количества булевых функций, так для трех переменных все функции перечислены в таблице 2, а лишь половина из всех функций для четырех переменных приведены в Приложении Б.

Время работы программы для k продемонстрировано на рисунке 5.

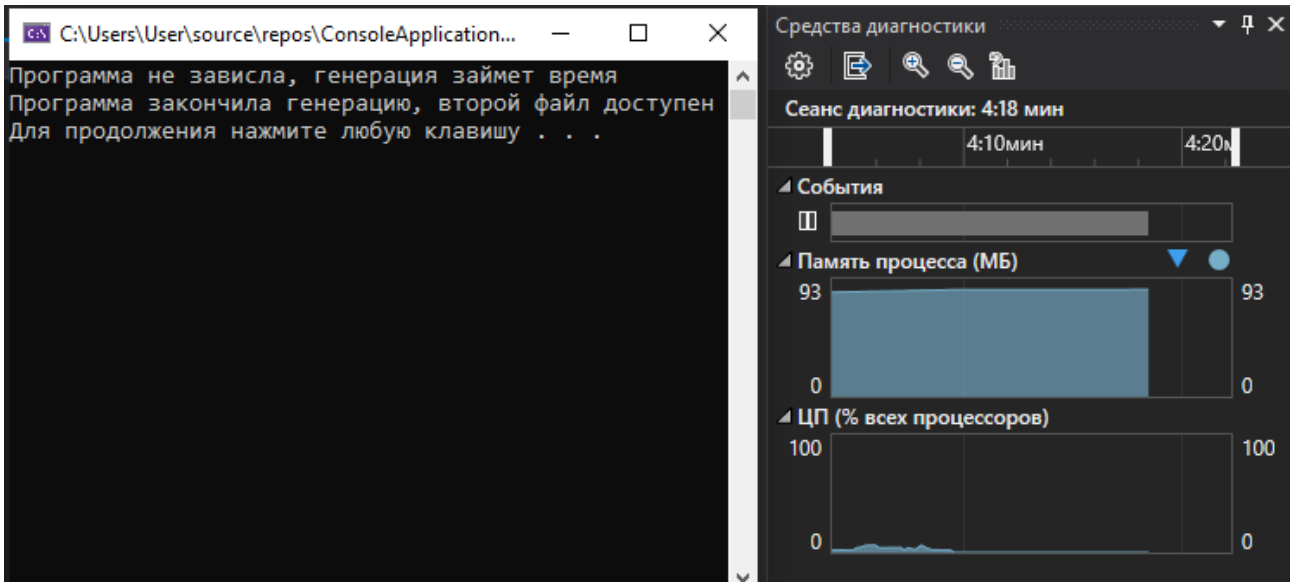

Рисунок 5 – Время работы программы для трех переменных

Поэтому было принято решение генерировать блоки для трех переменных длинной $N = 8$, которые впоследствии увеличиваются с помощью схемы Кима до $N = 16$.

3.2 Исследование полученных результатов

В результате работы программы получили два файла, в одном из них содержатся функции, удовлетворяющие сбалансированности и строгому лавинному критерию, все эти функции были приведены в таблице 2. На основе этих функций были сгенерированы блоки замен, содержащиеся во втором файле. Также во втором файле содержатся данные о матрице коэффициентов корреляции и расстоянии нелинейности. Пример содержимого второго файла приведен на рисунке 6.

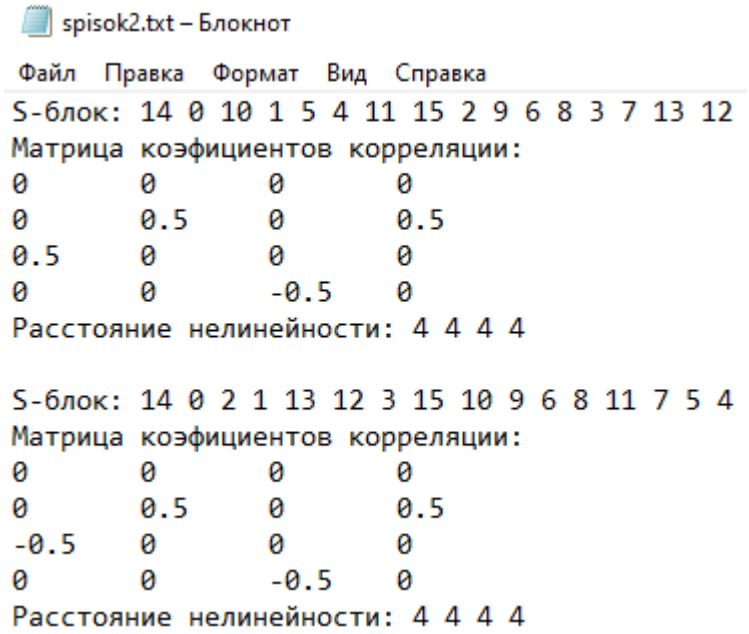

Рисунок 6 – Блок замен и его параметры

Выбранные критерии оценки качества являются одними из основных и позволяют достаточно полно оценить пригодность того или иного блока замен для выполнения своих задач. Именно поэтому важно найти зависимость между расстоянием нелинейности и корреляционной матрицей, постараться определить их взаимное влияние друг на друга и определить рекомендации по выбору блоков замен.

В первую очередь, согласно полученным данным расстояние нелинейности зависит от длины блока замен, чем он длиннее, тем выше $max\{|S|\}$ для формулы расстояния нелинейности и тем больше удаленность компонентных функций от линейных. Пример расстояния нелинейности для булевых функций трех и четырех переменных приведен на рисунке 7.

```
S-блок: 1 2 0 4 3 6 5 7
Матрица коэфициентов корреляции:
0.5     0.5     0.5
0.5     -0.5    0
0.5     0.5     -0.5
Расстояние нелинейности: 2 2 2

S-блок: 9 2 0 4 11 6 13 15 10 1 12 8 14 3 7 5
Матрица коэфициентов корреляции:
0       0       0       0
0       0.5     0.5     0.5
0       0.5     -0.5    0
-0.5    0       0       0
Расстояние нелинейности: 4 4 4 4
```

Рисунок 7 – Расстояние нелинейности

Для дальнейших исследований были внесены изменения в программу и сгенерирован еще один файл, в который записывались данные, для которых выполнение строгого лавинного критерия было необязательным.

Полученные блоки имеют расстояние нелинейности ноль, что очень сближает их с линейными функциями и делает непригодными с точки зрения криптографического качества. Примеры блоков замен без обязательного строгого лавинного критерия приведены на рисунке 8.

```
S-блок: 0 9 2 11 4 13 15 6 10 3 8 1 7 14 12 5
Матрица коэфициентов корреляции:
0       0       0       0
0       1       0       0
0       0       0       0
0       0       0       0.5
Расстояние нелинейности: 4 0 0 4

S-блок: 0 1 10 11 4 5 15 14 2 3 8 9 7 6 12 13
Матрица коэфициентов корреляции:
0       0       0       0
0       1       0       0
1       0       0       0
0       0       0       0.5
Расстояние нелинейности: 0 0 0 4
```

Рисунок 8 – Блоки замен без СЛК

Данные блоки наглядно демонстрируют необходимость СЛК при выборе функций для генерации блоков замен.

Статистическая независимость выхода S-блока подстановки от его входа, то есть матрица коэффициентов корреляции является хорошим качеством шифра тогда, когда ее элементы состоят из нулей.

Исследуя матрицы коэффициентов корреляции полученных блоков замен без СЛК, можно заметить, что у большинства матриц большая часть элементов состоит из нулей, однако наличие даже одной единицы приводит к тому, что блок замен становится близок к линейному. Если сравнить эти данные с матрицами удовлетворяющими СЛК в которых отсутствуют единицы, то можно заметить, что их расстояние нелинейности выше, чем у матриц, которые состоят из нулей, но имеют хотя бы одну единицу. Из этого можно сделать вывод о том, что нули в матрице коэффициентов корреляции не влияют на расстояние нелинейности, а вот единицы снижают нелинейность практически до линейных функций. Пример сравнения таких данных показан на рисунке 9.

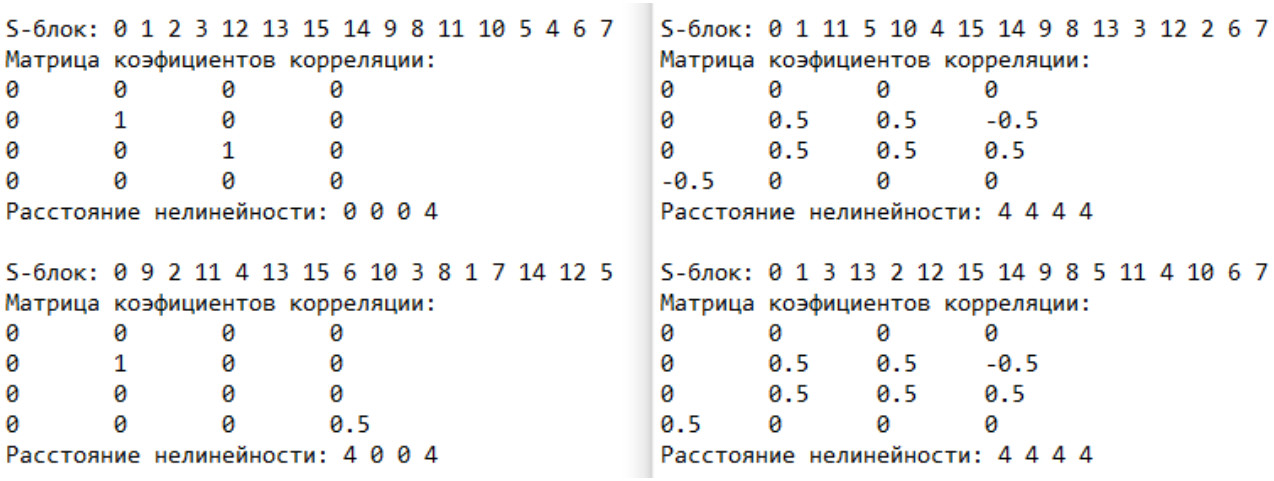

Рисунок 9 – Сравнение данных

Так как на основании полученных блоков для трех и четырех переменных не найдено ни одного *S*-блока удовлетворяющего СЛК и имеющего такую матрицу коэффициентов корреляции, чтобы все значения были равны нулю, что обозначало бы оптимальный с точки зрения этого критерия блок замен, рекомендуем использовать такие блоки, в которых элементы матрицы имеют равномерное распределение, то есть надо стремиться минимизировать разности их абсолютных значений. Исходя из этого стоит избегать матриц со значениями близкими к единице и отбрасывать матрицы в которых такая единица содержится. Часть подходящих блоков замен, созданных программой, представлена в Приложении В.

Согласно недавним исследованиям блоки со всеми нулевыми элементами можно получить только при больших k, рекомендуется $k = 8$, но первые нулевые матрицы появляются при $k = 5$ [9].

3.3 Выводы

В результате проведенного исследования можно сделать следующие рекомендации для генерации блоков замен:

– при генерации необходимо обязательно проверять соответствие функций строгому лавинному критерию, иначе велик шанс блок замен с слабыми криптографическими свойствами;

– матрицу коэффициентов корреляции рекомендуется использовать только со значениями близкими к нулю и отбрасывать матрицы с единичками,

так как наличие даже одной единички в матрице, в которой все остальные элементы равны нулю сделает этот блок близким к линейному;

– несмотря на то, что данная работа носит исследовательский характер, имеющаяся программа может использоваться для получения таких блоков замен в том числе и в ГОСТ 28147-89, так как имеют такую же длину $N = 16$, однако для достаточной эффективности требуется большое количество вычислительных мощностей;

– при наличии достаточных вычислительных мощностей рекомендуется использовать большее количество переменных, от пяти, рекомендуемое количество от восьми.

ЗАКЛЮЧЕНИЕ

Таким образом, в ходе практики была разработана и протестирована программа, генерирующая блоки замен для булевых функций трех и четырех переменных. Программа выбирает булевы функции удовлетворяющие СЛК и сбалансированности и на их основе создает блоки замен для которых рассчитываются матрицы коэффициентов корреляции.

Функции для трех и часть функций для четырех переменных приведены в таблице 1 и приложении Б соответственно.

На основе полученных данных была установлена зависимость между матрицей коэффициентов корреляции и расстоянием нелинейности, точнее ухудшение нелинейности при наличии в матрице коэффициентов корреляции хотя бы одной единички. Также был сделан вывод о малом влиянии большого количества нулей в матрице коэффициентов корреляции и необходимость при малом количестве переменных придерживаться матриц с распределенными значениями близкими к нулю.

При учете вышеперечисленных критериев обязательным является удовлетворение булевой функции строгому лавинному критерию, так как в противном случае у полученных блоков замен теряется нелинейность, что свидетельствует о плохом качестве данных *S*-блоков с точки зрения криптографической стойкости.

Было порекомендовано использование блоков замен большей длины при наличии достаточных вычислительных ресурсов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Игошин, В.И. Математическая логика и теория алгоритмов / В.И. Игошин. – Москва : Издательский центр «Академия», 2008. – 448 с. – Текст : непосредственный.

2. Агафонова, И.В. Криптографические свойства нелинейных булевых функций / И.В. Агафонова. – Текст : электронный // URL: http://dha.spb.ru/PDF/cryptoBOOLEAN.pdf (дата обращения 27.03.2020).

3. Грушо, А.А Анализ и синтез криптоалгоритмов. / А. А. Грушо, Э. А. Применко, Е.Е Тимонина. – Изд-во Марийского филиала Московского открытого социального университета, 2000. – 110 с. – Текст : непосредственный.

4. Токарева, Н. Н. Бент-функции: результаты и приложения. Обзор работ № 1(3) / Н.Н. Токарева. – ПДМ, 2009. – Текст : электронный // URL: http://mi.mathnet.ru/pdm50 (дата обращения 01.04.2020).

5. Сидоренко, А.В. Лавинный эффект в алгоритмах шифрования на основе динамического хаоса // А. В. Сидоренко, К. С. Мулярчик БГУ, Минск. – Текст : электронный // URL: http://elib.bsu.by/handle/123456789/52134 (дата обращения 12.04.2020).

6. Vergili, Işıl. Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen n × n S-Boxes. / I. Vergili. – 2001. – Текст: электронный // URL: https://pdfs.semanticscholar.org/187c/14aaae1ef93090716c37d79dbe0b5ff41178.pdf (дата обращения 13.04.2020).

7. Жданов, О.Н. Криптографические конструкции на основе функций многозначной логики: монография / А.В. Соколов, О.Н. Жданов. – Москва : ИНФРА-М, 2020. – 174 с. – Текст : непосредственный.

8. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования: Переиздание, апрель 1996 : утвержден и введён в действие постановлением Государственного комитета СССР по стандартам от 02.06.89 № 1409 – Текст : электронный // URL: http://docs.cntd.ru/document/1200007350 (дата обращения 14.04.2020).

9. Соколов, А.В. Метод синтеза S-блоков по критерию нулевой корреляции между выходными и входными векторами данных и строгому лавинному критерию / М.И. Мазурков, А.В. Соколов. – НТУУ «КПИ», 2014. – Текст : непосредственный.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Код программы

```
#include <iostream>
#include <time.h>
#include <fstream>
using namespace std;
//сравнение компонентных функций
int srav(int* a, int* b, int n)
{
	int x = 0;
	for (int i = 0; i < n; i++)
	{
		if (a[i] != b[i])
			x++;
	}

	return x;
}
//перестановка массива задом наперед
void obratka(int* s, int k)
{
	int buff = 0;
	for (int i = 0; i < k / 2; i++)
	{
		buff = s[i];
		s[i] = s[k - i - 1];
		s[k - i - 1] = buff;
	}
}
//перевод из 10-й системы счисления в 2-ю
void iz10v2(int y, int* s, int k)
{
	for (int i = 0; i < k; i++)	{s[i] = 0;}
	int x = 0;
	while (y > 0)
	{
		s[k - 1 - x] = y % 2;
		y = y / 2;
		x++;
	}
```

```
}
//перевод из 2-й системы счисления в 10-ю
int	iz2v10(int s[], int k)
{
	int y = 0;
	for (int i = 0; i < k; i++) {y = y + s[i] * pow(2, (k - 1) - i);}
	return y;
}
//расстояние нелинейности
void nelineinost(int** vessbox, int n, int k, int* nel)
{
	//f=axmod3
	int* a = new int[k];
	int* x = new int[k];
	int* F = new int[n];
	int symma;
	int zn, minzn;

	for (int f = 0; f < k; f++)
	{
		minzn = n;
		for (int b = 0; b < 2; b++)
		{
			for (int i = 0; i < n; i++)
			{
				for (int j = 0; j < n; j++)
				{
					symma = 0;
					iz10v2(i, a, k);
					iz10v2(j, x, k);
					for (int y = 0; y < k; y++)
					{
						symma=(symma+(a[y]*x[y])%2)%2;
					}
					F[j]=(symma+b)%2;
				}
				zn = srav(vessbox[k + 1 + f], F, n);
				if (zn < minzn)
				{
					minzn = zn;
```

```
                        }
                    }
                }
                nel[f] = minzn;
        }
}
 //матрица коэф-в корреляции
void matrkoef(int* sbox, int n, int k)
{
        int* s = new int[k];
        int** vessbox = new int* [2 * k + 1];
        for (int count = 0; count < 2 * k + 1; count++)
                vessbox[count] = new int[n];
        for (int i = 0; i < n; i++)
        {
                iz10v2(i, s, k);
                for (int h = 0; h < k; h++)
                {
                        vessbox[h][i] = s[h];
                }

                vessbox[k][i] = sbox[i];

                iz10v2(sbox[i], s, k);
                for (int h = 0; h < k; h++)
                {
                        vessbox[h + k + 1][i] = s[h];
                }
        }
        double x = 0,y = 0,x2 = 0,y2 = 0,xy = 0;
        //матрица корреляций-выделение памяти
        double** matrix = new double* [k];
        for (int i = 0; i < k; i++)
                matrix[i] = new double[k];
        for (int i = 0; i < k; i++)
        {
                for (int j = 0; j < k; j++)
                {
                        for (int q = 0; q < n; q++)
                        {
```

```
                        x = x + vessbox[i + 0][q];
                        y = y + vessbox[j + k + 1][q];
                        x2 = x2 + vessbox[i + 0][q] * vessbox[i + 0][q];
                        y2 = y2 + vessbox[j + k + 1][q] * vessbox[j + k + 1][q];
                        xy = xy + vessbox[i + 0][q] * vessbox[j + k + 1][q];
                }
                matrix[j][i] = (xy-(x*y)/n)/sqrt((x2-(x*x)/n)*(y2-(y*y)/n));
                x=0;y=0;x2=0;y2=0;xy=0;
        }
}
int* nel = new int[k];
nelineinost(vessbox, n, k, nel);//nelineinost
ofstream fout("spisok2.txt", ios::app);
fout<<"S-блок: ";
//cout << "S-блок: ";
for (int h = 0; h < n; h++)
{
         fout<<vessbox[k][h]<<" ";
         //cout<<vessbox[k][h]<<" ";
}
fout << endl;
//cout << endl;
fout << "Матрица коэфициентов корреляции: ";
//cout << "Матрица коэфициентов корреляции: ";
fout << endl;
//cout << endl;
for (int j= 0; j < k; j++)
{
        for (int i = 0; i < k; i++)
        {
                fout<<matrix[i][j]<<"\t";
                //cout<<matrix[i][j]<<"\t";
        }
        fout << endl;
        //cout << endl;
}
fout << "Расстояние нелинейности: ";
//cout << "Расстояние нелинейности: ";
for (int i = 0; i < k; i++)
{
```

```
                fout << nel[i] << " ";
                //cout << nel[i] << " ";
        }
        fout << endl << endl;
        //cout << endl << endl;
}
//увеличение s-блока по схеме Кима
void kim(int s[], int n, int k, int kolvokim)
{
        if (kolvokim > 0)
        {
                int n2 = pow(2, k + 1); //новый размер s-блока
                int* s2 = new int[n2];
                int* mas1 = new int[k];
                int* mas2 = new int[k + 1];
                int zn = 0, mesto = 0, por1 = 0,v1 = 0, q = 0;
                int* buff = new int[k + 1];
                for (int x = 0; x < k; x++)
                {
                        for (int y = 0; y < k; y++)
                        {
                                for (int i = 0; i < pow(2, k + 1); i++)
                                {
                                        iz10v2(i, buff, k + 1);
                                        q = buff[k];
                                        obratka(buff, k + 1);
                                        mesto = iz2v10(buff, k + 1);
                                        obratka(buff, k + 1);
                                        for (int j = 0; j < k; j++)
                                        {
                                                mas1[j] = buff[j];
                                        }
                                        mas1[x] = (mas1[x] + q) % 2;
                                        obratka(mas1, k);
                                        por1 = iz2v10(mas1, k);
                                        v1 = s[por1];
                                        iz10v2(v1, mas1, k);
                                        obratka(mas1, k);
                                        for (int j = 0; j < k; j++)
                                        {
```

```
                                        mas2[j] = mas1[j];
                                }
                                mas2[k] = (mas2[y] + q) % 2;
                                obratka(mas2, k + 1);
                                zn = iz2v10(mas2, k + 1);
                                s2[mesto] = zn;
                        }
                        matrkoef(s2, n2, k + 1);
                        kim(s2, n2, k + 1, kolvokim - 1);
                }
        }
}
}
//перебор для слк
void perebor(int s[], int n, int* flag)
{
        if (n >= 0)
        {
                if (s[n] == 2)
                {
                        s[n] = 0;
                        perebor(s, n - 1, flag);
                }
                else
                        s[n]++;

        }
        else *flag = 0;
}
//слк
bool slk(int* s, int n, int k)
{
        int* dvoich = new int[k];
        int* SLK = new int[2];
        bool slk = true;
        int rez = 0;
        for (int j = 0; j < k; j++)
        {
                for (int i = 1; i < 2; i++)
                {
```

```
                for (int z = 0; z < 2; z++)
                {
                        SLK[z] = 0;
                }
                for (int x = 0; x < n; x++)
                {
                        iz10v2(x, dvoich, k);
                        dvoich[j] = (dvoich[j] + i) % 2;
                        rez = s[x] - s[iz2v10(dvoich, k)];
                        for (int z = 0; z < 2; z++)
                        {
                                if (rez < 0)
                                {
                                        rez = rez + 2;
                                }
                                SLK[rez]++;
                        }
                }
        }
        for (int z = 1; z < 2; z++)
        {
                if (SLK[0] != SLK[z])
                {
                        slk = false;
                        break;
                }
        }
    }
    return slk;
}
//сбалансированность
bool balanced(int s[], int n)
{
    bool flag = true;
    int kolvo = 0;
    for (int i = 0; i < 2; i++)
    {
        kolvo = 0;
        for (int j = 0; j < n; j++)
        {
```

```
                        if (s[j] == i)
                        {
                                kolvo++;
                        }
                }
                if (kolvo != n / 2)
                {
                        flag = false;
                        break;
                }
        }
        return flag;
}

int main()
{
        setlocale(LC_ALL, "Russian");
        int k = 3; //kolvo peremennih
        int n = pow(2, k); //dlina
        int* s = new int[n];
        //gen:
        int* dvoich = new int[k];
        int* sbox = new int[n];
        int notS;
        //int kolvokima = k-3;

        for (int i = 0; i < n; i++)
        {
                s[i] = 0;
        }
                for (int i = 0; i <2; i++)//00001111
        {
                for (int j = 0; j < n /2; j++)
                {
                        s[i * (n / 2) + j] = i;
                }
        }
        int** f = new int* [10000];
        for (int i = 0; i < 10000; i++)
                f[i] = new int[n];
```

```
int x = 0;
//balans,slk
ofstream fout("spisok1.txt");
int flag = 1;
while (flag>0)
{
	if (balanced(s, n))
	{
		if (slk(s, n, k))
		{
			for (int y = 0; y < n; y++)
			{
				f[x][y] = s[y];
				fout<< f[x][y];
				//cout << f[x][y];
			}
			x++;
			fout << endl;
			//cout << endl;
		}
	}
	perebor(s, n - 1, &flag);
}
//generatia sbox
cout<<"Программа не зависла, генерация займет время"<<endl;
for (int i = 0; i < x; i++)
{
	for (int y = 0; y < x; y++)
	{
		for (int z = 0; z < x; z++)
		{


			notS = 0;
			for (int j = 0; j < n; j++)
			{
				dvoich[0] = f[i][j];
				dvoich[1] = f[y][j];
				dvoich[2] = f[z][j];
				sbox[j] = iz2v10(dvoich, k);
```

```
                                        for (int h = 0; h < j; h++)
                                        {
                                                if (sbox[j] == sbox[h])
                                                {
                                                        notS = 1;
                                                        break;
                                                }
                                        }

                                        if (notS == 1)
                                                break;
                                }
                                if (notS == 0)
                                {
                                        matrkoef(sbox, n, k);
                                        kim(sbox, n, k, 1);
                                }
                        }

                }
        }
        cout << "Программа закончила генерацию, второй файл доступен" << endl;
        system("pause");
}
```

ПРИЛОЖЕНИЕ Б

Булевы функции четырех переменных, удовлетворяющие СЛК и сбалансированности

0000011001101111	0000011001111110	0000011010011111	0000011010110111	0000011010111110
0000011011010111	0000011011011110	0000011011100111	0000011011110110	0000011101101110
0000011101110110	0000011110011011	0000011110011101	0000011111100110	0000100101101111
0000100101111011	0000100101111101	0000100110011111	0000100110111101	0000100111011011
0000100111101011	0000100111101101	0000100111111001	0000101101100111	0000101101101110
0000101110011101	0000101110111001	0000101111011001	0000110101100111	0000110101101110
0000110110011011	0000110110111001	0000110111011001	0000111001100111	0000111001110110
0000111010011011	0000111010011101	0000111011100110	0000111101100110	0000111110011001
0001001001111011	0001001001111110	0001001010011111	0001001010110111	0001001010111110
0001001011010111	0001001011011011	0001001011011110	0001001011110110	0001001101011110
0001001101111010	0001001110100111	0001001110110101	0001001111011010	0001010001111101
0001010001111110	0001010010011111	0001010010110111	0001010010111101	0001010010111110
0001010011010111	0001010011011110	0001010011110110	0001010100111110	0001010101111100
0001010110111100	0001010111000111	0001010111010011	0001011100101011	0001011100101110
0001011100111010	0001011101001101	0001011101001110	0001011101011100	0001011101110001
0001011101110010	0001011101110100	0001011110001011	0001011110001101	0001011110001110
0001011110100011	0001011110110001	0001011110110010	0001011111000101	0001011111010001
0001011111010100	0001100001111011	0001100001111101	0001100001111110	0001100010011111
0001100010111101	0001100010111110	0001100011011011	0001100011011110	0001100011111001
0001100100101111	0001100101001111	0001100110001111	0001100111110001	0001100111111000
0001101000110111	0001101001110011	0001101010110011	0001101011001110	0001101011011100
0001101100100111	0001101100101110	0001101100110101	0001101101000111	0001101101001101
0001101101001110	0001101101011100	0001101101110001	0001101101110010	0001101110001101
0001101110001110	0001101110100011	0001101110110001	0001101110110010	0001101110111000
0001101111001010	0001101111010001	0001101111011000	0001110001010111	0001110001110101
0001110010101110	0001110010111010	0001110011010101	0001110100100111	0001110100101011
0001110100101110	0001110100111010	0001110101000111	0001110101001110	0001110101010011
0001110101110001	0001110101110100	0001110110001011	0001110110001110	0001110110101100
0001110110110001	0001110110111000	0001110111000101	0001110111010001	0001110111010100
0001110111011000	0001111100100110	0001111101000110	0001111110001001	0001111110010001
0001111110011000	0010000101101111	0010000101111011	0010000101111101	0010000110110111
0010000110111101	0010000111100111	0010000111101011	0010000111101101	0010000111111001
0010001101011011	0010001101111010	0010001110101101	0010001110110101	0010001111100101
0010010001101111	0010010001111101	0010010001111110	0010010010110111	0010010010111101
0010010010111110	0010010011100111	0010010011101101	0010010011110110	0010010100111011
0010010101110011	0010010110110011	0010010111001101	0010010111101100	0010011000011111
0010011001001111	0010011010001111	0010011011110010	0010011011110100	0010011100011011
0010011100011101	0010011100111010	0010011101001101	0010011101001110	0010011101010011
0010011101110001	0010011101110010	0010011101110100	0010011110001011	0010011110001101
0010011110001110	0010011110101100	0010011110110001	0010011110110010	0010011111000101
0010011111100010	0010011111100100	0010100001101111	0010100001111011	0010100001111101
0010100001111110	0010100010111101	0010100010111110	0010100011101011	0010100011101101
0010100011111001	0010101000111101	0010101001111100	0010101010111100	0010101011001011
0010101011100011	0010101100010111	0010101100011101	0010101100110101	0010101101000111
0010101101001101	0010101101001110	0010101101010011	0010101101110001	0010101101110010
0010101110001101	0010101110001110	0010101110101100	0010101110110001	0010101110110010
0010101110111000	0010101111001010	0010101111100010	0010101111101000	0010110001011101
0010110001110101	0010110010101011	0010110010111010	0010110011101010	0010111000010111
0010111000011011	0010111000011101	0010111000110101	0010111001000111	0010111001001101
0010111001011100	0010111001110010	0010111001110100	0010111010001011	0010111010001101
0010111010100011	0010111010110010	0010111010111000	0010111011001010	0010111011100010
0010111011100100	0010111011101000	0010111100011001	0010111101000110	0010111101100010

0010111101100100	0010111110001001	0011000101011011	0011000101111010	0011000110100111
0011000110101101	0011000111100101	0011001001011011	0011001001011110	0011001010100111
0011001010110101	0011001011011010	0011001101011010	0011001110100101	0011010001010111
0011010001011101	0011010010101110	0011010010111010	0011010011010101	0011010100011011
0011010100101011	0011010100101110	0011010100111010	0011010101000111	0011010101001101
0011010101010011	0011010101011100	0011010101110010	0011010110001101	0011010110100011
0011010110101100	0011010110110010	0011010110111000	0011010111000101	0011010111010001
0011010111010100	0011010111100100	0011011100011010	0011011101010010	0011011110000101
0011011110100001	0011011110100100	0011100001011101	0011100001110101	0011100010101011
0011100010101110	0011100011101010	0011101000010111	0011101000011101	0011101000100111
0011101000110101	0011101001001110	0011101001010011	0011101001011100	0011101001110001
0011101001110100	0011101010001011	0011101010001110	0011101010100011	0011101010101100
0011101010110001	0011101011001010	0011101011011000	0011101011100010	0011101011101000
0011101100100101	0011101101001010	0011101101010010	0011101101011000	0011101110100001
0011110001010101	0011110010101010	0011110100101010	0011110101000101	0011110101010001
0011110101010100	0011110110101000	0011111000010101	0011111001010100	0011111010001010
0011111010100010	0011111010101000	0100000101101111	0100000101111011	0100000101111101
0100000111010111	0100000111011011	0100000111100111	0100000111101011	0100000111101101
0100000111111001	0100001001101111	0100001001111011	0100001001111110	0100001011010111
0100001011011011	0100001011011110	0100001011100111	0100001011101011	0100001011110110
0100001101011101	0100001101110101	0100001110101011	0100001111010101	0100001111101010
0100010100111101	0100010101111100	0100010111001011	0100010111010011	0100010111100011
0100011000011111	0100011000101111	0100011010001111	0100011011110010	0100011011110100
0100011100011011	0100011100011101	0100011100101011	0100011100101110	0100011100110101
0100011101011100	0100011101110001	0100011101110010	0100011101110100	0100011110001011
0100011110001101	0100011110001110	0100011110100011	0100011111001010	0100011111010001
0100011111010100	0100011111100010	0100011111100100	0100100001101111	0100100001111011
0100100001111101	0100100001111110	0100100011011011	0100100011011110	0100100011101011
0100100011101101	0100100011111001	0100101000111011	0100101001110011	0100101011001101
0100101011011100	0100101011101100	0100110001011011	0100110001111010	0100110010101101
0100110011011010	0100110011100101	0100110100010111	0100110100011011	0100110100100111
0100110100101011	0100110100101110	0100110100110101	0100110101010011	0100110101110001
0100110101110100	0100110110001011	0100110110001110	0100110110101100	0100110111001010
0100110111010001	0100110111010100	0100110111011000	0100110111100100	0100110111101000
0100111000010111	0100111000011011	0100111000011101	0100111000100111	0100111000101011
0100111000111010	0100111001010011	0100111001110010	0100111001110100	0100111010001011
0100111010001101	0100111010101100	0100111011000101	0100111011010100	0100111011011000
0100111011100010	0100111011100100	0100111011101000	0100111100011001	0100111100100110
0100111101100010	0100111101100100	0100111110001001	0101000100111101	0101000101111100
0101000111000111	0101000111001011	0101000111100011	0101001000110111	0101001000111011
0101001010110011	0101001011001110	0101001011011100	0101001100011101	0101001100100111
0101001100101011	0101001100110101	0101001100111010	0101001101001101	0101001101001110
0101001101011100	0101001101110100	0101001110001011	0101001110100011	0101001110110001
0101001110110010	0101001111000101	0101001111001010	0101001111010100	0101001111011000
0101001111100010	0101010000111101	0101010000111110	0101010010111100	0101010011000111
0101010011010011	0101010100111100	0101010111000011	0101011100011100	0101011100110100
0101011110000011	0101011111000001	0101011111000010	0101100000111011	0101100001110011
0101100011001101	0101100011001110	0101100011101100	0101101000110011	0101101011001100
0101101100100011	0101101100110001	0101101100110010	0101101101001100	0101101111001000
0101110000010111	0101110000011011	0101110000101110	0101110000110101	0101110000111010
0101110001000111	0101110001010011	0101110001110001	0101110001110010	0101110010001101
0101110010001110	0101110010101100	0101110010111000	0101110011000101	0101110011001010
0101110011010001	0101110011100100	0101110011101000	0101110100101100	0101110100110100
0101110100111000	0101110101000011	0101110111000001	0101111000010011	0101111000110010
0101111010001100	0101111011000100	0101111011001000	0110000001101111	0110000001111011
0110000001111101	0110000001111110	0110000011100111	0110000011101011	0110000011101101
0110000011110110	0110000011111001	0110001000101111	0110001001001111	0110001011110001

0110001011110100	0110001011111000	0110010000101111	0110010001001111	0110010011110001
0110010011110010	0110010011111000	0110011000001111	0110011011110000	0110011100001011
0110011100001101	0110011100001110	0110011101110000	0110011111100000	0110111000000111
0110111000001011	0110111000001101	0110111001110000	0110111011100000	0110111100000110
0110111100001001	0110111100100001	0110111100100100	0110111100101000	0110111101000001
0110111101000010	0110111101001000	0110111101100000	0111000001100111	0111000001101110
0111000010111001	0111000011011001	0111000011100110	0111000100010111	0111000100011011
0111000100011101	0111000100100111	0111000100101011	0111000100111010	0111000101000111
0111000101001101	0111000101011100	0111000110100011	0111000110110010	0111000110111000
0111000111000101	0111000111010100	0111000111011000	0111000111100010	0111000111100100
0111000111101000	0111001000010111	0111001000011011	0111001000100111	0111001000101011
0111001000101110	0111001000110101	0111001001000111	0111001001001110	0111001001011100
0111001010100011	0111001010110001	0111001010111000	0111001011001010	0111001011010001
0111001011010100	0111001011011000	0111001011100100	0111001011101000	0111001100011010
0111001100100101	0111001101001010	0111001101011000	0111001110100001	0111010000010111
0111010000011101	0111010000100111	0111010000101110	0111010000111010	0111010001000111
0111010001001101	0111010001001110	0111010001010011	0111010010101100	0111010010110001
0111010010110010	0111010010111000	0111010011000101	0111010011010001	0111010011011000
0111010011100010	0111010011101000	0111010100011100	0111010100101100	0111010100111000
0111010101000011	0111010111000001	0111011000000111	0111011000001110	0111011010110000
0111011011010000	0111011011100000	0111101000010011	0111101000100011	0111101000110001
0111101001001100	0111101011001000	0111101100001001	0111101100010010	0111101100011000
0111101100100001	0111101100101000	0111101101000001	0111101101000010	0111101101001000
0111101101100000	0111110000010101	0111110000101010	0111110001000101	0111110001010001
0111110010101000	0111110100001001	0111110100010100	0111110100011000	0111110100100001
0111110100100100	0111110100101000	0111110101000001	0111110101001000	0111110101100000
0111111000000110	0111111000010010	0111111000010100	0111111000011000	0111111000100100
0111111000101000	0111111001000010	0111111001001000	0111111001100000	0111111000101000

ПРИЛОЖЕНИЕ В

Качественные блоки замен на основе булевых функций трех и четырех переменных

S-блок: 0 9 2 4 11 13 6 15 1 8 12 10 5 3 7 14
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 10 4 11 5 14 15 9 8 12 2 13 3 7 6
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
-0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 2 12 3 13 14 15 9 8 4 10 5 11 7 6
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 9 2 4 11 13 6 15 10 12 8 1 14 7 3 5
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
-0.5 0 0 0
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 10 4 11 5 14 15 2 12 8 9 6 7 3 13
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0.5 0 0 0
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 2 12 3 13 14 15 10 4 8 9 6 7 11 5
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0.5 0 0 0
0 0.5 -0.5 0.5

Расстояние нелинейности: 4 4 4 4

S-блок: 0 9 2 4 11 13 6 15 3 5 14 7 8 1 10 12
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 10 4 11 5 14 15 3 13 6 7 8 9 2 12
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 2 12 3 13 14 15 11 5 6 7 8 9 10 4
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 2 4 3 5 7 6
Матрица коэффициентов корреляции:
0.5 0.5 0.5
0.5 0.5 -0.5
0.5 -0.5 0
Расстояние нелинейности: 2 2 2

S-блок: 0 9 2 4 11 13 15 6 1 8 12 10 5 3 14 7
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
0 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 10 4 11 5 15 14 9 8 12 2 13 3 6 7
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
-0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 2 12 3 13 15 14 9 8 4 10 5 11 6 7
Матрица коэффициентов корреляции:
0 0 0 0

0 0.5 0.5 0.5
0 0.5 0.5 -0.5
0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 9 2 4 11 13 15 6 10 12 8 1 7 14 3 5
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
-0.5 0 0 0
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 10 4 11 5 15 14 2 12 8 9 7 6 3 13
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0.5 0 0 0
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 2 12 3 13 15 14 10 4 8 9 7 6 11 5
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0.5 0 0 0
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 9 2 4 11 13 15 6 3 5 7 14 8 1 10 12
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 10 4 11 5 15 14 3 13 7 6 8 9 2 12
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 2 12 3 13 15 14 11 5 7 6 8 9 10 4
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 3 5 2 4 6 7
Матрица коэффициентов корреляции:
0.5 0.5 -0.5
0.5 0.5 0.5
0.5 -0.5 0.5
Расстояние нелинейности: 2 2 2

S-блок: 0 9 11 13 2 4 6 15 1 8 5 3 12 10 7 14
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0 0.5 0.5 0.5
0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 11 5 10 4 14 15 9 8 13 3 12 2 7 6
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0 0.5 0.5 0.5
-0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 3 13 2 12 14 15 9 8 5 11 4 10 7 6
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0 0.5 0.5 0.5
0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 9 11 13 2 4 6 15 3 5 8 1 14 7 10 12
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0.5 0 0 0
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 11 5 10 4 14 15 3 13 8 9 6 7 2 12
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0.5 0 0 0
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 3 13 2 12 14 15 11 5 8 9 6 7 10 4
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5

0.5 0 0 0
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 9 11 13 2 4 6 15 10 12 14 7 8 1 3 5
Матрица коэффициентов корреляции:
0 0 0 0
-0.5 0 0 0
0 0.5 0.5 0.5
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 11 5 10 4 14 15 2 12 6 7 8 9 3 13
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 0.5
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 3 13 2 12 14 15 10 4 6 7 8 9 11 5
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 0.5
0 0.5 -0.5 0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 3 5 2 4 7 6
Матрица коэффициентов корреляции:
0.5 0.5 -0.5
0.5 0.5 0.5
0.5 -0.5 0
Расстояние нелинейности: 2 2 2

S-блок: 0 9 11 13 2 4 15 6 1 8 5 3 12 10 14 7
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0 0.5 0.5 0.5
0 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 11 5 10 4 15 14 9 8 13 3 12 2 6 7
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0 0.5 0.5 0.5
-0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 3 13 2 12 15 14 9 8 5 11 4 10 6 7

Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0 0.5 0.5 0.5
0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 9 11 13 2 4 15 6 3 5 8 1 7 14 10 12
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0.5 0 0 0
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 11 5 10 4 15 14 3 13 8 9 7 6 2 12
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0.5 0 0 0
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 3 13 2 12 15 14 11 5 8 9 7 6 10 4
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 -0.5
0.5 0 0 0
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 9 11 13 2 4 15 6 10 12 7 14 8 1 3 5
Матрица коэффициентов корреляции:
0 0 0 0
-0.5 0 0 0
0 0.5 0.5 0.5
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 11 5 10 4 15 14 2 12 7 6 8 9 3 13
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 0.5
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 0 1 3 13 2 12 15 14 10 4 7 6 8 9 11 5
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 0.5

0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 2 4 3 5 6 7
Матрица коэффициентов корреляции:
0.5 0.5 0.5
0.5 0.5 -0.5
0.5 -0.5 0
Расстояние нелинейности: 2 2 2

S-блок: 9 0 2 4 11 13 6 15 8 1 12 10 5 3 7 14
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
0 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 10 4 11 5 14 15 8 9 12 2 13 3 7 6
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
-0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 2 12 3 13 14 15 8 9 4 10 5 11 7 6
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 9 0 2 4 11 13 6 15 10 12 1 8 14 7 3 5
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
-0.5 0 0 0
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 10 4 11 5 14 15 2 12 9 8 6 7 3 13
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0.5 0 0 0
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 2 12 3 13 14 15 10 4 9 8 6 7 11 5
Матрица коэффициентов корреляции:

0 0 0 0
0 0.5 0.5 0.5
0.5 0 0 0
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 9 0 2 4 11 13 6 15 3 5 14 7 1 8 10 12
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 10 4 11 5 14 15 3 13 6 7 9 8 2 12
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 2 12 3 13 14 15 11 5 6 7 9 8 10 4
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 2 4 3 5 7 6
Матрица коэффициентов корреляции:
0.5 0.5 0.5
0.5 0.5 -0.5
0.5 -0.5 -0.5
Расстояние нелинейности: 2 2 2

S-блок: 9 0 2 4 11 13 15 6 8 1 12 10 5 3 14 7
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
-0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 10 4 11 5 15 14 8 9 12 2 13 3 6 7
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
-0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 2 12 3 13 15 14 8 9 4 10 5 11 6 7
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0 0.5 0.5 -0.5
0.5 0 0 0
Расстояние нелинейности: 4 4 4 4

S-блок: 9 0 2 4 11 13 15 6 10 12 1 8 7 14 3 5
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
-0.5 0 0 0
0 0.5 -0.5 -0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 10 4 11 5 15 14 2 12 9 8 7 6 3 13
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0.5 0 0 0
0 0.5 -0.5 -0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 2 12 3 13 15 14 10 4 9 8 7 6 11 5
Матрица коэффициентов корреляции:
0 0 0 0
0 0.5 0.5 0.5
0.5 0 0 0
0 0.5 -0.5 -0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 9 0 2 4 11 13 15 6 3 5 7 14 1 8 10 12
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 -0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 10 4 11 5 15 14 3 13 7 6 9 8 2 12
Матрица коэффициентов корреляции:
0 0 0 0
0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 -0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 2 12 3 13 15 14 11 5 7 6 9 8 10 4
Матрица коэффициентов корреляции:
0 0 0 0

0.5 0 0 0
0 0.5 0.5 -0.5
0 0.5 -0.5 -0.5
Расстояние нелинейности: 4 4 4 4

S-блок: 1 0 3 5 2 4 6 7
Матрица коэффициентов корреляции:
0.5 0.5 -0.5
0.5 0.5 0.5
0.5 -0.5 0
Расстояние нелинейности: 2 2 2