**Лабораторная работа №8**

**Тема:** Блочное шифрование на основе сети Фейстеля.

**Цель работы:** Исследование структуры алгоритма и методики практической реализации блочного шифра Twofish.

1. **Основные теоретические положения**

Twofish — симметричный алгоритм блочного шифрования с размером блока 128 бит и длиной ключа до 256 бит. Число раундов 16. Разработан группой специалистов во главе с Брюсом Шнайером. Являлся одним из пяти финалистов второго этапа конкурса AES. Отличительными особенностями алгоритма являются использование предварительно вычисляемых и зависящих от ключа узлов замены и сложная схема развёртки подключей шифрования. Половина n-битного ключа шифрования используется как собственно ключ шифрования, другая — для модификации алгоритма (от неё зависят узлы замены).

Алгоритм Twofish возник в результате попытки модифицировать алгоритм Blowfish для 128-битового входного блока. Новый алгоритм должен был быть легко реализуемым аппаратно (в том числе использовать таблицы меньшего размера), иметь более совершенную систему расширения ключа (key schedule) и иметь однозначную функцию F.

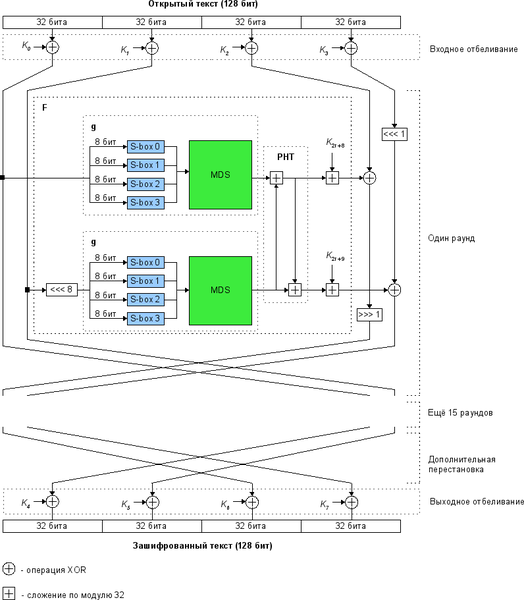
В результате, алгоритм был реализован в виде смешанной сети Фейстеля с четырьмя ветвями, которые модифицируют друг друга с использованием криптопреобразования Адамара (Pseudo-Hadamar Transform, PHT).

Возможность эффективной реализации на современных (для того времени) 32-битных процессорах (а также в смарт-картах и подобных устройствах) — один из ключевых принципов, которым руководствовались разработчики Twofish.

Например, в функции F при вычислении PHT и сложении с частью ключа K намеренно используется сложение, вместо традиционного XOR. Это дает возможность использовать команду LEA семейства процессоров Pentium, которая за один такт позволяет вычислить преобразование Адамара.

Алгоритм Twofish не запатентован и может быть использован кем угодно без какой-либо платы или отчислений. Он используется во многих программах шифрования, хотя и получил меньшее распространение, чем Blowfish.

1. **Схема алгоритма шифрования данных Twofish**



Рассмотрим более подробно алгоритм формирования раундовых подключей, а также зависящей от ключа функции g. Как для формирования подключей, так и для формирования функции g в Twofish используется одна основная функция:

h(X, L0, L1, … , Lk).

Здесь X, L0, L1, … , Lk — 32 битные слова, а k = N / 64, где N — длина исходного ключа в битах. Результатом функции является одно 32-битное слово.

**Функция h**

Функция выполняется в k этапов. То есть для длины ключа 256 бит будет 4 этапа, для ключа в 192 бита — 3 этапа, для 128 бит — 2 этапа. На каждом этапе входное 32-битное слово разбивается на 4 байта, и каждый байт подвергается фиксированной перестановке битов q0 или q1. Результат представляется в виде вектора из 4 байтов и умножается на MDS матрицу. Вычисления производятся в поле Галуа GF(28) по модулю неприводимого многочлена x^8 + x^6 + x^5 + x^3 + 1.

Матрица MDS имеет вид:

~\mbox{MDS} = \begin{pmatrix} 
01 & EF & 5B & 5B \\ 
5B & EF & EF & 01 \\ 
EF & 5B & 01 & EF \\
EF & 01 & EF & 5B
\end{pmatrix}

**Перестановки q0 и q1**

q0 и q1 — фиксированные перестановки 8 битов входного байта x.

Байт x разбивается на две 4-битные половинки a0 и b0, над которыми проводятся следующие преобразования:

~a_0 = x/16            ~b_0 = x\bmod{16}

~a_1 = a_0 \oplus b_0            ~b_1 = a_0 \oplus \mbox{ROR}_4(b_0,1) \oplus 8a_0 \bmod{16}

~a_2 = t_0[a_1]            ~b_2 = t_1[b_1]

~a_3 = a_2 \oplus b_2            ~b_3 = a_2 \oplus \mbox{ROR}_4(b_2,1) \oplus 8a_2 \bmod{16}

~a_4 = t_2[a_3]            ~b_4 = t_3[b_3]

~y = 16b_4 + a_4

Здесь ~\mbox{ROR}_4 — 4-битный циклический сдвиг вправо, а t0, t1, t2, t3 — табличные замены 4-битных чисел.

Для q0 таблицы имеют вид:

t0 = [ 8 1 7 D 6 F 3 2 0 B 5 9 E C A 4 ]

t1 = [ E C B 8 1 2 3 5 F 4 A 6 7 0 9 D ]

t2 = [ B A 5 E 6 D 9 0 C 8 F 3 2 4 7 1 ]

t3 = [ D 7 F 4 1 2 6 E 9 B 3 0 8 5 C A ]

Для q1 таблицы имеют вид:

t0 = [ 2 8 B D F 7 6 E 3 1 9 4 0 A C 5 ]

t1 = [ 1 E 2 B 4 C 3 7 6 D A 5 F 9 0 8 ]

t2 = [ 4 C 7 5 1 6 9 A 0 E D 8 2 B 3 F ]

t3 = [ B 9 5 1 C 3 D E 6 4 7 F 2 0 8 A ]

**Генерация ключей**

Пусть M — исходный ключ, N — его длина в битах. Генерация подключей происходит следующим образом:

* Исходный ключ разбивается на 8\*k байтов m_0,...,m_{8k-1}, где k = N / 64.
* Эти 8\*k байтов разбиваются на слова по четыре байта, и в каждом слове байты переставляются в обратном порядке. В итоге получается 2\*k 32-битных слова M_i

~M_i=\sum^3_{j=0}m_{(4i+j)}\cdot2^{8j}~~~~~i=0,...,2k-1

* Полученные 2\*k 32-битных слов разбиваются на два вектора ~M_eи ~M_oразмером в k 32-битных слов.

~M_e = (M_0, M_2, ... , M_{2k-2})

~M_o = (M_1, M_3, ... , M_{2k-1})

Подключи для i-го раунда вычисляются по формулам:

~\rho=2^{24} + 2^{16} + 2^8 + 2^0

~A_i = h(2i\rho, M_e)

~B_i = \mbox{ROL}(h((2i+1)\rho, M_0), 8)

~K_{2i} = (A_i + B_i)\bmod{2^{32}}

~K_{2i+1} = \mbox{ROL}((A_i+2B_i)\bmod{2^{32}},9)

**Функция g и S-box’ы**

Функция g определяется через функцию h: ~g(X) = h(X,S)

Вектор S, как и вектора Me и Mo, тоже формируется из исходного ключа и состоит из k 32-битных слов. Исходные байты ключа разбиваются на k групп по восемь байт. Каждая такая группа рассматривается как вектор с 8-ю компонентами и умножается на фиксированную RS матрицу размером 4x8 байт. В результате умножения получается вектор, состоящий из четырех байт. Вычисления проводятся в поле Галуа GF(2^8)по модулю неприводимого многочлена x^8 + x^6 + x^3 + x^2 + 1.

RS-матрица имеет вид:

~\mbox{RS} = \begin{pmatrix} 
01 & A4 & 55 & 87 & 5A & 58 & DB & 9E \\ 
A4 & 56 & 82 & F3 & 1E & C6 & 68 & E5 \\
02 & A1 & FC & C1 & 47 & AE & 3D & 19 \\
A4 & 55 & 87 & 5A & 58 & DB & 9E & 03
\end{pmatrix}

**3. Содержание отчёта**

3.1. Составить блок-схему алгоритма шифрования Twofish.

3.2. Листинг программы шифрования и дешифрования заданного сообщения с использованием алгоритма Twofish.

3.3. Выводы.