Задание по практикуму на ЭВМ
2010-2011 год, первый семестр

13 сентября 2010 г.

1. Общая формулировка задания.

Требуется реализовать криптографический калькулятор. На вход криптографического калькулятора подаётся файл, пароль и идентификатор режима работы — либо "e", либо "d". Из пароля по определённому правилу получается ключ шифрования. После выработки ключа шифрования калькулятор должен в зависимости от режима работы либо зашифровать файл на ключе (режим "e"), либо расшифровать файл (режим "d"). Шифрование/расшифрование должно производиться некоторым шифром, построенным по архитектуре шифра Фейстеля. Шифр задаётся в конфигурационном файле.

Пример. *Пусть программа имеет название cryptocalc.exe. Схема работы программы должна быть следующей:*

cryptocalc.exe -f filename.abc -p password -e $\rightarrow$ *filename.abc.enc*

cryptocalc.exe -f filename.abc.enc -p password -d $\rightarrow$ *filename.abc*

2. Сеть Файстеля.

Сетью Файстеля называется один из методов построения блоковых шифров. Как правило сначала выбирается длина блока данных, которое будет преобразоваться за один раз. В шифрах DES и ГОСТ блок данных равен 64 битам. Далее блок данных делится на две, обычно равные, части. В шифрах DES и ГОСТ блок делится на две равные 32 битные части. Далее левая часть данных не изменяется, а правая подвергается преобразованию. Далее левая и правая половина меняются местами и всё повторяется заново.

Математически сеть Файстеля можно записать следующие системой.

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus F_i(R_{i-1}, k_i) \end{cases}, i = 1, 2, \ldots, r$$

При этом предполагается, что L_0R_0 — блок открытого текста, разделённый на две половины — L_0и R_0, а k_i — раундовый ключ, который

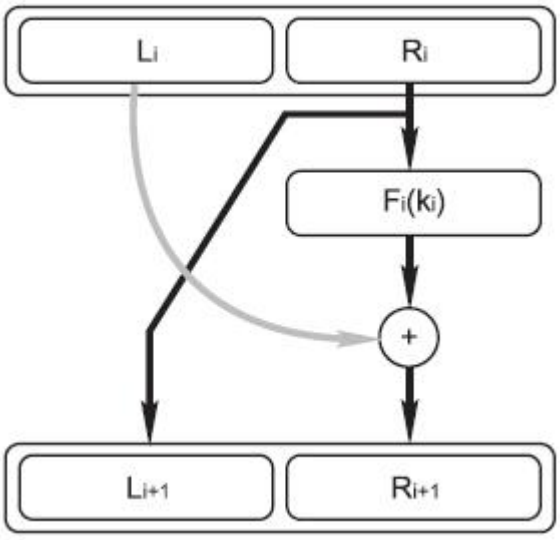

Рис. 1.

получается из исходного ключа K.

При построении функции F_i используются следующие функции: $\oplus$ — сложение по модулю два двух векторов, $>>(x, n)$ — циклический сдвиг двоичного вектора x на n битов вправо, $P(x)$ — перестановка битов вектора x, $E(x)$ — перестановка битов вектора x с раширением, то есть на выходе вектор имеет большую длину, чем вектор x, $S(x)$ — нелинейное преобразование, применяемое к вектору x, так накзываемый узел замены.

Достоинства сети Файстеля состоит в том, что какой бы ни было преобразование F_i сеть Файстеля обратима всегда!!! Обратное преобразование задаётся следующими соотношениями:

$$\begin{cases} R_{i-1} = L_i \\ L_{i-1} = R_i \oplus F_i(R_i, k_i) \end{cases}, i = r, r-1, \ldots, 1$$

3. Конфигурационный файл программы

Опишем структуру конфигурационного файла программы. Предполагается, что блок данных, передаваемый в сеть Файстеля далее делится на правую и левую части поровну.

r = 16 — указывается число раундов в сети Фейстеля

n = 64 — указывается длина блока шифрования (должно быть всегда чётным)

k = 64 — указывается длина ключа шифрования

F = P1(S1(E(r)+P2(k),0,5)|S2(E(r)+P2(k),6,11)|S3(E(r)+P2(k),12,17)|
S4(E(r)+P2(k),18,23)|S5(E(r)+P2(k),24,29)|S6(E(r)+P2(k),30,35)|
S7(E(r)+P2(k),36,41)|S8(E(r)+P2(k),42,47)) — указывается суперпозиция функций

S1={14 4 13 1 2 15 11 8 3 10 6 12 5 9 0 7 0 15 7 4 14 2 13 1 10 6 12 11 9 5 3 8 4 1 14 8 13 6 2 11 15 12 9 7 3 10 5 0 15 12 8 2 4 9 1 7 5 11 3 14 10 0 6 13} — указывается каждый узел замены в виде списка.

S2 ={15 1 8 14 6 11 3 4 9 7 2 13 12 0 5 10 3 13 4 7 15 2 8 14 12 0 1 10 6 9 11 5 0 14 7 11 10 4 13 1 5 8 12 6 9 3 2 15 13 8 10 1 3 15 4 2 11 6 7 12 0 5 14 9}

S3 ={10 0 9 14 6 3 15 5 1 13 12 7 11 4 2 8 13 7 0 9 3 4 6 10 2 8 5 14 12 11 15 1 13 6 4 9 8 15 3 0 11 1 2 12 5 10 14 7 1 10 13 0 6 9 8 7 4 15 14 3 11 5 2 12}

S4 ={7 13 14 3 0 6 9 10 1 2 8 5 11 12 4 15 13 8 11 5 6 15 0 3 4 7 2 12 1 10 14 9 10 6 9 0 12 11 7 13 15 1 3 14 5 2 8 4 3 15 0 6 10 1 13 8 9 4 5 11 12 7 2 14}

S5 ={2 12 4 1 7 10 11 6 8 5 3 15 13 0 14 9 14 11 2 12 4 7 13 1 5 0 15 10 3 9 8 6 4 2 1 11 10 13 7 8 15 9 12 5 6 3 0 14 11 8 12 7 1 14 2 13 6 15 0 9 10 4 5 3}

S6 ={12 1 10 15 9 2 6 8 0 13 3 4 14 7 5 11 10 15 4 2 7 12 9 5 6 1 13 14 0 11 3 8 9 14 15 5 2 8 12 3 7 0 4 10 1 13 11 6 4 3 2 12 9 5 15 10 11 14 1 7 6 0 8 13}

S7 ={4 11 2 14 15 0 8 13 3 12 9 7 5 10 6 1 13 0 11 7 4 9 1 10 14 3 5 12 2 15 8 6 1 4 11 13 12 3 7 14 10 15 6 8 0 5 9 2 6 11 13 8 1 4 10 7 9 5 0 15 14 2 3 12}

S8 ={13 2 8 4 6 15 11 1 10 9 3 14 5 0 12 7 1 15 13 8 10 3 7 4 12 5 6 11 0 14 9 2 7 11 4 1 9 12 14 2 0 6 10 13 15 3 5 8 2 1 14 7 4 10 8 13 15 12 9 0 3 5 6 11}

E = {32 1 2 3 4 5 4 5 6 7 8 9 8 9 10 11 12 13 12 13 14 15 16 17 16 17 18 19

20 21 20 21 22 23 24 25 24 25 26 27 28 29 28 29 30 31 32 1} — указывается перестановка с расширением в виде списка.

P1 = {16 7 20 21 29 12 28 17 1 15 23 26 5 18 31 10 2 8 24 14 32 27 3 9 19 13 30 6 22 11 4 25} — указывается перестановка в виде списка.

P2 = {14 17 11 24 1 5 3 28 15 6 21 10 23 19 12 4 26 8 16 7 27 20 13 2 41 52 31 37 47 55 30 40 51 45 33 48 44 49 39 56 34 53 46 42 50 36 29 32}

K = (>> (P3(k),I[i],0,31)) | (>>(P3(k),I[i],32,63)) - указывается преобразование выработки раундового ключа.

I = {1,2,4,6,8,10,12,14,15,17,19,21,23,25,27,28} - раундовые константы.

4. Описание функций и резервных слов

В конфигурационными словами являются буквы F, K, R и i. Остальные слова - переменные и функции. Слово F — основное преобразование в раунде сети Файстеля, K — функция выработки раундовых ключей, R — правая половина входного слова в раунде, а i — номер раунда.

a+b - сложение по модулю 2 векторов.

a#b - сложение по модулю 2^{32}двух чисел.

>>(x,i,n0,n1) - циклический сдвиг вправо отрезка [n0,n1] битов в векторе x.

a|b - объединение (конкатенация) векторов.

P(x) - применение к вектору x перестановки битов. Функция P задаётся в виде списка. При этом бит с номером i в векторе x переходит на место бита с номером P[i].

Example 4.1. Пусть P - P1 из конфигурационного файла. Тогда бит с номером 1 перейдёт на место бита с номером 16, бит номер 2 - на место бита с номером 7 и так далее.

S(x,n0,n1) - применение к вектору x узла замены S. Узел замены задаётся в виде списка. Применение происходит следующим образом. Из вектора x биты на отрезке [n0,n1] извлекаются в новый вектор y длины n1-n0+1. Этот двоичный вектор является двоичным представлением

некоторого числа |y|. Выходом S(x,n0,n1) является двоичное представление числа S[|y|].

Example 4.2. Пусть S - S1 из конфигурационного файла. Пусть

$x = 10101000010010001101001111100010$

Вычислим S(x,0,5). Образуем новый вектор $y = x_0x_1x_2x_3x_4x_5 = 101010$. Тогда $|y| = 42$, S[42]=9=1001, поэтому S(x,0,5)=1001.

E(x) - применение к вектору x перестановки битов с расширением. Функция E задаётся в виде списка. Действие E аналогично действию перестановки P за тем исключением, что в E могут быть повторение битов, то есть биты могут дублироваться (см., например, E в конфигурационном файле).

I - массив констант. Задаётся в виде списка. Доступ осуществляется по индексу - I[i].

Кроме того допустимо вводить несколько функции F (F<10>, F<1-9>) и несколько функций K (K<1>, K<2-7>). При этом запись F означает функцию применяемую по умолчанию. Запись F<n> означает, что F применяется только в раунде с номером n, а запись F<m-n> означает, что функция будет выполняться в раундах с m по n включительно. Аналогично и для функции K. Правило выбора функции в каждом раунде следующее: если для раунда n имеется функция явно назначенная этому раунду (F<n>), то выполняется именно эта функция, если такой функции нет, но есть такаю функция F<m0-m1>, что n принадлежит отрезку [m0,m1], то выполняется эта функция, во всех оставльных случаях выполняется функция по умолчанию. Такое же правило следует использовать и при выборе функций K.