Лекция 10

Шифрование информации. Стандартные алгоритмы шифрования

DES
(Data Encryption Standard)

- Общеправительственный стандарт шифрования некритичной информации

DES
(Data Encryption Standard)

- симметричный алгоритм шифрования, разработанный фирмой IBM и утвержденный правительством США в 1977 году как официальный стандарт (FIPS 46-3).
- DES имеет
  - блоки по **64** бита и
  - 16 цикловую структуру сети Фейстеля,
  - для шифрования использует ключ с длиной **56** бит.
- Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP^{-1}) преобразований.

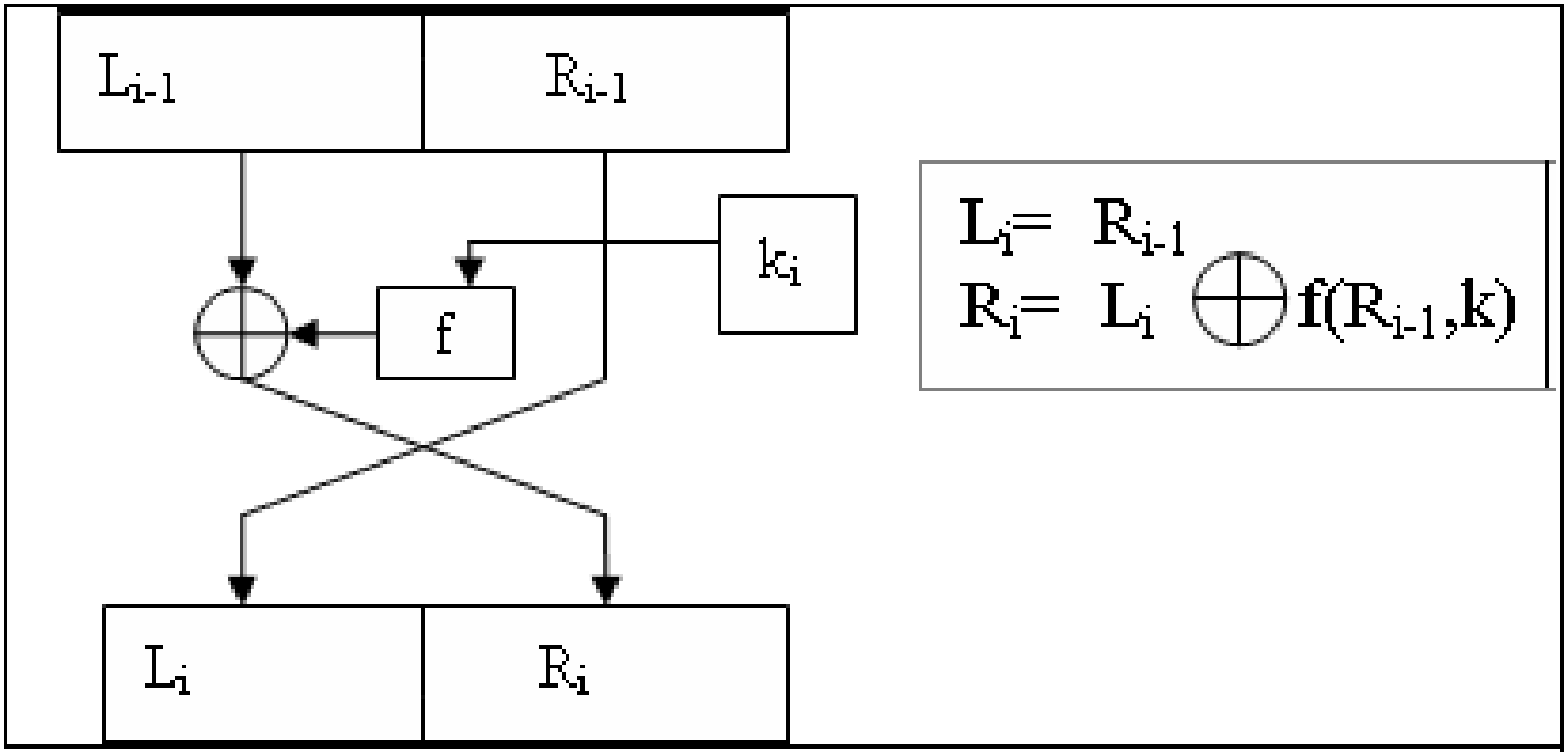
Блочный шифр

- **Вход:** блок размером n бит и k-битный ключ
- **Алгоритм:** шифрующее преобразование
- **Выход:** n-битный зашифрованный блок
- **Примечание_1:** преобразование таково, что незначительные различия входных данных приводят к существенному изменению результата.
- **Примечание_2:** Блочные шифры реализуются путём многократного применения к блокам исходного текста некоторых базовых преобразований.
- **Примечание_3:** Входной файл - бинарный

Преобразования Сетью Фейстеля

- Преобразование над векторами (блоками) представляющими собой левую и правую половины регистра сдвига.
- В алгоритме DES используются
  - прямое преобразование сетью Фейстеля в шифровании и
  - обратное преобразование сетью Фейстеля в расшифровании

Прямое преобразование сетью Фейстеля



Обратное преобразование сетью Фейстеля

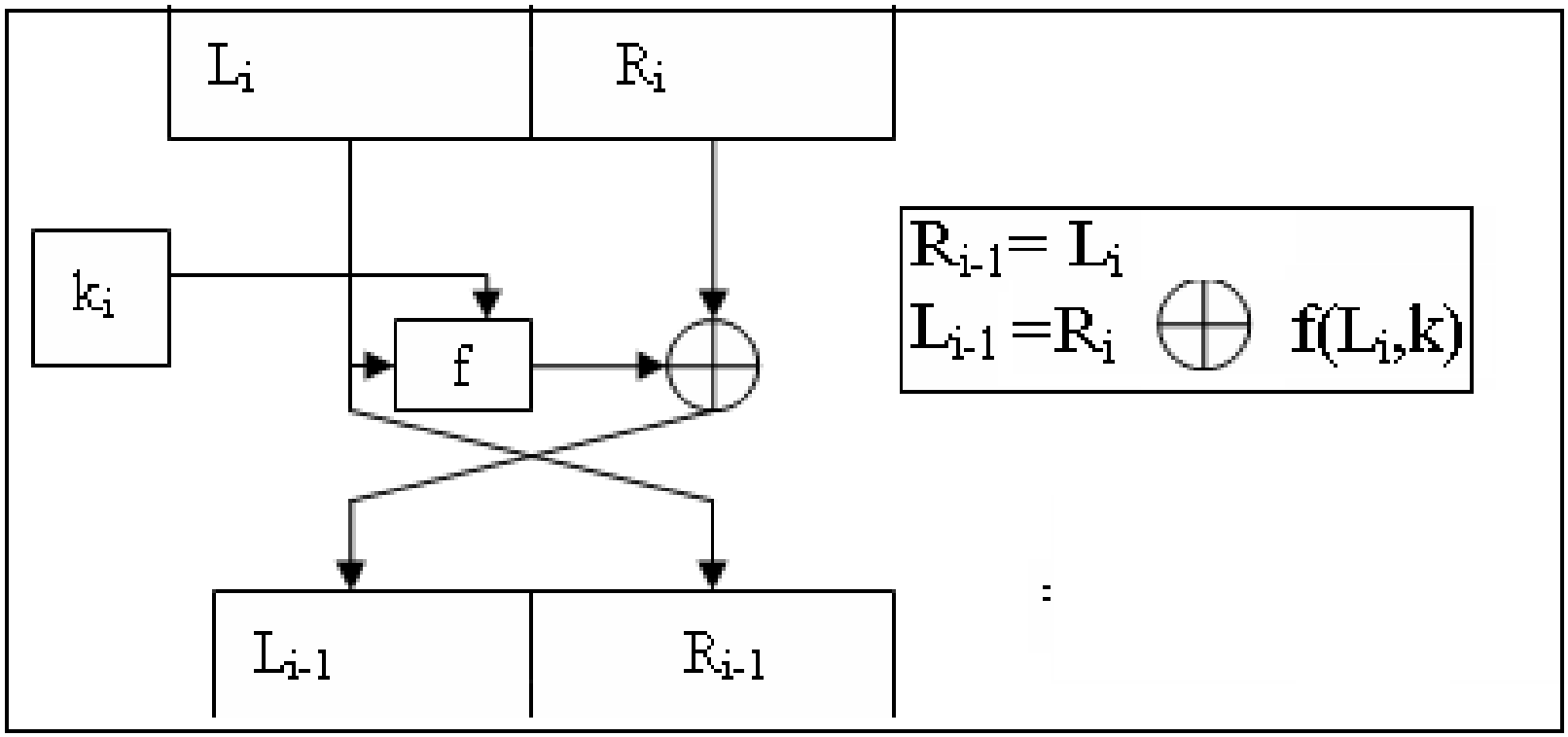
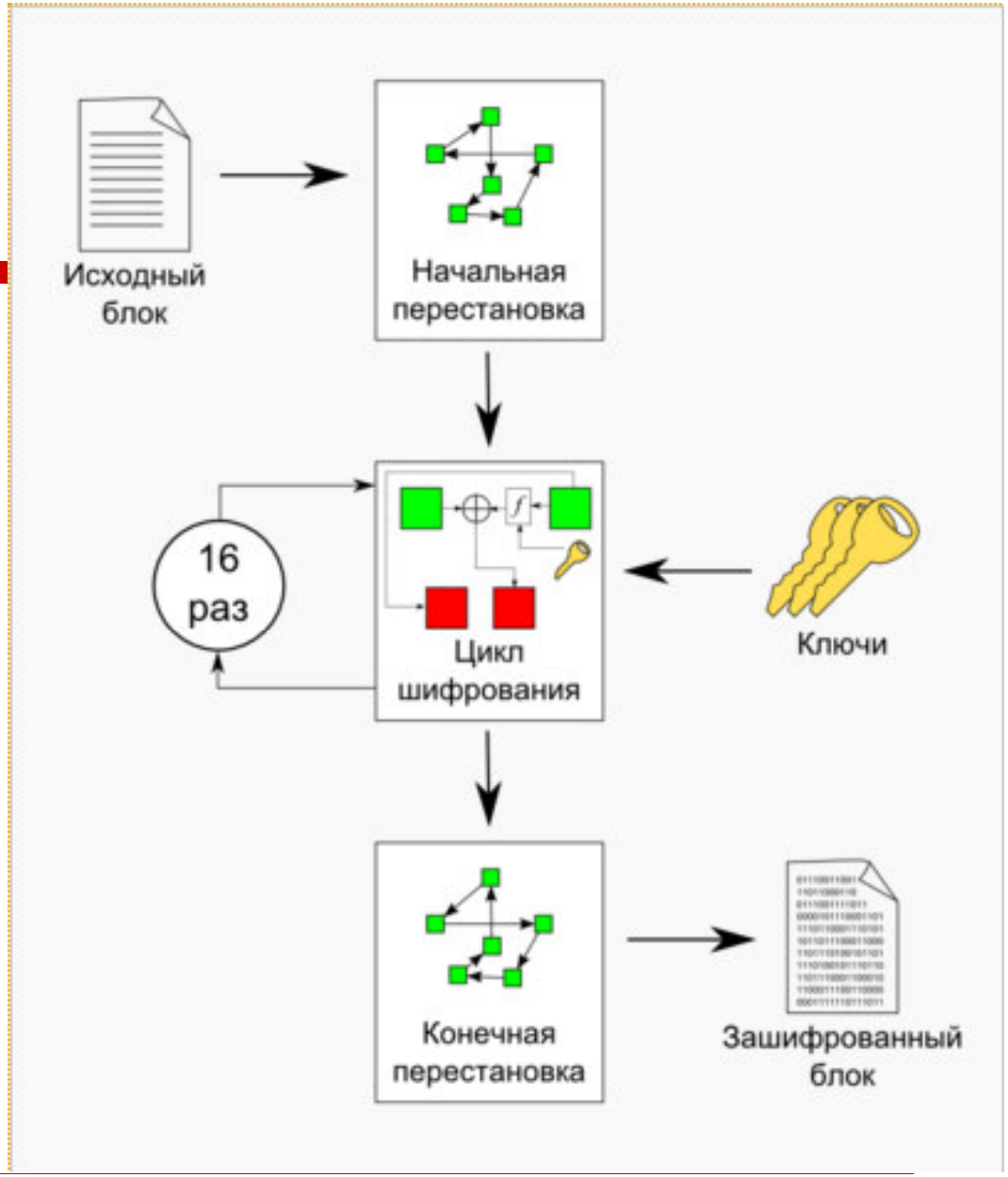


Схема шифрования алгоритма DES

- Исходный текст — блок 64 бит.
- Процесс шифрования состоит в
  - начальной перестановке,
  - 16 циклах шифрования и
  - конечной перестановке.



Начальная перестановка IP

- Исходный текст **T** (блок 64 бит) преобразуется с помощью начальной перестановки **IP**, определяемой таблицей

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

- По таблице первые 3 бита результирующего блока **IP(T)** после начальной перестановки IP являются битами 58, 50, 42 входного блока T, а его 3 последние бита являются битами 23, 15, 7 входного блока

Циклы шифрования - 16 циклов преобразования Фейстеля

- $T_0 = IP(T) = L_0R_0$ (L_0-32 bit, R_0–32 bit)
- После i-й итерации: $T_{i-1} = L_{i-1}\, R_{i-1}$
- $T_i = L_i\, R_i$: $L_i = R_{i-1}$, $R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$

- f – функция шифрования

f – функция шифрования. $f(R_{i-1}, k_i)$

- Аргументы:
  R_{i-1} - 32 bit,
  k_i - 48 bit
- Результат:
  $f(R_{i-1}, k_i)$ - 32 bit
- Включает:
  - Функцию расширения E,
  - преобразование S, состоящее из 8 преобразований S-блоков S1,…,S8
  - перестановку P

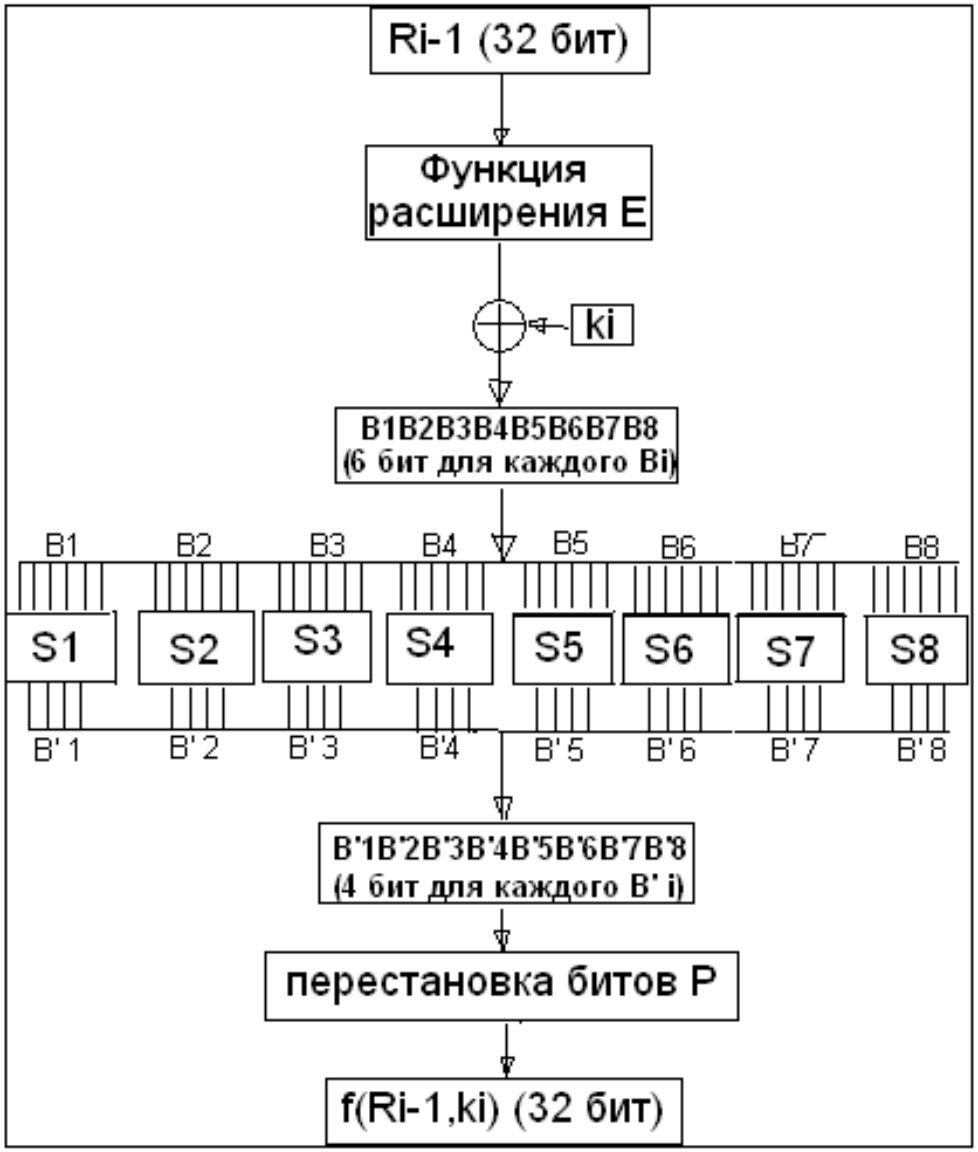


Схема работы функции f

Функция расширения E

- Расширяет 32-битовый вектор R_{i-1} до 48-битового вектора $E(R_{i-1})$ путём дублирования некоторых битов из R_{i-1}
- Порядок битов вектора $E(R_{i-1})$:

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

- Первые три бита вектора $E(R_{i-1})$ являются битами 32, 1, 2 вектора R_{i-1}. Последние 3 бита вектора $E(R_{i-1})$ — это биты 31, 32, 1 вектора R_{i-1}.
- Биты 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29, 32 дублируются.
- Полученный после перестановки блок $E(R_{i-1})$ складывается по модулю 2 с ключами k_i и затем представляется в виде восьми последовательных блоков $B_1,B_2,...B_8$.
- $E(R_{i-1}) = B_1B_2...B_8$

Преобразование S

$$E(R_{i-1}) = B_1B_2...B_8 \Rightarrow B'_1B'_2...B'_8$$

- Каждый B_j является 6-битовым блоком.
- Каждый из блоков B_j трансформируется в 4-битовый блок B'_j с помощью преобразований S_j.
- Преобразования S_j определяются таблицей

	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	

Пусть B_1 = 101111, и мы хотим найти $B`_1$.
Первый и последний разряды B_1 являются двоичной записью числа a, 0<=a<=3, средние 4 разряда представляют число b, 0<=b<=15.
Двоичное представление числа (a,b) дает $B`_1$.
При $a = 11_2 = 3$, $b = 0111_2 = 7$, (3,7) = 7. Его двоичное представление $B`_1$=0111.

Перестановка P:

$f(R_{i-1}, k_i) = P(B'_1 B'_2 \ldots B'_8)$

$B'_1 B'_2 \ldots B'_8$ – 32 bit

$f(R_{i-1}, k_i)$ – 32 bit

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

Согласно таблице, первые четыре бита вектора $f(R_{i-1}, k_i)$ — это биты 16, 7, 20, 21 вектора $B'_1 B'_2 \ldots B'_8$

Генерирование ключей k_i
Расширение ключа k (56 бит) до 64 бит

- Восемь битов, находящих в позициях 8, 16, 24, 32, 40, 48, 56, 64 добавляются в ключ k таким образом чтобы каждый байт (64 бита = 8 байт) содержал нечетное число единиц.
- Это используется для обнаружения ошибок при обмене и хранении ключей.
- Затем делают перестановку для расширенного ключа (кроме добавляемых битов 8, 16, 24, 32, 40, 48, 56, 64)

Перестановки расширенного ключа

57	49	41	33	25	17	9	1	58	50	42	34	26	18	C_0
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	D_0
14	6	61	53	45	37	29	21	13	5	28	20	12	4	

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Число сдвига	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

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

1. Перестановка определяется двумя блоками C_0 и D_0 по 28 бит каждый. Первые 3 бита C_0 есть биты 57, 49, 41 расширенного ключа. А первые три бита D_0 есть биты 63, 55, 47 расширенного ключа.
2. C_i,D_i i=1,2,3…получаются из C_{i-1}, D_{i-1} одним или двумя левыми циклическими сдвигами
3. Ключ k_i, i=1,…16 состоит из 48 бит, выбранных из битов вектора C_iD_i (56 бит) Первый и второй биты k_i есть биты 14, 17 вектора C_iD_i

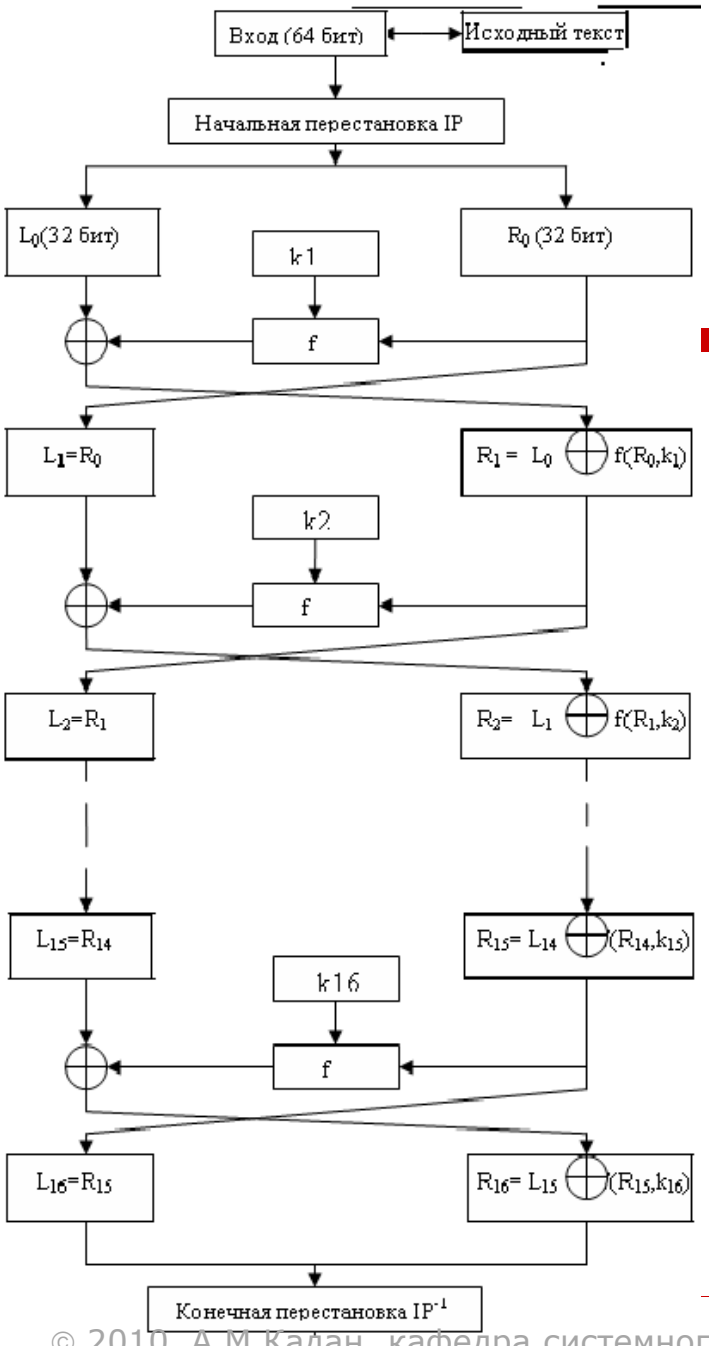
Конечная перестановка IP^{-1}

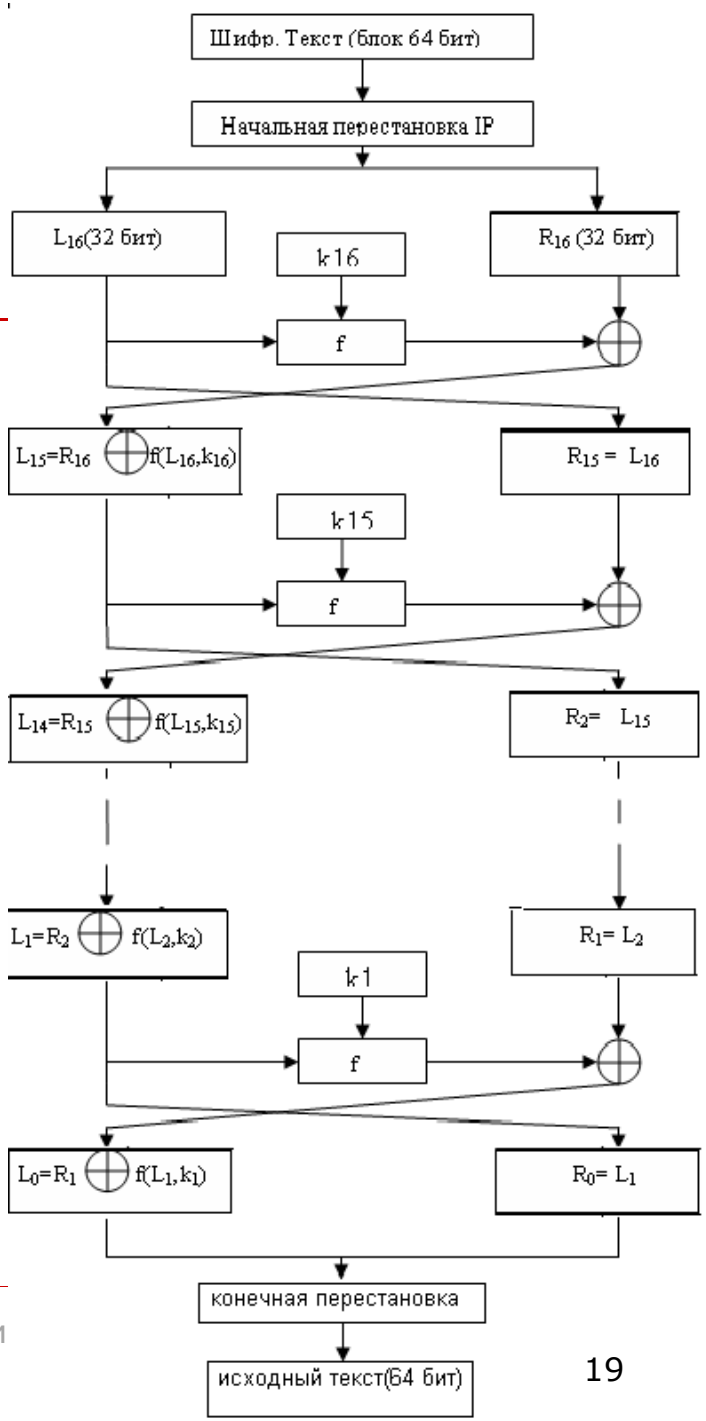
- Действует на T_{16} и используется для восстановления позиции

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

Схема расшифрования

- При расшифровании данных все действия выполняются в обратном порядке.
- В 16 циклах расшифрования используется обратное преобразование сетью Фейстеля:
  $R_{i-1} = L_i$
  $L_{i-1} = R_i + f(L_i, k_i)$
- При расшифровании ключ k_i, i=1,…,16, функция f, перестановка IP и IP^{-1} такие же как и в процессе шифрования.





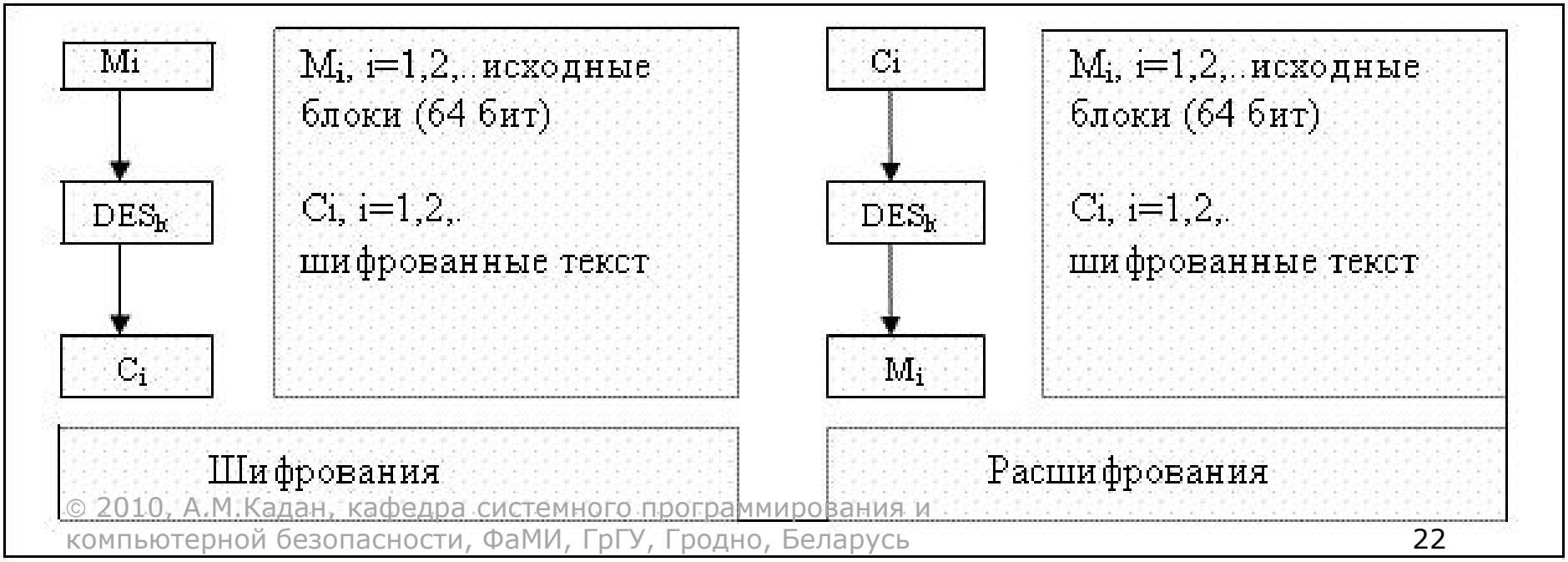
Режимы использования DES

Режимы использования DES

- Режим электронной кодовой книги — (ECB — Electronic Code Book)
- Режим сцепления блоков (CBC — Cipher Block Chaining)
- Режим обратной связи по шифротексту (CFB — Cipher Feed Back)
- Режим обратной связи по выходу (OFB — Output Feed Back)

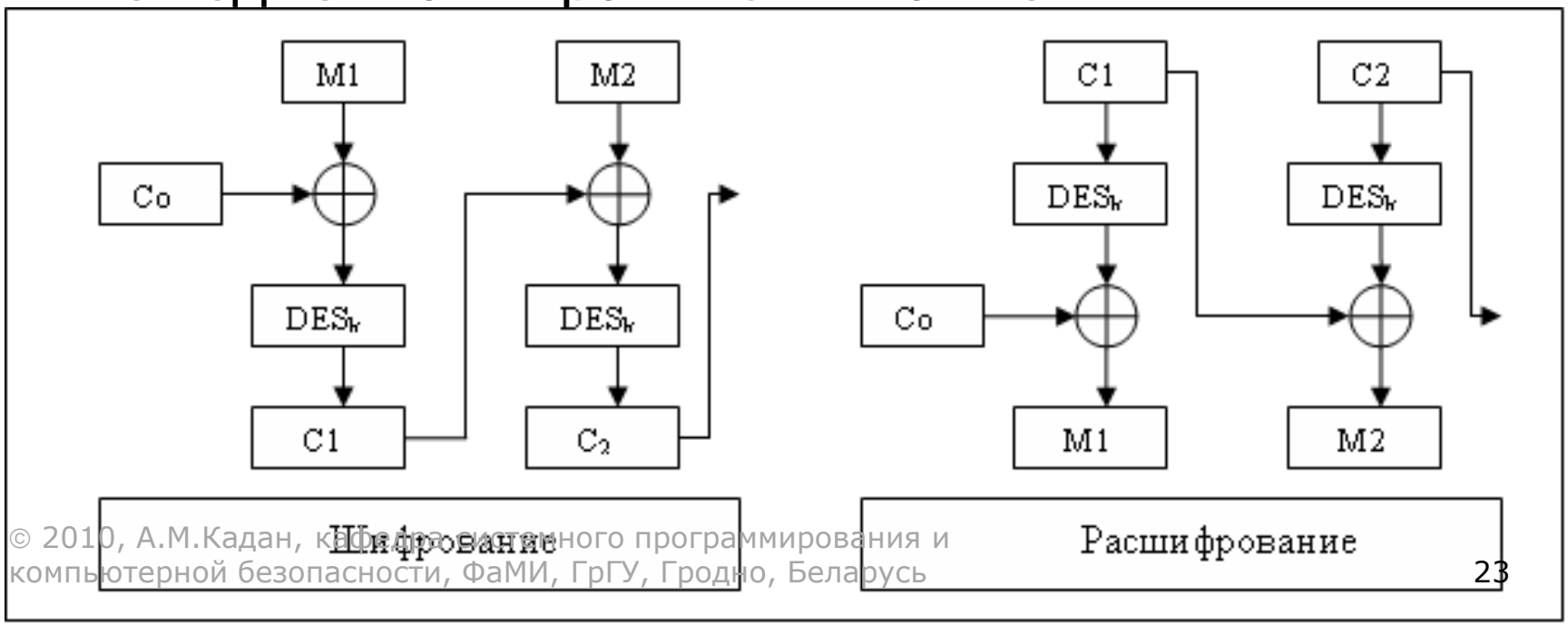
Режим электронной кодовой книги — ECB

Обычное использование DES как блочного шифра. Шифруемый текст разбивается на блоки, при этом, каждый блок шифруется отдельно, не взаимодействуя с другими блоками



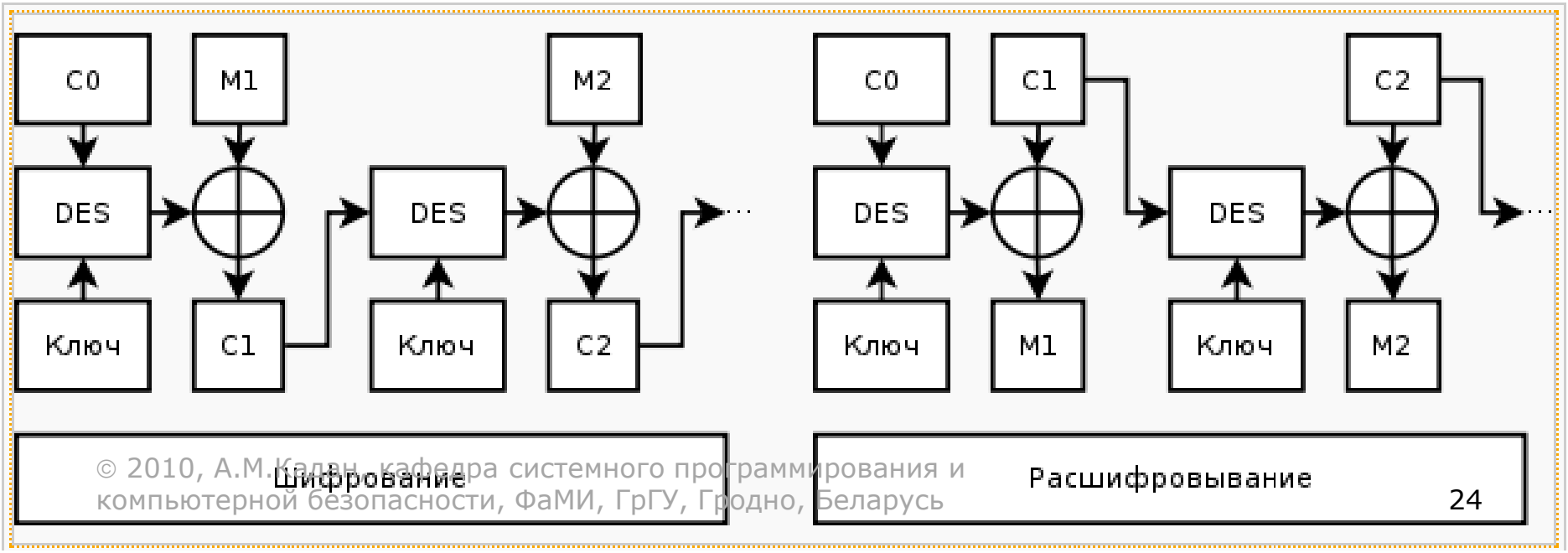
Режим сцепления блоков

- Каждый очередной блок C_i i>=1, перед зашифровыванием складывается по mod 2 со следующим блоком открытого текста M_{i+1}. Вектор C_0 — начальный вектор. Меняется ежедневно и хранится в тайне



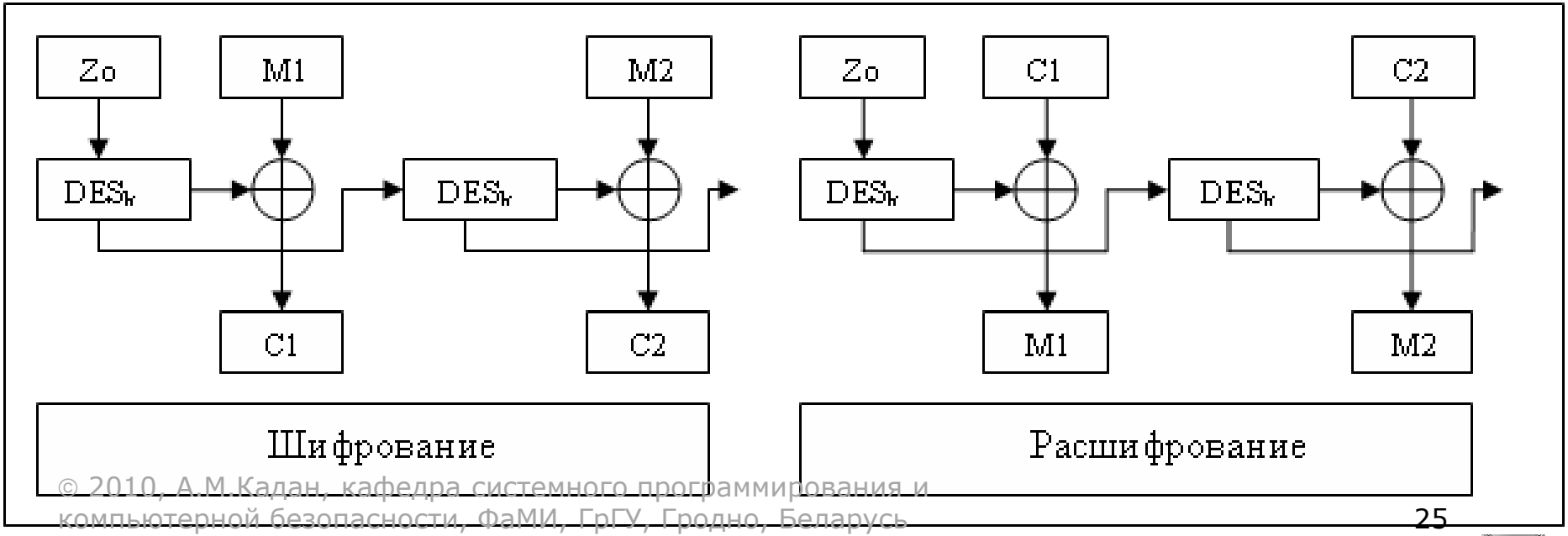
Режим обратной связи по шифротексту

- В режиме CFB вырабатывается блочная «гамма» $Z_0, Z_1, \ldots$
  $Z_i = DES_k(C_{i-1})$. $C_i = M_i \oplus Z_i$
  Начальный вектор C_0 сохраняется в



Режим обратной связи по выходу

- Вырабатывается блочная «гамма» $Z_0, Z_1, \ldots$,
- $Z_i = DES_k(Z_{i-1}) C_i = M_i \oplus Z_i$, i >= 1



Криптостойкость алгоритма DES

Выбор S-блоков. Требует соблюдения нескольких условий

- Каждая строка каждого блока должна быть перестановкой множества {0,1,2,……,15}
- S-блоки не должны являться линейной или афинной функцией своих аргументов.
- Изменение одного бита на входе S-блока должно приводить к изменению по крайней мере двух битов на выходе.
- Для каждого S-блока и любого аргумента x значение S(x) и $S(x \oplus 001100_2)$ должны различаться по крайней мере двумя битами.

Полный перебор ключей

- Ключей всего 2^{56} - возможность их перебора
- В 1998 году The Electronic Foundation на специальном компьютере DES-Cracker, удалось взломать DES за 3 дня.

Слабые ключи

- Слабыми ключами называется ключи k такие что $DES_k(DES_k(x)) = x$, x — блок 64 бит.
- Известны 4 слабых ключа. Для каждого слабого ключа существует 2^{32} «постоянные точки», то есть таких 64-битовых блоков x, в которых $DES_k(x) = x$

Слабые ключи(hexadecimal)	C_0	D_0
0101-0101-0101-0101	$[0]^{28}$	$[0]^{28}$
FEFE-FEFE-FEFE-FEFE	$[1]^{28}$	$[1]^{28}$
1F1F-1F1F-0E0E-0E0E	$[0]^{28}$	$[1]^{28}$
E0E0-E0E0-F1F1-F1F1	$[1]^{28}$	$[0]^{28}$

Частично-слабые ключи

- Пары ключей (k_1,k_2) такие что $DES_{k1}(DES_{k2}(x))=x$
- Существуют 6 частично-слабых пар ключей. Для каждого из 12 частично-слабых ключей существуют 2^{32} «анти-постоянные точки», то есть такие блоки x, что $DES_k(x) = \tilde{x}$

C_0	D_0	Пары частично-слабых ключей	C_0	D_0
$[01]^{14}$	$[01]^{14}$	01FE-01FE-01FE-01FE,----FE01-FE01-FE01-FE01	$[10]^{14}$	$[10]^{14}$
$[01]^{14}$	$[01]^{14}$	1FE0-1FE0-1FE0-1FE0,----E0F1-E0F1-E0F1-E0F1	$[10]^{14}$	$[10]^{14}$
$[01]^{14}$	$[0]^{28}$	01E0-01E0-01F1-01F1,----E001-E001-F101-F101	$[10]^{14}$	$[0]^{28}$
$[01]^{14}$	$[1]^{28}$	1FFE-1FFE-0EFE-0EFE,----FE1F-FE1F-FE0E-FE0E	$[0]^{28}$	$[1]^{28}$
$[0]^{28}$	$[01]^{14}$	O11F-011F-010E-010E,----1F01-1F01-0E01-0E01	$[0]^{28}$	$[10]^{14}$
$[1]^{28}$	$[01]^{14}$	E0FE-E0FE-F1FE-F1FE,----FEE0-FEE0-FEF1-FEF1	$[1]^{28}$	$[10]^{14}$

Известные атаки на DES

Методы атаки	Известные откр. тексты	Выбранные отк. тексты	Объём памяти	Количество операций
Полный поиск	1	-	Незначительный	2^{55}
Линейный Криптоанализ	2^{43}(85%)	-	Для текста	2^{43}
Линейный Криптоанализ	2^{38}(10%)	-	Для текста	2^{50}
Диффер. Криптоанализ	-	2^{47}	Для текста	2^{47}
Диффер. Криптоанализ	2^{55}	-	Для текста	2^{55}

Увеличение криптостойкости DES

- Методы 2DES и 3DES основаны на DES, но увеличивают длину ключей (2DES — 112 бит, 3DES — 168 бит).
- Схема 3DES имеет вид $DES(k_3,DES(k_2,DES(k_1,M)))$, где k_1,k_2,k_3 ключи для каждого шифра DES.
- Существует 3 типа алгоритма 3DES:

DES-EEE3: Шифруется три раза с 3 разными ключами.
DES-EDE3: 3DES операции шифровка-расшифровка-шифровка с 3 разными ключами.

$$C = E_{k_3}(E_{k_2}^{-1}(E_{k_1}(P))) \quad P = E_{k_1}^{-1}(E_{k_2}(E_{k_3}^{-1}(C)))$$

DES-EEE2 и DES-EDE2: Как и предыдущие, за исключением того, что первая и третья операции используют одинаковый ключ.