Симметричная криптография

Безопасность определяется:

- **стойкостью (криптостойкостью) алгоритма –** в малой степени,
- **длиной ключа:** *Пример*. При длине кл. 8 бит существует 2^8 ключей; при длине 56 бит – 2^{56} ключей. Если PC за 1 сек перебирает 1 млн ключей, то поиск требует 2285 лет, при 64 бит – 585000 лет; при 2048 бит – 1 млн PC, работая параллельно, потратят 10^{597} лет
- Вычислительная возможн. PC оценивается в MIPS-годах (million instructions per second) – миллион операций в 1 с за 1 год – $3 * 10^{13}$ операций

В симметричных системах Отправитель и Получатель используют один и тот же ключ, ***K***.

Пусть открытое сообщение ($M = \{m_i\}$) в двоичной форме имеет вид: $M = 10101100$.

Считаем, что выбран симметричный ключ $K_1 = K_2 = K = 1010$. Используется самая простая операция зашифрования:

$$c_i = m_i \oplus K$$

и операция расшифрования:

$$m_i = c_i \oplus K$$

Нетрудно убедиться, что сложение каждых четырех бит шифртекста с ключом восстанавливает исходное сообщение (здесь $\oplus$ - операция суммирования по модулю 2).

Краткая историческая информация и общая характеристика блочных шифров

- В **1972** г. Национальное бюро стандартов США (ныне – Национальный институт стандартов и технологии, National Institute of Standarts & Technology – NIST) инициировал программу защиты каналов связи и компьютерных данных. Одна из целей – разработка единого стандарта криптографического шифрования.
- Основными критериями оценки алгоритма являлись:

- алгоритм должен обеспечить высокий уровень защиты,
- алгоритм должен быть понятен и детально описан,
- криптостойкость алгоритма должна зависеть только от ключа,
- алгоритм должен допускать адаптацию к различным применениям,
- алгоритм должен быть разрешен для экспорта.

- В качестве начального варианта нового алгоритма рассматривался **Lucifer** – разработка компании IBM начала семидесятых годов.
- В основе указанного алгоритма использовались два запатентованных 1971 г. **Хорстом Фейстелем** (Horst Feistel) устройства, реализующие различные алгоритмы шифрования, позже получившие ***шифр (сеть) Фейстеля*** (Feistel cipher, Feistel network).
- В ноябре 1976 г. был утвержден стандарт DES (Data Encryption Standard – стандарт шифрования данных).
- В 1981 г. ANSI одобрил **DES** в качестве стандарта для публичного использования (стандарт ANSI X3.92), назвав его алгоритмом шифрования данных (Data Encryption Algorithm – DEA).
- 1987 году были разработаны алгоритмы FEAL и RC2

- Сети Фейстеля получили широкое распространение в 1990-е гг.: появление алгоритмов:

Blowfish (1993),
TEA (1994),
RC5 (1994),
CAST-128 (1996),
XTEA (1997),
XXTEA (1998),
RC6 (1998) и других.

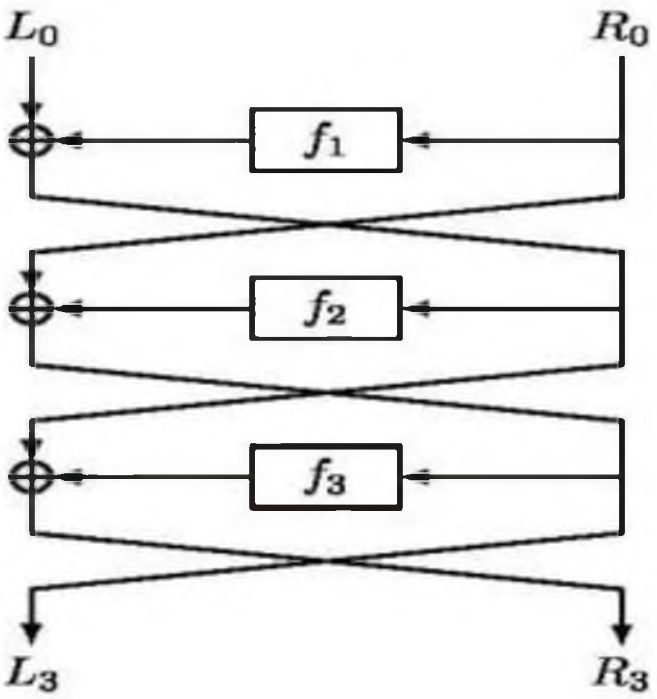
- На основе сети Фейстеля в 1990 году в СССР был принят в качестве **ГОСТ 28147-89** стандарт шифрования.
- **Блочное зашифрование (расшифрование)** - разбиение исходного открытого (зашифрованного) текста на **равные блоки**, к которым применяется однотипная процедура зашифрования (расшифрования).
- Указанная однотипность характеризуется, что **процедура зашифрования (расшифрования) состоит из совокупности повторяющихся наборов преобразований, называемых раундами**.

Сеть Фейстеля

- Название конструкции Фейстеля (сеть) означает ее ячеистую топологию.
- Формально одна ячейка сети соответствует одному раунду зашифрования или расшифрования сообщения.
- При зашифровании сообщение разбивается на блоки одинаковой (фиксированной) длины (как правило – 64 или 128 бит).
- Каждый входной блок шифруемого сообщения изначально делится на два подблока одинакового размера:
  левый (L_0) и правый (R_0).

В каждом i-ом раунде выполняются преобразования:

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} + f(R_{i-1}, K_i)$$



Показаны 3 ячейки сети

.....

Функции f_i (R_{i-1}, K_i):

блок подстановок (**S**-блок, англ. **S**-box);

блок перестановок (**P**-блок, англ. P-box)

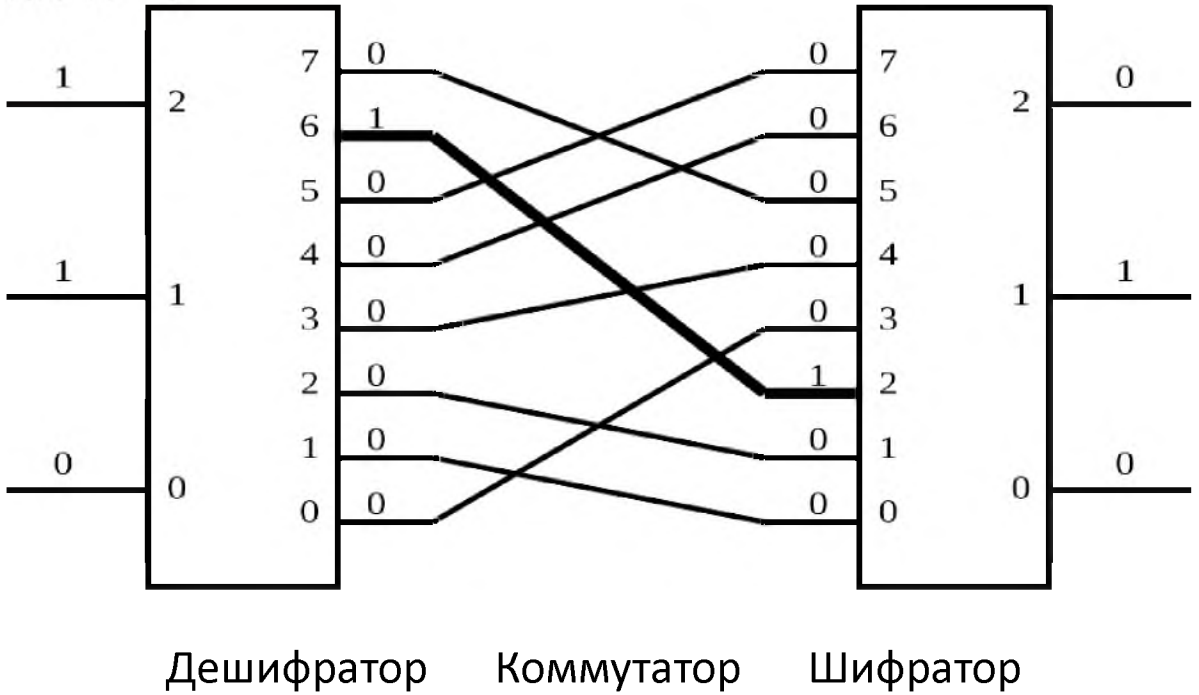
Блок подстановок состоит из:

дешифратора, преобразующего n-разрядное двоичное число в одноразрядное сигнал по основанию 2n;

внутреннего коммутатора;

шифратора, преобразующего сигнала из одноразрядного 2n-ричного в n-разрядный Двоичный

Пример для n=3



Симметричная криптография. Алгоритм **DES**

DES – Data Encryption Standard

С 23.11.1976 – национальный стандарт США, с 1981 – стандарт шифрования для частных лиц (ANSI X3.92; DEA – D. E. Algorithm)

DES – блочный шифр (64-битовый блок)

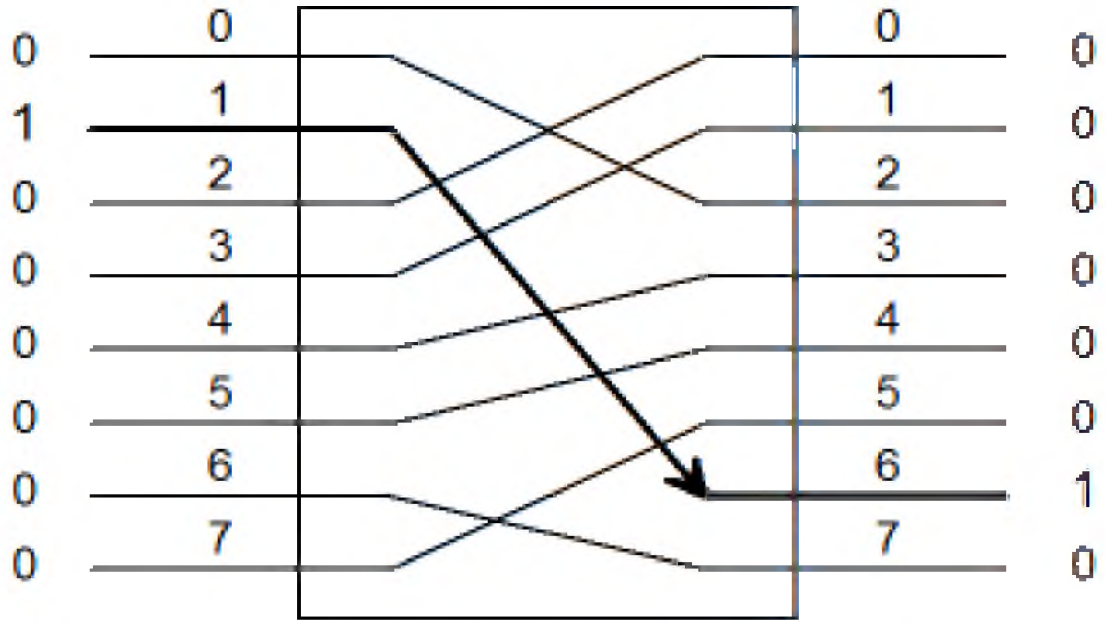
Длина кл. – 56 бит (из 64 бит – 8 – биты четности: расположены в позициях **8,16,24,32,40,48,56,64**.

Базовые методы – подстановка и перестановка данных : **1 под. + 1 перест. – раунд (цикл)**

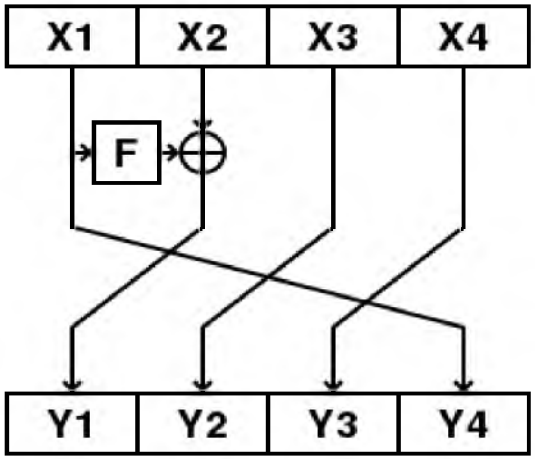
Алгоритм состоит из **16 раундов,** т.е. один блок данных дл. 64 бита обрабатывается 16 р., **в каждом из которых исп-ся новый ключ:** в кажд. р. биты ключа сдвигаются, затем из 56 бит выбир-ся 48 бит

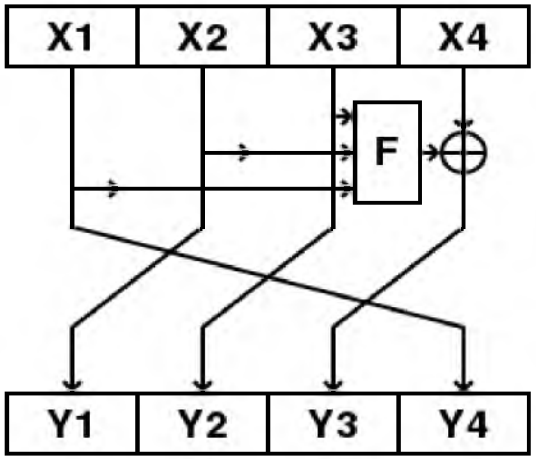
Блок перестановок изменяет положение цифр, т. е. является линейным устройством

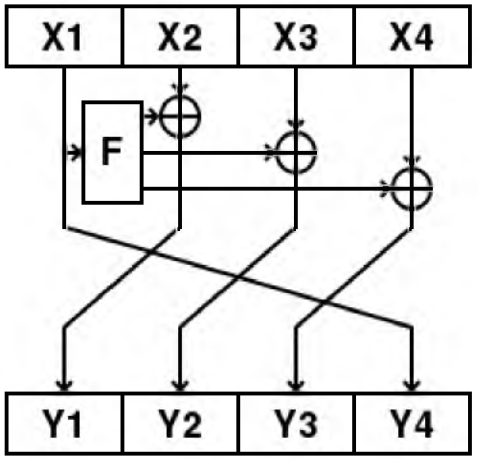
Пример блока



При большом размере блоков (128 бит и более) реализация сети Фейстеля на 32-разрядных архитектурах может вызвать затруднения, поэтому применяются модифицированные варианты сети. Такие модификации предусматривают использование не 2-х, а 4-х ветвей







Важное свойство блочного шифра — **эффект лавины** (*Avalanche effect*)

ЭЛ означает, что изменение значения малого количества битов во входном тексте или в ключе ведет к «лавинному» изменению значений выходных битов шифротекста. Другими словами, это зависимость всех (или хотя бы **половины**) выходных битов от каждого входного бита.

АЛГОРИТМИЧЕСКАЯ И МАТЕМАТИЧЕСКАЯ ОСНОВА алгоритма *DES*:

СЕТЬ ИЛИ **КОНСТРУКЦИЯ ФЕ́ЙСТЕЛЯ** (*FEISTEL NETWORK, FEISTEL CIPHER*) — ОДИН ИЗ МЕТОДОВ ПОСТРОЕНИЯ БЛОЧНЫХ ШИФРОВ

АЛГОРИТМ:

БЛОК ОТКР ТЕКСТА ДЕЛИТСЯ НА ДВЕ РАВНЫЕ ЧАСТИ: L_0 И R_0

В КАЖДОМ РАУНДЕ ВЫЧИСЛЯЮТСЯ:

$$L_i = R_{i-1};$$

$$R_i = L_{i-1} \oplus F(L_{i-1}, K_i),$$

ГДЕ: i — НОМЕР РАУНДА; $i = 1 \ldots N$

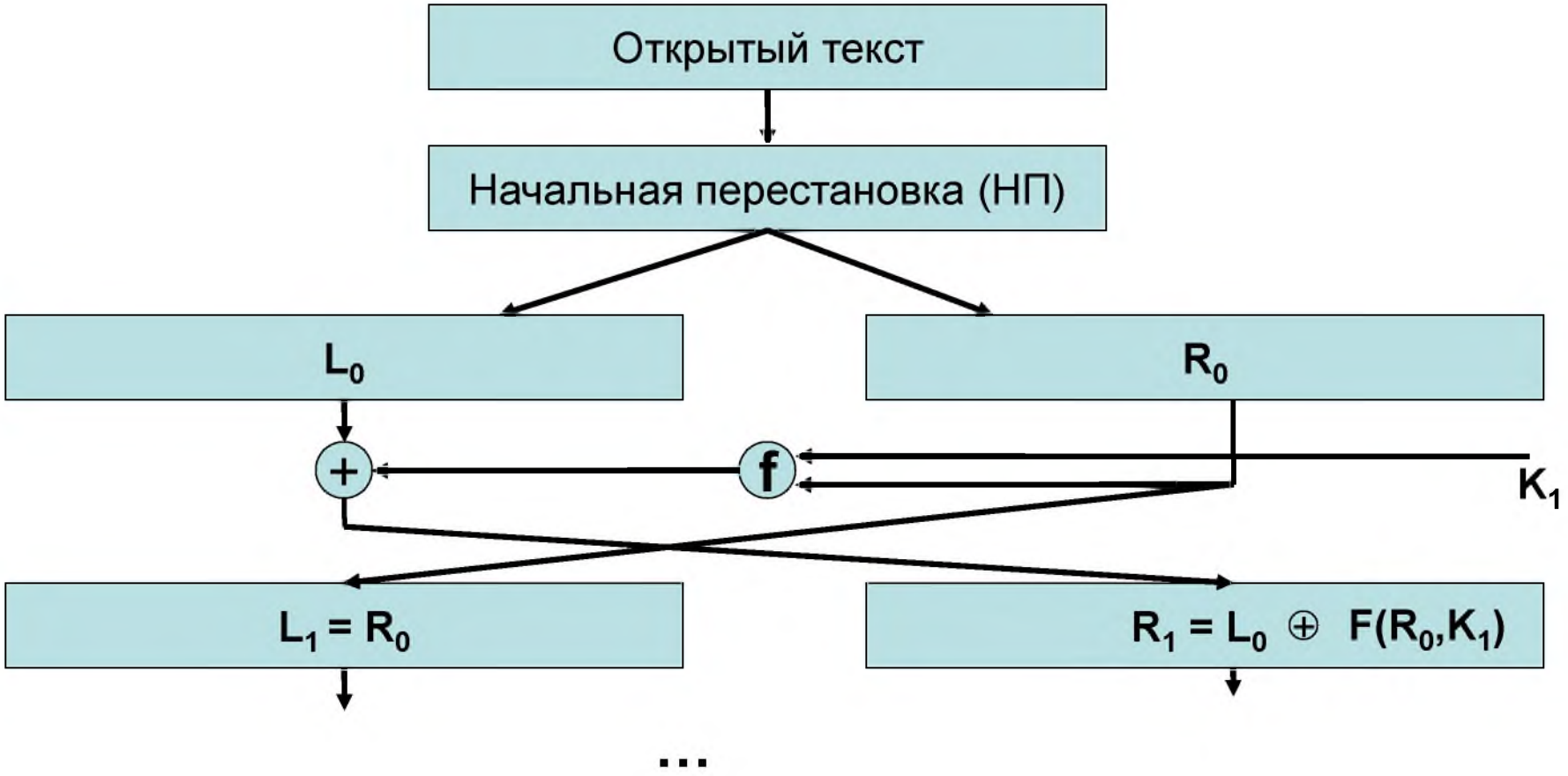
N — КОЛИЧЕСТВО РАУНДОВ;

F — НЕКОТОРАЯ ФУНКЦИЯ;

K_{i-1} — КЛЮЧ *i-1*-ГО РАУНДА (РАУНДОВЫЙ КЛЮЧ).

Алгоритм DES

Рис.3



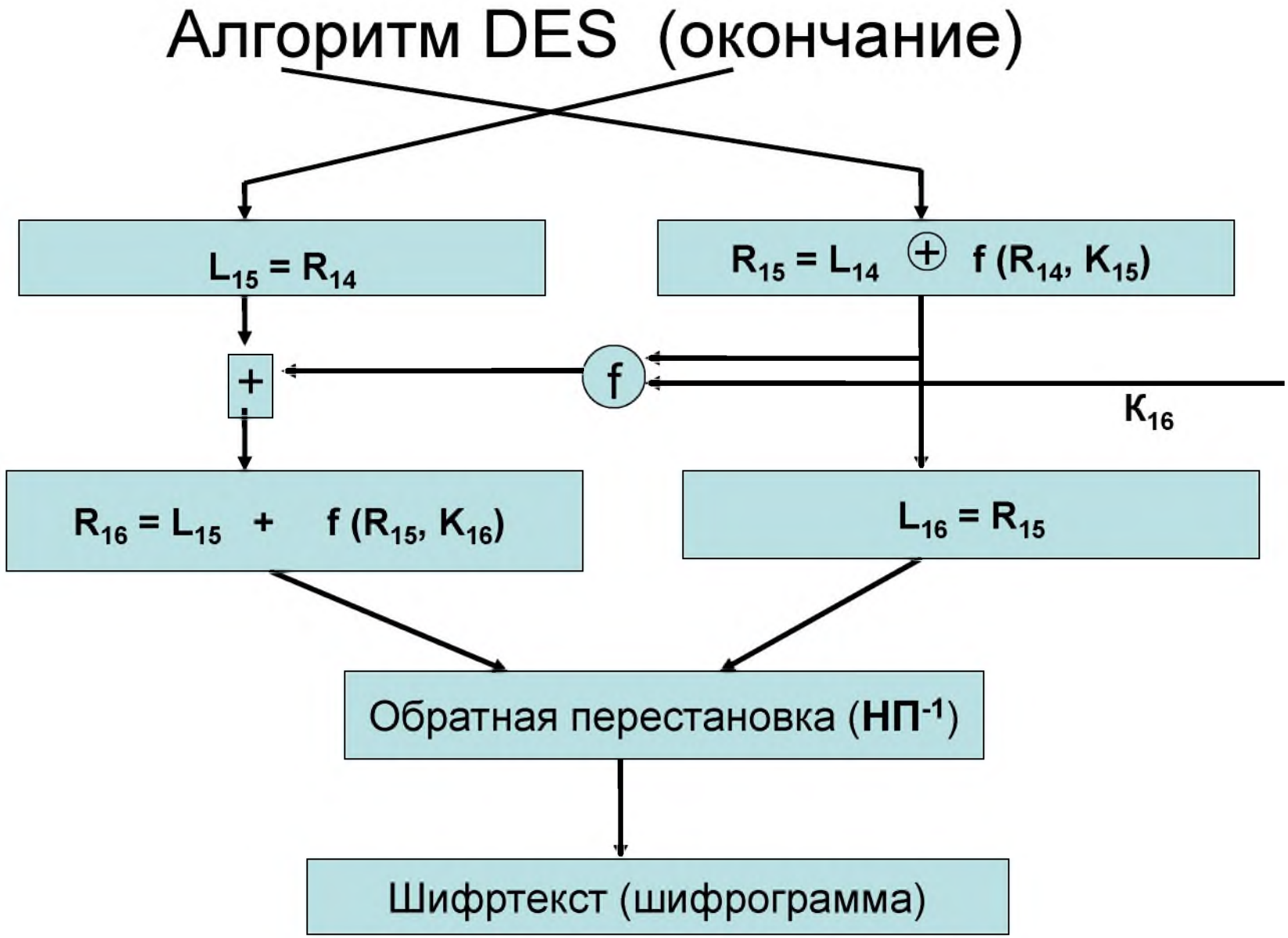


Рис.3 (окончание)

Расшифрование - идентичный алгоритм, но ключи – в обратном порядке

Начальная и конечная перестановки не имеют никакого значения для криптографии в DES.
Обе перестановки – без ключей и предопределенны

Начальная перестановка								Конечная перестановка							
58	50	42	34	26	18	10	02	40	08	48	16	56	24	64	32
60	52	44	36	28	20	12	04	39	07	47	15	55	23	63	31
62	54	46	38	30	22	14	06	38	06	46	14	54	22	62	30
64	56	48	40	32	24	16	08	37	05	45	13	53	21	61	29
57	49	41	33	25	17	09	01	36	04	44	12	52	20	60	28
59	51	43	35	27	19	11	03	35	03	43	11	51	19	59	27
61	53	45	37	29	21	13	05	34	02	42	10	50	18	58	26
63	55	47	39	31	23	15	07	33	01	41	09	49	17	57	25

Начальные и конечные перестановки – это прямые P -блоки, которые инверсны друг другу

Пример.
Найти выход начального блока перестановки, когда на его вход поступает последовательность:
0x**00 02 00 00 00 00 00 01**

Входной сигнал имеет только две единицы: биты 15 и 64);
Выход должен также иметь только две единицы (прямая перестановка).
Используя левую таблицу, мы можем найти выход, связанный с этими двумя битами.
Бит 15 на входе становится битом 63 в выходе.
Бит 64 во входе становится битом 25 в выходе.
На выходе будем иметь только две единицы: бит 25 и бит 63.
Результат в шестнадцатеричном исчислении:
0x **0000 0080 0000 0002**

Пример

Докажем, что начальные и финальные перестановки инверсны друг другу.

Преобразуем полученную выходную последовательность

0x **00 00 00 80 00 00 00 02**

во входную.

Единичные биты - это 25 и 63, другие биты равны нулю.

В конечной перестановке 25 -й бит переходит в 64 -й,

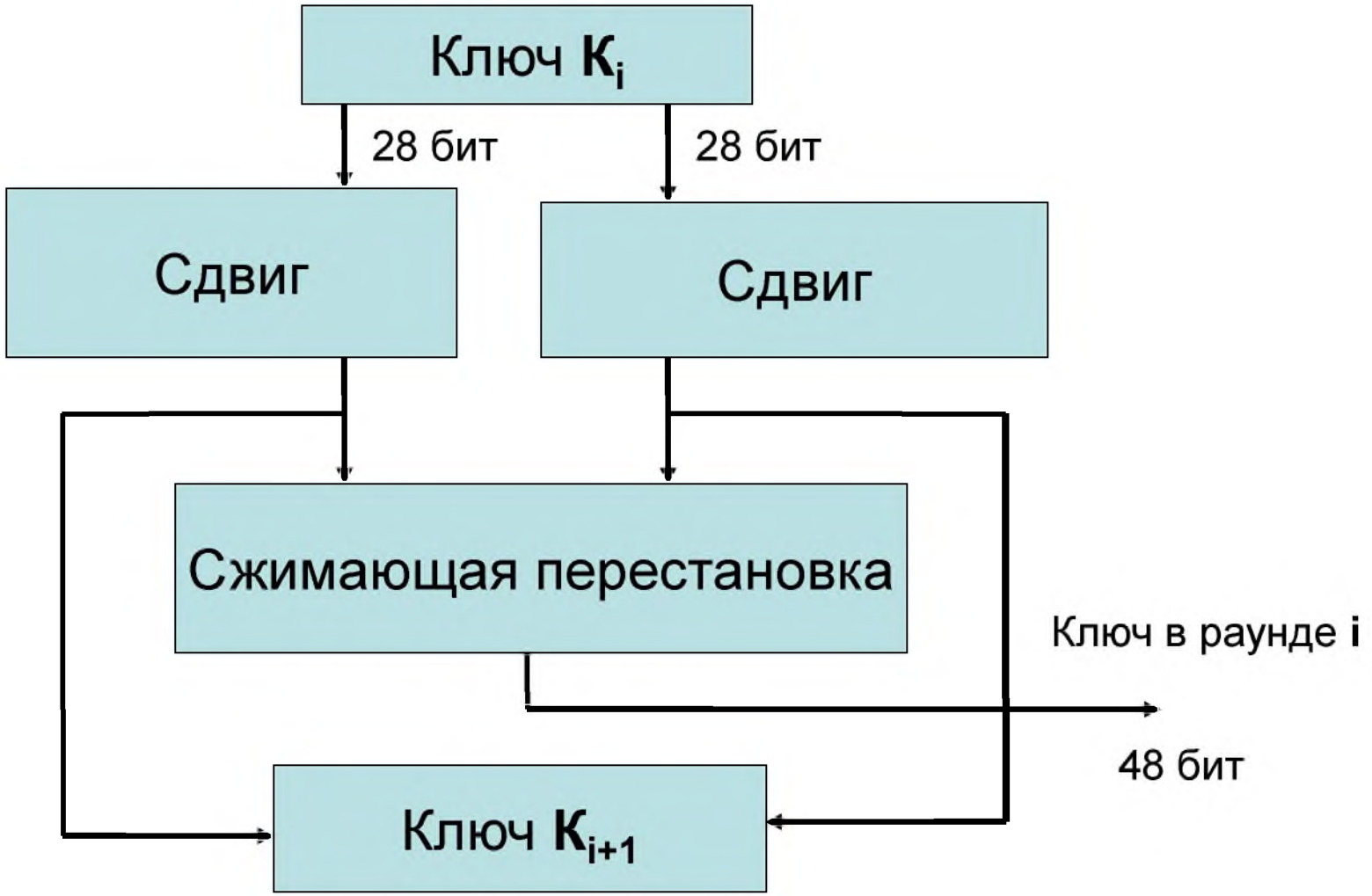
а 63 -й - в 15 -й.

Результат

0x **00 02 00 00 00 00 00 01**

Преобразование ключа в 1-м раунде

Рис.1



Матрица первоначальной подготовки ключа

(сжимающая перестановка перед первым раундом)

57 49 41 33 25 17 09

01 58 50 42 34 26 18 *28 бит*

10 02 59 51 43 35 27

19 11 03 60 52 44 36

63 55 47 39 31 23 15

07 62 54 46 38 30 22 *28 бит*

14 06 61 53 45 3729

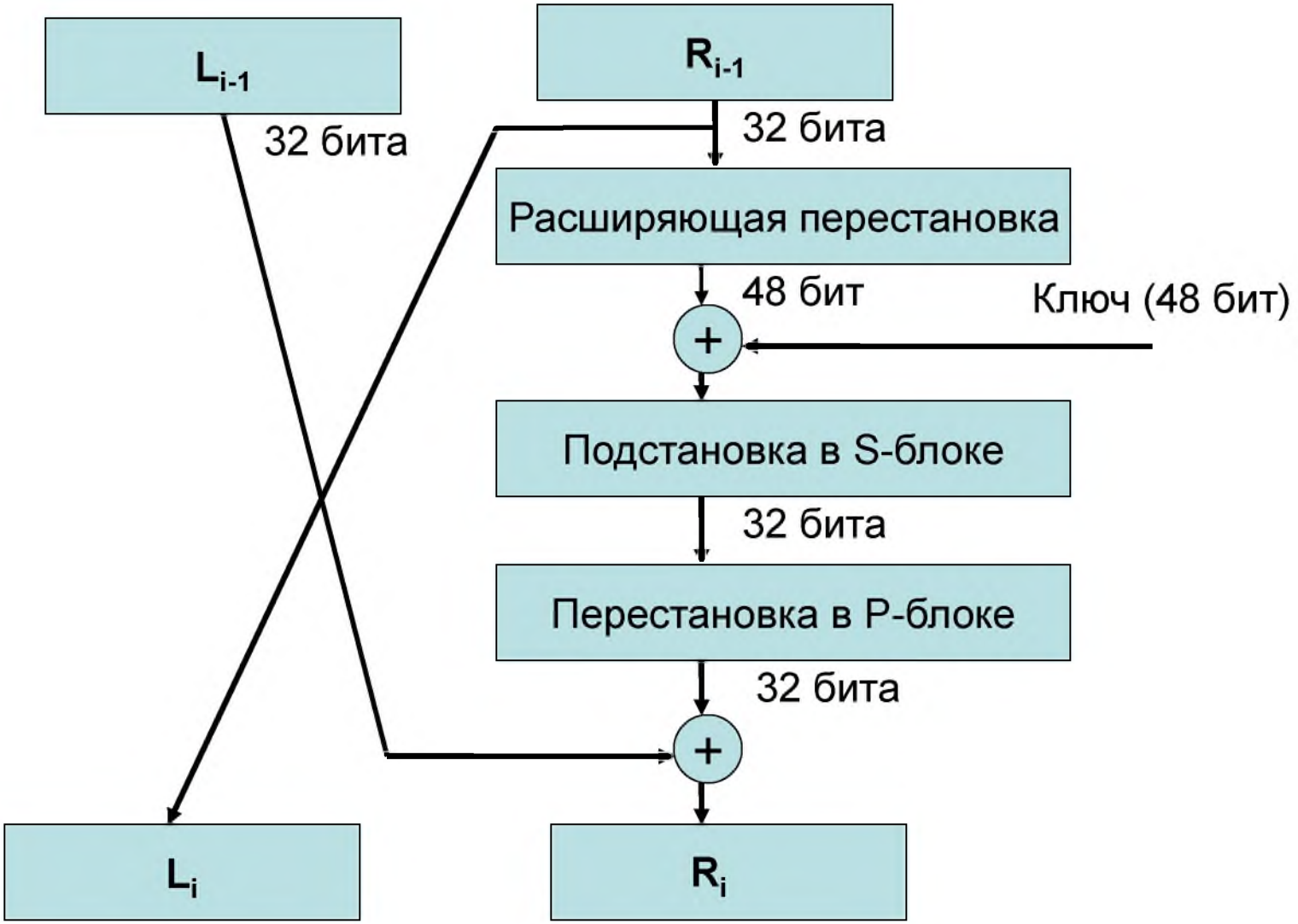
21 13 05 28 20 12 04

Таблица сдвигов для вычисления ключа

Номер раунда	Сдвиг влево (бит)
01	1
02	1
03	2
04	2
05	2
06	2
07	2
08	2
09	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Один раунд **DES**

Рис.2



Сжимающая перестановка ключа перед первым раундом

14 17 11 24 01 05

03 28 15 06 21 10

23 19 12 04 26 08

16 07 27 20 13 02

41 52 31 37 47 55

30 40 51 45 33 48

44 49 39 56 34 53

46 42 50 36 29 32

Один раунд **DES**

Расширяющаяся перестановка (некоторые символы повторяются)

1	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
1	13	14	15	16	17
2	17	18	19	20	21
3	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Один раунд **DES**

Подстановка с помощью S-блока

Производится в 8 блоках (каждый имеет 6-битовый вх и 4-битовый вых)

Каждый блок – таблица 4 стр х 16 стб: по 6-ти вх битам определяются номера стб и стр, под которыми ищем вых значения: биты b1 и b6 объединяются (b1b6) и соответствуют № строки таблицы, остальные биты - № столбца

Каждая таблица из 8 – известна

Пример. На вх 6-го S-блока: **110011**. № стр – **11** – третья строка, № стб – **1001** – 9-ый стб ; на их пересечении в таблице – число 14 – **1110** – эти данные на вых блока

Один раунд **DES**

Перестановка с помощью P-блока

Просто переставляются биты :

1 7 20 21 29 12 28 17 16 15 23 26 5 18 31 10

2 8 24 14 32 27 3 9 19 13 30 6 22 11 4 25

Последняя операция XOR – результаты перестановки в P-блоке суммируются с левой половиной исходного текста и образуют правую часть вх информация для след-го раунда

Реализация **DES**

- Аппаратная Самая скоростная реализация – на основе чипа DEC : скорость заш и расш – более 1 Гб/с (16.8 млн блоков за 1 сек)
- Программная: на мэйнфрейме IBM 3090 – более 32 тыс блоков за 1 сек

Оценка времени лобового взлома шифра DES в 1995 г.

Стоимость	*Длина ключа, бит*				
(долл)	*40*	*56*	*64*	*80*	*128*
100 тыс	2с	35ч	1г	70 000л	10^{19} л
1 млн	0.2с	3.5ч	37д	7 000л	10^{18} л
100 млн	2 мс	2 мин	9ч	70л	10^{16} л
10 млрд	0.02мс	1с	5.4мин	245д	10^{14} л

В 1997 г. конкурс: 64-битовый кл взломан за 96 дн, в 1999 - за 1сутки

Основной подход при защите: **$C_{информации} >= C_{взлома}$**

Особенности DES

DES работает с бинарными числами с 0 и 1. Каждая группа из 4х бит преобразовывается в шестнадцатиричное число: бинарное «0001» равно шестнадцатиричному «1», «1000» - «8», «1111» равно «F».

DES шифрует группы 64-битных сообщений (16 шестнад-ричных чисел).

DES использует ключи (16 hex или 64 бита; каждый 8-й ключевой бит игнорируется DES алгоритмом, так что эффективный размер ключей - 56 бит).

Пример 1. M=8787878787878787 (длина – 64 бита), K=0E329232EA6D0D73,

Зашифрование **DES:** C=0000000000000000.

Расшифрование тем же ключом: M=8787878787878787.

Пример 2. **M** = 0123456789ABCDEF (**M** - в шестн-чном формате). **M** в бинарный формате (64 бит блок текста):
M = *0000 0001 0010 0011 0100 0101 0110 0111* 1000 1001 1010 1011 1100 1101 1110 1111
L = *0000 0001 0010 0011 0100 0101 0110 0111*
R = 1000 1001 1010 1011 1100 1101 1110 1111

K = 133457799BBCDFF1

или **K** = 0001 0011 00110100 01010111 01111001 10011011 10111100 11011111 **1**1110001

каждый 8-й ключевой бит игнорируется DES в соответствии с таблицей

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

получим 56-битный пермутированный ключ:

K+ = 1111000 0110011 0010101 0101111

0101010 1011001 1001111 0001111

Пример 3. **M** = «*Your lips are smoother than vaseline*» - текстовое сообщение длиной 36 байт (72 hex чисел).

M_{16} =596F7572206C6970732061726520736D 6F6F746865722074 68616E2076617365 6C696E650D0A.

72 hex числа представляют строку на английском языке, «0D» — это hex для переноса каретки на следующую строку и «0A» — новая строка).

Сообщение должно быть дополнено байтами (чтобы длина была кратна 8 байтам (или 16 hex числам или 64 битам); обычно это «0».

Дополняем **M** нулями в конце чтобы получить длину 80 hex чисел: 596F7572206C6970 732061726520736D 6F6F746865722074 68616E2076617365 6C696E650D0A0000

Используем ключ: 0E329232EA6D0D73.

Зашифрованное сообщение : ***C*** = C0999FDDE378D7ED 727DA00BCA5A84EE 47F269A4D6438190 9DD52F78F5358499 828AC9B453E0E653.

$C = c_1, \ldots c_{10}$

c_1 = C0999FDD,

c_2 = E378D7ED,

.....

c_{10} = 53E0E653.

При расшифровании добавленные нули отбрасываются

Пример.

Входной текст:

123456ABCD 132536

После первоначальной перестановки:

14A7D67818CA18D

После разбиения:

L_0= 14A7D678 R_0 = 18CA18D

Начальный ключ:

AABB09182736CCDD

После 16-го раунда зашифрования:

19BA9212 CF26B472

Шифртекст (после обратной перестановки):

COB7ASD05F3AS29C

Источник:https://www.intuit.ru/studies/courses/552/408/lecture/9362?page=3

Раунд	Левая	Правая	Ключ раунда
Раунд 1	18CA18AD	5A78E394	194CD072DE8C
Раунд 2	5A78E394	4A1210F6	4568581ABCCE
Раунд 3	4A1210F6	B8089591	06EDA4ACF5B5
Раунд 4	B8089591	236779C2	DA2D032B6EE3
Раунд 5	236779C2	A15A4B87	69A629FEC913
Раунд 6	A15A4B87	2E8F9C65	C1948E87475E
Раунд 7	2E8F9C65	A9FC20A3	708AD2DDB3C0
Раунд 8	A9FC20A3	308BEE97	34F822F0C66D
Раунд 9	308BEE97	10AF9D37	84BB4473DCCC
Раунд 10	10AF9D37	6CA6CB20	02765708B5BF
Раунд 11	6CA6CB20	FF3C485F	6D5560AF7CA5
Раунд 12	FF3C485F	22A5963B	C2C1E96A4BF3
Раунд 13	22A5963B	387CCDAA	99C31397C91F
Раунд 14	387CCDAA	BD2DD2AB	251B8BC717D0
Раунд 15	BD2DD2AB	CF26B472	3330C5D9A36D
Раунд 16	19BA9212	CF26B472	181C5D75C66D

Проверка эффекта лавины в DES

Зашифруем два блока исходного текста, которые отличаются только одним битом текста, с помощью одного и того же ключа и определим разницу в числе бит в каждом раунде.

1) Исходный текст: **0000000000000000**
Ключ: **22234512987ABB23**
Зашифрованный текст: **4789FD476E82A5F1**
2)
Исходный текст: **0000000000000001**
Зашифрованный текст: **0A4ED5C15A63FEA3**

- Хотя два блока исходного текста отличаются только самым правым битом, блоки зашифрованного текста отличаются на **29** бит.
- Это означает, что изменение приблизительно в **1,5** процентах исходного текста создают изменение приблизительно **45** процентов зашифрованного текста.
- Таблица показывает изменение в каждом раунде. Можно увидеть, что существенные изменения возникают уже в третьем раунде.

Раунд	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Разница в битах	**1**	**6**	**20**	**29**	**30**	**33**	**32**	**29**	**32**	**39**	**33**	**28**	**30**	**31**	**30**	**29**

Достоинство DES– относительно высокая скорость (из-за малой длины ключа);

- бесплатное распространение по всему миру; -

- общедоступность и отсутствие лицензионных отчислений;

Недостатки DES– низкая криптостойкость (из-за малой длины ключа)

в январе 1999 года закодированное посредством DES сообщение было взломано с помощью связанных через Internet в единую сеть 100 тыс. персональных компьютеров за 24 часа;

-проблема хранения и распределения ключевой информации

*На основе **DES** строится протокол **Kerberos** (Цербер) **:** предполагает наличие высоконадежного **сервера**, хранящего исходные копии ключей для взаимодействия с каждым пользователем в сети.*

Криптостойкость DES

Среди предпринятых на DES атак три представляют интерес:

грубая сила (brute force): шифр (ключ) может быть взломан с числом испытаний 2^{55},

дифференциальный криптоанализ (differential cryptanalysis): основан на изучении преобразования разностей между шифруемыми значениями на различных раундах шифрования,

линейный криптоанализ (linear cryptanalysis): основан на использовании **линейных приближений** для описания работы шифра

Дифференциальный криптоанализ

Предложен в 1990 году Э. Бихамем и А. Шамиром.

Задача: определить ключ.

Сущность:

- Выбираются два открытых текста с фиксированной разностью. Можно выбрать и два произвольных открытых текста - лишь бы они удовлетворяли некоторому условию их разности. Криптоаналитику не нужно даже знать их значений. (Для DES термин «разность» определяется с помощью операции XOR. Для других алгоритмов этот термин может определяться иначе).
- Производится зашифрование открытых текстов.
- Затем, используя разности полученных шифртекстов, присвоим различные вероятности различным ключам.
- В процессе дальнейшего анализа следующих пар шифртекстов один из ключей станет наиболее вероятным. Это и есть правильный ключ.

Линейный криптоанализ

- Предложен Митцури Мацуи в 1993 году.
- Анализ использует знания исходного текста (в отличии от анализа с выборкой исходного текста в дифференциальном криптоанализе).
- Сущность:

  Нужно получить соотношения следующего вида:

  (m1+m2+… + mx) + (c1+c2+… + cy) = k1+k2+… + kz

  где: *i* – *i*-е биты текста, шифртекста и ключа соответственно,

  + - операция XOR

Если выполнить операцию XOR над некоторыми битами открытого текста, затем над некоторыми битами шифртекста, а затем над результатами, получим бит, который представляет собой XOR некоторых битов ключа.

Это называется **линейным приближением**, которое может быть верным с некоторой вероятностью *p*.

см. книгу: Б. Шнайер. Прикладная криптография

см.: https://www.intuit.ru/studies/courses/552/408/lecture/9360?page=6

3-DES

- 3DES устраняет недостатки DES (построен на основе DES),
- создан У. Диффи, М. Хеллманом, У. Тачманном, 1978,
- для его реализации возможно использовать программы, созданные для DES,
- формальная запись: $C = f(M,(DES\ (K_3, (DES(K_2, (DES, K_1))))))$
- существуют 3 типа алгоритма 3DES:

1 DES-EEE3: шифруется 3 раза с 3 разными ключами (операции шифрование-шифрование-шифрование).

2 DES-EDE3 : 3DES операции шифрование-расшифрование-шифрование с разными ключами:

3 DES-EEE2 и DES-EDE2 : как и предыдущие, однако, на первом и третьем шаге используется одинаковый ключ.

3DES реализован во многих приложениях, ориентированных на работу с Интернетом:PGP, S/mime

• стандартный **3DES** выполняет 3 раза алгоритм DES,

•длина ключа **3DES** равна 192 (64*3) битов; т.к. в каждом байте используется только 7 битов, на самом деле длина ключа 3DES равна 168 (56*3) битов,

• **Криптостойкость 3DES:**

• 3DES с различными ключами имеет длину ключа равную 168 бит, но из-за атак «встреча посередине» (известны ***M*** и ***C***, нужно найти ***K***) эффективная криптостойкость составляет только 112 бит;

• в варианте DES-EDE, в котором $\mathbf{K_1=K_3}$, эффективный ключ имеет длину 80 бит;

• для успешной атаки на 3DES потребуется около $\mathbf{2^{32}}$ **бит** известного открытого текста, $\mathbf{2^{113}}$ шагов, $\mathbf{2^{90}}$ циклов DES-шифрования и $\mathbf{2^{88}}$ бит памяти;

Применение 3DES

• **3DES** с тремя ключами реализован во многих Интернет-приложениях (в том числе в **PGP** (*Pretty Good Privacy* — позволяет выполнять операции шифрования и цифровой подписи сообщений, файлов и другой информации, представленной в электронном виде, например, на жёстком диске) и **S/mime** (предназначена для обеспечения криптографической безопасности электронной почты; для использования необходимо получить и установить индивидуальный ключ/сертификат от **центра сертификации**);

• **3DES** используется при управлении ключами в стандартах **ANSI X9.17** (*метод генерации 64-битных ключей*) и **ISO 8732 (***управление ключами в банковском деле***)** и в **PEM** (Privacy Enhanced Mail).

• Не известны успешные практические атаки на 3DES;

• **3DES** больше подходит для аппаратных реализаций;

• современная альтернатива **3DES –** *AES Rijndael, программная реализация которого работает в 6 раз быстрее* **3DES;**

- **ГОСТ 28147-89** — советский и российский стандарт симметричного шифрования (1990 год), также является стандартом СНГ.
- Полное название — «*ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования*».
- *Длина ключа – 256 бит*

Таблица ASCII

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1.	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2.	_	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6.	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL