Элементы криптографического анализа

Автор курса: Тимонина Елена Евгеньевна
Составитель: Смирнов Дмитрий Константинович

Версия от 12:08, 15 июня 2022 г.

Оглавление

Часть 1

Домашние задания

1.1 Определение шифра. Простейшие примеры.

Задача 1.1 Что такое подстановка?
Решение. Подстановка — это взаимно однозначная функция, которая переводит буквы алфавита в буквы того же самого алфавита.

Задача 1.2 Что такое группа, и почему множество S_m из примера 2.1 образует группу?
Решение. Множество $G \neq \varnothing$ с бинарной операцией "$\circ$", называется *группой*, если выполнены условия:

1. $\forall a, b \in G \;\; a \circ b \in G$;
2. $\forall a, b, c \in G \;\; a \circ (b \circ c) = (a \circ b) \circ c$;
3. $\exists e \in G \colon \forall a \in G \;\; e \circ a = a \circ e = a$;
4. $\forall a \in G \;\; \exists b \in G \colon a \circ b = b \circ a = e$

Множество S_m вводится как множество всех подстановок на конечном алфавите $A = \{a_1, ..., a_m\}$. Проверим выполнение аксиом группы:

1. Подстановка $k \in S_m$ – отображение $k \colon A \to A$. $\forall k_1, k_2 \in S_m$ рассмотрим суперпозицию $k_1 \circ k_2$. Так как $k_1 \circ k_2 \colon A \to A \to A$, то $k_1 \circ k_2 \in S_m$ и первая аксиома верна.

2. $\forall k_1, k_2, k_3 \in S_m \;\;\; k_1 \circ (k_2 \circ k_3) = k_1 \circ k_2(k_3(a)) = k_1(k_2(k_3(a))) = k_1(k_2(a)) \circ k_3(a) = (k_1 \circ k_2) \circ k_3$.

3. Поскольку S_m – множество всех подстановок, то найдётся тождественная подстановка: $\exists e \in S_m \colon \forall a \in A \;\; e(a) = a$. Тогда $\forall k \in S_m$ верно $e \circ k = e(k(a)) = k(a) = k(e(a)) = k \circ e$.

4. Так как подстановка – взаимно однозначная функция, то $\forall k \in S_m$ существует обратная функция: $\exists k^{-1} \colon A \to A \Rightarrow k^{-1} \in S_m$, для которой будет выполнено равенство $k \circ k^{-1} = k(k^{-1}(a)) = k^{-1}(k(a)) = k^{-1} \circ k$. При этом, $\forall a \in A \;\; k^{-1}(k(a)) = a = e(a)$.

Выполнены все аксиомы группы, следовательно S_m – группа.

Задача 1.3 Почему группа S_n из примера 2.2 является симметрической?
Решение. Симметрической группой n-го порядка называется множество S(X) всех биективных отображений $f \colon X \to X$, где X – конечное множество из n элементов. Группа S_n в примере 2.2 определяется как группа подстановок на множестве $X = \{1, ..., n\}$. Подстановка – это биективное отображение, X – конечное множество из n элементов. Следовательно, по определению, группа S_n является симметрической.

Задача 1.4 Что такое кольцо? Что такое кольцо вычетов по модулю m?

Решение. Множество K называется *кольцом*, если в K определены две операции "$+$" (сложение) и "$\cdot$" (умножение) и выполняются следующие условия $\forall a, b, c \in K$:

1. $a + b \in K, a \cdot b \in K$;
2. $a + (b + c) = (a + b) + c,\ a(bc) = (ab)c$;
3. $a + b = b + a$;
4. $(a + b)c = ac + bc$;
5. $\exists 0 \in K\colon a + 0 = a$.

Кольцом вычетов по модулю m называется такое кольцо $\mathbb{Z}_{/m} = \{C_0, C_1, ..., C_{m-1}\}$ (C_r – смежный класс вычетов по модулю m), в котором операции сложения и умножения определяются следующими правилами:

1. $C_a + C_b = C_r, \quad$ где $r \equiv (a + b)(\text{mod } m)$;
2. $C_a C_b = C_r, \quad$ где $r \equiv ab(\text{mod } m)$

То есть, $C_a + C_b$ – это класс, в который входит число $a + b$, а $C_a C_b$ – класс, в который входит число ab.

Задача 1.5 Какую алгебраическую структуру представляет собой кольцо $\mathbb{Z}_{/m}$ при $m = 2$?

Решение.

Теорема 1.1 Если p – простое число и $p \geq 2$, то $\mathbb{Z}_{/m}$ – поле характеристики p.

По приведённой выше теореме кольцо $\mathbb{Z}_{/2}$ является полем характеристики 2.

1.2 Стойкость шифров. Метод полного перебора.

Задача 2.1 Дан алфавит $A = \{1, 2, ..., n\}$, x – открытый текст в алфавите A. Ключ шифрования (T_1, T_2, T_3), где T_i – случайные подстановки. Алгоритм шифрования: $T_3(T_2(T_1(x))) = y$. Какова формула для расшифрования? Мощность пространства различных ключей? Сложность МПП?

Решение.

1. Формула для расшифрования – $x = T_1^{-1}(T_2^{-1}(T_3^{-1}(y)))$.

2. В каждой подстановке на первое место можно поставить n различных букв, на второе – $n-1$, и т.д. В итоге получаем $n!$ вариантов на каждую подстановку, следовательно, $|K| = (n!)^3$ для трёх подстановок.

3. Пусть в тексте a букв. Тогда необходимо провести $3a$ операций подстановки, чтобы проверить один ключ. В среднем нужно проверить количество ключей, равное средней трудоёмкости МПП: $E\tau = \frac{|K|+1}{2} = = \frac{(n!)^3+1}{2}$. Следовательно, сложность МПП равна $\frac{3}{2}a[(n!)^3 + 1]$.

Задача 2.2 Найти минимальную среднюю трудоёмкость в следующей схеме шифрования:

$$\xrightarrow{x} \boxed{DES(x, k_1)} \longrightarrow \boxed{DES(x, k_2)} \longrightarrow \boxed{DES(x, k_3)} \xrightarrow{y}$$

Решение.

В предложенной схеме используется три блока DES с разными ключами. Для одного блока DES $|K| = 2^{56}$, тогда для всей схемы:
$|K| = (2^{56})^3 = 2^{168}$. Окончательно, $E\tau = \frac{|K|+1}{2} = \frac{2^{168}+1}{2} \approx 2^{167}$.

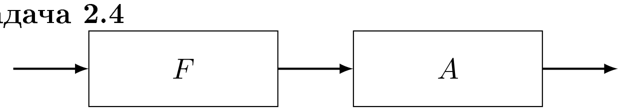
Задача 2.3 В сообщении каждая буква записывается два раза. Для шифрования используется шифр перестановки длины $2n$. Сложность МПП?

Решение.

В данной схеме используется две подстановки, причём для каждой нечётной буквы применяется первая подстановка, а для каждой чётной – вторая: $T(x) = T(x_1, x_2, ..., x_{2l-1}, x_{2l}) = (T_1(x_1), T_2(x_2), ..., T_1(x_{2l-1}), T_2(x_{2l}))$, где l – половина длины сообщения. Тогда длина ключа для каждой из подстановок будет равна n, а мощность пространства различных ключей для всей системы будет равна $|K| = (n!)^2$.

Для проверки одного ключа (T_1, T_2) требуется $2l$ операций подстановки. Тогда сложность МПП равна $2lE\tau = 2l\frac{|K|+1}{2} = l[(2n)! + 1]$.

Задача 2.4



В данной схеме байт ОТ $x = x_1 x_2 ... x_8$ шифруется с помощью функции F следующим образом:

$x'_1 = x_1$;
$x'_2 = x_2 + f_1(x_1)$;
...
$x'_8 = x_8 + f_8(x_1, x_2, ..., x_7)$,

где $f_1, ..., f_7$ – случайные булевы функции, A – невырожденная матрица. Ключом являются F и A. Оценить сложность нахождения ключа с помощью МПП.

Решение.

Определим мощность пространства ключей для F. Так как количество функций, зависящих от n переменных, равно 2^{2^n}, то

$$|K_F| = \prod_{i=1}^{7} 2^{2^i} = 2^{\sum_{i=1}^{7} 2^i} = 2^{\frac{2(2^7-1)}{2-1}} = 2^{2^8-2} = 2^{254}.$$

Теперь рассмотрим матрицу A. Оценим мощность пространства ключей индуктивно по строкам. Для первой строки подходит $2^n - 1$ вариантов (все, кроме нулевой строки). Для следующей строки не подойдёт предыдущий вариант заполнения (иначе будет линейная зависимость, следовательно, вырожденность матрицы) и нулевое заполнение, то есть, $2^n - 2$ вариантов. Теперь, для третьей строки нужно не допустить линейной комбинации первых двух: $\alpha a_1 + \beta a_2 \neq a_3$. Вариантов выбрать коэффициенты α и β – 2^2 (при этом, тут уже считается и нулевой случай). Далее, для четвёртой строки, аналогично, 2^3. Таким образом, получаем формулу:

$$|K| = \prod_{i=0}^{n-1} 2^n - 2^i$$

На матрицу A мы умножаем вектор длины 8 и на выходе тоже получаем вектор длины 8. Следовательно, $n = 8$, $|K_A| \approx 2^{62.21}$
Таким образом,

$$|K| = |K_F| \cdot |K_A| \approx 2^{254} \cdot 2^{62.21} = 2^{316.21}$$

Если бы нам были известны функции $f_1, ..., f_7$, то можно было бы рассчитать количество операций на каждый ключ точно. Но нам они неизвестны, поэтому примем за общее число операций для проверки одного ключа за p. Тогда сложность МПП равна $\frac{|K|+1}{2}p \approx 2^{315.21}p$.

Комментарий к задачам о многочлене Жегалкина.

В полином Жегалкина степени не выше m от функции n переменных входит C_n^k различных мономов степени k. При этом перед каждым из них стоит коэффициент, следовательно, $2^{C_n^k}$ – количество различных вариантов выбрать 0 или 1 перед мономами.

Если полином степени ровно m, то хотя бы при одном мономе этой степени стоит коэффициент 1. Это означает, что число различных вариантов выбрать 0 или 1 перед мономами степени m в таком полиноме равно $2^{C_n^m-1}$.

Используя полином Жегалкина степени не выше m, будем считать, что $n = m$.

Задача 2.5 Ключ шифрования k – многочлен Жегалкина степени 2. Мощность пространства различных ключей? Сложность МПП?
Решение.
$|K| = 2^{C_n^0+C_n^1+C_n^2-1} = 2^{n+\frac{(n-1)n}{2}} = 2^{\frac{n^2+n}{2}}$.
Количество операций $p = C_n^1(1+1)+C_n^2(1+2) = 2n+3\frac{(n-1)n}{2} = \frac{3}{2}n^2+\frac{1}{2}n$
Сложность: $pE\tau = (\frac{3}{2}n^2+\frac{1}{2}n)\frac{2^{\frac{n^2+n}{2}}+1}{2} \approx (3n^2+n)2^{\frac{n^2+n-4}{2}}$

С учётом последнего комментария получим $|K| = 8$, $pE\tau = 31.5$.

Задача 2.6 Ключ шифрования k – многочлен Жегалкина степени не выше m. Мощность пространства различных ключей? Сложность МПП?
Решение.
$|K| = 2^{\sum_{i=0}^m C_n^i}$.
Количество операций $p = \sum_{i=1}^m C_n^i(i+1)$
Сложность: $pE\tau = [\sum_{i=1}^m C_n^i(i+1)]\frac{2^{\sum_{i=0}^m C_n^i}+1}{2} \approx [\sum_{i=1}^m C_n^i(i+1)]2^{\sum_{i=1}^m C_n^i}$

Задача 2.7 Ключ шифрования k – многочлен вида:

$$\sum_{1\le i<j\le n} a_{ij}x_ix_j, a_{ij} \in \{0,1\}.$$

Мощность пространства различных ключей? Сложность МПП?
Решение.

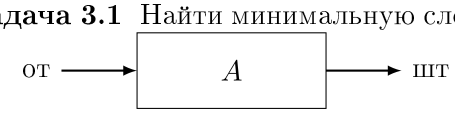
Множество a_{ij} образует верхнетреугольную матрицу без главной диагонали. Следовательно, $|K| = 2^{(n-1)+(n-2)+\ldots+1+0} = 2^{\frac{(n-1)n}{2}}$.

Количество операций $p = \frac{(n-1)n}{2}(1+2) - 1 = \frac{3}{2}n^2 - \frac{3}{2}n - 1$

Сложность: $pE\tau = (\frac{3}{2}n^2 - \frac{3}{2}n - 1)\frac{2^{\frac{(n-1)n}{2}} + 1}{2} \approx (3n^2 - 3n - 2)2^{\frac{n^2-n-4}{2}}$

1.3 Аналитический метод криптоанализа.

Задача 3.1 Найти минимальную сложность нахождения ключа в схеме



Ключом является невырожденная двоичная матрица A размером $n \cdot n$. Сравнить со сложностью МПП.

Решение.

При решении СЛАУ методом Гаусса сложность оценивается в $\frac{n^3}{3}$ операций. Количество операций, необходимое для проверки одного ключа, равно $p = (n+(n-1)) \cdot n = 2n^2 - n$ – такое количество операций сложения и умножения нужно проделать для умножения вектора на квадратную матрицу. Было установлено, что:

$$|K| = \prod_{i=0}^{n-1} 2^n - 2^i = (2^n)^n + ... = O(2^{n^2})$$

Следовательно, сложность МПП:

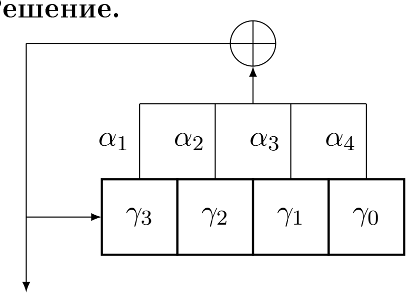
$$E\tau = p\frac{|K|+1}{2} = (2n^2 - n)\frac{2^{n^2} + ...}{2} = O(n^2 \cdot 2^{n^2})$$

Пусть $n = 10$, тогда для МПП потребуется порядка $10^2 \cdot 2^{10^2} \approx 10^{32.10}$ операций, тогда как для аналитического метода получится $\frac{10^3}{3} \approx 3 \cdot 10^2$ операций.

Задача 3.2 Для ЛРП, задаваемой с помощью характеристического многочлена

$F(x) = x^4 \oplus x^2 \oplus x \oplus 1$, построить ЛРС, определить матрицу A, и для выходной (после 4-х тактов работы ЛРС) последовательности $\gamma = (1, 0, 1, 0)$ найти начальное заполнение регистра.

Решение.



Из характеристической функции следует, что $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 0, \alpha_4 = 1$.

Тогда $\gamma_4 = 1\cdot\gamma_0 + 0\cdot\gamma_1 + 1\cdot\gamma_2 + 1\cdot\gamma_3$. Значит, матрица $A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$.

Решим следующее уравнение: $A^4\gamma^T(0) = \gamma^T$.

$$A^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}^2 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \\ 2 & 1 & 3 & 3 \\ 3 & 2 & 4 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\left[\begin{array}{cccc|c} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{array}\right] \sim \left[\begin{array}{cccc|c} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{array}\right] \sim \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

Следовательно, $\gamma(0) = (0, 0, 0, 1)$.

Задача 3.3 Объяснить равенства (4.11) и (4.12).

Решение.

Пусть f имеет следующую структуру:

$$f(\gamma_n, \gamma_{n+1}, ..., \gamma_{n+r-1}) = \gamma_n \oplus g(\gamma_{n+1}, \gamma_{n+1}, ..., \gamma_{n+r-1}).$$

Тогда:

$$f(0, x_2, ..., x_r) \oplus f(1, x_2, ..., x_r) = 0 \oplus g(x_2, ..., x_r) \oplus 1 \oplus g(x_2, ..., x_r) = 1$$

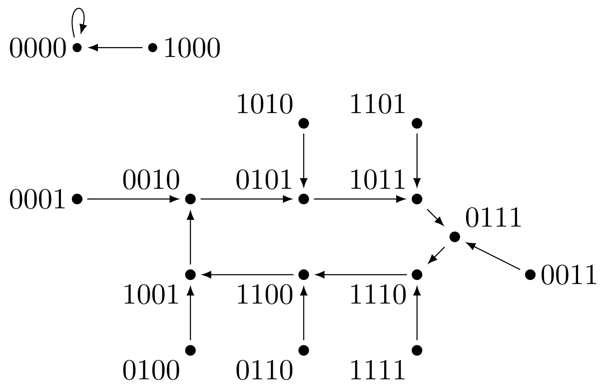
Следовательно, $f(0, x_2, ..., x_r) = 1 \oplus f(1, x_2, ..., x_r)$.

Равенство $f(x_1, x_2, ..., x_r) = x_1 f(1, x_2, ..., x_r) \oplus (1 \oplus x_1) f(0, x_2, ..., x_r)$ проверяется непосредственной подстановкой x_1. В самом деле, при $x_1 = 0$ первое слагаемое обращается в ноль, и имеем $f(0, x_2, ..., x_r) = f(0, x_2, ..., x_r)$. А при $x_1 = 1$ – второе: $f(1, x_2, ..., x_r) = f(1, x_2, ..., x_r)$

Задача 3.4 Построить графы отображений для РС, обратные связи которых задаются функциями от 4 переменных:
$f_1 = x_2 \oplus x_3$, $f_2 = x_1 \oplus x_2 \oplus x_3$, $f_3 = x_3 \oplus x_2 * x_4$, $f_4 = x_1 \oplus x_3 * x_4$, $f_5 = x_1 * x_3 \oplus x_2 * x_4$.
Прокомментировать результаты.

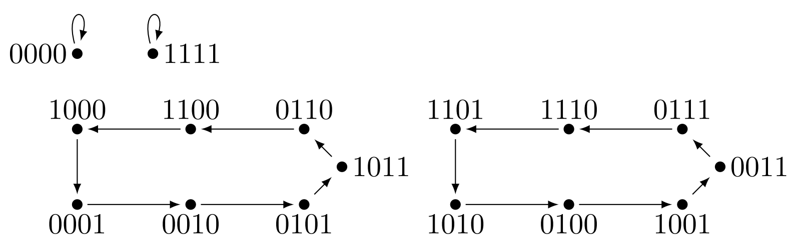
Решение.

ЛРС $F_1 : (x_1, x_2, x_3, x_4) \to (x_2, x_3, x_4, x_2 \oplus x_3)$



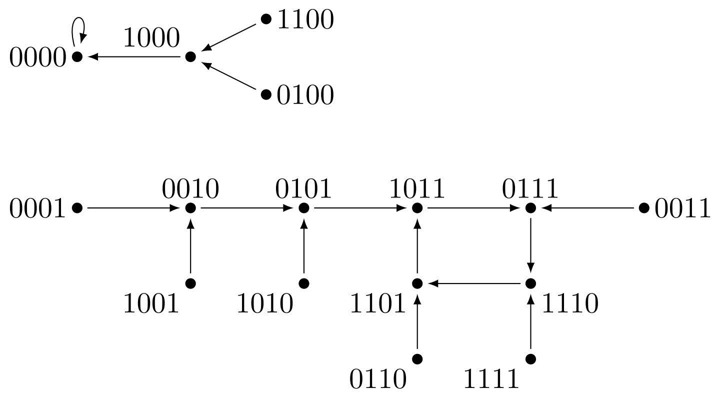
Данный граф имеет структуру "циклы с подходами". Длины циклов: 1, 7. Это отображение не является взаимно однозначным.

ЛРС $F_2 : (x_1, x_2, x_3, x_4) \to (x_2, x_3, x_4, x_1 \oplus x_2 \oplus x_3)$



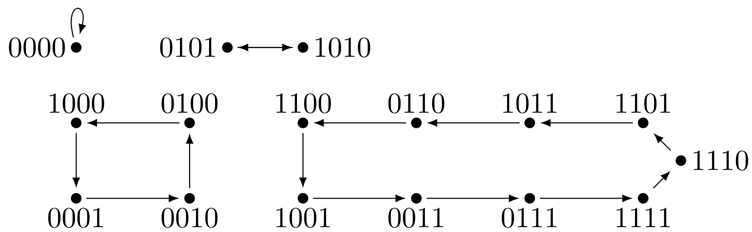
У этого графа полностью цикловая структура. Длины циклов: 1, 1, 7 и 7. Это отображение является взаимно однозначным.

ЛРС $F_3 : (x_1, x_2, x_3, x_4) \to (x_2, x_3, x_4, x_3 \oplus x_2 * x_4)$



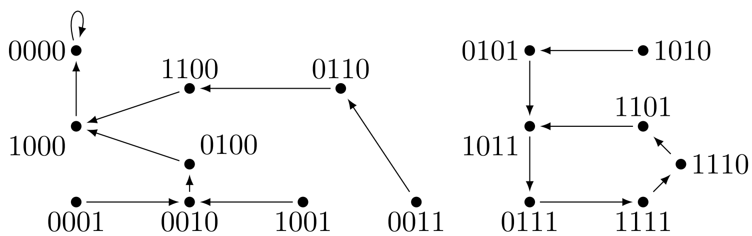
Данный граф имеет структуру "циклы с подходами". Длины циклов: 1, 4. Это отображение не является взаимно однозначным.

ЛРС $F_4 : (x_1, x_2, x_3, x_4) \to (x_2, x_3, x_4, x_1 \oplus x_3 * x_4)$



Граф имеет полностью цикловую структуру. Длины циклов: 1, 2, 4 и 9. Это отображение является взаимно однозначным.

ЛРС $F_5 : (x_1, x_2, x_3, x_4) \to (x_2, x_3, x_4, x_1 * x_3 \oplus x_2 * x_4)$



Данный граф имеет структуру "циклы с подходами". Длины циклов: 1, 5. Это отображение не является взаимно однозначным.

1.4 Перекрытия гаммы. Криптоанализ при неравновероятной гамме.

Задача 4.1 Два текста x и x' на русском языке зашифрованы шифром гамирования по $\mod 30$ с помощью одной и той же гаммы γ. Использована следующая таблица соответствия букв числами (здесь – означает пробел):

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Э	Ю	Я	–
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

Получено два шифротекста y = КЛОВБЛЖЗФ и y' = ВУПЗЕРСВЖ, известна тематика x и x': 'времена года'. Применяя 'протяжку вероятного слова' найти x, x', γ.

Решение.

Переведём векторы y и y' в числа и найдём их разность:
$y - y' = x + \gamma - x' - \gamma = x - x' = (9-2, 10-18, 13-14, 2-7, 1-5, 10-15, 6-16, 7-2, 19-6) = (7, 22, 29, 25, 26, 25, 20, 5, 13) =$ ЗЧ–ЫЭЫЧЕО.

Попробуем подставить в начало x' слово 'ЗИМА–':
$x = (x - x') + x' =$ АСНЕГ $****$

Видно, что получается осмысленное предложение. Посмотрим, какая гамма:

$\gamma = y' - x' =$ ВВВВВ $****$

Предположим, что гамма состоит только из этих букв, продлим и получим окончательный ответ:
$x =$ ЗИМА – ИДЕТ
$x' =$ АСНЕГОПАД
$\gamma =$ ВВВВВВВВВ

Задача 4.2 Пусть в шифре гаммирования по mod 30 используется только 6 знаков гаммы $\{17, 05, 02, 15, 08, 14\}$ (соответствие букв и чисел в таблице):

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Ъ	Э	Ю	Я
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

Получен шифртекст $y =$ ШАССЧАТАИЦОС. Используя "зигзагообразное"чтение дешифровать открытый текст и восстановить гамму.

Решение.

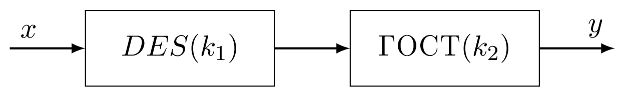
Составим таблицу из возможных результатов гаммирования:

17	Ж	О	Я	Я	Е	**О**	А	О	Ц	Д	Ъ	**Я**
05	У	Ы	М	М	**Т**	Ы	Н	Ы	Г	С	**И**	М
02	Ц	Ю	П	**П**	Х	Ю	Р	Ю	Ж	**Ф**	М	П
15	И	**Р**	Б	Б	З	Р	В	**Р**	Ш	Ж	Ю	Б
08	Р	Ч	**И**	И	П	Ч	К	Ч	**А**	О	Е	И
14	**К**	С	В	В	И	С	**Г**	С	Щ	З	Я	В

Легко видеть, $x =$ КРИПТОГРАФИЯ, $\gamma =$ ПРИВЕТПРИВЕТ.

1.5 Методы "встреча посередине" и "разделяй и властвуй".

Задача 5.1 Найти минимальную среднюю трудоёмкость нахождения ключа в следующей схеме шифрования, длина ключа ГОСТ $= 256$ бит. Сравнить с МПП.



Решение.

Средняя трудоёмкость метода "встречи посередине":

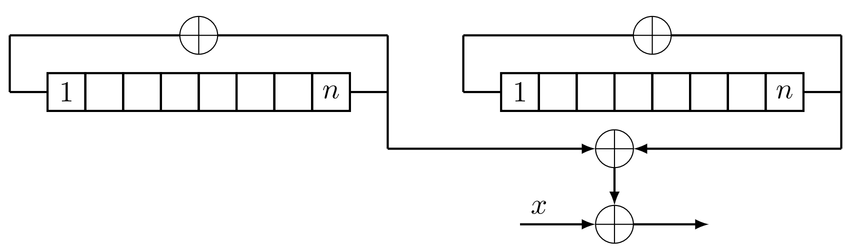
$$(|K_1|+|K_2|)(1+\ln(|K_1|+|K_2|)) = (2^{56}+2^{256})(1+\ln(2^{56}+2^{256})) \approx 10^{79.47}$$

Средняя трудоёмкость полного перебора: $\frac{|K_1||K_2|}{2} = \frac{2^{56}\cdot 2^{256}}{2} = 2^{311} \approx 10^{93.62}$

Метод "встречи посередине" оказывается на 14 порядков эффективнее МПП.

Если предположить, что у нас имеется эффективный критерий, отбраковывающий ключи из K_1, то можно воспользоваться методом "разделяй и властвуй", средняя трудоёмкость которого равна $\frac{|K_1|+|K_1|}{2} = 2^{55}+2^{255} \approx 10^{76.76}$. Этот метод ещё эффективнее в 1000 раз.

Задача 5.2 Ключом являются начальные заполнения ЛРС в алгоритме получения γ для шифра гаммирования. Предполагается, что имеется необходимое количество пар (x, y). Оценить сложность нахождения ключа с помощью метода "встречи посередине" и сравнить с МПП.



Решение.

Для каждого ЛРС оценим мощность множеств ключей: $N = |K_1| = |K_2| = 2^n$. Тогда средняя трудоёмкость метода "встречи посередине":

$$\sqrt{N}\ln N = 2^{\frac{n}{2}}\ln 2^n$$

Средняя трудоёмкость полного перебора:

$$\frac{|K_1||K_2|}{2} = \frac{2^n \cdot 2^n}{2} = 2^{2n-1}$$

При $n = 8$ метод "встречи посередине" эффективнее, чем МПП, в 369 раз, а при $n = 256$ – примерно в 10^{113} раз.

Задача 5.3 В задаче 3.4 найти минимальную среднюю трудоёмкость нахождения ключа и сравнить с МПП. Предполагается, что имеется необходимое количество пар (x, y).

Решение.

Было установлено, что $|K_F| = 2^{254}$ и $|K_A| \approx 2^{62.21}$.

Метод "разделяй и властвуй": $\frac{|K_F|+|K_A|}{2} \approx 10^{76.16}$.

Метод "встречи посередине": $(|K_A|+|K_F|)(1+\ln(|K_A|+|K_F|)) \approx 10^{78.71}$.

МПП: $\frac{|K_F||K_A|}{2} \approx 10^{95.19}$.

Если предположить, что у нас есть эффективный критерий, отбраковывающий ключи из K_F, то минимальная средняя трудоёмкость достигается первым методом, иначе – вторым. Разница по эффективности с МПП от $10^{16.48}$ до $10^{19.03}$ раз.

Часть 2

Контрольные работы

2.1 Шифры перестановки.

Задача 1.1 Раскрыть шифр простой замены:
56 73 31 68 52 88 52 70 16 78 16 90 40 49 16 31 78 56 46 28 88 31 40 88 70
68 52 40 19 56 70 73 88 19 94 00 52 31 49 68 78 88 56 90 73 16 31 49 94 88
88 46 36 49 88 52 88 46 68 74 49 16 78 64 94 88 52 40 68 19 94 16 03 20 49
64 46 88 78 64 13 16 90 40 49 03 16 52 31 78 16 70 88 73 68 78 88 90 40 49
20 94 56 66 46 00 88 49 40 68 78 88 73 31 74 87 88 16 83 16 78 68 94 56 16
16 52 20 90 68 73 56 70 88 73 68 49 64 49 03 87 56 94 16 73 16 31 16 78 56
78 56 31 64 46 00 88 94 56 40 88 40 88 73 88 70 20 16 28 88 73 16 03 94 00
66 94 16 70 88 19 68 90 20 52 16 94 56 82 31 83 16 94 11 56 94 68 52 56 90
40 49 90 94 68 74 90 40 49 03 49 88 31 78 68 73 88 82 70 68 52 31 87 88 28
88 20 28 88 70 94 56 87 68 83 68 87 88 46 74 90 68 94 46 88 74 90 94 56 31
40 68 49 64 73 88 70 56 94 88 03 16 31 49 73 16 90 40 49 68 94 16 40 19 56
19 88 70 94 88 82 88 90 68 46 88 03 16 94 94 88 31 49 56 49 03 87 68 31 94
16 70 68 73 94 56 66 40 88 19 13 20 49 56 73 88 73 31 16 31 49 19 68 13 56
78 31 74 90 68 31 00 40 68 49 64 56 90 90 68 40 88 31 49 88 74 94 94 00 66
87 88 13 52 68 19 88 73 49 03 87 66 88 19 88 13 88 16 11 16 90 40 49 03 49
88 88 94 40 88 94 68 49 20 19 16 03 16 78 88 73 16 87 78 16 28 87 88 52 00
31 78 16 94 94 00 82 56 94 16 31 40 88 31 88 46 94 00 82 90 68 97 56 87 78
56 73 68 49 64 31 74 94 68 03 16 52 78 56 46 88 90 40 49 56 94 68 03 16 52
88 83 94 88 56 20 52 88 52 49 19 88 94 20 49 64 31 74

Решение.

Для более простого воспроизведения описанных действий буду приводить код на языке Python.

Проанализируем частоты монограмм.

```
>>> sorted(zip(*np.unique(cipher, return_counts = True)), key =
    lambda x: x[1], reverse = True)[:10]
[('88', 58), ('16', 37), ('94', 36), ('68', 33), ('49', 31), ('56',
    29), ('31', 26), ('40', 21), ('73', 19), ('90', 19)]
```

Теперь рассмотрим биграммы:

```
>>> bigram = np.array([cipher[i] + ' ' + cipher[i+1] for i in
    range(len(cipher) - 1) ])
>>> sorted(zip(*np.unique(bigram, return_counts = True)), key =
    lambda x: x[1], reverse = True)[:10]
[('40 49', 8), ('88 73', 8), ('90 40', 8), ('40 88', 7), ('94 56',
    7), ('03 16', 6), ('16 31', 6), ('31 49', 6), ('49 03', 6),
    ('49 64', 6)]
```

Наиболее частые моно- и биграммы русского языка:

О	**Е**	**А**	**И**	**Н**	**Т**	**С**	**Р**	**В**	**Л**

СТ	**НО**	**ЕН**	**ТО**	**НА**	**ОВ**	**НИ**	**РА**	**ВО**	**КО**

Предположим, что 88 – это О. В биграммах из текста эта буква встречается дважды: 88 73 и 40 88. В справочной таблице единственное сочетание, в котором О стоит на первом месте – это ОВ. Сравнивая позицию буквы 73 с первой таблицей, можем убедиться, что В действительно подходит.

Допустим также, что 16 – это Е. Поскольку в шифротексте нет явных знаков препинания, предположим, что они записаны в виде ЗПТ и ТЧК. Запятых, скорее всего, больше, чем точек, поэтому рассмотрим триграммы текста и самую частую определим как ЗПТ.

```
>>> trigram = np.array([cipher[i] + ' ' + cipher[i+1] + ' ' +
    cipher[i+2] for i in range(len(cipher) - 2) ])
>>> sorted(zip(*np.unique(trigram, return_counts = True)), key =
    lambda x: x[1], reverse = True)[:5]
[('90 40 49', 8), ('16 90 40', 4), ('68 49 64', 4), ('03 16 52',
    3), ('16 31 49', 3)]
```

Тогда 49 – это Т. Попробуем найти среди биграмм наиболее частую – СТ: единственный вариант, заканчивающийся на 49, – это 31 49 (40 49 уже занято – ПТ). Пусть 31 будет С.

Итак, попробуем подставить:

О	**В**	**Е**	**З**	**П**	**Т**	**С**
88	73	16	90	40	49	31

```
>>> key = {'88': 'О', '73': 'В', '16': 'Е', '90': 'З', '40': 'П',
    '49': 'Т', '31': 'С'}
>>> ' '.join([key[x] if x in key else x for x in cipher])
'56 В С 68 52 О 52 70 Е 78 Е З П Т Е С 78 56 46 28 О С П О 70 68 52
    П 19 56 70 В О 19 94 00 52 С Т 68 78 О 56 З В Е С Т 94 О О 46
    36 Т О 52 О 46 68 74 Т Е 78 64 94 О 52 П 68 19 94 Е 03 20 Т 64
    46 О 78 64 13 Е З П Т 03 Е 52 С 78 Е 70 О В 68 78 О З П Т 20 94
    56 66 46 00 О Т П 68 78 О В С 74 87 О Е 83 Е 78 68 94 56 Е Е 52
    20 З 68 В 56 70 О В 68 Т 64 Т 03 87 56 94 Е В Е С Е 78 56 78 56
    С 64 46 00 О 94 56 П О П О В О 70 20 Е 28 О В Е 03 94 00 66 94
    Е 70 О 19 68 З 20 52 Е 94 56 82 С 83 Е 94 11 56 94 68 52 56 З П
    Т З 94 68 74 З П Т 03 Т О С 78 68 В О 82 70 68 52 С 87 О 28 О
```

```
    20 28 О 70 94 56 87 68 83 68 87 О 46 74 З 68 94 46 О 74 З 94 56
    С П 68 Т 64 В О 70 56 94 О 03 Е С Т В Е З П Т 68 94 Е П 19 56
    19 О 70 94 О 82 О З 68 46 О 03 Е 94 94 О С Т 56 Т 03 87 68 С 94
    Е 70 68 В 94 56 66 П О 19 13 20 Т 56 В О В С Е С Т 19 68 13 56
    78 С 74 З 68 С 00 П 68 Т 64 56 З З 68 П О С Т О 74 94 94 00 66
    87 О 13 52 68 19 О В Т 03 87 66 О 19 О 13 О Е 11 Е З П Т 03 Т О
    О 94 П О 94 68 Т 20 19 Е 03 Е 78 О В Е 87 78 Е 28 87 О 52 00 С
    78 Е 94 94 00 82 56 94 Е С П О С О 46 94 00 82 З 68 97 56 87 78
    56 В 68 Т 64 С 74 94 68 03 Е 52 78 56 46 О З П Т 56 94 68 03 Е
    52 О 83 94 О 56 20 52 О 52 Т 19 О 94 20 Т 64 С 74'
```

Обратим внимание на 'ЗПТЕС 78 56', 'ПОПОВО 70 20', 'ПОСТО 74 94 94 *', 'СПОСО 46'. Всё это похоже на ', если', 'по поводу', 'постоянн*' и 'способ'. Попробуем добавить в ключ следующие замены:

Л	И	Д	У	Я	Н	Б
78	56	70	20	74	94	46

```
>>> key.update(**{'78': 'Л', '56': 'И','70': 'Д','20': 'У','74':
    'Я','94': 'Н','46': 'Б'})
>>> ' '.join([key[x] if x in key else x for x in cipher])
'И В С 68 52 О 52 Д Е Л Е З П Т Е С Л И Б 28 О С П О Д 68 52 П 19 И
    Д В О 19 Н 00 52 С Т 68 Л О И З В Е С Т Н О О Б 36 Т О 52 О Б
    68 Я Т Е Л 64 Н О 52 П 68 19 Н Е 03 У Т 64 Б О Л 64 13 Е З П Т
    03 Е 52 С Л Е Д О В 68 Л О З П Т У Н И 66 Б 00 О Т П 68 Л О В С
    Я 87 О Е 83 Е Л 68 Н И Е Е 52 У З 68 В И Д О В 68 Т 64 Т 03 87
    И Н Е В Е С Е Л И Л И С 64 Б 00 О Н И П О П О В О Д У Е 28 О В
    Е 03 Н 00 66 Н Е Д О 19 68 З У 52 Е Н И 82 С 83 Е Н 11 И Н 68
    52 И З П Т З Н 68 Я З П Т 03 Т О С Л 68 В О 82 Д 68 52 С 87 О
    28 О У 28 О Д Н И 87 68 83 68 87 О Б Я З 68 Н Б О Я З Н И С П
    68 Т 64 В О Д И Н О 03 Е С Т В Е З П Т 68 Н Е П 19 И 19 О Д Н О
    82 О З 68 Б О 03 Е Н Н О С Т И Т 03 87 68 С Н Е Д 68 В Н И 66 П
    О 19 13 У Т И В О В С Е С Т 19 68 13 И Л С Я З 68 С 00 П 68 Т
    64 И З З 68 П О С Т О Я Н Н 00 66 87 О 13 52 68 19 О В Т 03 87
    66 О 19 О 13 О Е 11 Е З П Т 03 Т О О Н П О Н 68 Т У 19 Е 03 Е Л
    О В Е 87 Л Е 28 87 О 52 00 С Л Е Н Н 00 82 И Н Е С П О С О Б Н
    00 82 З 68 97 И 87 Л И В 68 Т 64 С Я Н 68 03 Е 52 Л И Б О З П Т
    И Н 68 03 Е 52 О 83 Н О И У 52 О 52 Т 19 О Н У Т 64 С Я'
```

Видно, что 'С Т 68 Л О И З В Е С Т Н О О Б 36 Т О 52 О Б 68 Я Т Е Л 64 Н О 52 П 68 19 Н Е' похоже на 'стало известно об этом обаятельном парне', а 'В Е С Е Л И Л И С 64 Б 00 О Н И П О П О В О Д У Е 28 О' – на 'веселились бы они по поводу его', 'В О Д И Н О 03 Е С Т В Е' – 'в одиночестве'

А	Э	М	Ь	Р	Ы	Г	Ч
68	36	52	64	19	00	28	03

```
>>> key.update(**{'68': 'А', '36': 'Э','52': 'М','64': 'Ь','19':
    'Р','00': 'Ы','28': 'Г', '03': 'Ч'})
>>> ' '.join([key[x] if x in key else x for x in cipher])
'И В С А М О М Д Е Л Е З П Т Е С Л И Б Г О С П О Д А М П Р И Д В О
    Р Н Ы М С Т А Л О И З В Е С Т Н О О Б Э Т О М О Б А Я Т Е Л Ь Н
    О М П А Р Н Е Ч У Т Ь Б О Л Ь 13 Е З П Т Ч Е М С Л Е Д О В А Л
    О З П Т У Н И 66 Б Ы О Т П А Л О В С Я 87 О Е 83 Е Л А Н И Е Е
    М У З А В И Д О В А Т Ь Т Ч 87 И Н Е В Е С Е Л И Л И С Ь Б Ы О
    Н И П О П О В О Д У Е Г О В Е Ч Н Ы 66 Н Е Д О Р А З У М Е Н И
    82 С 83 Е Н 11 И Н А М И З П Т З Н А Я З П Т Ч Т О С Л А В О 82
    Д А М С 87 О Г О У Г О Д Н И 87 А 83 А 87 О Б Я З А Н Б О Я З Н
    И С П А Т Ь В О Д И Н О Ч Е С Т В Е З П Т А Н Е П Р И Р О Д Н О
    82 О З А Б О Ч Е Н Н О С Т И Т Ч 87 А С Н Е Д А В Н И 66 П О Р
    13 У Т И В О В С Е С Т Р А 13 И Л С Я З А С Ы П А Т Ь И З З А П
    О С Т О Я Н Н Ы 66 87 О 13 М А Р О В Т Ч 87 66 О Р О 13 О Е 11
    Е З П Т Ч Т О О Н П О Н А Т У Р Е Ч Е Л О В Е 87 Л Е Г 87 О М Ы
    С Л Е Н Н Ы 82 И Н Е С П О С О Б Н Ы 82 З А 97 И 87 Л И В А Т Ь
    С Я Н А Ч Е М Л И Б О З П Т И Н А Ч Е М О 83 Н О И У М О М Т Р
    О Н У Т Ь С Я'
```

'ЧУТЬБОЛЬ 13 ЕЗПТ' – 'чуть больше,', 'УНИ 66 БЫ' – 'у них бы', 'В С Я 87 О Е 83 Е Л А Н И Е' – 'всякое желание', 'Н Е Д О Р А З У М Е Н И 82 С 83 Е Н 11 И Н А М И' – 'недоразумений с женщинами', 'З А 97 И 87 Л И В А Т Ь С Я' – 'зацикливаться'.

Ш	Х	К	Ж	Й	Щ	Ц
13	66	87	83	82	11	97

```
>>> key.update(**{'13': 'Ш', '66': 'Х','87': 'К','83': 'Ж','82':
    'Й','11': 'Щ','97': 'Ц'})
>>> ' '.join([key[x] if x in key else x for x in cipher])
'И В С А М О М Д Е Л Е З П Т Е С Л И Б Г О С П О Д А М П Р И Д В О
    Р Н Ы М С Т А Л О И З В Е С Т Н О О Б Э Т О М О Б А Я Т Е Л Ь Н
    О М П А Р Н Е Ч У Т Ь Б О Л Ь Ш Е З П Т Ч Е М С Л Е Д О В А Л О
    З П Т У Н И Х Б Ы О Т П А Л О В С Я К О Е Ж Е Л А Н И Е Е М У З
    А В И Д О В А Т Ь Т Ч К И Н Е В Е С Е Л И Л И С Ь Б Ы О Н И П О
    П О В О Д У Е Г О В Е Ч Н Ы Х Н Е Д О Р А З У М Е Н И Й С Ж Е Н
    Щ И Н А М И З П Т З Н А Я З П Т Ч Т О С Л А В О Й Д А М С К О Г
    О У Г О Д Н И К А Ж А К О Б Я З А Н Б О Я З Н И С П А Т Ь В О Д
    И Н О Ч Е С Т В Е З П Т А Н Е П Р И Р О Д Н О Й О З А Б О Ч Е Н
    Н О С Т И Т Ч К А С Н Е Д А В Н И Х П О Р Ш У Т И В О В С Е С Т
    Р А Ш И Л С Я З А С Ы П А Т Ь И З З А П О С Т О Я Н Н Ы Х К О Ш
```

```
    М А Р О В Т Ч К Х О Р О Ш О Е Щ Е З П Т Ч Т О О Н П О Н А Т У Р
    Е Ч Е Л О В Е К Л Е Г К О М Ы С Л Е Н Н Ы Й И Н Е С П О С О Б Н
    Ы Й З А Ц И К Л И В А Т Ь С Я Н А Ч Е М Л И Б О З П Т И Н А Ч Е
    М О Ж Н О И У М О М Т Р О Н У Т Ь С Я'
>>> key
{'88': 'О', '73': 'В', '16': 'Е', '90': 'З', '40': 'П', '49': 'Т',
    '31': 'С', '78': 'Л', '56': 'И', '70': 'Д', '20': 'У', '74':
    'Я', '94': 'Н', '46': 'Б', '68': 'А', '36': 'Э', '52': 'М',
    '64': 'Ь', '19': 'Р', '00': 'Ы', '28': 'Г', '03': 'Ч', '13':
    'Ш', '66': 'Х', '87': 'К', '83': 'Ж', '82': 'Й', '11': 'Щ',
    '97': 'Ц'}
```

Задача 1.2 Раскрыть шифр вертикальной перестановки:

АЕЧСЕ ЛЫЯИЛ ОПЗИЕ СТЫБД ТТДРД ОВИГР ЙВКАЛ МАШЛУ ПЗЖТЯ РОСЗГ ЕНОПЫ ИОМЕО ОЯТТХ ОДАЛР УИВИО ООННИ ОВЫЫБ ИАОРС ОТГАБ СОЕЧД ВУНЛУ НИМОЕ ШШАВН ЕАВМЙ

Решение.

Длина текста 120 букв. Наиболее целесообразно было бы использовать ключ длины 10 или 12 (близкой к $\sqrt{120}$). Проверим различные длины ключей на основе известного соотношения гласных к согласным: 44% к 56%.

```
>>> def get_mse(text, n):
...     vn = lambda row: sum([x in list('АЕЁИОУЫЭЮЯ') for x in row])
...     table = np.array(list(text)).reshape((n, len(text) // n)).T
...     ratio =  np.array([vn(row) / len(row) for row in table])
...   return sum((ratio - 0.44) ** 2) / (len(text) // n)
...
>>> mse = [(round(get_mse(text, i), 5), i) for i in [6, 8, 10, 12,
    15]]
>>> sorted(mse, key = lambda x: x[0])
[(0.00216, 15), (0.02229, 12), (0.02577, 10), (0.03514, 8),
    (0.03966, 6)]
```

Видим, что наименьшая среднеквадратичная ошибка достигается при ключе длины 15.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
А	И	Т	Д	К	П	З	О	Х	В	О	Р	О	У	А
Е	Л	Ы	О	А	З	Г	М	О	И	В	С	Е	Н	В
Ч	О	Б	В	Л	Ж	Е	Е	Д	О	Ы	О	Ч	И	Н
С	П	Д	И	М	Т	Н	О	А	О	Ы	Т	Д	М	Е
Е	З	Т	Г	А	Я	О	О	Л	О	Б	Г	В	О	А
Л	И	Т	Р	Ш	Р	П	Я	Р	Н	И	А	У	Е	В
Ы	Е	Д	Й	Л	О	Ы	Т	У	Н	А	Б	Н	Ш	М
Я	С	Р	В	У	С	И	Т	И	И	О	С	Л	Ш	Й

Обратим внимание на столбцы, в которых есть буква 'Ы' – с ними будет проще всего найти невстречающиеся биграммы. Например, столбец 11 сочетается только с 3 и 5 столбцами. Так как, например, 'ДЫМ' встретится чаще, чем 'МЫД', поставим столбцы в порядке 3 - 11 - 5. Во второй строке получаем триграмму 'ЫВА', после которой может быть 'Н', 'Т', 'Е', 'Ю', 'Л', 'Я'. Отметим кандидатами 1, 2, 13 и 14 столбец. В последней строке получается 'РОУЯ', если выбрать первый столбец – отбраковываем, при 14-м столбце в 5-й строке получится 'ТБАО' – отбраковываем. На третьей строке скорее будет 'БЫЛО', чем 'БЫЛЧ', поэтому остановимся на варианте 3 - 11 - 5 - 2.

1	6	3	11	5	2	7	8	9	10	4	12	13	14	15
А	П	**Т**	**О**	**К**	**И**	З	О	Х	В	Д	Р	О	У	А
Е	З	**Ы**	**В**	**А**	**Л**	Г	М	О	И	О	С	Е	Н	В
Ч	Ж	**Б**	**Ы**	**Л**	**О**	Е	Е	Д	О	В	О	Ч	И	Н
С	Т	**Д**	**Ы**	**М**	**П**	Н	О	А	О	И	Т	Д	М	Е
Е	Я	**Т**	**Б**	**А**	**З**	О	О	Л	О	Г	Г	В	О	А
Л	Р	**Т**	**И**	**Ш**	**И**	П	Я	Р	Н	Р	А	У	Е	В
Ы	О	**Д**	**А**	**Л**	**Е**	Ы	Т	У	Н	Й	Б	Н	Ш	М
Я	С	**Р**	**О**	**У**	**С**	И	Т	И	И	В	С	Л	Ш	Й

В первой строке видно слово 'ВОЗДУХ', 10 - (8, 13) - 7 - 4 - 14 - 9. На третьей строке оказывается 'ОЕЕ', если выбрать 8-й столбец, и 'ОЧЕ', если выбрать 13-й. Установим столбцы по второму варианту.

1	6	3	11	5	2	12	8	15	10	13	7	4	14	9
А	П	**Т**	**О**	**К**	**И**	Р	О	А	**В**	**О**	**З**	**Д**	**У**	**Х**
Е	З	**Ы**	**В**	**А**	**Л**	С	М	В	**И**	**Е**	**Г**	**О**	**Н**	**О**
Ч	Ж	**Б**	**Ы**	**Л**	**О**	О	Е	Н	**О**	**Ч**	**Е**	**В**	**И**	**Д**
С	Т	**Д**	**Ы**	**М**	**П**	Т	О	Е	**О**	**Д**	**Н**	**И**	**М**	**А**
Е	Я	**Т**	**Б**	**А**	**З**	Г	О	А	**О**	**В**	**О**	**Г**	**О**	**Л**
Л	Р	**Т**	**И**	**Ш**	**И**	А	Я	В	**Н**	**У**	**П**	**Р**	**Е**	**Р**
Ы	О	**Д**	**А**	**Л**	**Е**	Б	Т	М	**Н**	**Н**	**Ы**	**Й**	**Ш**	**У**
Я	С	**Р**	**О**	**У**	**С**	С	Т	Й	**И**	**Л**	**И**	**В**	**Ш**	**И**

Видно, что эти два блока можно объединить. Кроме того, можно заметить слова 'ПОТОКИ' и 'ОЧЕВИДНО': 9 - 15 - 12, 6 - 8 - 3. Остаётся последний столбец, для которого становится ясно, что он должен находиться в конце таблицы.

Окончательный ответ:

П	**О**	**Т**	**О**	**К**	**И**	**В**	**О**	**З**	**Д**	**У**	**Х**	**А**	**Р**	**А**
З	**М**	**Ы**	**В**	**А**	**Л**	**И**	**Е**	**Г**	**О**	**Н**	**О**	**В**	**С**	**Е**
Ж	**Е**	**Б**	**Ы**	**Л**	**О**	**О**	**Ч**	**Е**	**В**	**И**	**Д**	**Н**	**О**	**Ч**
Т	**О**	**Д**	**Ы**	**М**	**П**	**О**	**Д**	**Н**	**И**	**М**	**А**	**Е**	**Т**	**С**
Я	**О**	**Т**	**Б**	**А**	**З**	**О**	**В**	**О**	**Г**	**О**	**Л**	**А**	**Г**	**Е**
Р	**Я**	**Т**	**И**	**Ш**	**И**	**Н**	**У**	**П**	**Р**	**Е**	**Р**	**В**	**А**	**Л**
О	**Т**	**Д**	**А**	**Л**	**Е**	**Н**	**Н**	**Ы**	**Й**	**Ш**	**У**	**М**	**Б**	**Ы**
С	**Т**	**Р**	**О**	**У**	**С**	**И**	**Л**	**И**	**В**	**Ш**	**И**	**Й**	**С**	**Я**

2.2 Корреляционный анализ.

Задача 2.1 Дано:

1) Схема, в которой ЛРС длины 5 задаётся характеристической функцией $F(x) = 1 + x^2 + x^5$,

2) Функция усложнения $f(x_1, x_2, x_3, x_4, x_5) = x_4 x_5$ (операции сложения и умножения в $GF(2)$),

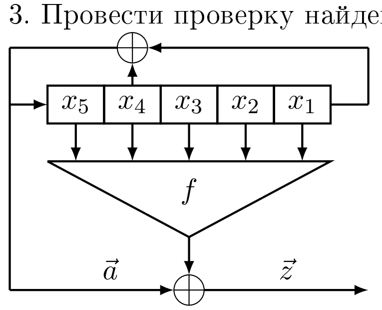
3) $z = 0101011011110011000101111001000$.

Задание:

1. Провести полный расчет корреляционного метода, включая нахождение требуемого числа линейных соотношений $m, E_0(p^*), E_1(p^*)$.

2. Применяя корреляционный метод, найти неизвестное начальное заполнение ЛРС $(x_1, x_2, x_3, x_4, x_5)$.

3. Провести проверку найденного решения.



Решение.

Проведём расчёт метода. Длина вектора $\vec{z}$: $N = 31$. Длина ЛРС: $r = 5$. Количество слагаемых в линейной рекурренте: $t = 2$. Вероятность того, что функция усложнения будет равна нулю: $P(f = 0) = \frac{3}{4}$. Поскольку эта вероятность $\approx 75\%$, можно эффективно применить корреляционную атаку Мейера-Штаффельбаха (алгоритм A) [1]. Необходимое количество уравнений в системе: $m \approx (t+1)\left[\log_2 \frac{N}{r}\right] = 6$.

Получим линейную рекурренту генератора:

$$a_{n+r} = a_n + a_{n+3}$$

Заменой $n + r$ и $n + 3$ на n дополнительно получим 2 уравнения:

$$\begin{cases} a_n = a_{n+r} + a_{n+3}, \\ a_n = a_{n-r} + a_{n-r+3}, \\ a_n = a_{n-3} + a_{n+r-3}. \end{cases}$$

[1] Meier, W., Staffelbach, O.: Fast correlation attacks on certain stream ciphers. J. Cryptol. 1(3), 159–176 (1989)

Подставим a_{n+r} из первого уравнения вместо a_{n-r} во второе уравнение и a_{n+3} вместо a_{n-3} в третье.

$$\begin{cases} a_n = a_{n+r} + a_{n+3},\ (1) \\ a_n = a_{n-r} + a_{n-r+3},\ (2) \\ a_n = a_{n-3} + a_{n+r-3},\ (3) \\ a_n = a_{n-2r} + a_{n-2r+3} + a_{n-r+3},\ (1+2) \\ a_n = a_{n-6} + a_{n+r-6} + a_{n+r-3}.\ (1+3) \end{cases}$$

Теперь подставим второе уравнение в пятое.

$$\begin{cases} a_n = a_{n+r} + a_{n+3},\ (1) \\ a_n = a_{n-r} + a_{n-r+3},\ (2) \\ a_n = a_{n-3} + a_{n+r-3},\ (3) \\ a_n = a_{n-2r} + a_{n-2r+3} + a_{n-r+3},\ (1+2) \\ a_n = a_{n-6} + a_{n+r-6} + a_{n+r-3}.\ (1+3) \\ a_n = a_{n-r-6} + a_{n-r-3} + a_{n+r-6} + a_{n+r-3}.\ (1+2+3) \end{cases}$$

Таким образом, мы получили систему из $m = 6$ уравнений. Теперь подставим r, вместо членов последовательности a подставим члены последовательности z, опустим индексы n и получим систему линейных форм:

$$\begin{cases} z + z_5 + z_3 = L_1, \\ z + z_{-5} + z_{-2} = L_2, \\ z + z_{-3} + z_2 = L_3, \\ z + z_{-10} + z_{-7} + z_{-2} = L_4, \\ z + z_{-6} + z_{-1} + z_2 = L_5, \\ z + z_{-11} + z_{-8} + z_{-1} + z_2 = L_6. \end{cases}$$

Каждый z_i представляет собой $a_i \oplus \gamma_i$, где γ_i – это н.о.р.с.в. с $P(\gamma = 0) = P(f = 0) = \frac{3}{4}$. Пусть b_{ij} – это слагаемые правой стороны уравнений системы с a_i, а y_{ij} – слагаемые левой стороны уравнений системы с z_i, не содержащие z. Тогда уравнения первой системы принимают вид $a + \sum_{j=0}^{t} b_{ij} = 0$, а второй – $z + \sum_{j=0}^{t} y_{ij} = L_i$. Заметим, что в таком случае $P(z_i = a_i) = P(y_{ij} = b_{ij}) = \frac{3}{4} = p$.

Пусть вероятность $s = s(t,p) = P(y_i = b_i)$ не зависит от i. По формуле полной вероятности получим рекуррентное соотношение:

$$\begin{cases} s(t,p) = p \cdot s(t-1,p) + (1-p)(1-s(t-1,p)), \\ s(1,p) = p. \end{cases}$$

Поскольку $t = 2$, то $s = s(2, \frac{3}{4}) = \frac{3}{4} \cdot \frac{3}{4} + (1-\frac{3}{4})(1-\frac{3}{4}) = \frac{5}{8}$. Определим апостериорную вероятность того, что $z = a$ при условии события B_k: k из m линейных форм L_i равны нулю.

$$P(z = a|B_k) = \frac{\binom{m}{k} p s^k (1-s)^{m-k}}{\binom{m}{k} p s^k (1-s)^{m-k} + \binom{m}{k}(1-p) s^{m-k} (1-s)^k} = p^*$$

Найдём матожидания этой величины в двух разных случаях: $z = a$ и $z \neq a$:

$$E_0(p^*) = E(p^*|z = a) =$$
$$= \sum_{k=0}^{m} \binom{m}{k} \frac{p s^k (1-s)^{m-k}}{p s^k (1-s)^{m-k} + (1-p) s^{m-k} (1-s)^k} s^k (1-s)^{m-k} =$$
$$= \sum_{k=0}^{6} \binom{6}{k} \frac{\frac{3}{4} \cdot (\frac{5}{8})^k (\frac{3}{8})^{6-k}}{\frac{3}{4} \cdot (\frac{5}{8})^k (\frac{3}{8})^{6-k} + \frac{1}{4} \cdot (\frac{5}{8})^{6-k} (\frac{3}{8})^k} \left(\frac{5}{8}\right)^k \left(\frac{3}{8}\right)^{6-k} \approx 0.81$$

$$E_1(p^*) = E(p^*|z \neq a) =$$
$$= \sum_{k=0}^{m} \binom{m}{k} \frac{p s^k (1-s)^{m-k}}{p s^k (1-s)^{m-k} + (1-p) s^{m-k} (1-s)^k} s^{m-k} (1-s)^k =$$
$$= \sum_{k=0}^{6} \binom{6}{k} \frac{\frac{3}{4} \cdot (\frac{5}{8})^k (\frac{3}{8})^{6-k}}{\frac{3}{4} \cdot (\frac{5}{8})^k (\frac{3}{8})^{6-k} + \frac{1}{4} \cdot (\frac{5}{8})^{6-k} (\frac{3}{8})^k} \left(\frac{5}{8}\right)^{6-k} \left(\frac{3}{8}\right)^k \approx 0.56$$

Составим таблицу в соответствии с последней системой. Записываем последовательность z в том же порядке, в котором она была задана в условии. Далее добавляем столбцы z_i, участвующие в СЛАУ в качестве слагаемых: это будет та же последовательность, но со сдвигом i. i положительное – сдвиг "вверх", i отрицательное – сдвиг "вниз". Потом заполняем L_i, исходя из их равенств, уже зная все слагаемые в них.

N	z	z_5	z_3	z_{-5}	z_{-2}	z_{-3}	z_2	z_{-10}	z_{-7}	z_{-6}	z_{-1}	z_{-11}	z_{-8}	L_1	L_2	L_3	L_4	L_5	L_6
1	0	1	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1
2	1	1	0	1	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0
3	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	0	1	0	0
4	1	1	1	0	1	0	1	1	1	0	0	1	0	1	0	0	0	0	1
5	0	1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	0	0	0
6	1	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1
7	1	1	1	1	0	1	1	1	0	0	1	0	0	1	0	1	0	1	1
8	0	0	1	0	1	0	1	0	0	1	1	1	0	1	1	1	1	1	1
9	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1	0	0
10	1	1	0	0	0	1	1	0	0	1	1	0	1	0	1	1	1	0	0
11	1	1	0	1	1	0	0	0	1	0	1	0	0	0	1	1	1	0	0
12	1	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	1	1
13	0	0	1	0	1	1	1	0	1	1	1	1	0	1	1	0	0	1	1
14	0	0	0	1	1	1	1	1	1	0	0	0	1	0	0	0	1	1	0
15	1	1	0	1	0	1	0	0	0	1	0	1	1	0	0	0	1	0	1
16	1	0	0	1	0	0	0	1	1	1	1	0	0	1	0	1	1	1	0
17	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	1	0	1
18	0	1	0	0	1	1	1	0	1	1	0	1	1	1	1	0	0	0	1
19	0	1	1	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	1
20	1	1	1	1	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0
21	0	0	1	1	0	0	1	1	0	1	1	1	0	1	1	1	1	1	1
22	1	0	1	0	1	0	1	1	1	1	0	1	0	0	0	0	0	1	1
23	1	1	0	0	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1
24	1	0	0	0	1	0	0	0	0	0	1	0	1	1	0	1	0	0	1
25	1	0	1	1	1	1	0	1	0	0	1	0	0	0	1	0	1	0	0
26	0	0	0	0	1	1	1	1	0	1	1	1	0	0	1	0	0	1	1
27	0	0	0	1	1	1	0	0	1	0	0	1	0	0	0	1	0	0	1
28	1	1	0	1	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0
29	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	0	1
30	0	1	1	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1	1
31	0	0	0	0	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1

Выберем r строк, в которых L_i принимают наибольшее количество нулей. Это строки 2, 1, 5, 20, 28. Выразим a с соответствующими номерами через начальное заполнение регистров.

$$\begin{cases} a_2 = x_2 + x_5 = 1, \\ a_1 = x_1 + x_4 = 0, \\ a_5 = x_1 + x_3 + x_4 + x_5 = 0, \\ a_{20} = x_3 + x_4 + x_5 = 1, \\ a_{28} = x_2 = 1. \end{cases}$$

Решив систему уравнений, получим $\vec{x} = (1, 1, 0, 1, 0)$. Выполним проверку:

```
>>> def check_solution(x):
...   true_z = [int(c) for c in '0101011011110011000101111001000']
...   z = []
...   for i in range(len(true_z)):
...      gamma = x[3] * x[4]
...      a = (x[0] + x[3]) % 2
...      z += [(gamma + a) % 2]
...      x = x[1:] + [a]
...   return z == true_z
...
>>> x = [1, 1, 0, 1, 0]
>>> check_solution(x)
False
```

К сожалению, не повезло. Значит, надо выбрать другие строки. Попробуем взять 2, 1, 5, 3, 28. Тогда четвёртое уравнение в системе изменится на $a_3 = x_1 + x_3 + x_4 = 0$. Новая система будет иметь два решения: $\vec{x} = (0, 1, 0, 0, 0)$ и $\vec{x} = (1, 1, 0, 1, 0)$. Второе из них уже было проверено выше, проверим первое:

```
>>> x = [0, 1, 0, 0, 0]
>>> check_solution(x)
True
```

Победа.

Ответ: $m = 6$, $E_0(p^*) \approx 0.81$, $E_1(p^*) \approx 0.56$, $\vec{x} = (0, 1, 0, 0, 0)$.

2.3 Дифференциальный криптоанализ.

Задача 3.1 N – количество слов длины l в алфавите A, n – количество пар вариантов сообщений M и M', p – вероятность успешной атаки.

Доказать теорему: Пусть $N, n \to \infty$, но $\frac{n^2}{N} \to t > 0$, тогда:

$$p = (1 - e^{-t})(1 + o(1)).$$

Решение.

Известно, что: $1 - p = \frac{[(N-n)!]^2}{N!(N-2n)!}$. Следовательно:

$$p = 1 - \frac{[(N-n)!]^2}{N!(N-2n)!} =$$

$$= 1 - \frac{\left[\left(\frac{N-n}{e}\right)^{N-n}\sqrt{2\pi(N-n)}\left(1 + \frac{1}{12(N-n)} + O\left(\frac{1}{(N-n)^2}\right)\right)\right]^2}{\left(\frac{N}{e}\right)^N\sqrt{2\pi N}\left(1 + \frac{1}{12N} + O\left(\frac{1}{N^2}\right)\right)\left(\frac{N-2n}{e}\right)^{N-2n}\sqrt{2\pi(N-2n)}\left(1 + \frac{1}{12(N-2n)} + O\left(\frac{1}{(N-2n)^2}\right)\right)} =$$

$$= 1 - \frac{\left(\frac{1}{e}\right)^{2N-2n}}{\left(\frac{1}{e}\right)^{N+N-2n}} \cdot \frac{(N-n)^{2N-2n}}{N^N (N-2n)^{N-2n}} \cdot \sqrt{\frac{(N-n)^2}{N(N-2n)}} \cdot \frac{\left(1 + \frac{1}{12(N-n)} + O\left(\frac{1}{(N-n)^2}\right)\right)^2}{\left(1 + \frac{1}{12N} + O\left(\frac{1}{N^2}\right)\right)\left(1 + \frac{1}{12(N-2n)} + O\left(\frac{1}{(N-2n)^2}\right)\right)} =$$

$$= \left\{ \frac{(N-n)^{2N-2n}}{N^N (N-2n)^{N-2n}} = \left(\frac{N^2 - 2nN + n^2}{N^2 - 2nN}\right)^N \cdot \left(\frac{N^2 - 4nN + 4n^2}{N^2 - 2nN + n^2}\right)^n = \left(1 + \frac{n^2}{N} \cdot \frac{1}{N-2n}\right)^N \cdot \right.$$

$$\cdot \left(1 - \frac{2nN - 3n^2}{N^2 - 2nN + n^2}\right)^n = \left(1 + \frac{n^2}{N} \cdot \frac{1}{N-2n}\right)^N \cdot \left(1 - \frac{2}{\frac{N}{n^2}n - 1} + \left(\frac{1}{\frac{N}{n^2}n - 1}\right)^2\right)^n =$$

$$= \exp\left\{N \ln\left(1 + \frac{n^2}{N} \cdot \frac{1}{N-2n}\right) + n \ln\left(1 - \frac{2}{\frac{N}{n^2}n - 1} + \left(\frac{1}{\frac{N}{n^2}n - 1}\right)^2\right)\right\} =$$

$$= \left\{\frac{n^2}{N} \cdot \frac{1}{N-2n} \sim \frac{t}{N - 2\sqrt{tN}} \xrightarrow{t \ll N} 0, \quad \frac{1}{\frac{N}{n^2}n - 1} \sim \frac{t}{n-t} \xrightarrow{t \ll n} 0\right\} = \{\ln(1+\alpha) = \alpha + o(\alpha)\} =$$

$$= \exp\left\{N\left(\frac{n^2}{N} \cdot \frac{1}{N-2n} + o\left(\frac{1}{N - 2\sqrt{tN}}\right)\right) + n\left(-\frac{2}{\frac{N}{n^2}n - 1} + \left(\frac{1}{\frac{N}{n^2}n - 1}\right)^2 + o\left(\frac{1}{n-t}\right)\right)\right\} =$$

$$= \exp\left\{\frac{n^2}{N} \cdot \frac{1}{1 - 2\frac{n^2}{N}\frac{1}{n}} - \frac{n^2}{N} \cdot \frac{2}{1 - \frac{n^2}{N}\frac{1}{n}} - \left(\frac{n^2}{N}\right)^2 \frac{1}{n - 2\frac{n^2}{N} + (\frac{n^2}{N})^2\frac{1}{n}} + o\left(\frac{N}{(N - 2\sqrt{tN})^3}\right) + o\left(\frac{n}{(n-t)^3}\right)\right\} =$$

$$= \exp\{\frac{n^2}{N}\cdot\frac{3\frac{n^2}{N}\frac{1}{n}-1}{1-3\frac{n^2}{N}\frac{1}{n}+2(\frac{n^2}{N})^2\frac{1}{n^2}}+o\,(1)\} = \exp\{-\frac{n^2}{N}\cdot\frac{1+o(1)}{1+o(1)}+o\,(1)\}\Big\} =$$

$$= 1-\exp\{-\frac{n^2}{N}\cdot\frac{1+o(1)}{1+o(1)}+o\,(1)\}\cdot$$

$$\cdot\sqrt{\frac{(N-n)^2}{N(N-2n)}}\cdot\frac{(1+\frac{1}{12(N-n)}+O\left(\frac{1}{(N-n)^2}\right))^2}{(1+\frac{1}{12N}+O\left(\frac{1}{N^2}\right))(1+\frac{1}{12(N-2n)}+O\left(\frac{1}{(N-2n)^2}\right))} =$$

$$= \Big\{\sqrt{\frac{(N-n)^2}{N(N-2n)}} = \sqrt{1+\frac{n^2}{N}\frac{1}{N-2\sqrt{tN}}} = 1+O(\frac{n^2}{N}\frac{1}{N-2\sqrt{tN}}) = 1+o(1)\Big\} =$$

$$= 1-\exp\{-\frac{n^2}{N}\cdot\frac{1+o(1)}{1+o(1)}+o\,(1)\}\cdot(1+o(1))\cdot\frac{(1+\frac{1}{12(N-n)}+O\left(\frac{1}{(N-n)^2}\right))^2}{(1+\frac{1}{12N}+O\left(\frac{1}{N^2}\right))(1+\frac{1}{12(N-2n)}+O\left(\frac{1}{(N-2n)^2}\right))} =$$

$$= 1-\exp\{-\frac{n^2}{N}\cdot\frac{1+o(1)}{1+o(1)}+o\,(1)\}\cdot(1+o(1))\cdot\frac{(1+o(1))^2}{(1+o(1))^2}\xrightarrow{\frac{n^2}{N}\to t}(1-e^{-t})(1+o(1)).$$

2.4 Линейный криптоанализ

Поскольку мы не связаны никакими ограничениями в выборе тех или иных функций, поисследуем, как изменится эффективность метода Мицуру Мацуи, если допустить хотя бы малейшие необдуманные изменения в оригинальном алгоритме DES.

Пусть функция расширения будет иметь вид:

$$E(\vec{X}) = (X[4], X[3], X[1], X[3], X[2], X[4], X[6], X[7], X[5], X[7], X[8], X[6])$$

Функция перестановки:

$$P(\vec{X}) = (X[2], X[6], X[4], X[7], X[3], X[8], X[5], X[1])$$

Возьмём следующие S-боксы:

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

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

Построим таблицу значений $NS_1(\alpha, \beta)$:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	-2	-2	-4	-4	2	-2	4	2	0	0	6	6	-4	0	-2
3	-2	-6	0	-4	2	2	0	2	8	-4	2	6	4	-12	2
4	4	2	-2	2	2	-8	0	2	2	0	8	0	-4	2	6
5	-4	-6	6	-2	-2	-4	-4	-2	6	-4	4	0	-4	2	-2
6	-6	0	2	-2	4	-2	-4	0	-2	4	2	2	-4	-2	-8
7	-6	4	6	2	0	-2	4	4	-6	-12	-6	2	-4	2	-4
8	8	2	6	-4	-4	-2	2	-4	-4	-2	2	0	0	2	-2
9	0	2	-2	-4	-4	6	-6	0	0	2	6	4	-4	-2	2
10	-6	0	-2	0	2	4	2	-6	4	2	0	-6	4	-2	4
11	2	-4	-6	0	-14	0	6	-2	0	2	0	-2	-8	-2	-4
12	-4	4	4	-2	6	-2	2	2	2	2	-2	4	-8	0	8
13	4	4	4	2	2	-6	-2	2	2	10	-2	0	4	-4	-4
14	-2	2	-4	2	4	-4	2	4	6	2	0	2	0	0	2
15	-2	-2	8	-2	0	4	-6	-4	-2	-2	-4	-2	-4	0	2
16	6	6	0	0	2	2	0	-2	8	-4	2	2	0	-4	-18
17	2	-6	0	0	-2	-2	8	-2	4	8	-6	2	-4	0	-2
18	0	0	-4	-8	4	0	0	-4	0	4	4	8	0	0	-4
19	4	0	-8	0	0	-8	-12	-4	-4	-4	0	0	4	0	0
20	2	4	2	-6	-4	-2	4	0	10	0	-2	2	4	2	0
21	-2	8	-6	-2	4	-2	0	12	2	0	2	-6	0	6	0
22	4	6	-2	2	-2	-4	4	2	6	-4	4	-4	0	2	2
23	0	6	-6	6	6	0	-4	-10	-2	0	-4	-4	-4	2	-2
24	2	8	2	0	6	4	2	6	-4	6	4	2	-4	-2	0
25	-2	4	-6	0	-6	0	2	2	-4	6	8	-2	4	-2	-4
26	0	-6	2	0	-4	2	6	0	-4	-6	6	-4	4	6	-2
27	4	2	6	-8	0	-6	2	-4	-4	6	-2	0	4	2	-2
28	-2	6	4	2	0	0	-2	-4	2	-2	4	-2	4	8	-2
29	2	-6	4	-2	0	0	-6	4	6	-6	4	-6	-4	-8	2
30	-4	0	-4	2	-2	-2	-6	2	6	-2	-6	4	-8	4	0
31	0	0	0	-2	-2	2	2	2	-6	-2	-2	-8	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	2	-6	4	0	2	6	0	2	-4	-4	6	10	4	8	2
35	2	-2	0	0	2	2	4	2	4	0	-6	-6	-4	4	-2
36	0	-6	-6	6	2	4	0	2	-2	0	-4	-4	4	-2	-10
37	-8	2	2	2	-2	-8	-4	-2	2	-4	8	-4	4	-2	-2
38	2	4	-2	6	-4	2	0	0	-2	0	-2	2	4	2	4
39	2	0	-6	-6	8	-6	0	4	-6	-8	-2	2	4	-2	0
40	4	-2	-2	0	-4	-2	-2	4	0	2	2	-4	-8	10	2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
41	-4	-2	-10	0	-4	-10	6	0	-4	-2	-2	8	-4	-2	-2
42	2	0	-2	0	2	-4	2	2	-4	-6	0	2	4	-2	4
43	-6	4	2	0	2	0	14	-2	0	-6	0	-2	0	-2	-4
44	4	0	0	6	6	2	6	-6	2	-2	10	-4	-8	-4	4
45	-4	0	0	-6	2	-2	2	2	-6	-2	2	0	-4	0	0
46	-6	2	8	6	-4	-8	2	-4	2	2	4	6	0	4	2
47	-6	6	-4	2	-8	8	2	-4	2	-2	0	10	4	4	2
48	-2	-2	0	0	2	2	0	-2	0	4	2	2	0	-4	-2
49	-6	2	0	0	-2	-2	-8	-2	-4	0	-6	2	-4	0	-2
50	-4	-4	-4	-4	4	0	4	-4	4	0	-4	-4	8	0	8
51	0	4	0	4	0	0	0	-4	0	0	0	4	-4	-8	4
52	6	-4	6	6	4	-6	4	0	-2	0	-6	6	4	-2	0
53	2	0	-2	-6	-4	-6	0	-4	6	0	-2	-2	0	2	0
54	4	2	-6	2	-2	8	0	2	-2	0	0	4	0	-2	6
55	0	-6	-2	6	6	4	0	6	6	-4	0	4	-4	6	-6
56	-10	4	2	-4	-2	4	-2	6	0	2	4	-2	4	-2	-4
57	2	0	-6	-4	2	0	-2	-6	8	-6	0	2	4	6	0
58	0	2	2	8	4	2	-2	0	4	2	-2	4	4	6	-2
59	4	2	-2	0	8	2	2	-12	-4	-2	6	0	-4	2	-2
60	-2	10	0	-6	0	4	2	-4	2	-6	-8	-2	-4	-4	2
61	2	-2	0	-10	0	4	-2	-4	-2	-2	0	2	-4	4	-2
62	0	0	0	-10	6	2	2	2	2	6	-2	0	0	8	0
63	-12	-8	-4	2	6	-2	2	-6	-2	6	2	-4	0	4	0

Наибольшее по модулю число в этой таблице находится на позиции (16, 15), оно равно -18. Тогда уравнение

$$\vec{X}[2] \oplus \vec{Y}[1,2,3,4] = \vec{K}[2]$$

является эффективным линейным статистическим аналогом 1-го S-бокса в классе всех линейных статистических аналогов вида

$$(\vec{Y}, \vec{j}) = (\vec{X} \oplus \vec{K}, \vec{i})$$

с вероятностью $p_1 = \frac{-18+32}{64} = \frac{7}{32}$ и $\Delta_1 = |1 - \frac{7}{16}| = \frac{9}{16}$

Повторим то же самое со вторым S-боксом:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	-2	2	-2	2	0	-8	-4	4	-6	-2	-2	2
3	0	0	-8	2	6	2	-2	0	0	-4	4	-2	10	2	6
4	0	2	2	6	2	0	4	4	4	-2	-2	-2	2	0	12
5	4	2	-2	6	-2	0	-8	0	-4	2	6	-6	2	4	-4
6	-4	-2	-6	-4	0	2	-2	-4	0	-2	2	0	-4	2	6
7	0	-2	-2	0	0	6	-2	0	8	10	2	-8	0	2	2
8	4	4	0	2	2	-2	-2	0	-8	-4	-4	-2	2	10	-2
9	4	0	-4	2	2	2	2	-4	4	-4	-4	2	6	-6	-2
10	0	4	4	4	4	0	-8	0	4	0	4	4	0	-4	0
11	0	0	8	0	0	0	0	4	0	8	-4	-4	0	0	4
12	0	6	-2	0	0	-2	6	-4	0	2	-2	4	0	-6	-2
13	-4	2	-2	0	4	2	-10	4	-4	6	-2	4	-4	-2	6
14	0	2	2	2	-2	-4	8	-4	0	2	6	2	2	0	0
15	-4	-2	-6	-2	-2	-4	12	4	4	6	6	-2	2	0	4
16	0	-6	2	4	0	-6	-2	4	0	6	2	0	-8	-2	-10
17	4	2	-2	4	-4	2	2	0	0	2	2	-4	0	10	-2
18	4	2	6	2	6	0	4	-4	4	-6	10	2	-14	4	12
19	0	-6	2	-2	6	4	4	0	-4	-2	-6	2	-2	4	0
20	-4	-8	-4	-2	2	-6	-2	-4	4	4	-4	6	-2	2	2
21	12	0	-4	-2	2	2	-2	4	4	4	-4	-2	-2	2	2
22	4	-4	0	4	-4	4	4	4	-4	4	4	8	4	4	0
23	12	4	0	0	0	-8	0	-4	-4	4	-4	-4	0	0	4
24	-8	-2	-2	2	2	4	4	0	0	6	6	-2	6	0	0
25	4	-6	6	2	6	8	-4	0	-4	2	-2	6	2	-4	0
26	8	-2	-2	-4	0	-2	2	0	0	2	2	-4	0	2	-2
27	-4	10	6	0	0	6	-2	0	-4	-2	2	-8	0	2	-6
28	0	4	-4	4	0	-4	0	8	0	-4	4	4	0	4	0
29	0	0	0	4	0	0	-4	4	4	-4	4	8	-4	4	0
30	-4	0	4	-2	2	2	-2	0	4	-4	0	2	-2	2	-2
31	4	-4	8	2	-2	-6	-2	-12	0	4	0	2	6	-2	2
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	-4	4	6	-6	2	-2	4	4	-4	-12	-2	-6	-2	2
35	0	-4	-4	2	6	-2	2	4	-4	-4	4	-6	-2	-6	-2
36	0	-2	-2	-2	-6	4	-8	0	0	-2	-2	2	6	8	4
37	-4	-2	2	-2	-2	4	4	-4	0	2	-2	-2	-2	12	-4
38	4	-10	2	-12	0	2	-2	-4	0	-2	10	0	-4	2	-2
39	0	6	-2	0	0	-2	-2	0	0	-6	2	0	0	-6	-6
40	0	4	-4	2	-2	-2	-6	0	-4	4	8	6	-2	-6	2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
41	0	0	-8	2	-2	2	-2	-12	0	-4	0	2	-6	2	-6
42	4	0	-4	4	-8	-4	0	-4	4	0	0	0	0	-4	-4
43	4	-4	0	8	12	4	0	-8	8	0	0	-8	0	0	0
44	-4	2	6	0	-4	-6	-2	0	-8	2	2	0	0	2	10
45	0	-2	-2	0	-8	-2	-10	0	4	-2	2	-8	-4	-2	2
46	-4	-6	-2	10	2	-4	-4	-4	-4	-6	2	-6	6	0	4
47	0	6	-2	-2	2	4	0	-4	0	6	2	6	-2	0	0
48	4	-2	2	-4	-4	6	6	0	-8	-2	-2	-4	0	-10	2
49	0	-2	-2	4	-8	-2	2	-4	0	2	-2	0	-8	2	2
50	0	2	2	2	2	0	0	-4	0	2	-2	2	-2	-4	0
51	4	2	-2	-2	-6	-4	0	0	0	-2	-2	2	2	4	4
52	-8	0	-8	6	6	-6	2	4	0	4	-8	-2	-6	2	-2
53	-8	0	0	-2	-2	2	2	-4	0	-4	0	-2	2	2	-2
54	0	0	0	-4	0	0	-4	0	4	4	0	-4	4	-4	4
55	8	0	-8	-8	4	-4	0	8	4	-4	0	0	0	0	0
56	0	2	2	-6	2	0	0	4	-4	-2	-2	-6	-6	0	0
57	4	6	-6	2	6	4	0	-4	-8	10	-2	2	-2	4	0
58	0	-2	6	4	0	-10	2	0	0	2	2	-4	0	2	-2
59	-4	2	-2	-8	8	-10	-2	-8	-4	-2	-6	0	0	2	2
60	0	-4	-4	4	0	4	0	0	-8	4	4	-4	-8	-4	0
61	0	0	8	-4	8	-8	-4	4	4	4	4	0	4	4	-8
62	4	-12	0	6	2	-2	2	4	-8	-4	0	6	2	2	-2
63	-4	-8	-4	-6	-2	-2	-6	0	-4	4	0	-2	2	-2	2

Наибольшее по модулю число в этой таблице находится на позиции (18, 13), оно равно -14. Тогда уравнение

$$\vec{X}[2,5] \oplus \vec{Y}[1,2,4] = \vec{K}[2,5]$$

является эффективным линейным статистическим аналогом 2-го S-бокса в классе всех линейных статистических аналогов вида

$$(\vec{Y}, \vec{j}) = (\vec{X} \oplus \vec{K}, \vec{i})$$

с вероятностью $p_2 = \frac{-14+32}{64} = \frac{9}{32}$ и $\Delta_2 = |1 - \frac{9}{16}| = \frac{7}{16}$.

Получаем $\Delta_1 > \Delta_2$, значит, эффективным линейным статистическим аналогом произвольного раунда DES является уравнение:

$$\vec{X}_i[2] \oplus \vec{Y}_i[1,2,3,4] = \vec{K}_i[2]$$

.

С учётом расширения и перестановки:

$$\vec{X}_i[3] \oplus \vec{Y}_i[8, 1, 5, 3] = \vec{K}_i[2]$$

Запишем для первого и третьего раунда:

$$\vec{P}_L[3] \oplus (\vec{X}_2 \oplus \vec{P}_H)[8, 1, 5, 3] = \vec{K}_1[2]$$

$$(\vec{C}_L \oplus \vec{Y}_4)[3] \oplus (\vec{X}_2 \oplus \vec{C}_L)[8, 1, 5, 3] = \vec{K}_3[2]$$

Ещё нам понадобится уравнение, содержащее $\vec{Y}_4[3]$. До перестановки это четвёртый бит первого S-бокса. То есть, надо искать по первому столбцу. На позиции (63, 1) максимальный по модулю элемент -12. Тогда $p_* = \frac{-12+32}{64} = \frac{5}{16}$ и $\Delta_* = |1 - \frac{5}{8}| = \frac{3}{8}$

$$\vec{X}_i[1, 2, 3, 4, 5, 6] \oplus \vec{Y}_i[4] = \vec{K}_i[1, 2, 3, 4, 5, 6]$$

Учтя расширение и перестановку, получим:

$$\vec{X}_i[1, 2] \oplus \vec{Y}_i[3] = \vec{K}_i[1, 2, 3, 4, 5, 6]$$

Запишем для четвёртого раунда

$$\vec{C}_L[1, 2] \oplus \vec{Y}_4[3] = \vec{K}_4[1, 2, 3, 4, 5, 6]$$

Сложив все уравнения, получим:

$$\vec{P}_L[3] \oplus \vec{P}_H[1, 3, 5, 8] \oplus \vec{C}_L[2, 5, 8] = \vec{K}_1[2] \oplus \vec{K}_3[2] \oplus \vec{K}_4[1, 2, 3, 4, 5, 6]$$

Получим результирующую эффективность и вероятность:

$$\Delta = \Delta_1 \cdot \Delta_1 \cdot \Delta_* = \frac{9}{16} \cdot \frac{9}{16} \cdot \frac{3}{8} = \frac{243}{2048} \approx 0.12$$

$$p = \frac{1 - \Delta}{2} = \frac{1805}{4096} \approx 0.44$$

Тогда можно раскрыть 8 бит ключа $K = (K_1, K_2, K_3, K_4)$, зная $|p - \frac{1}{2}|^{-2} = 284$ открытых текста с вероятностью успеха 97.7%. Из статьи Мицуру Мацуи можно сделать вывод, что лучший стат. аналог для оригинального $DES/4$ требует 269 открытых текстов для раскрытия 2 бит ключа. То есть, стало хуже примерно в 4 раза.

Это означает, что "мудрить" с алгоритмами нельзя, а подбирать все параметры нужно крайне обдуманно, иначе можно значительно ухудшить стойкость криптографических алгоритмов.

Часть 3

Экзамен

1. Определение шифра. Шифр простой замены, перестановки, гаммирования. Основные условия криптоанализа.

Отображение $T : X \times K \to Y$ называется **шифром**, если $\forall k \in K$ $\exists T^{-1}(y, k) = x$.

Пусть $A = \{a_1, \ldots, a_m\}$ – конечный алфавит, S_m – множество всех подстановок на A. Для некоторого натурального n положим $X = A^n$. Если $x = (a_{i_1}, \ldots, a_{i_n})$, $k \in S_m$, то определим **шифр простой замены** следующим образом:

$$T(x, k) = (k(a_{i_1}), k(a_{i_2}), \ldots, k(a_{i_n})) = y = (b_{i_1}, \ldots, b_{i_n}).$$

Пусть $A = \{a_1, \ldots, a_m\}$ – конечный алфавит, n – натуральное число, S_n – симметрическая группа подстановок на множестве $\{1, \ldots, n\}$, $X = A^n$. Если $x = (a_{i_1}, \ldots, a_{i_n})$, $k = \left(\begin{smallmatrix} 1 & \cdots & n \\ j_1 & \cdots & j_n \end{smallmatrix}\right) \in S_n$, то **шифр перестановки** на X определяется следующим образом:

$$T(x, k) = (a_{i_{j_1}}, a_{i_{j_2}}, \ldots, a_{i_n}) = y.$$

Пусть $A = \{0, \ldots, m-1\}$ – алфавит, $X = A^n$ – множество открытых текстов. Рассмотрим кольцо вычетов $\mathbb{Z}_m$. Положим $K = A^n$ и $\forall x \in X, k \in K$, определим **шифр гаммирования**:

$$y = T(x, k) = (x + k) \pmod{m},$$

где сложение происходит в кольце $\mathbb{Z}_m$.

Основные условия криптоанализа:

1. Известен шифртекст y, один или несколько. Задачи:
   а) Нахождение T – преобразования зашифрования;
   б) Нахождение T, T^{-1}, x – дешифрование по шифртексту.
2. Известны одна или несколько пар (x, y). Определить $T(T^{-1})$ и найти k – ключ шифрования.
3. Известны $T(T^{-1})$, один или несколько шифртекстов y. Найти:
   а) x – бесключевое чтение;
   б) k, x – дешифрование по шифртексту при известной шифрсистеме.
4. Известны $T, T^{-1}, (x, y)$. Найти k.
   а) Известны особые x – атака выбранного открытого текста;
   а) Известны особые y – атака с использованием шифртекста.
5. Известны T, T^{-1}, шифртекст y или пары (x, y), некоторая форма преобразования $T(., k)$, но неизвестны k и $T^{-1}(., k)$ – системы с открытым ключом (Диффи и Хеллман, 1976 г.)

2. Теоретическая стойкость по Шеннону. Практическая стойкость. Пример совершенного шифра.

Теоретическая стойкость (совершенная секретность) – Система является безопасной против атак противника с неограниченным временем и ресурсами.

Практическая стойкость (вычислительная) – Система является безопасной против атак противника в ограниченный период времени с ограниченными ресурсами.

Шеннон определил совершенную секретность условием:

$$P(x|y) = P(x) \quad \forall x \in X, y \in Y,$$

где X, Y – множества открытых сообщений и возможных шифртекстов.

Пример совершенного шифра – шифр гаммирования, в котором равновероятный ключ имеет ту же длину, что и открытый текст. Пусть $X = \{0, 1\},\ Y = \{0, 1\}, K = \{0, 1\}$, $T(x, k) = (x \oplus k) \pmod 2$, $T^{-1}(y, k) = (y \oplus k) \pmod 2$, $X \sim \begin{pmatrix} 0 & 1 \\ p & q \end{pmatrix}$, $q = 1 - p$, $Y \sim \begin{pmatrix} 0 & 1 \\ 0.5 & 0.5 \end{pmatrix}$, $K \sim \begin{pmatrix} 0 & 1 \\ 0.5 & 0.5 \end{pmatrix}$, $P(Y|X) = \frac{P(X,Y)}{P_X(X)} = \begin{pmatrix} y/x & 0 & 1 \\ 0 & 0.5 & 0.5 \\ 1 & 0.5 & 0.5 \end{pmatrix}$.

По формуле Байеса:

$$P(X = x|Y = y) = \frac{P_X(x)P(y|x)}{P(y)} = \frac{P_X(x)}{\frac{1}{2}2} = P_X(x).$$

3. Метод полного перебора, его средняя трудоемкость. Параллельное опробование с помощью случайно выбираемого на каждом шагу ключа.

Пусть известны $T, T^{-1}, (x, y)$, а ключ неизвестен. **Методом полного перебора** называется поиск решения уравнения $T(x, k) = y$ перебором по всем $k \in K, \quad |K| < \infty$.

Определим случайные величины: τ – количество опробований ключа включительно до момента обнаружения, $\xi_i =$ [ключ на i-м месте]. Ключ равновероятен, тогда $P(\xi_i = 1) = \frac{1}{|K|}$ для всех $i = \overline{1, |K|}$. **Средняя трудоёмкость МПП**:

$$\mathbb{E}\tau = \sum_{i=1}^{|K|} iP(\xi_i = 1) = \frac{1}{|K|}\sum_{i=1}^{|K|} i = \frac{|K|(|K|+1)}{|K| \cdot 2} = \frac{|K|+1}{2}.$$

Пусть параллельно работают N машин. Если t – число шагов работы машин, то Nt – число опробований. Число тактов опробования – случайная величина η. Аналогом является задача о размещениях: в каждую из $|K|$ ячеек может попасть от 0 до N частиц. Вероятность того, что из комплекта i ни одна частица не попадёт в данную ячейку, равна $q = (1 - \frac{1}{|K|})^N$. Первое попадание в данную ячейку на комплекте с номером t, означающее, что ключ получен, имеет вероятность $P(\eta = t) = q^{t-1}(1-q)$ – с.в. η имеет геометрическое распределение. Тогда **средняя трудоёмкость МПП при параллельном опробовании** равна:

$$\mathbb{E}\eta = \frac{1}{1-q} = \frac{1}{1-(1-\frac{1}{|K|})^N} \approx \left\{N << |K|\right\} \approx \frac{1}{1-1+\frac{N}{|K|}} = \frac{|K|}{N}.$$

4. Аналитический метод криптоанализа. Треугольные системы и их решение. Линейные системы. Сложность решения методом Гаусса.

Пусть $K = K_1 \times K_2 \times \ldots \times K_r$, $x = (x_1 x_2 \ldots x_s)$, $y = (y_1 y_2 \ldots y_s)$, $x_i, y_i \in A$. **Идея аналитического метода** заключается в том, чтобы записать систему уравнений и решить её относительно ключа:

$$\begin{cases} y_1 = f_1(x_1, \ldots, x_s, k_1, \ldots, k_r) \\ \ldots \\ y_s = f_s(x_1, \ldots, x_s, k_1, \ldots, k_r) \end{cases}$$

Известны f_i и (x, y).

Пусть эту систему можно преобразовать к **треугольной системе**:

$$\begin{cases} g_1(x,y) = h_1(x,y,k_1) \\ g_2(x,y) = h_2(x,y,k_1,k_2) \\ \dots \\ g_r(x,y) = h_r(x,y,k_1,\dots,k_r) \end{cases}$$

1. Опробуем $k_1 \in K_1$, число опробований $\le |K_1| \Rightarrow$ восстанавливаем k_1.
2. Опробуем $k_2 \in K_2$, число опробований $\le |K_2| \Rightarrow$ восстанавливаем k_2.
...
r. Опробуем $k_r \in K_r$, число опробований $\le |K_r| \Rightarrow$ восстанавливаем k_r.

Таким образом, сложность $\le |K_1| + |K_2| + \ldots + |K_r|$.

Пусть имеется **линейная система**:

$$\begin{cases} b_{11}k_1 + b_{12}k_2 + \ldots + b_{1r}k_r = g_1(x,y) \\ b_{21}k_1 + b_{22}k_2 + \ldots + b_{2r}k_r = g_2(x,y) \\ \dots \\ b_{r1}k_1 + b_{r2}k_2 + \ldots + b_{rr}k_r = g_r(x,y) \end{cases}$$

Методом Гаусса решается за $\sum_{k=1}^{r} k^2 = \frac{r(r+1)(2r+1)}{6} \approx \frac{r^3}{3}$ операций.

5. Регистр сдвига с нелинейной обратной связью. Линейная сложность. Условия регулярности (теорема с доказательством).

Последовательность γ называется **линейной рекуррентной последовательностью** (ЛРП) порядка $r > 0$ над $GF(2)$, если она описывается **законом рекурсии**:

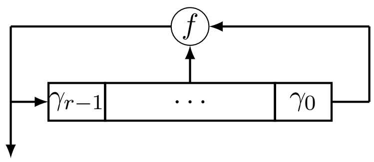
$$\gamma_{n+r} = \sum_{i=0}^{r-1} \alpha_i \gamma_{n+i}, \quad n = 0, 1, \ldots$$

$\alpha_i \in GF(2),\ i = \overline{0, r-1}$ и все операции выполняются в поле $GF(2)$.

Нелинейная рекуррентная последовательность (НЛРП) определяется выражением:

$$\gamma_{n+r} = f(\gamma_n, \gamma_{n+1}, \ldots, \gamma_{n+r-1}), \quad n = 0, 1, \ldots$$

Регистр сдвига с нелинейной обратной связью (НЛРС) выглядит следующим образом:



Нелинейный регистр сдвига называется **регулярным**, если порождаемая им выходная последовательность γ периодична при любом начальном заполнении регистра.

Условие регулярности: если НЛРС регулярен, то для любого начального заполнения существует ЛРС (вида) размера v ($v \geq r$) такой, что порождаемая им последовательность совпадает с последовательностью, порождаемой при этом начальном заполнении НЛРС.

◩ НЛРС регулярен $\Rightarrow$ при любом начальном заполнении последовательность γ периодична $\Rightarrow$ ЛРС вида $\gamma_{n+T} = \gamma_n$, $n = 0, 1, \ldots$ порождает ту же последовательность.

◪

6. Метод “встреча посередине”. Трудоемкость метода. Пример реализации метода “встреча посередине”.

Пусть даны два шифра: $T_1(x, k_1)$ и $T_2(z, k_2)$. Положим

$$y = T_2(T_1(x, k_1), k_2).$$

Ключ $k = (k_1, k_2)$, где $k_1 \in K_1,\ k_2 \in K_2,\ K = K_1 \times K_2$. Считаем, что в T_1 и T_2 согласованы области определения и области значений.

Описание метода. Пусть для пары (x, y) существует единственный ключ. Составим две таблицы вида:

$$z_1 = T_1(x, k_1^{(1)}) \ldots\ldots z_{|K_1|} = T_1(x, k_1^{(|K_1|)}),$$

$$z'_1 = T_2^{-1}(y, k_2^{(1)}) \ldots\ldots z'_{|K_2|} = T_2^{-1}(y, k_2^{(|K_2|)}),$$

затем объединим их и упорядочим. Пара $z_i = z'_j$ определяет искомый ключ $k = (k_1^{(i)}, k_2^{(j)})$.

Трудоёмкость метода. Составление таблицы требует $|K_1| + |K_2|$ операций опробования. Упорядочивание таблицы размера M оценивается в $M \ln M$ операций. Таким образом, средняя трудоёмкость метода равна:

$$(|K_1| + |K_2|)(1 + \ln(|K_1| + |K_2|)).$$

Если $|K| = N,\ |K_1| = |K_2|$, можно сделать такую оценку: $\sqrt{N}\ln N$.

Пример. Рассмотрим двойной DES на ключах k_1 и k_2.

$$\xrightarrow{x} \boxed{DES(k_1)} \longrightarrow \boxed{DES(k_2)} \xrightarrow{y}$$

Оценка трудоёмкости $2^{56}\ln 2^{112} \approx 10^{19} \ll 10^{34} \approx \frac{2^{56}\cdot 2^{56}}{2}$. Памяти потребуется $2N \approx 10^{17}$.

7. Метод "разделяй и побеждай". Трудоемкость метода. Пример реализации метода.

Ключ $k = (k_1, k_2)$, где $k_1 \in K_1,\ k_2 \in K_2,\ K = K_1 \times K_2$.

Пусть существует критерий h:

$$h(x, y, k_1) = \begin{cases} 1, & \exists k_2 \in K_2:\ T(x, (k_1, k_2)) = y \\ 0, & \forall k_2 \in K_2:\ T(x, (k_1, k_2)) \neq y \end{cases}$$

Пусть известна пара (x, y) достаточной длины, что $\exists! k:\ T(x, k) = y$.

Описание метода. Первым шагом отбракуем элементы множества K_1, используя критерий h, и получим единственный k_1. На это потребуется $\frac{|K_1|}{2}$ опробований. На втором шаге применяем МПП относительно k_2, на это уйдёт $\frac{|K_2|}{2}$ опробований.

Трудоёмкость метода равна $\frac{|K_1|+|K_2|}{2}$.

Пример. Рассмотрим двойной DES на ключах k_1 и k_2, устроенный таким образом, что открытый текст разбивается на блоки и каждый блок перед шифрованием складывается с предыдущим зашифрованным блоком. При этом каждый нечётный блок шифруется с использованием ключа k_1, а каждый чётный – с k_2:

$$\text{ОТ } x = x_1 x_2 \ldots x_{2N},\ |x_i| = 64,\ i = \overline{1, 2N},$$

$$\text{ШТ } y = y_1 y_2 \ldots y_{2N},\ y_0 = 0,$$

$$DES(x_{2i+1} \oplus y_{2i}, k_1) = y_{2i+1},\ i = \overline{0, N-1},$$

$$DES(x_{2i} \oplus y_{2i-1}, k_2) = y_{2i},\ i = \overline{1, N}.$$

Пусть известна пара (x, y). Получим следующие два множества:

$$A = \{(x_{2i+1} \oplus y_{2i}, y_{2i+1}),\ i = \overline{0, N-1},\ y_0 = 0\}$$

$$B = \{(x_{2i} \oplus y_{2i-1}, y_{2i}),\ i = \overline{1, N}\}$$

Определим критерий $h(x, y, k_1) = 1 \Leftrightarrow DES(x_{2i+1} \oplus y_{2i}, k_1) = y_{2i+1}$, $i = \overline{0, N-1}$.

Трудоёмкость метода составит 2^{56}.

8. Методы криптоанализа при неравновероятной гамме.

Пусть $x = x_1 x_2 \ldots x_n$ – ОТ, $y = y_1 y_2 \ldots y_n$ – ШТ, $\gamma = \gamma_1 \gamma_2 \ldots \gamma_n$ – ключ, используется шифр гаммирования.

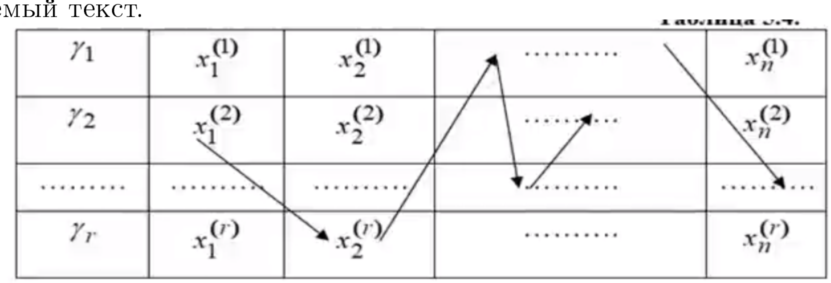
Метод протяжки вероятностного слова. Пусть для (x, y) и (x', y') использовался один и тот же ключ γ. Тогда:

$$y - y' = (x + \gamma) - (x' + \gamma) = x - x'.$$

Значит, если угадано (или предполагается), что начиная с некоторого места i в x стоит слово $a = a_1 a_2 \ldots a_r$, то в x' на том же месте можно прочитать слово $a' = a'_1 a'_2 \ldots a'_r$, где $a'_j = y'_{i+j} - y_{i+j} + a_j$, $j = \overline{1, r}$.

Метод чтения в колонках (Зигзагообразное чтение).

Пусть всего используется r значений гаммы. Составляется таблица, где в каждой строке находится сумма ШТ с некоторым значением гаммы γ_i. Всего таких строк r штук. Криптоаналитик пытается восстановить ОТ, выбирая буквы из столбцов так, чтобы получался осмысленный читаемый текст.



γ_1	$x_1^{(1)}$	$x_2^{(1)}$	....	$x_n^{(1)}$
γ_2	$x_1^{(2)}$	$x_2^{(2)}$	....	$x_n^{(2)}$
....	....	....	....	....
γ_r	$x_1^{(r)}$	$x_2^{(r)}$	....	$x_n^{(r)}$

9. Первая теорема Шеннона (теорема с доказательством).

Пусть $p(a_1) \ldots p(a_m)$ – вероятности появления букв на фиксированном месте i в открытом сообщении длины n. Предположим, что буквы в сообщении появляются независимо друг от друга с одним и тем же распределением. Обозначим через ν_i, $i = \overline{1, m}$ частоты букв $a_1 \ldots a_m$ в последовательности x (ОТ). Тогда вероятность выбора x в нашей схеме равна

$$P(x) = p^{\nu_1}(a_1) \cdot \ldots \cdot p^{\nu_m}(a_m).$$

Будем считать, что $p(a_i) > 0$, $H = -\sum_{i=1}^{m} p(a_i) \log_2 p(a_i)$.

Теорема 1. Для любых $\epsilon > 0$ и $\delta > 0$ можно найти такое n_0, что для любого $n > n_0$ последовательности из V_n распадаются на два непересекающихся класса B и $\overline{B}$ так, что:

$$1) P(\overline{B}) < \epsilon$$

$$2) \left| \frac{\log_2 P^{-1}(x)}{n} - H \right| < \delta, \ \forall x \in B$$

◪ Возьмём произвольные малые $\epsilon > 0$ и $\delta > 0$ и рассмотрим события

$$\overline{B}_i = \{x \in V_n, |\nu_i - np(a_i)| > \delta n\}, \ i = \overline{1,m}.$$

Эти события означают, что в слове длины n реальная частота встречаемости буквы a_i отличается от её теоретической встречаемости больше, чем на δn. Из ЗБЧ следует, что $\exists n_0^{(i)} : \ \forall n > n_0^{(i)} \Rightarrow P(\overline{B}_i) < \frac{\epsilon}{m}$ (точнее, из неравенства Чебышёва: $P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$). Определим $\overline{B} = \bigcup_{i=1}^m \overline{B}_i$. Тогда:

$$P(\overline{B}) = P(\bigcup_{i=1}^m \overline{B}_i) \leq \sum_{i=1}^m P(\overline{B}_i) < \epsilon, \quad \forall n > \max_i(n_0^{(i)}).$$

Первое утверждение доказано. Рассмотрим теперь следующее представление множества B:

$$B = \overline{\overline{B}} = \overline{\bigcup_{i=1}^m \overline{B}_i} = \bigcap_{i=1}^m \overline{\overline{B}}_i = \bigcap_{i=1}^m B_i,$$

$$B_i = \{x \in V_n, |\nu_i - np(a_i)| \leq \delta n\}, \ i = \overline{1,m}.$$

Обозначим за $\alpha_i = \nu_i - np(a_i), \ i = \overline{1,m}$. Тогда $|\alpha_i| \leq \delta n$. Выразим вероятность выбора ОТ через α_i (см. выражение перед теоремой):

$$P(x) = p^{\alpha_1 + np(a_1)}(a_1) \cdot \ldots \cdot p^{\alpha_m + np(a_m)}(a_m) = \prod_{i=1}^m p^{\alpha_i + np(a_i)}(a_i).$$

Тогда получим:

$$\log_2 \frac{1}{P(x)} = \log_2 \prod_{i=1}^m p^{-\alpha_i - np(a_i)}(a_i) = -\sum_{i=1}^m (\alpha_i + np(a_i)) \log_2 p(a_i) =$$

$$= -n \sum_{i=1}^m p(a_i) \log_2 p(a_i) - \sum_{i=1}^m \alpha_i \log_2 p(a_i) = nH - \sum_{i=1}^m \alpha_i \log_2 p(a_i)$$

Следовательно,

$$\left|\frac{\log_2 P^{-1}(x)}{n} - H\right| = \left| -\frac{1}{n}\sum_{i=1}^{m}\alpha_i \log_2 p(a_i)\right| \leq \frac{1}{n}\sum_{i=1}^{m}|\alpha_i||\log_2 p(a_i)| < \\ < \delta \sum_{i=1}^{m}|\log_2 p(a_i)| = \delta \cdot c$$

Поскольку $p(a_i) > 0$, сумма конечна и является константой (от n не зависит).

◪

10. Вторая теорема Шеннона (теорема с доказательством).

Упорядочим множество всех возможных последовательностей по убыванию вероятности возможности появления их в качестве открытого текста. Определим множество наиболее вероятных ОТ для $0 < \epsilon < 1$ таким образом:

$$Q_n(\epsilon) \subseteq V_n : P(Q_n(\epsilon)) \geq 1 - \epsilon, \ \forall y \in Q_n(\epsilon) : P(Q_n(\epsilon) \setminus y) < 1 - \epsilon.$$

Обозначим $\beta_n(\epsilon) = |Q_n(\epsilon)|$.

Докажем, что множество $Q_n(\epsilon)$ содержит минимальное число последовательностей среди всех множеств C, таких что $P(C) \geq 1 - \epsilon$.

От противного. Пусть $Q_n(\epsilon)$ – не минимальное множество. Это означает $|C \setminus Q_n(\epsilon)| < |Q_n(\epsilon) \setminus C|$. По определению $Q_n(\epsilon)$, в нём содержатся наиболее вероятные последовательности. Тогда $\forall x \in Q_n(\epsilon), x \notin C$ и $\forall y \notin Q_n(\epsilon), y \in C$ справедливо $P(x) \geq P(y)$. Следовательно:

$$\sum_{y \in C \setminus Q_n(\epsilon)} P(y) < \sum_{x \in Q_n(\epsilon) \setminus C} P(x),$$

поскольку справа слагаемых хотя бы на 1 больше и все они не меньше слагаемых в левой сумме.

Пусть $x_0 = \min\limits_{x \in Q_n(\epsilon) \setminus C} P(x)$. Значит,

$$P(C \setminus Q_n(\epsilon)) = \sum_{y \in C \setminus Q_n(\epsilon)} P(y) \leq \sum_{x \in Q_n(\epsilon) \setminus (C \cup \{x_0\})} P(x) = P(Q_n(\epsilon) \setminus (C \cup \{x_0\})).$$

$$Q_n(\epsilon) \setminus \{x_0\} = (Q_n(\epsilon) \cap C) \cup ((Q_n(\epsilon) \setminus C) \setminus \{x_0\}) \text{ (очев.)}$$

$$P(Q_n(\epsilon) \setminus \{x_0\}) < 1 - \epsilon \text{ (опр.)}$$

$$P(C) = P(C \cap Q_n(\epsilon)) + P(C \setminus Q_n(\epsilon)) \leq P(C \cap Q_n(\epsilon)) + \\ + P(Q_n(\epsilon) \setminus (C \cup \{x_0\})) = P(Q_n(\epsilon) \setminus \{x_0\}) < 1 - \epsilon$$

Получили $P(C) < 1 - \epsilon$ – противоречие, значит, $Q_n(\epsilon)$ – минимальное множество.

Теорема 2. $\forall \epsilon > 0$

$$\lim_{n \to \infty} \frac{\log_2 \beta_n(\epsilon)}{n} = H$$

◩ Возьмём малое $\delta > 0$ и рассмотрим множество B из первой теоремы Шеннона. Тогда, $\forall x \in B$ по этой теореме получим:

$$\left| \frac{\log_2 P^{-1}(x)}{n} - H \right| < \delta \Rightarrow H - \delta < \frac{-\log_2 P(x)}{n} < H + \delta \Rightarrow \\ \Rightarrow -n(H + \delta) < \log_2 P(x) < -n(H - \delta) \Rightarrow 2^{-n(H+\delta)} < P(x) < 2^{-n(H-\delta)}$$

Так как B и $\overline{B}$ не пересекаются, можно записать:

$$P(Q_n(\epsilon)) = \sum_{x \in Q_n(\epsilon)} P(x) = \sum_{x \in Q_n(\epsilon) \cap B} P(x) + \sum_{x \in Q_n(\epsilon) \cap \overline{B}} P(x) < \\ < \sum_{x \in Q_n(\epsilon) \cap B} 2^{-n(H-\delta)} + P(\overline{B}) = \big|Q_n(\epsilon) \cap B\big| 2^{-n(H-\delta)} + P(\overline{B}) < \\ < \big|Q_n(\epsilon)\big| 2^{-n(H-\delta)} + \epsilon = \beta_n(\epsilon) 2^{-n(H-\delta)} + \epsilon$$

По определению:

$$1 - \epsilon \leq P(Q_n(\epsilon)) < \beta_n(\epsilon) 2^{-n(H-\delta)} + \epsilon \Rightarrow \beta_n(\epsilon) > 2^{n(H-\delta)}(1 - 2\epsilon)$$

С другой стороны,

$$\beta_n(\epsilon) = |Q_n(\epsilon)| \leq |B| = \sum_{x \in B} 1 < \sum_{x \in B} \frac{P(x)}{2^{-n(H+\delta)}} \leq 2^{n(H+\delta)}$$

Тогда

$$2^{n(H-\delta)} < \beta_n(\epsilon) < 2^{n(H+\delta)} \Rightarrow H - \delta < \frac{\log_2 \beta_n(\epsilon)}{n} < H + \delta \Rightarrow \\ \Rightarrow \left| \frac{\log_2 \beta_n(\epsilon)}{n} - H \right| < \delta \Rightarrow \lim_{n \to \infty} \frac{\log_2 \beta_n(\epsilon)}{n} = H.$$

◪

11. Модель открытого текста, оценка числа открытых текстов.

Модель открытого текста. Пусть $p(a_1)\dots p(a_m)$ – вероятности появления букв на фиксированном месте i в открытом сообщении длины n. Предположим, что буквы в сообщении появляются независимо друг от друга с одним и тем же распределением. Обозначим через $\nu_i,\ i=\overline{1,m}$ частоты букв $a_1\dots a_m$ в последовательности x (ОТ). Тогда вероятность выбора x в нашей схеме равна

$$P(x) = p^{\nu_1}(a_1)\cdot\ldots\cdot p^{\nu_m}(a_m).$$

Будем считать, что $p(a_i)>0,\ H=-\sum_{i=1}^m p(a_i)\log_2 p(a_i)$.

Возьмём произвольные малые $\epsilon>0$ и $\delta>0$ и рассмотрим события

$$B_i = \{x\in V_n, |\nu_i - np(a_i)|\le \delta n\},\ i=\overline{1,m},$$

$$\overline{B}_i = \{x\in V_n, |\nu_i - np(a_i)| > \delta n\},\ i=\overline{1,m}.$$

Оценка по теореме Шеннона. Множество ОТ можно представить как $X=B\cup\overline{B}$. Из первой теоремы Шеннона следует, что $\overline{B}$ имеет очень малую вероятность ($P(\overline{B})\ll 1$). Из той же теоремы следует свойство равнораспределённости: для каждого $x\in B$ справедливо $P(x)\approx 2^{-nH}$. Таким образом, $|\overline{B}|\approx 0$ и $|B|\approx 2^{nH}$. Тогда число ОТ можно оценить как $|X|\approx 2^{nH}$.

12. Перекрытия гаммы. Средняя длина цикла с данной точкой в случайной подстановке.

Перекрытия гаммы. Пусть $x=x_1x_2\dots x_n$ – ОТ, $y=y_1y_2\dots y_n$ – ШТ, $\gamma=\gamma_1\gamma_2\dots\gamma_n$ – ключ, используется шифр гаммирования. Есть две различные пары (x,y) и (x',y'). Какова вероятность перекрытия?

Пусть $p(a_1)\dots p(a_m)$ – вероятности появления букв на фиксированном месте i в открытом сообщении длины n (распределение P). Тогда с.в. $\xi = x - x'$ имеет распределение $P^* = P * P$ – свёртка P с P.

В случае, если имеется перекрытие, получим $y-y'=x-x'$ – имеет то же распределение P^*.

Если перекрытия нет, то $y-y'=(x-x')+(\gamma-\gamma')$. Поскольку гамма выбирается равновероятно, то $\gamma-\gamma'$ тоже имеет равновероятное распределение, а следовательно, и $y-y'$.

Пусть есть статистический критерий, проверяющий гипотезу H_0 о равновероятности $y - y'$ против альтернативы H_1, что $y - y'$ имеет распределение P^*. Тогда принятие гипотезы H_0 будет означать отсутствие перекрытия гаммы, а принятие H_1 – наличие перекрытия.

Средняя длина цикла с данной точкой в случайной подстановке. Гамму получают с помощью конечного автомата A без входа, в котором начальное состояние является ключом k. Из-за конечности множества состояний автомата обязательно возникнет период. Пусть A – равновероятная подстановка на множестве $\{1, \ldots, n\}$. Оценим количество шагов автомата до того, как он попадёт обратно в состояние k.

Пусть t – длина полученного цикла. Введём случайную величину

$$\xi_i = \begin{cases} 1, & t = i \\ 0, & t \neq i \end{cases}.$$

Тогда $t = \sum_{i=1}^{n} i\xi_i$ и $\mathbb{E}t = \sum_{i=1}^{n} i\mathbb{E}\xi_i$.

Получим $P(\xi_i = 1)$. Для того, чтобы цикл был длины i, можно выбрать $i-1$ состояний случайным образом (начальное состояние фиксировано – k) – C_{n-1}^{i-1}, расположить их же случайным образом – $(i-1)!$ и случайно расположить оставшиеся состояния – $(n-i)!$. При этом всего различных автоматов (-перестановок) длины n – $n!$ штук.

$$P(\xi_i = 1) = \frac{C_{n-1}^{i-1}(i-1)!(n-i)!}{n!} = \frac{(n-1)!(i-1)!(n-i)!}{(i-1)!(n-i)!n!} = \frac{1}{n}$$

$$\mathbb{E}\xi_i = 1 \cdot P(\xi_i = 1) + 0 \cdot P(\xi_i = 0) = P(\xi_i = 1) = \frac{1}{n}$$

Следовательно,

$$\mathbb{E}t = \sum_{i=1}^{n} i\mathbb{E}\xi_i = \frac{1}{n}\sum_{i=1}^{n} i = \frac{1}{n} \cdot \frac{n(n+1)}{2} = \frac{n+1}{2}.$$

13. Линейный криптоанализ блочных шифров.

Рассмотрим схему произвольного итеративного блочного шифра в i-ом раунде: $\vec{Y}(i) = E(\vec{X}(i), \vec{K}(i))$, где E – функция шифрования, $\vec{X}(i)$ – блок открытого текста в i-ом раунде, $\vec{Y}(i)$ – блок шифртекста, $\vec{K}(i)$ - подключ, используемый в i-ом раунде. $\vec{Y}(i), \vec{X}(i) \in V_n$, $\vec{K}(i) \in V_m$, n – размер блока, m – размер подключа.

Обозначим через $(\vec{X},\vec{\alpha}) = X_1\alpha_1 \oplus \ldots \oplus X_n\alpha_n = X_{i_1} \oplus \ldots \oplus X_{i_k} = X[i_1,\ldots,i_k]$ – скалярное произведение двоичных векторов $\vec{X}$ и $\vec{\alpha}$, где $(\alpha_{i_1},\ldots,\alpha_{i_k})$ – единичные координаты вектора $\vec{\alpha}$.

Линейным статистическим аналогом нелинейной функции E (ЛСА) называется случайная величина

$$S(i) = (\vec{Y}(i),\alpha(i)) \oplus (\vec{X}(i),\beta(i)) \oplus (\vec{K}(i),\gamma(i)),$$

для которой $P(S(i)=1) = p \neq \frac{1}{2}$ для произвольного $\vec{X}(i)$.

$\Delta(S(i)) = |1-2p|$ – **эффективность линейного стат. аналога**.

Эффективным линейным статистическим аналогом (ЭЛСА) называется линейный статистический аналог

$$S_{1\ldots n} = (\vec{X}(1),\vec{\alpha}) \oplus (\vec{Y}(n),\vec{\beta}) \oplus \sum_{i=1}^{n}(\vec{K}(i),\vec{\gamma}(i))$$

из заданного множества с наибольшим Δ.

Задачи линейного криптоанализа:

1. Найти ЭЛСА и вычислить его вероятность.
2. Определить несколько или все биты ключа с помощью ЭЛСА.

Задача нахождения ЭЛСА для S-боксов DES. $\vec{Y} \in V_4$, $\vec{X},\vec{K} \in V_6$. Нелинейная функция, реализующая S-бокс может быть записана в виде

$$\vec{Y} = F_a(\vec{X} \oplus \vec{K}),\ a = \overline{1,8}$$

Пусть $1 \leq i < 64$, $1 \leq j < 16$, а $\vec{k}$ – двоичное представление числа $k \in \mathbb{N}$. ЛСА для каждого из таких уравнений будет уравнение вида

$$(\vec{Y},\vec{j}) = (\vec{X} \oplus \vec{K},\vec{i}).$$

Обозначим через $S_a(i,j)$ число ненулевых $\vec{X} \in V_6$ для a-го S-бокса DES таких, что выполняется указанное уравнение. Пусть

$$S_a^*(i^*,j^*):\ |S_a^*(i^*,j^*) - 32| = \max_{1\leq i<64, 1\leq j<16}|S_a(i,j) - 32|.$$

Тогда уравнение

$$(\vec{Y},\vec{j^*}) = (\vec{X} \oplus \vec{K},\vec{i^*})$$

является ЭЛСА a-го S-бокса в классе всех ЛСА указанного вида с вероятностью

$$p_a = \frac{S_a^*(i^*,j^*)}{64}.$$

14. Дифференциальный криптоанализ блочных шифров.

Пусть есть схема блочного шифра, состоящая из r блоков длины N, где выход одного блока соединяется с входом другого, ключи $\vec{Z} = (Z(1), \dots Z(r))$ получаются по некоторой схеме из Z_0 или выбираются независимо равновероятно. Пусть $X(1), X^*(1)$ – пара ОТ, $Y(i), Y^*(i)$ – соответствующие им ШТ на i-том цикле,

$$\Delta \mathrm{X}(1) = X(1) - X^*(1)$$

$$\Delta Y(i) = Y(i) - Y^*(i)$$

Идея дифференциального криптоанализа заключается в том, чтобы найти такие $\Delta\mathrm{X}(1)$, что при случайном равновероятном выборе $\mathrm{X}(1), Z(1), \dots, Z(r-1)$ с вероятностью более $\frac{1}{2^N}$ появится $\Delta Y(r-1)$.

Преобразование f называется **криптографически слабым**, если по $\Delta Y(r-1), Y(r)$ и $Y^*(r)$ для некоторого (малого) числа пар $(\mathrm{X}(1), \mathrm{X}^*(1))$ можно найти (хотя бы часть) $Z(r)$.

Пара (α, β) возможных значений вектора $(\Delta X(1), \Delta Y(i))$ называется **дифференциалом i-го цикла**.

Пусть f определяет операции в Δ и f криптографически слаба. Тогда возможна следующая атака.

1. Выбираем дифференциал $(r-1)$-го цикла (α, β), для которого вероятность $\mathrm{P}(\Delta Y(r-1) = \beta \mid \Delta X(1) = \alpha)$ большая.
2. Случайно выбираем $X(1)$ и подбираем $X^*(1)$, чтобы $\Delta X(1) = \alpha$. Пусть известны $Y(r)$ и $Y^*(r)$.
3. Делаем предположение, что $\Delta Y(r-1) = \beta$ и, зная $Y(r)$ и $Y^*(r)$, находим $Z(r)$.
4. Повторяем 2 и 3, пока один (частичный) ключ не начнет появляться чаще других. Это и будет $Z(r)$.
5. Повторяем 1-4 до нахождения полного ключа.

15. Метод коллизий для хэш-функций (теорема с доказательством).

Пусть $A = \{a_1, \dots, a_m\}$ – алфавит, A^* - множество слов конечной длины в алфавите A. Пусть $H: \ A^* \leftarrow A^l$ – хэш-функция. $N = |A^l|$.

Используется задача о днях рождения. Пусть у злоумышленника есть два сообщения: M – то, которое жертва подпишет, и M' – то, которое злоумышленнику нужно подписать. Варьируя стилем, шрифтом, интервалами и т.д. получаем n различных вариантов каждого из сообщений с сохранением смысла. Затем, просматривая пары, злумышленник ищет совпадение:

$$H(M_i) = H(M'_j),\ i = \overline{1,n},\ j = \overline{1,n}$$

Теорема. Пусть $N, n \to \infty$, но $\frac{n^2}{N} \to t > 0$, тогда:

$$p = (1 - e^{-t})(1 + o(1)).$$

◩ Найдём вероятность того, что при наборе из n таких пар не окажется ни одного совпадения. Мы выбираем n хэшей $H(M_i)$ из множества A^l (C_N^n вариантов). Поскольку нет ни одного совпадения, то по правой стороне выбираем n хэшей из множества $A^l \setminus \{H(M_1), \ldots, H(M_n)\}$ (C_{N-n}^n вариантов). Всего возможных вариантов выбора $(C_N^n)^2$. Таким образом,

$$1 - p = \frac{C_N^n C_{N-n}^n}{(C_N^n)^2} = \frac{(N-n)!n!(N-n)!}{n!(N-2n)!N!} = \frac{[(N-n)!]^2}{N!(N-2n)!}.$$

Используем формулу Стирлинга:

$$1 - p = \frac{[(N-n)!]^2}{N!(N-2n)!} =$$

$$= \frac{\left[\left(\frac{N-n}{e}\right)^{N-n}\sqrt{2\pi(N-n)}\right]^2 (1+o(1))}{\left(\frac{N}{e}\right)^N \sqrt{2\pi N}\left(\frac{N-2n}{e}\right)^{N-2n}\sqrt{2\pi(N-2n)}(1+o(1))} =$$

$$= \frac{(1-\frac{n}{N})^{2N-2n}}{(1-\frac{2n}{N})^{N-2n}}(1+o(1))$$

Отсюда, используя разложение логарифма в ряд:

$$\ln(1-p) = [(2N-2n)\ln(1-\frac{n}{N}) - (N-2n)\ln(1-\frac{2n}{N})](1+o(1)) =$$

$$= [(2N-2n)(-\frac{n}{N} - \frac{n^2}{2N^2} + O(\frac{n^3}{N^3})) - (N-2n)(-\frac{2n}{N} - \frac{2n^2}{N^2} + O(\frac{n^3}{N^3}))](1+o(1)) =$$

$$= -\frac{n^2}{N}(1+o(1)) = -t(1+o(1))$$

Следовательно,

$$1 - p = e^{-t}(1+o(1)),$$

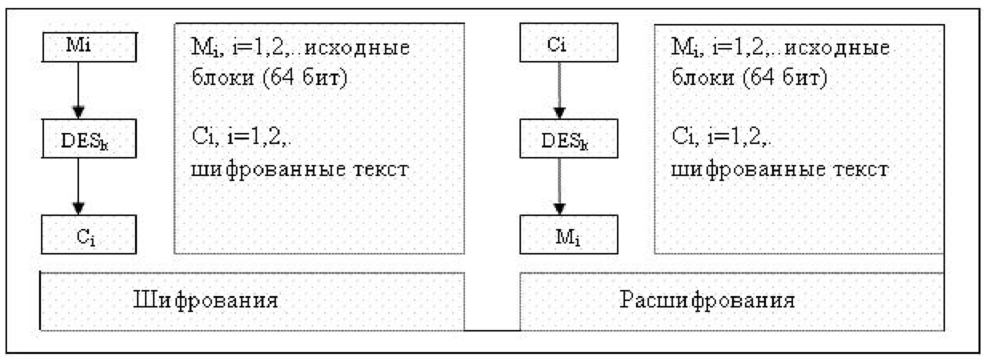
$$p = (1 - e^{-t})(1 + o(1)).$$

◪

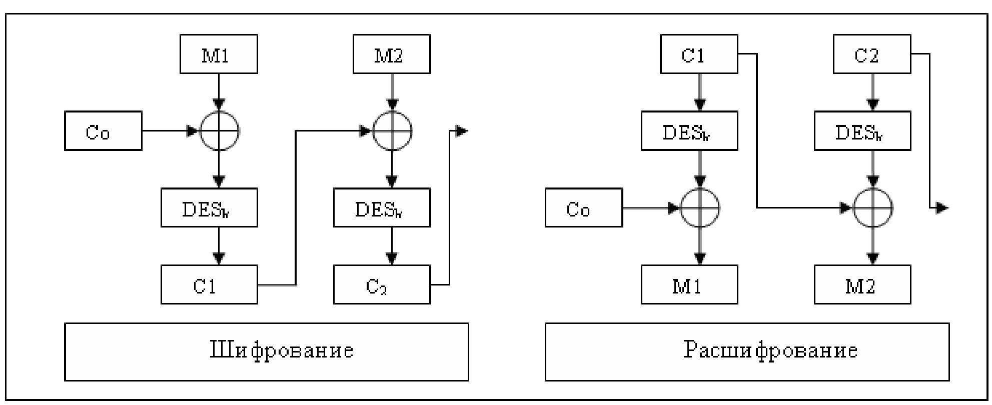
16-20. Атаки на тройной DES.

Режимы использования DES.

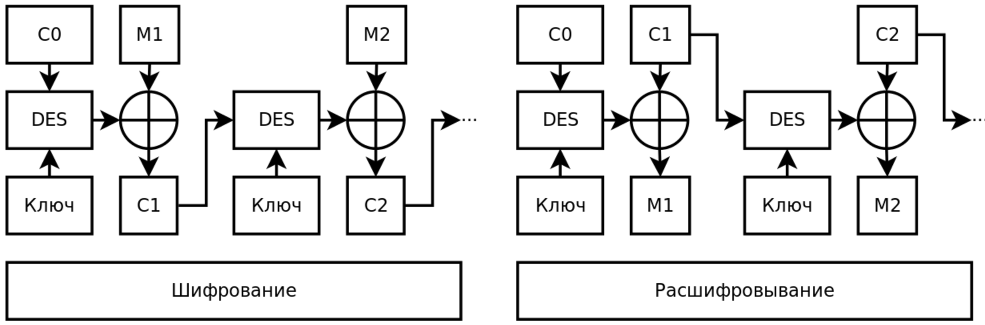
Режим электронной кодовой книги (ECB – Electronic Codebook):



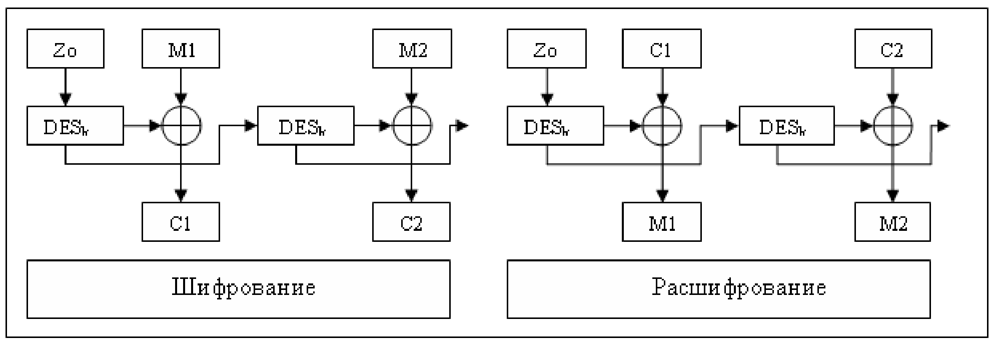
Режим сцепления блоков шифротекста (CBC – Cipher Block Chaining):



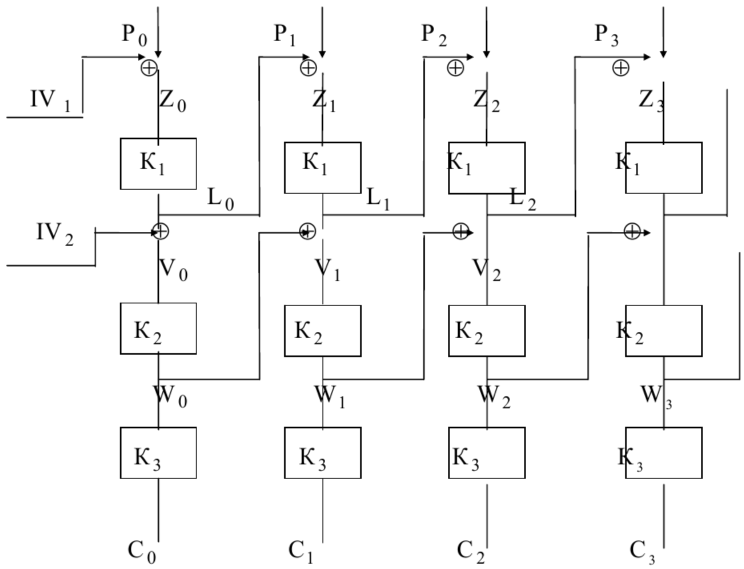
Режим обратной связи по шифротексту (CFB – Cipher Feedback):



Режим обратной связи по выходу (OFB – Output Feedback):



16. Атака на тройной DES в режиме CBC/CBC/ECB.



Используется атака выбранного шифротекста. Сначала будем искать ключ K_3. Среди всех ШТ выберем две тройки блоков следующего вида: (C_0, C_1, C_2) и (C_0^*, C_1, C_2), где $C_0 \neq C_0^*$.

Для первой тройки обозначим за Z_i, V_i, W_i – входы первого, второго и третьего блоков соответственно, L_i – выход первого блока, P_i – ОТ. Те же обозначения для второй тройки со звездочкой: $Z_i^*, V_i^*, W_i^*, L_i^*, P_i^*$.

Так как $C_1 = C_1^*, C_2 = C_2^*$, то $W_1 = W_1^*, W_2 = W_2^*, V_1 = V_1^*, V_2 = V_2^*$. Тогда $L_2 = L_2^*$ и $Z_2 = Z_2^*$. Видно, что

$$W_0 \oplus L_1 = V_1,\ W_0^* \oplus L_1^* = V_1^* \Rightarrow W_0 \oplus W_0^* = L_1 \oplus L_1^*$$

$$P_2 \oplus L_1 = Z_2,\ P_2^* \oplus L_1^* = Z_2^* \Rightarrow P_2 \oplus P_2^* = L_1 \oplus L_1^*$$

Окончательно получим:

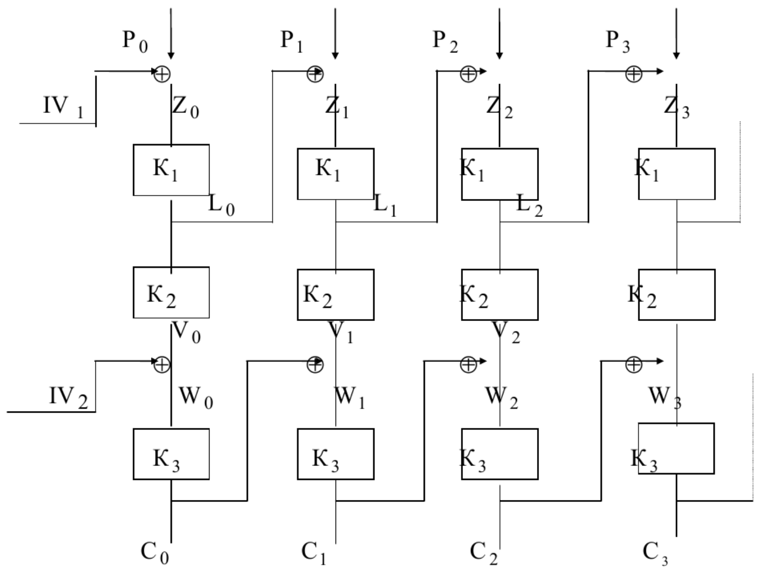
$$P_2 \oplus P_2^* = W_0 \oplus W_0^* = DES_{K_3}^{-1}(C_0) \oplus DES_{K_3}^{-1}(C_0^*)$$

Поскольку мы выбрали ШТ (а значит, знаем ОТ), мы можем совершить 2^{55} опробований в среднем, чтобы получить K_3.

Вероятность найти $C_1 = C_1^*, C_2 = C_2^*$ равна $\frac{1}{2^{128}}$. А значит, по задаче о днях рождения вероятность хоть какой-нибудь пары $\sqrt{\frac{1}{2^{128}}} = \frac{1}{2^{64}}$. Тогда потребуется 2^{64} блоков ШТ.

Для извлечения всего ключа требуется $3*2^{55}$ операций опробования и $3*2^{64}$ блоков ШТ.

17. Атака на тройной DES в режиме CBC/ECB/CBC.



Используется линейный криптоанализ. Пусть найдено достаточно много шифртекстов (C_0, C_1, C_2), где C_1 и C_2 фиксированы, а C_0 произвольные и различные.

1. Ищем K_2. Можно заметить, что $W_1 = C_0 \oplus V_1,\ V_1 = DES_{K_2}(L_1)$. Тогда $L_1 = DES^{-1}_{K_2}(C_0 \oplus W_1)$. Учитывая $L_1 \oplus P_2 = Z_2$, получим

$$P_2 = DES^{-1}_{K_2}(C_0 \oplus W_1) \oplus Z_2$$

Пусть $D*$ – ЛСА $DES^{-1}_{K_2}$. Тогда, набрав достаточно уравнений, статистически выделим решение K_2, W_1 и Z_2.

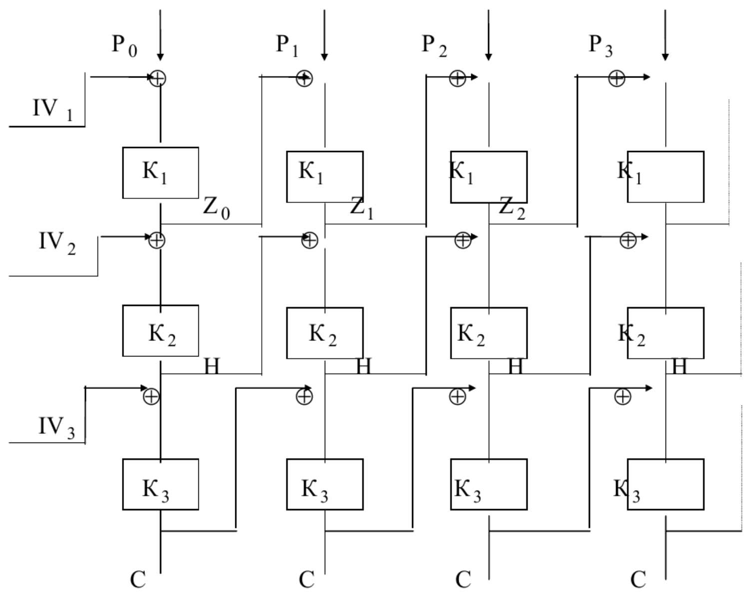
2. Ищем K_3. С помощью МПП $DES_{K_3}(W_1) = C_1$.

3. Ищем K_1. Получим L_1:

$$L_2 = DES^{-1}_{K_2}(C_1 \oplus W_2) = DES^{-1}_{K_2}(C_1 \oplus DES^{-1}_{K_3}(C_2))$$

И далее МПП $DES_{K_1}(Z_2) = L_2$.

18. Атака на тройной DES в режиме CBC/CBC/CBC.



Атака, основанная на задаче о днях рождения. Пусть найдено $2^{3}3$ шифртекстов вида (C, C, C, C). Сначала ищем K_3.

На выходе второго блока имеем $(?, H, H, H)$, где $H = C + DES^{-1}_{K_3}(C)$. H – не взаимно-однозначное отображение, тогда для одного и того же H с большой вероятностью (по задаче о днях рождения) найдутся различные C и C^*, при этом у них будет один и тот же P_3:

$$Z_2 = H \oplus DES^{-1}_{K_2}(H)$$

$$DES_{K_1}(P_3 \oplus Z_2) = H \oplus DES^{-1}_{K_2}(H)$$

Тогда

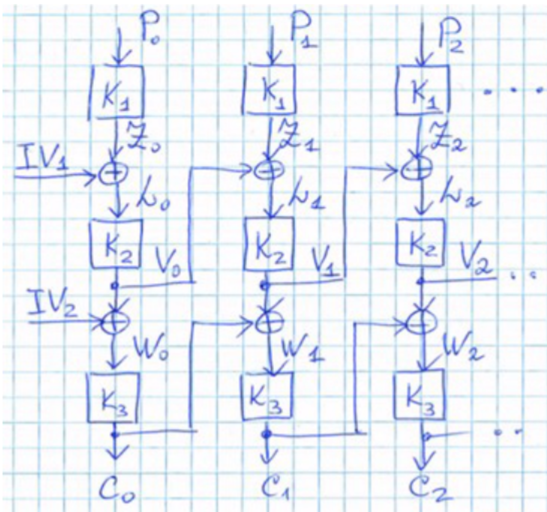
$$P_3 = DES^{-1}_{K_1}(H \oplus DES^{-1}_{K_2}(H)) \oplus DES^{-1}_{K_2}(H) \oplus H$$

Для такой пары C, C^* получим уравнение:

$$C + DES^{-1}_{K_3}(C) = C^* + DES^{-1}_{K_3}(C^*)$$

Решив его МПП, получим K_3. Аналогично остальные ключи.

19. Атака на тройной DES в режиме ECB/CBC/CBC.



Атака использует дифференциальный криптоанализ. Сначала будем искать ключ K_1. Среди всех ШТ выберем две тройки блоков следующего вида: (C_0, C_1, C_2) и (C_0^*, C_1, C_2), где $C_0 \neq C_0^*$. Пусть $\Delta = C_0 \oplus C_0^*$.

Для первой тройки обозначим за Z_i, V_i, W_i – входы первого, второго и третьего блоков соответственно, L_i – выход первого блока, P_i – ОТ. Те же обозначения для второй тройки со звездочкой: $Z_i^*, V_i^*, W_i^*, L_i^*, P_i^*$.

Так как $C_1 = C_1^*, C_2 = C_2^*$, то $W_1 = W_1^*, W_2 = W_2^*, V_2 = V_2^*, L_2 = L_2^*$. Тогда

$$V_1 = C_0 \oplus DES_{K_3}^{-1}(C_1), V_1^* = C_0 \oplus \Delta \oplus DES_{K_3}^{-1}(C_1) \Rightarrow V_1^* = V_1 \oplus \Delta$$

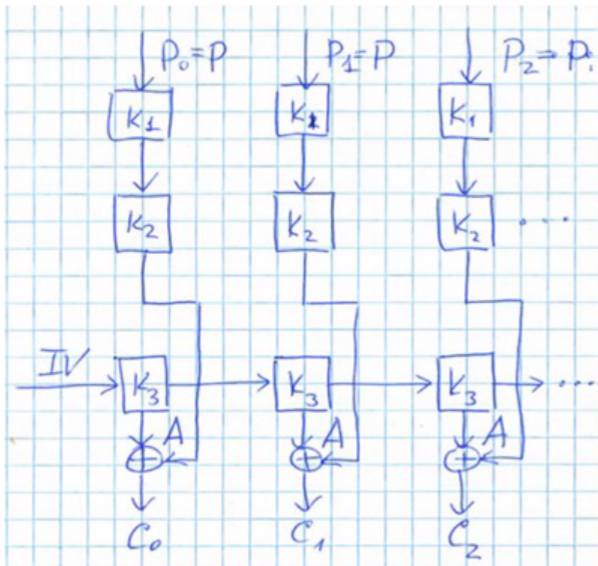
$$Z_2 = L_2 + V_1,\ Z_2^* = L_2^* + V_1^* \Rightarrow Z_2^* = Z_2 \oplus \Delta$$

Далее находим K_1, используя дифференциальный криптоанализ из

$$DES_{K_1}(P_2) \oplus DES_{K_1}(P_2^*) = C_0 \oplus C_0^*$$

Затем последовательно находим остальные ключи.

20. Атака на тройной DES в режиме ECB/ECB/OFB.



Сначала ищем ключ K_3. Выбираем произвольное $\vec{P} = (P, \ldots, P)$ из 2^{64} одинаковых блоков, пусть ему соответствует $C = (C_0, \ldots, C_{2^{64}-1})$. Период режима OFB $\leq 2^{64}$. Обозначим $A = DES_{K_2}(DES_{K_1}(P))$ и поток OFB $= v_0, \ldots, v_{2^{64}-1}$. Тогда $C_i = v_i \oplus A$. Таким образом, можно найти разности OFB-блоков:

$$C_0 \oplus C_1 = v_0 \oplus v_1, \ldots, C_{2^{64}-2} \oplus C_{2^{64}-1} = v_{2^{64}-2} \oplus v_{2^{64}-1}$$

1. Выбираем произвольное $u_0 = v_i$.

2. Перебирая K, вычисляем $u_1 = DES_K(u_0)$, $u_2 = DES_K(u_1)$. Далее находим $u_0 \oplus u_1$, $u_1 \oplus u_2$, расположенные последовательно, в указанном выше ряде. Если такой пары нет, то либо $K \neq K_3$, либо u_0 не принадлежит периоду OFB. Перебрав все K, но не найдя такой пары, меняем u_0. Ожидается, что опробований u_0 будет 2^{64}/порядок OFB.

3. Как только нашли такие u_0 и K: $K_3 = K$, $v_i = u_0$, $v_{i+1} = u_1$. Получаем весь цикл OFB, что даёт нам возможность атаковать двойной DES в режиме ECB методом "встречи посередине".

Сложность атаки: Кол-во ОТ/кол-во шагов/память $= 2^{64}/2^{58}/2^{56}$.