Немного про блочные шифры

ОПР(Блочный шифр)

Блочный шифр это криптосистема $(\{0,1\}^n, \{0,1\}^k, \{0,1\}^n, E, D)$ где:

- $\mathcal{M} = \{0,1\}^n$
- $\mathcal{K} = \{0,1\}^k$
- $\mathcal{C} = \{0,1\}^n$
- n - длина блока
- k - длина ключа

Идея применять к маленьким кусочкам открытого текста сложные функции(которые нужно задать таблицей)

Затем перемешаем эти блоки(например с помощью линейного преобразования либо другая простая функция). Действуем этой функцией на весь большой блок открытого текста

Итеративная схема блочного шифра

Есть

- $f : \{0,1\}^n >\!\!\twoheadrightarrow \{0,1\}^n$ - сложное, локальное преобразование
- $g : \{0,1\}^n >\!\!\twoheadrightarrow \{0,1\}^n$ - простое, глобальное преобразование
- $h : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ берёт че-то и ключ и возвращает че-то другое
- $h_k = h(\_, k)$ **т.е в преобразование закладываем ключ k**

По итогу получаем формулу для криптограммы

$$c = (h_k \circ f \circ g)^r(m)$$

- r - это число раундов

Конструкция фейстеля

Открытый текст разобъем его на 2 части(n - длина открытого текста - четное число)

$m = L_0 R_0$, где:

- $L_0, R_0 \in \{0,1\}^{\frac{n}{2}}$

теперь преобразовываем эти полублоки

$\forall i = \{1, \cdots, r\}$:

- $L_i = R_{i-1}$
- $R_i = L_{i-1} \bigoplus f(R_{i-1}, k_i)$
  - k_i - раундовый ключ, как-то получается из основного ключа

Проделываем процедуру и в конце получаем L_r, R_r.

Формула для криптограмы c:

$$c = R_r L_r$$

Как расшифровывать криптограмму c?

пусть $c = u_0 v_0$

$\forall i \in \{1, \cdots, r\}$

- $u_i = v_{i-1}$
- $v_i = u_{i-1} \bigoplus f(v_{i-1}, k_{r+1-i})$

Проделываем процедуру и получаем v_r, u_r

Тогда формула для открытого текста это

$$m = v_r u_r$$

Д-ВО

С помощью индукции по i нужно показать, что:

- $u_i = R_{r-i}$
- $v_i = Lr - i$

Б.И

$i = 0 : \begin{cases} u_0 = R_r \\ v_0 = L_r \end{cases}$

Ш.И от $(i-1) \to i$

- по опр конструкции Фейстеля $L_i = R_{i-1}$

$u_i = v_{i-1} = [\text{П.И}] = L_{r-i+1} = R_{r-i}$

$v_i = u_{i-1} \bigoplus f(v_{i-1}, k_{r+1-i})$

- **по П.И:**
  - $u_{i-1} = R_{r-i+1} =$
  - $v_{i-1} = L_{r-i+1} = R_{r-i}$
  - подставим

$v_i = R_{i-i+1} \bigoplus f(R_{r-i}, k_{r+1-i})$

из $R_i = L_{i-1} \bigoplus f(R_{i-1}, k_i) \Rightarrow$

$L_{i-1} = R_i \bigoplus f(R_{i-1}, k_i) \Rightarrow$

По итогу получаем, что

$L_{r-i} = R_{r-i+1} + (R_{r-i}, k_{r-i+1})$

■

Конструкция расшифрования такая же как и шифрования, кроме порядка ключей

- **При шифровании ключи используются по возрастанию**
- **При расшифровании ключи используются по убыванию**

Нам не важно какую функцию f использовать, т.к не имеет значение её обратимость → **можем выбрать сколь угодно сложную функцию**

DES(Data Encryption Standard)

Блочный шифр это конструкция, которая позволяет реализовать ШПЗ на большом алфавите

- Блок - 64 бит
  - это длина буквы в алфавите(в битах)
- Ключ - 56 бит

Устройство ключа

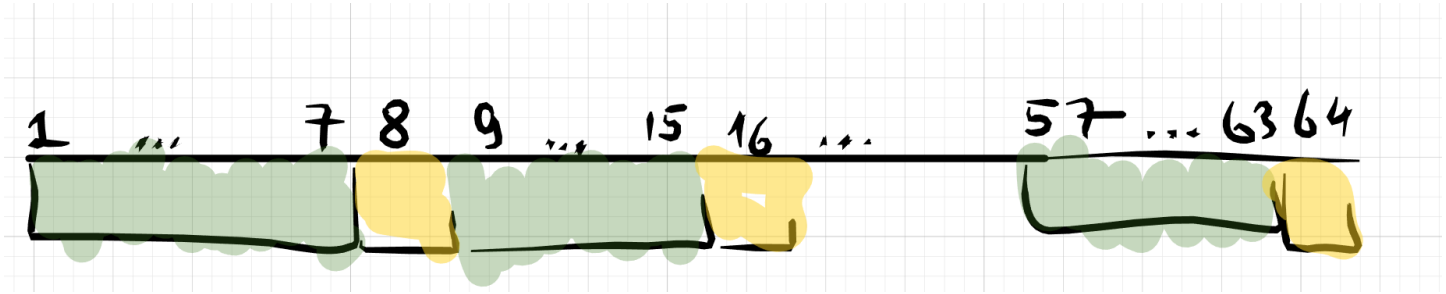

Рис. 1: alt text

- биты с 1 по 7, с 9 по 15, с 17 по 23 ... с 57 по 63 были **ключевыми**(зеленые)
- 8,16,24, ..., 64 биты были **проверками на нечетность**(сумма іого байта должна быть 1 по модулю 2) (желтые)

Построение раундовых ключей

Берётся 56 битов ключа и сначала переставляется 2 блока$({}_0, D_0)$ по 28 бит

Надо ли запоминать это разбиение на 2 таблицы

Заметь, что здесь пропущены биты проверки на нечетность

Это просто перестановка битов. Записаны они в порядке номеров, но читать их нужно по строкам. C_0 отдельно от D_0 изготавливаем $\forall n \in \{1, \cdots, 16\}$:

- $C_n = \ll_\alpha C_{n-1}$
- $D_n = \ll_\alpha D_{n-1}$

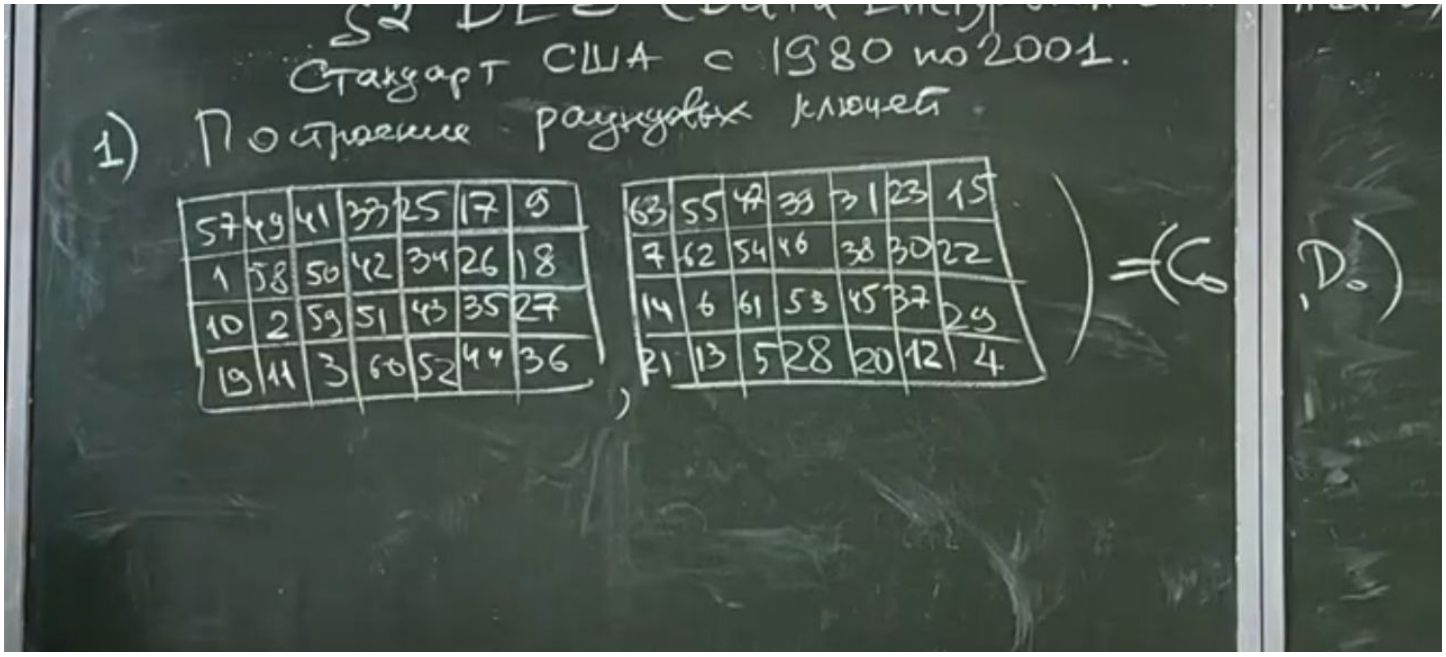

Рис. 2: разбиение на (C_0, D_0)

где $\alpha = \begin{cases} 1, \text{ при } n = 1, 2, 9, 16 \\ 2, \text{иначе} \end{cases}$

Затем из каждой пары (C_n, D_n) изготавливается ключ K_n. K_n - это следующие 48 бит строки $(C_n||D_n)$

- $(C_n||D_n)$ - сцепленные строки

По итогу получаем 16 48-битных цепочек $K_1, \cdots, K_16$, изготовленных из ключа K

Шифрование

- На входе берётся открытый текст $m \in \{0,1\}^{64}$
- На выходе выдаётся криптограмма $c \in \{0,1\}^{64}$

1. в качестве m' = P(m) - начальная перестановка
2. L_0R_0 =m' (разбили m' на 2 полублока)
3. $\forall n \in 1, \cdots, 16 : \begin{cases} L_n = R_{n-1} \\ R_n = L_{n-1} \bigoplus f(R_{n-1}, k_n) \end{cases}$
   - k_n- получен на предыдущем этапе
4. $c = P^{-1}(R_{16}L_{16})$

- это в чистом виде конструкция фейстеля, но с добавленными двумя перестановками P и P^{-1}
- Добавление в начале и в конце двух перестановок не нарушает свойства конструкции фейстеля обратимости.

Чтобы расшифровать нужно сначала получить $R_{16}L_{16} = P(c)$

Затем проделать ту же процедуру расшифрования, что и в конструкции Фейстеля

Затем получаем $m = P^{-1}(m')$

опишем p

p - это перестановка, а значит задаётся таблицей 8 на 8.

- можно заполнить только последний правый столбик, дальше строить строчки, прибавляя к числам по 8

опишем f

$f : \{0,1\}^{32} \times \{0,1\}^{48} \to \{0,1\}^{32}$

f(x,y) = z

Рис. 3: таблица для изготовления ключа K_n

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Рис. 4: перестановка p

Таблица 2. Функция расширения 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

1. происходит раздувание. $x \to x'$ по правилу
- видим, что (32,1), (4,5), (8,9), (12,13), (16,17), (20,21), (24,25), (28,29) биты используются дважды
2. $t = x' \bigoplus y$
3. $t_1 t_2 \cdots t_8 = t$(разбили на 8 кусочков, каждый по 6 бит)
4. $\alpha_1, \alpha_2, \cdots \alpha_6 = t_i (i \in \{1, \cdots, 8\})$
    - $\alpha_1 \alpha_6 \in \{0, 1, 2, 3\}$
    - $\alpha_2, \alpha_3, \alpha_4, \alpha_5 \in 0, \cdots, 15$
5. $r_i = S_i(\alpha_1 \alpha_6, \alpha_2 \alpha_3 \alpha_4 \alpha_5)$
    - имеется 8 таблиц значений S_i для каждого t
    -
    -
    1. каждая строка таблицы это перестановка
    2. S - нелинейная от аргументов
    3. изменение одного бита входа ведёт к изменению не менее 2 битов выхода
    4. $\forall x : S(x)$ и $S(x \bigoplus 001100)$ - эти строки отличаются не менее чем в 2 битах.
6. на выходе $z = q(r_1, r_2, \cdots, r_8)$
    - q - перестановка, задаваемая таблицей
    -

Достоинства

1. Первый опубликованный
2. Зашита конструкция фейстеля ⇒ не нужно изготавливать дополнительное расшифровывающее устройство
3. Быстрый т.к всего 16 раундов

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	

Рис. 5: таблицы S_i

0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Рис. 6: таблицы S_i

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

Рис. 7: q - перестановка, задаваемая таблицей

Недостатки

1. Очень много битовых операций
2. Слишком короткий ключ
3. Чтобы бороться с коротким ключом, то используется тройной DES