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

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
канд. техн. наук, доцент
_____ В.Н. Тренькаев
«17» 01 2022 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА (ДИПЛОМНАЯ РАБОТА)

ИССЛЕДОВАНИЕ СТОЙКОСТИ АСИММЕТРИЧНОЙ КРИПТОСИСТЕМЫ НА БУЛЕВЫХ ФУНКЦИЯХ

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

Кондратьев Вадим Андреевич

Научный руководитель ВКР
зав. лаб. компьютерной криптографии
_____ И.А. Панкратова
«17» 01 2022 г.

Автор работы
студент группы № 1165
_____ В.А. Кондратьев
«17» 01 2022 г.

Томск - 2022

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

УТВЕРЖДАЮ
Руководитель ООП
канд. техн. наук, доцент

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

ЗАДАНИЕ

по выполнению выпускной квалификационной работы специалиста обучающемуся
Кондратьева Вадима Андреевича
Фамилия Имя Отчество обучающегося
по специальности 10.05.01 Компьютерная безопасность, специализация (профиль) «Анализ безопасности компьютерных систем»

1 Тема выпускной квалификационной работы:

Исследование атак на асимметричный шифр на булевых функциях.

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:
а) в учебный офис / деканат – 17 января 2022 г. б) в ГЭК – 28 января 2022 г.

3 Исходные данные к работе:
Объект исследования – Асимметричный шифр на булевых функциях
Предмет исследования – Атаки на шифр
Цель исследования – Анализ атак на шифр

Задачи:

Разработать и реализовать алгоритмы, осуществляющие варианты атак на шифр с одной и двумя перестановками в качестве ключевых параметров. Провести исследования и сравнения алгоритмов. Сделать выводы относительно стойкости шифра.

Методы исследования:

Теоретический (с использованием комбинаторики, теории булевых функций и теории вероятностей) и экспериментальный (на базе ЭВМ).

Организация или отрасль, по тематике которой выполняется работа, –

Лаборатория компьютерной криптографии ТГУ.

4 Краткое содержание работы

Описание исследуемого шифра, описание атак на него, описание разработанных алгоритмов, результаты исследований и сравнений атак и разработанных алгоритмов.

Научный руководитель выпускной квалификационной работы
зав. лаб. компьютерной криптографии _____ / *И.А.Панкратова*
должность, место работы *подпись* *И.О. Фамилия*

Задание принял к исполнению
студент группы 1165 _____ / *В.А.Кондратьев*
должность, место работы *подпись* *И.О. Фамилия*

АННОТАЦИЯ

Дипломная работа содержит 32 страницы, 2 рисунка, 4 таблицы, 4 источника литературы и 1 приложение.

АСИММЕТРИЧНЫЙ ШИФР НА БУЛЕВЫХ ФУНКЦИЯХ, КРИПТОАНАЛИЗ, ПЕРЕСТАНОВКИ

Актуальность:

Стойкость большинства классических шифров с открытым ключом основана на том, что проблема дискретного логарифма вычислительно неразрешима, но такие шифры уязвимы по отношению к квантовым атакам. Поэтому следует исследовать криптостойкость альтернативных шифров.

Объекты исследования: асимметричный шифр на булевых функциях, атаки на этот шифр с двумя перестановками.

Цель работы: исследование атак на асимметричный шифр на булевых функциях.

Задачи:

Разработать и реализовать алгоритмы, осуществляющие варианты атак на шифр с одной и двумя перестановками в качестве ключевых параметров. Провести исследования и сравнения алгоритмов. Сделать выводы относительно стойкости шифра.

Методы исследования: теоретический (с использованием комбинаторики, теории булевых функций и теории вероятностей) и экспериментальный (на базе ЭВМ).

Результаты: Разработаны и исследованы атаки на асимметричный шифр, когда ключевой параметр состоит из одной или двух перестановок. Разработан и реализован алгоритм построения матрицы $D = T(A,B)$. Разработаны и реализованы алгоритмы, которые позволяют выяснить, является ли матрица $D = T(A,B)$ матрицей перестановок, т.е. проверить, получена ли матрица B из матрицы A перестановкой столбцов. Разработан и реализован алгоритм, который в случае, когда матрица $D = T(A,B)$ является матрицей перестановок, позволяет извлечь одну случайную перестановку в матричном виде. Проведены исследования алгоритмов.

Проведены исследования и сравнения атак на асимметричный шифр с двумя перестановками. Выдвинуто и доказано два утверждения с оценками о количестве возможных пар кандидатов в (π_1, π_2) в качестве ответа. Сделана вероятностная оценка количества открытых текстов, когда с вероятностью 90% в качестве возможного ответа будет выступать одна пара (π_1, π_2). В результате сравнения двух вариантов атак по скорости установлено, что второй вариант эффективнее и быстрее. Обосновано это отбрасыванием неподходящей перестановки прежде, чем перейдем к построению D. При этом эффективность второй атаки относительно самой себя зависит от функции g. Установлено, что для $n>3$ в среднем требуется построить одну строку матрицы C', прежде чем отбросить неподходящую $\pi_1^{'}$, и сортировка перед построением $C^{'}$ играет незначительную роль в эффективности атаки, когда g - случайна.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В настоящее время большинство классических шифров с открытым ключом описывается на основе групп, в которых групповая операция легко применяется и проблема дискретного логарифма вычислительно неразрешима, но такие шифры уязвимы по отношению к квантовым атакам. Поэтому была поставлена цель изучить альтернативный асимметричный шифр, основанный на алгебре биективных векторных булевых функций с операциями отрицания и перестановки на множествах их переменных и координатных функций, предложенный Агибаловым Геннадием Петровичем в [1].

В качестве знакомства был реализован и исследован алгоритм шифрования ElGamal на булевых функциях в сравнении с классической схемой в группе F_p^*. В результате экспериментов над алгоритмами установлено, что схема на булевых функциях выполняется быстрее. Однако вопросы относительно криптостойкости асимметричного шифра на булевых функциях остаются открытыми.

Данная работа посвящена исследованию и реализации атак на асимметричный шифр на векторных булевых функциях, в частности, когда ключевой параметр состоит из одной или двух перестановок аргументов и/или координат функции.

1 Асимметричный шифр на булевых функциях

1.1 Определения

Определение 1. *Векторной булевой функцией* от n переменных называется упорядоченный набор функций $F(x) = (f_1(x), \ldots, f_m(x))$, где $x = (x_1, \ldots, x_n)$ и $f_i(x)$ - булева функция, $i = 1, \ldots, m$; $F : \mathrm{F}_2^n \to \mathrm{F}_2^m$.

Пусть n есть натуральное целое, $n \geq 2$, и S_n есть множество всех перестановок ряда $(1,\ldots,n)$, т. е. $\mathrm{S}_n = \{(i_1 i_2 \ldots i_n) : i_j \in \{1,\ldots,n\}, j \neq r \Rightarrow i_j \neq i_r; j,r \in \{1,\ldots,n\}\}$.

Определение 2. Перестановка $\pi = (i_1 i_2 \ldots i_n) \in \mathrm{S}_n$ называется *операцией перестановки,* если результат её применения к любому слову $w = w_1 w_2 \ldots w_n$ есть слово $\pi(w) = w_{i_1} w_{i_2} \ldots w_{i_n}$.

Определение 3. Булев вектор $\sigma = b_1 b_2 \ldots b_n \in \mathrm{F}^n_2$ задает *операцию инверсии*; результат ее применения к строке $a = a_1 a_2 \ldots a_n$ булевых величин (констант, переменных или функций) $a_1, \ldots, a_n$ есть строка $a^\sigma = a_1^{b_1} a_2^{b_2} \ldots a_n^{b_n}$, где для a и b из F_2 верно: $a^b = a$, если $b = 1$, и $a^b = \neg a$, если $b = 0$.

1.2 Построение $g^\alpha(x)$ и $g^{\alpha^{-1}}(x)$

Пусть $x = (x_1, x_2, \ldots, x_n)$ есть строка из n различных булевых переменных, g: $\mathrm{F}_2^n \to \mathrm{F}_2^n$ есть n-мерная векторная булева функция $g(x)$ и g_i: $\mathrm{F}_2^n \to \mathrm{F}_2$, $i \in \{1,2,\ldots,n\}$, суть координатные функции этой g.

Пусть p_1, p_2 и s_1, s_2 суть символы переменных со значениями, соответственно, операций перестановки в S_n и операций отрицания в F_2^n, а именно: p_1, s_1 — над переменными в x и p_2, s_2 — над координатами в $g(x)$.

Для любого $a = (s_1 p_1 s_2 p_2)$ формула $g^a(x)$ определяется так: $g^a(x) = p_2(g^{s_2}(p_1(x^{s_1})))$.

Схематически вычисления в соответствии с этой формулой могут быть выражены следующей цепочкой:

$$x \xrightarrow{s_1} x^{s_1} \xrightarrow{p_1} p_1(x^{s_1}) \xrightarrow{g} g(p_1(x^{s_1})) \xrightarrow{s_2} g^{s_2}(p_1(x^{s_1})) \xrightarrow{p_2} g^a(x).$$

В каждом случае, когда $g(x)$ является биективной векторной булевой функцией на F_2^n, таковой будет и функция $g^a(x)$. Её обратная функция $g^{a^{-1}}(x)$ удовлетворяет соотношению тождества $g^{a^{-1}}(g^a(x)) = x$ и может быть построена следующим образом: если $y = g^a(x)$, то $x = g^{a^{-1}}(y) = (p_1^{-1}(g^{-1}((p_2^{-1}(y))^{s_2})))^{s_1}$.

Схематически вычисления в соответствии с этой формулой могут быть выражены следующей цепочкой:

$$y \xrightarrow{p_2^{-1}} g^{s_2}(p_1(x^{s_1})) \xrightarrow{s_2} g(p_1(x^{s_1})) \xrightarrow{g^{-1}} p_1(x^{s_1}) \xrightarrow{p_1^{-1}} x^{s_1} \xrightarrow{s_1} x.$$

Шифрование и расшифрование в асимметричном шифре происходит посредством вычисления значения векторной булевой функции $g^a(x)$ и $g^{a^{-1}}(x)$ соответственно. Открытым ключом являются функции $g(x)$ и $g^a(x)$, закрытым ключом - $g^{a^{-1}}(x)$ и секретным параметром - a. Криптостойкость шифра основана на сложности задачи нахождения обратной векторной булевой функции.

В качестве ключевых параметров шифра могут выступать элементы любого непустого подмножества $J \subseteq \{p_1, p_2, s_1, s_2\}$ (всего 15 вариантов).

1.3 Пример асимметричной схемы шифрования на булевых функциях - схема шифрования ElGamal [2]

Параметры: n — натуральное число, $n \geq 2$; $g(x) = g_1(x)g_2(x)...g_n(x)$ — биективная векторная булева функция размерности n с координатными

функциями $g_1(x),\ldots,g_n(x)$, заданными конструктивно и вычислимыми с полиномиальной (от n) временной сложностью, g: $F_2^n \to F_2^n$;

p_1, p_2 и s_1, s_2 — символы переменных со значениями, равными соответственно операциям перестановки из S_n и операциям отрицания из F_2^n;

$a = (s_1p_1s_2p_2)$ и $g^a(x) = p_2(g^{s2}(p_1(x^{s1})))$.

Открытый ключ — $(g(x),g^a(x))$, *закрытый ключ* - $g^{a^{-1}}(x)$, *секретный параметр* - a.

Зашифрование: m — *открытый текст*, $m \in F_2^n$; k — *параметр рандомизации,* $k = (r_1.q_1,r_2,q_2)$, r_1,r_2 из S_n, q_1,q_2 из F_2^n; $\gamma(m) = g^k(m) = q_2(g^{r2}(q_1(m^{r1})))$, $\delta(m) = g^k(m) \oplus g^a(m)$; $(\gamma(m),\delta(m))$ — *шифртекст.*

Расшифрование: $m = g^{a^{-1}}(\gamma(m) \oplus \delta(m))$.

Обоснование решения: $g^{a^{-1}}(\gamma(m) \oplus \delta(m)) = g^{a^{-1}}(g^k(m) \oplus g^k(m) \oplus g^a(m)) = g^{a^{-1}}(g^a(m)) = m$.

2 Атака на асимметричный шифр с одной перестановкой

Рассмотрим вспомогательную задачу 1. Пусть даны две булевых матрицы, одна из которых получена перестановкой столбцов второй матрицы. Требуется найти эту перестановку.

Пусть $\pi_1 \in S_n$, A и B – булевы матрицы размера m x n, A_i, B_i – их строки, $A(j)$ и $B(j)$ – вектор-столбцы, причём $B(j) = A(\pi(j))$ для $j = 1, \ldots, n$.

Для решения данной задачи разработан и реализован следующий алгоритм.

Матрица D размера n x n строится следующим образом:

$$d_{jk} = \bigwedge_{i=1}^{m} a_{ik}^{b_{ij}}, \quad j,k = 1,\ldots,n. \tag{1}$$

Будем обозначать $D = T(A, B)$. Тогда в матрице $D = T(A, B)$ элемент $d_{jk} = 1$, если и только если существует перестановка $\pi \in S_n$, такая, что $B = \pi(A)$ и $\pi(j) = k$.

Пример 1:

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix},$$

$$D = T(A, B) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \qquad \pi(1) = 2;\ \pi(2), \pi(4) \in \{1, 3\};\ \pi(3) = 4,$$

т. е. все перестановки π, такие, что $\pi(A) = B$, — это $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$ и $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$.

Мы будем рассматривать атаки с известным открытым текстом в следующих предположениях:

1. для любого $x \in F_2^n$ криптоаналитик может вычислить $g(x)$ и $g^{-1}(x)$;
2. криптоаналитику известно, что множество J включает в себя π_1 и/или π_2, но не их конкретные значения.

Атака на асимметричный шифр в случае $J = \{\pi_1\}$.

Функция f строится так: $f(x) = g(\pi_1(x))$. Следовательно, $\pi_1(x) = g^{-1}(x)$. Пусть имеется m пар открытых текстов и шифртекстов длины n вида (P_i, C_i), $i = 1, ..., m$.

Атака с известным открытым текстом может быть проведена следующим образом:

1) строим матрицу C со строками C_i, $i = 1, ... , m$;
2) строим матрицу P' со строками $g^{-1}(C_i)$, $i = 1, ... , m$;
3) находим матрицу $D = T(P, P')$, в которой содержатся все возможные перестановки π_1, удовлетворяющие системе:
   $C_i = g(\pi_1(P_i))$, $i = 1, \ldots, m$.

Атака на асимметричный шифр в случае $J = \{\pi_2\}$.

Функция f строится так: $f(x) = \pi_2(g(x))$. Пусть имеется m пар открытых текстов и шифртекстов длины n вида (P_i, C_i), $i = 1, ..., m$.

Атака с известным открытым текстом может быть проведена следующим образом:

1) строим матрицу C со строками C_i, $i = 1, ... , m$;
2) строим матрицу C' со строками $g(P_i)$, $i = 1, ... , m$;
3) находим матрицу $D = T(C', C)$, в которой содержатся все возможные перестановки π_2, удовлетворяющие системе:
   $C_i = \pi_2(g(P_i))$, $i = 1, \ldots, m$.

Сложность приведенных атак равна O(nm), так как для построения матрицы перестановок для m пар (открытый текст, шифртекст) длины n требуется nm действий. Следовательно, данные атаки эффективны в сравнении с атакой грубой силы, сложность которой равна O(n!).

Выводы:

В данной главе представлен алгоритм построения матрицы $D = T(A, B)$, содержащей все перестановки π такие, что $B = \pi(A)$. Описаны две атаки на асимметричный шифр на булевых функциях в случае, когда $J = \{\pi_1\}$ и $J = \{\pi_2\}$. Обе атаки сводятся к решению вспомогательной задачи 1.

3 Первый вариант атаки на асимметричный шифр с двумя перестановками

Рассмотрим вспомогательную задачу 2. Пусть даны две булевых матрицы A и B размера $m \times n$. Требуется выяснить, получена ли матрица B из A с помощью какой-либо перестановки.

Строим матрицу $D = T(A, B)$ по формуле (1). Матрица D обладает следующими свойствами [3]: Если в матрице A все столбцы различны, то существует единственная перестановка π со свойством $B = \pi(A)$. Если множество столбцов матрицы A можно разбить на классы $Q_1, ..., Q_k$ одинаковых столбцов, $1 \le k \le n$, так, что $|Q_j| = r_j$, $j = 1, ..., k$, $r_1 + ... + r_k = n$, то множество строк матрицы $D = T(A, B)$ тоже разбивается на k классов одинаковых строк мощностей $r_1, ..., r_k$. Вес строк в каждом классе равен мощности класса, и все возможные перестановки π, для которых верно $\pi(A) = B$, удовлетворяют условию $\pi(i) = j \Leftrightarrow d_{ij} = 1$. Количество таких перестановок $Q = \prod_{i=1}^{k} r_i!$. В таком случае матрица B может быть получена из A с помощью какой-либо перестановки, находящейся в D. В примере 1 это можно наглядно увидеть.

Рассмотрим случай, когда семейства столбцов в матрицах A и B различны, т. е. матрицу B невозможно получить из матрицы A никакой перестановкой столбцов. Пусть $B(j_1) = \ldots = B(j_r) = A(k_1) = ... = A(k_s)$ и $r \neq s$ (в частности, может быть $s = 0$), тогда в матрице $D = T(A, B)$ строки $D_{j_1}, ..., D_{j_r}$ одинаковы и имеют по s единиц — мощность класса не равна весу строк в нём. В таком случае D не является матрицей перестановок и, соответственно, матрица B не может быть получена из A с помощью какой-либо перестановки.

Пример 2:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \qquad D = T(A,B) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

D не является матрицей перестановок, так как вес строк в классе $\{D_1, D_2\}$ равен 1, а в классе D_3 равен 2, т.е. вес не совпадает с мощностью классов.

Решая вспомогательные задачи 1 и 2 мы можем осуществить атаку на асимметричный шифр с двумя перестановками.

Множество $J = \{\pi_1, \pi_2\}$. Функция f строится так: $f(x) = \pi_2(g(\pi_1(x)))$.

Пусть имеется m пар открытых текстов и шифртекстов вида (P_i, C_i), $i = 1, ..., m$. Составим матрицы P и C размера $m \times n$ со строками P_i и C_i соответственно:

$$C_i = \pi_2(g(\pi_1(P_i))),\ i = 1, \ldots, m. \qquad (2)$$

Атака с известным открытым текстом может быть проведена следующим образом:

1) строим матрицу C со строками C_i, $i = 1, \ldots, m$;

2) перебираем $n!$ перестановок π_1' («кандидатов» в π_1);

3) для каждой π_1' строим матрицу C' со строками $g(\pi_1'(P_i))$, $i = 1, \ldots, m$;

4) находим $D = T(C', C)$;

5) если D не является матрицей перестановок, то π_1' не может быть частью ключа, иначе все пары (π_1, π_2), где $\pi_1 = \pi_1'$ и π_2 содержатся в D, удовлетворяют системе (2).

Здесь на шаге 4 и 5 возникают вспомогательные задачи 1 и 2.

Сложность приведенной атаки равна O(n!), так как происходит перебор n! перестановок в кандидаты π_1. Следовательно, данная атака эффективна в сравнении с атакой грубой силы, сложность которой равна O($n!^2$).

Выводы:

В данной главе описано свойство матрицы $D = T(A, B)$ о мощностях классов одинаковых строк, на которые разбивается D. Представлен первый вариант атаки на асимметричный шифр на булевых функциях в случае, когда $J = \{\pi_1, \pi_2\}$.

4 Алгоритмы проверки матрицы перестановок

Задача: выяснить, является ли матрица D размера n на n матрицей перестановок.

4.1 Алгоритм 1

Краткое описание: формируем массив от 0 до 2^n такой, что элемент с индексом i будет содержать количество строк в матрице D, которые равны индексу i; затем проверяем, чтобы элемент по индексу i равнялся весу i.

Вход: матрица D, d_j – строки матрицы D, $j = 0,\ldots, n\text{-}1$.

Выход: r - ответ на вопрос в задаче; $r = 1$, если «да», и $r = 0$ иначе.

1. Построим массив N мощностью 2^n, где n – длина строки в D.
2. Для $j = 0, \ldots, n\text{-}1$ выполняем: $N[d_j] := 0$.
3. Для $j = 0, \ldots, n\text{-}1$ выполняем: $N[d_j] := N[d_j] + 1$.
4. Для $j = 0, \ldots, n\text{-}1$: если $(N[d_j]) \neq w(d_j)$, то $r := 0$ и выход.
5. $r := 1$ и выход.

4.2 Алгоритм 2

Краткое описание: сортируем массив строк из D и проверяем, чтобы в массиве подряд находилось такое число равных элементов, которое равно весу этих элементов.

Вход: матрица D, d_j – строки матрицы D, $j = 0,\ldots, n\text{-}1$.

Выход: r - ответ на вопрос в задаче, $r = 1$, если «да», и $r = 0$ иначе.

1. Копируем D в K и сортируем D.
2. $j := 0$.
3. Пока $j < n$, выполняем шаги 4 – 9.
4. Если $w(d_j) = 1$, то переходим на шаг 9.
5. Если $w(d_j) = 0$, то $r := 0$ и выход.
6. $i := j + w(d_j) - 1$.
7. Пока $j < i$, выполняем шаг 8.
8. Если $d_j = d_{j+1}$, то $j := 1 + j$, иначе $r := 0$ и выход.
9. Если $d_j \neq d_{j+1}$, то $j := 1 + j$, иначе $r := 0$ и выход..
10. $r := 1$ и выход.

4.3 Сравнение двух алгоритмов

Сравнение алгоритмов по сложности и затрачиваемой памяти приведено в таблице 1.

Таблица 1 - Сравнение алгоритмов по сложности и затрачиваемой памяти, n - порядок D

Алгоритм	**Характеристика**	
	Сложность алгоритма	**Затрачиваемая память**
Алгоритм 1	O(n) Выполняются два последовательных цикла из n итераций: первый для заполнения массива N, второй для проверки весов.	Хранение массива 2^n элементов
Алгоритм 2	O(n^2) Последовательно выполняются сортировка сложностью O(n^2) и цикл из n итераций.	Хранение копии массива D из n элементов

Алгоритмы реализованы на языке ЛЯПАС [приложение А] и проведено их экспериментальное исследование.

Сравнение алгоритмов по времени работы при разных n и одинаковых матрицах D представлено в таблице 2. Данные приведены о времени работы алгоритмов для 10^7 запусков в секундах.

Таблица 2 - Сравнение алгоритмов по времени работы

n	2-5	6-9	10-11	12-15	17	18-19	20-21	22-24	25	26	27	28	29
Алг. 1	0	0	1	1	1	1	1	1	2	2	2	2	2
Алг. 2	0	1	1	2	3	4	5	6	7	8	8	9	9

Таким образом, получаем, что первый алгоритм выполняется быстрее второго, но проигрывает по памяти, и первый алгоритм работает для $n <= 29$ (ограничение по памяти), а второй для $n <= 31$ (ограничение по разрядности переменной в ЛЯПАСе).

4.4 Выводы

В данной главе представлено описание двух алгоритмов проверки матрицы $D = T(A, B)$ – является ли D матрицей перестановок. Проведены исследования и сравнения двух алгоритмов и представлены результаты.

5 Алгоритм извлечения перестановок

Задача: извлечь из матрицы перестановок $D = T(P, C)$ одну случайную перестановку в матричном виде.

Краткое описание: берем строку матрицы D, выбираем случайную единицу в ней и обнуляем столбец D, в котором она находится, кроме нее самой. Затем в текущей строке обнуляем все позиции кроме выбранной единицы. Полученную строку под тем же индексом записываем в матрицу G и переходим к очередной строке в D. В итоге в матрице G будет храниться единственная случайная перестановка.

Вход: матрица D, d_j – строки матрицы D, $j = 0,\ldots, n\text{-}1$.

Выход: матрица G, g_i – строки матрицы G, $i = 0,\ldots, n\text{-}1$.

1. Копируем D в G.
2. $j := 0$.
3. Пока $j < n$, выполняем шаги 4 – 9.
4. Если $w(g_j) = 1$, то переходим на шаг 9.
5. Пусть m – булев вектор с одной единице на k-м месте, где k – индекс случайной единицы в векторе g_j.
6. $g_j := m$.
7. $i := j + 1$.
8. Для всех $i = j+1, \ldots, n\text{-}1$: если $g_i \,\&\, m \neq 0$, то $g_i := g_i \oplus m$.
9. $j := 1 + j$.

Чтобы получить все перестановки, следует повторить алгоритм многократно. Среднее количество запусков для нахождения всех перестановок в матрице из всех единиц при разных n приведены в таблице 3.

Таблица 3 - Среднее количество запусков для нахождения всех матриц перестановок в матрице из всех единиц

N	2	3	4	5	6	7	8	9
Кол. запусков	3,5	15	92	660	5216	45335	416477	5020459

Обсудим полученные экспериментальные данные. Матрица из всех единиц содержит $n!$ перестановок и при каждом запуске мы извлекаем одну случайную. При первом запуске вероятность извлечь новую матрицу $p_1 = 1$, и соответственно, количество запусков равно 1. При втором запуске вероятность извлечения новой перестановки $p_2 = (n!-1)/n!$, и соответственно, запусков требуется $1/p_2$. Продолжая так до последней матрицы, мы получим, что общее количество запусков $M_n = 1 + \frac{n!}{n!-1} + \frac{n!}{n!-2} + \ + n! = n!(\frac{1}{n!} + \cdots + 1) = n!H_{n!}$, где $H_{n!}$ – это частичная сумма гармонического ряда, причем $\ln(n!) < H_{n!} < \ln(n!) + 1$. Для небольших n значения посчитаны и в соответствии с ними $M_3 = 14{,}7$, $M_4 \approx 87$, $574 < M_5 < 694$, $42966 < M_7 < 48006$ и так далее. Таким образом, результаты экспериментов согласуются с теоретическими оценкам, а так как реализация была на языке ЛЯПАС, то это обоснование подтверждает, что алгоритм случайных чисел реализован в языке хорошо.

Сложность алгоритма зависит от матрицы D. В лучшем случае, когда в D содержится одна перестановка, производится n вычислений веса. В худшем случае, когда в D содержится $n!$ перестановок, выполняются два цикла, один в другом. Внешний из n итераций, а внутренний из n-i итераций, где i – текущая итерация внешнего цикла. Следовательно, получаем арифметическую прогрессию и соответствующую сложность: $n(n+1)/2$, $O(n^2)$.

Выводы:

В данной главе представлено описание алгоритма извлечения перестановки из $D = T(P, C)$. Сделана оценка сложности алгоритма и проведены эксперименты.

6 Второй вариант атаки на асимметричный шифр с двумя перестановками

Множество $J = \{\pi_1, \pi_2\}$. Функция f строится так: $f(x) = \pi_2(g(\pi_1(x)))$.

Пусть имеется m пар открытых текстов и шифртекстов вида (P_i, C_i), $i = 1, ..., m$. Составим матрицы P и C размера $m \times n$ со строками P_i и C_i соответственно:

$$C_i = \pi_2(g(\pi_1(P_i))), i = 1, \ldots, m. \quad (3)$$

Заметим, что мы можем попытаться сократить перебор всех кандидатов в π_1, учитывая, что вес булева вектора не меняется после применения перестановки. Для этого на шаге 3 при построении C' мы можем проверять условие: $w(g(\pi_1'(P_i))) = w(C_i)$, $i = 1,\ldots,m$. И, следовательно, использовать только те π_1' в кандидаты π_1, которые удовлетворяют этому условию.

После преобразования атака будет выглядеть следующим образом:

1) строим матрицу C со строками C_i, $i = 1, ... , m$;
2) перебираем $n!$ перестановок π_1' («кандидатов» в π_1);
3) для $i = 1,\ldots,m$, если $w(g(\pi_1'(P_i))) = w(C_i)$, то строим $C_i' = g(\pi_1'(P_i))$, иначе переходим к следующему π_1';
4) находим $D = T(C', C)$;
5) если D не является матрицей перестановок, то π_1' не может быть частью ключа; иначе все пары (π_1, π_2) где $\pi_1 = \pi_1'$ и π_2 содержится в D, удовлетворяют системе (3).

Также, в качестве предположения, можно сократить и перебор всех пар (P_i, C_i) при проверке равенства весов путем упорядочивания их так, чтобы сначала шли шифртексты с максимально далеким от $n/2$ весом. Так как, например, векторов с $n/2$ единицами больше, чем векторов с одной единицей.

7 Исследование и сравнение атак на асимметричный шифр с двумя перестановками

Для исследования и сравнения двух вариантов атак поставим следующие задачи:

1) выявить зависимость количества возможных пар кандидатов в (π_1, π_2) в качестве ответа (т.е. пар, которые могли использоваться в качестве ключа в шифре) от количества открытых текстов при разных n;
2) исследовать зависимость количества возможных пар кандидатов в (π_1, π_2) в качестве ответа от значений ключа (π_1, π_2) и функции g;
3) выяснить, что трудозатратнее – вычисление m весов векторов $g(\pi_1'(P_i))$ во втором варианте или проверка того, является ли матрица D матрицей перестановок, сравнить атаки по скорости выполнения;
4) сколько в среднем строк матрицы C' требуется построить, прежде чем отбросить неподходящую π_1' в втором варианте атаки;
5) стоит ли перед построением C' переупорядочить пары (P_i, C_i) так, чтобы сначала шли шифртексты с далеким от $n/2$ весом;
6) выяснить зависимость эффективности второго варианта атаки от функции g.

7.1 Исследование 1

Задача: исследовать зависимость количества возможных пар кандидатов в (π_1, π_2) в качестве ответа от значений ключа (π_1, π_2) и функции g.

Утверждение 1. Пусть функция g – тождественная, и имеется m пар открытых текстов и шифртекстов вида (P_i, C_i), $i = 1, ..., m$. Составим матрицы P и C размера $m \times n$ со строками P_i и C_i соответственно. Тогда количество пар

кандидатов в (π_1, π_2) будет $L = n!\prod_{i=1}^{k} r_i!$, где $r_i = |Q_i|$, $i = 1, ..., k$, $r_1 + ... + r_k = n$ и $Q_1, ..., Q_k$ – классы одинаковых столбцов в C.

Доказательство

Так как g – тождественная, то $C_i = \pi_2(g(\pi_1(P_i))) = \pi_2(\pi_1(P_i)) = \pi(P_i)$, $i = 1, \ldots, m$. Возьмем случайную перестановку π_1' и построим матрицу $C_i' = g(\pi_1'(P_i)) = \pi_1'(P_i)$. Так как матрица C получена с помощью перестановки π из P, то C можно получить с помощью какой-либо перестановки из C'. Тогда можем построить матрицу $D = T(C', C)$ и по свойству D, если множество столбцов матрицы C можно разбить на классы $Q_1, ..., Q_k$ одинаковых столбцов так, что $|Q_i| = r_i$, $i = 1, ..., k$, $r_1 + ... + r_k = n$, то множество строк матрицы D тоже разбивается на k классов одинаковых строк мощностей $r_1, ..., r_k$ и количество таких перестановок π_2', что $C_i = \pi_2'(C_i')$, равно $Q = \prod_{i=1}^{k} r_i!$. Т.е. для каждого π_1' существует Q штук π_2'. И так как количество π_1' равно $n!$ (всевозможные перестановки), то всего пар кандидатов в (π_1, π_2) будет $L = n!\prod_{i=1}^{k} r_i!$.

Утверждение 2. Предположим, что функция g – случайна, и имеется m пар открытых текстов и шифртекстов вида (P_i, C_i), $i = 1, ..., m$. Составим матрицы P и C размера $m \times n$ со строками P_i и C_i соответственно, $C_i = \pi_2(g(\pi_1(P_i)))$, $i = 1, \ldots, m$. Тогда количество пар кандидатов в (π_1, π_2) будет не меньше, чем $L = \prod_{j=1}^{l} s_i! \prod_{i=1}^{k} r_i!$, где $r_i = |Q_i|$, $i = 1, ..., k$, $r_1 + ... + r_k = n$, $s_j = |G_j|$, $j = 1, ..., l$, $s_1 + ... + r_l = n$ и $Q_1, ..., Q_k$ – классы одинаковых столбцов в C, а $G_1, ..., G_l$ – классы одинаковых столбцов в P.

Доказательство

Возьмем перестановку π_1' такую, что выполняется равенство $C_i = \pi_2(g(\pi_1'(P_i)))$ и построим матрицу $C_i' = g(\pi_1'(P_i))$. Так как матрицу C можно получить с помощью перестановки π_2 из C', то мы можем построить матрицу $D = T(C', C)$ и по свойству D, если множество столбцов матрицы C можно разбить на классы $Q_1, ..., Q_k$ одинаковых столбцов так, что $|Q_i| = r_i$, $i = 1, ..., k$,

$r_1 + ... + r_k = n$, то множество строк матрицы D тоже разбивается на k классов одинаковых строк мощностей $r_1, ..., r_k$ и количество таких перестановок π_2, что $C_i = \pi_2(C_i')$ равно $Q = \prod_{i=1}^{k} r_i!$. Т.е. для каждого π_1' существует Q штук π_2'. И так как количество π_1' равно $n!$, то всего пар кандидатов в (π_1, π_2) будет $L = n!\prod_{i=1}^{k} r_i!$. Теперь для π_1' построим матрицу $P_i' = g^{-1}(C_i') = \pi_1'(P_i)$, т.е. P' получена с помощью перестановки π_1' из P, и аналогично рассуждая, по свойству D таких перестановок π_1' существует $\prod_{j=1}^{l} s_i!$. Следовательно, всего пар кандидатов в (π_1, π_2) будет не меньше, чем $L = \prod_{j=1}^{l} s_i! \prod_{i=1}^{k} r_i!$.

Были проведены соответствующие эксперименты с алгоритмами атак и данные двух атак совпали между собой и согласуются с оценками из утверждений 1 и 2.

7.2 Эксперимент 1

Задача: определить, сколько требуется в среднем открытых текстов (строк), чтобы все столбцы в матрице P были разные с вероятностью больше 90%.

Проведен эксперимент, в котором алгоритм работал для 10000 построений случайных P с n столбцами и m строками и ответом выдавал количество матриц P, в которых все столбцы были различны. В таблице 4 приведены минимальные значения m для каждого n, при которых с вероятностью 90% в P все столбцы были различными

Таблица 4 - Минимальные значения m, при которых с вероятностью 90% в A все столбцы различны:

n	2	3	4	5	6-8	9-11	12-16	17-21	22-31
m	4	5	6	7	8	9	10	11	12
2*(*log*(*n*)+1)	4	6	6	8	8	10	10	12	12

Таким образом, мы можем сделать следующую предположительную оценку сложности атаки: если будет использоваться в среднем $2(\log(n)+1)$ открытых текстов, то матрица D будет содержать одну перестановку и шаг 5 атаки будет осуществлен за полиномиальное время. Т.е. для осложнения атаки следует не допускать, чтобы у взломщика был доступ более, чем к $2(\log(n)+1)$ парам (открытый текст, шифртекст) или менять ключи после шифрования $2(\log(n)+1)$ блоков.

7.3. Эксперимент 2

Задача: сравнить оба варианта атаки по скорости выполнения.

Эксперимент проводился при одинаковых данных (перестановки π_1 и π_2, функция g, открытые текст). Алгоритм эксперимента запускался по 10000 раз для каждого n при разных значениях m и выдавал скорость выполнения каждой атаки. Результаты при разных m были одинаковые, поэтому на рисунке 1 представлен график для фиксированного m= 10.

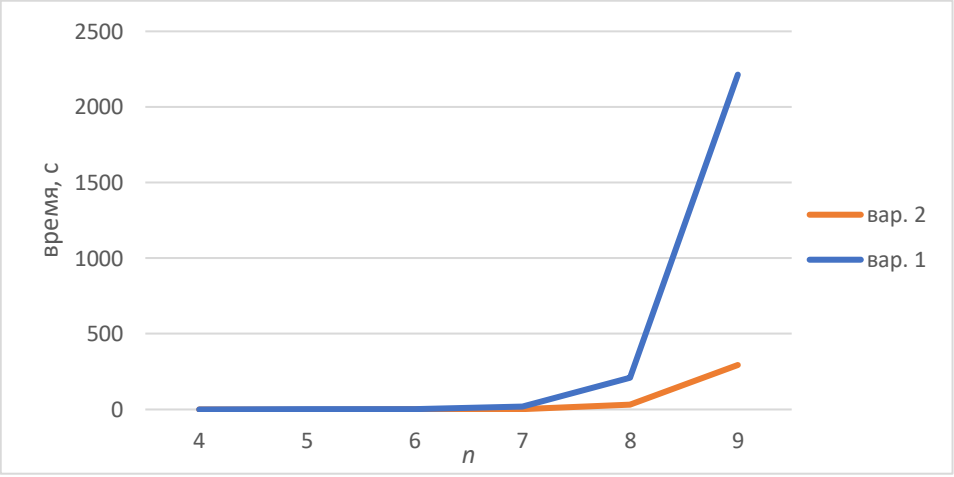


Рисунок 1 – Разница скоростей выполнения двух вариантов атак

Таким образом, можно сделать вывод, что второй вариант атаки эффективнее и быстрее справится с работой в случае, когда функция g

случайная. Связано это с тем, что мы отбрасываем неподходящие перестановки прежде, чем переходим к построению D.

В случае, когда функция g сохраняет вес для каждого аргумента и, соответственно, нам приходится строить матрицу C' полностью, то скорость выполнения второй атаки становится равна скорости первой с точностью до секунды.

Также были проведены эксперименты, проверяющие, зависит ли скорость выполнения первого варианта атаки от ключа (π_1, π_2) и функции g. Результаты показали, что скорости одинаковые при разных значениях перестановок и g.

Таким образом, можно сделать вывод, что второй вариант атаки эффективнее первого, но эффективность второй атаки относительно самой себя зависит от функции g. Т.е. для осложнения осуществления данной атаки можно использовать функцию, сохраняющую веса, но, возможно это упростит другие атаки на шифр. К примеру, если открытый текст будет с одной единицей, то злоумышленник сразу будет знать, что зашифрованный текст - это вектор с одной единицей. Эту уязвимость мы можем устранить тем, что будем использовать функцию, которая сохраняет веса на векторах с весом близким к $n/2$ и не сохраняет для далеких.

7.4 Эксперимент 3

Задача: выяснить, сколько в среднем строк матрицы C' требуется построить, прежде чем отбросить неподходящую π_1', и определить, стоит ли перед построением C' переупорядочить пары (P_i, C_i) так, чтобы сначала шли шифртексты с далеким от $n/2$ весом.

Эксперимент проводился при одинаковых данных (перестановки π_1 и π_2, функция g, открытые текст). Алгоритм запускался по 100 раз для каждого n и выдавал среднее количество строк, у которых приходилось считать вес, прежде чем отбросить неподходящую перестановку.

Эксперимент показал, что для $n > 3$ это среднее число равно одному, что является хорошей новостью для атакующего и объясняет эффективность данного варианта атаки. Но как уже было сказано, эту уязвимость закрывает сохраняющая вес функция.

Чтобы выяснить, стоит ли перед построением C' переупорядочить пары (P_i, C_i) так, чтобы сначала шли шифртексты с далеким от $n/2$ весом, использовался этот же алгоритм, но чтобы увидеть разницу, он выдавал не среднее количество сравнений, а сумму всех сравнений для каждого запуска и для каждого кандидата в π_1'. Результаты эксперимента представлены в виде графика на рисунке 2.

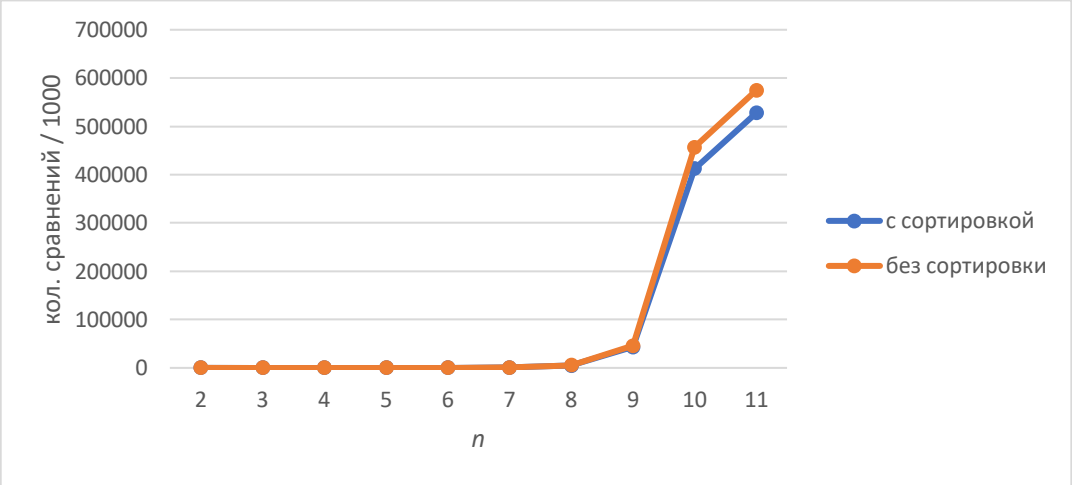


Рисунок 2 – Зависимость эффективности выполнения второго варианта атаки с переупорядочиванием и без

Таким образом, можно сделать вывод, что сортировка перед построением C' играет незначительную роль. Но, вспоминая эксперимент 3, если мы будем использовать функцию, которая сохраняет веса на векторах с весом близким к $n/2$ и не сохраняет для далеких, то данная сортировка значительно поможет в

осуществлении атаки, вернув среднее значение построений строк в C' к одному.

7.5 Выводы

В данной главе поставлены задачи для исследования двух вариантов атак на асимметричный шифр на булевых функциях в случае, когда $J = \{\pi_1, \pi_2\}$. Проведены исследования зависимости количества возможных пар кандидатов в (π_1, π_2) в качестве ответа от значений ключа (π_1, π_2) и функции g, в результате чего доказано два утверждения.

Описаны эксперименты, решающие задачи, поставленные в начале главы, и представлены их результаты, на основе которых сделаны соответствующие выводы в конце каждого эксперимента.

ЗАКЛЮЧЕНИЕ

В рамках данной работы разработаны и исследованы атаки на асимметричный шифр, когда ключевой параметр состоит из одной или двух перестановок. Разработан и реализован алгоритм построения матрицы $D = T(A,B)$. Разработаны и реализованы алгоритмы, которые позволяют выяснить, является ли матрица $D = T(A,B)$ матрицей перестановок, т.е. проверить, получена ли матрица B из матрицы A перестановкой столбцов. Разработан и реализован алгоритм, который в случае, когда матрица $D = T(A,B)$ является матрицей перестановок, позволяет извлечь одну случайную перестановку в матричном виде. Проведены исследования алгоритмов.

Проведены исследования и сравнения атак на асимметричный шифр с двумя перестановками. Выдвинуто и доказано два утверждения с оценками о количестве возможных пар кандидатов в (π_1, π_2) в качестве ответа. Сделана вероятностная оценка количества открытых текстов, когда с вероятностью 90% в качестве возможного ответа будет выступать одна пара (π_1, π_2). В результате сравнения двух вариантов атак по скорости установлено, что второй вариант эффективнее и быстрее. Обосновано это отбрасыванием неподходящей перестановки прежде, чем перейдем к построению D. При этом эффективность второй атаки относительно самой себя зависит от функции g. Установлено, что для $n>3$ в среднем требуется построить одну строку матрицы C', прежде чем отбросить неподходящую $\pi_1^{'}$, и сортировка перед построением $C^{'}$ играет незначительную роль в эффективности атаки, когда g - случайна.

ЛИТЕРАТУРА

1. Agibalov G.P. and Pankratova I. A. Asymmetric cryptosystems on Boolean functions // Прикладная дискретная математика. 2018. №40. С.23–33.
2. Agibalov G.P. ElGamal cryptosystems on Boolean functions // Прикладная дискретная математика. 2018. №42. С.57–65.
3. Боровкова И.В., Кондратьев В.А., Панкратова И.А. Криптоанализ асимметричного шифра на булевых функциях // Прикладная дискретная математика. 2020. №50. С.42–50.
4. Боровкова И.В., Панкратова И.А. Криптоанализ шифрсистемы ACBF // Прикладная дискретная математика. Приложение. 2019. №12. С.91–92.

ПРИЛОЖЕНИЕ А

Программный код функций, необходимых для осуществления атак на асимметричный шифр на булевых функциях.

Построение матриц *D* = *T*(*A*,*B*).

```
alg_D(L3,L4,n/L7) ***на входе матрицы A, B и количество переменных n
¯j ¯h Q3⇒g            ***на выходе матрица D
  §2 Δj⊕n↪5 In-1⇒c ¯i
  §3 Δi⊕g↪6 L4i&Ij↪4 c&L3i⇒c →3
  §4 L3i¬&c⇒c →3
  §6 c⇒L7j →2
  §5 **
```

Проверка матрицы *D* = *T*(*A*,*B*), т.е. является ли *D* матрицей перестановок.

Алгоритм 1.

```
alg_one(L7/r)                  ***на вход матрица D
  1<Q7⇒m @+L8(m) m⇒Q8          ***на выходе 0, если D не является матрицей
  Oi Or                        ***перестановок, и 1 иначе
  §4 L7i⇒k OL8k  Δi ↑(i<Q7)4 Oi
  §1 L7i⇒k ΔL8k  Δi ↑(i<Q7)1 Oi
  §2 L7i%⇒w L7i⇒k ↑(L8k≠w)3 Δi ↑(i<Q7)2 Δr
  §3 **
```

Алгоритм 2.

```
alg_two(L9/r)                  ***на вход матрица D
  *sort(L9/L9)                   ***на выходе 0, если D не является матрицей
  Oi Ow Or                       ***перестановок, и 1 иначе
  §1 L9i%⇒w ↑(w=1)4 w↪6 i+w-2⇒k
  §2 ↑(i>k)4 i⇒t Δi L9i⊕L9t ↪2 →6
  §4 i⇒j Δi↑(i=Q9)5 L9i⊕L9j ↪6  →1
  §5 Δr
  §6 **
```

Извлечение одной случайноq перестановки из *D*.

```
ext_per(L9/L9)          ***на входе матрица D
    ¯i Q9⇒n 32-n⇒e      ***на выходе одна случайная перестановка, записанная в D
  §2 Δi ↑(i≥n)6 L9i%⇒w ↑(w=1)2 X>e;w⇒j Ol
  §3 ↑(j=l)4 L9i!⇒k Ik⊕L9i⇒L9i Δl →3
  §4 L9i!⇒k Ik&L9i⇒L9i i⇒j
  §5 Δj ↑(j≥n)2 Ik&L9j↪5 Ik⊕L9j⇒L9j →5
  §6 **
```

Сортировка массива *D*.

```
sort(L9/L9)              ***на входе массив D
  Q9-1⇒k k↪4 Oi     ***на выходе отсортированный массив D
  §1 ↑(i=k)3 i+1⇒j ↑(L9i>L9j)2 Δi →1
  §2 ⇔(L9ij) Δi →1
  §3 ∇k Oi ↑(k>0)1
  §4 **
```

Перестановка столбцов булевой матрицы *A*.

```
perestanovka(L1,L2/L3)    ***на входе булева матрица A и перестановка π
  Oi                              ***на выходе булева матрица π(A)
  §1 L1i⇳L2⇒L3i Δi ↑(i<Q1)1
  **
```

Генерация случайной перестановки $\pi \in S_n$.

```
permutation(L2/L2)   ***на входе упорядоченный массив из n значений равных от 0 до n-1
  Q2⇒m-1⇒l           ***применяется тасование Кнута и на выходе неупорядоченный
                     ***входной массив из n значений равных от 0 до n-1
  §1  ∇l ↑(l=0)2 X>8;m⇒j ⇔(L2jl) →1
  §2  **
```

Вычисление количества перестановок в $D = T(A,B)$.

```
vol_per(L9/l)     ***на входе D
  *sort(L9/L9)    ***на выходе количество перестановок в D
  Oi Ow Or 1⇒l
  §1 i+w⇒i ↑(i≥Q9)2 L9i%⇒w ↑(w=1)1 *faktorial(w/h) h*l⇒l →1
  §2 **
```

Вычисление факториала числа *n*.

```
faktorial(n/l)   ***на входе число n
  Oi 1⇒l         ***на выходе факториал числа n
  §1 ↑(i≥n)2 Δi*l⇒l →1
  §2 **
```

Построение функции *f* сохраняющей вес аргументов.

```
func_w(L2,n/L2)          *** на входе упорядоченный массив из 2^n значений равных
  Q2⇒l @+F1(10)          *** от 0 до 2^n-1 и количество переменных функции n
  Oi @+L15(n) n⇒Q15   ***на выходе функция f
  §1 i⇒L15i Δi ↑(i<n)1 Oi       ***берем очередной элемент входного массива, делаем случайную
  §2 *permutation(L15/L15)   ***перестановку его битов до тех пор, пока не получится элемент,
  §3 L2i⇳L15⇒t i↪6 i⇒j       ***которого нет ранее в массиве
  §5 ∇j t⊕L2j↪2 j↪6 →5
  §6 t⇒L2i Δi ↑(i≥l)7 →2
  §7 **
```

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: Кондратьев Вадим
Проверяющий: Кондратьев Вадим (wadim.condratjev@yandex.ru / ID: 6400099)

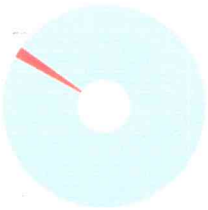
Отчет предоставлен сервисом «Антиплагиат» - users.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 57
Начало загрузки: 19.01.2022 11:58:56
Длительность загрузки: 00:00:01
Имя исходного файла:
ВКР_6_курс_Кондратьев_В_А (5).pdf
Название документа:
ВКР_6_курс_Кондратьев_В_А (5)
Размер текста: 38 кБ
Символов в тексте: 38760
Слов в тексте: 5383
Число предложений: 269

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 19.01.2022 11:58:58
Длительность проверки: 00:00:02
Комментарии: не указано
Модули поиска: Интернет Free



ЗАИМСТВОВАНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
1,55%	0%	0%	98,45%

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

№	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	1,11%	http://vital.lib.tsu.ru/vital/access/services/Download/vital:6890/SOURCE01 http://vital.lib.tsu.ru	07 Сен 2020	Интернет Free
[02]	0,44%	http://vital.lib.tsu.ru/vital/access/services/Download/vital:8382/SOURCE01 http://vital.lib.tsu.ru	24 Янв 2020	Интернет Free
[03]	0%	http://vital.lib.tsu.ru/vital/access/services/Download/vital:11008/SOURCE01 http://vital.lib.tsu.ru	24 Янв 2020	Интернет Free