ПРАКТИЧЕСКАЯ КРИПТОЛОГИЯ
ЛЕКЦИЯ 6
Специальность: 6.170101 – Бсіт
Лектор: Сушко С.А.

§6. АЛГОРИТМ DES

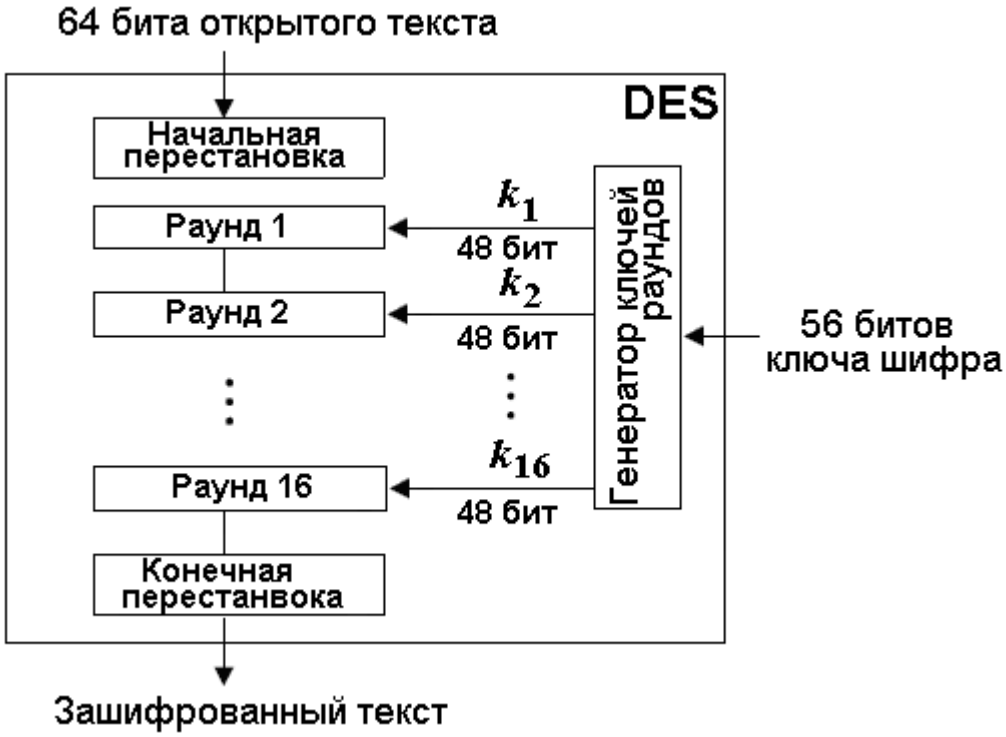
Стандарт шифрования данных DES (DATA ENCRYPTION STANDARD) – блочный шифр с симметричными ключами, разработан Национальным Институтом Стандартов и Технологии (NIST – National Institute of Standards and Technology).

История: в l973 году NIST издал запрос для разработки предложения национальной криптографической системы с симметричными ключами Предложенная IBM модификация проекта, названная LUCIFER, была принята как DES. В марте 1975 криптоалгоритм DES был издан в эскизном виде в Федеральном Регистре *года как Федеральный Стандарт Обработки Информации (FIPS – Federal Information Processing Standard).*

После публикации алгоритм жестко критиковался по двум причинам. Первая: критиковалась сомнительно маленькая длина ключа 56 битов, что могло сделать шифр уязвимым к атаке "грубой силой". Вторая причина: критики были обеспокоены некоторым скрытым построением внутренней структуры DES. Они подозревали, что S-боксы имеет скрытую лазейку, которая позволит Национальному агенству по безопасности США расшифровывать сообщения без ключа. Впоследствии проектировщики IBM сообщили, что внутренняя структура была доработана, чтобы предотвратить криптоанализ. Федеральный Регистр объявил DES стандартом шифрования и он быстро стал наиболее широко используемым блочным шифром. Позже NIST предложил новый стандарт, рекомендующий использовать трехкратно повторенный шифр DES. В 2000 г. новый стандарт AES заменил DES.

Общие положения

Для шифрования DES принимает 64-битовый открытый текст и порождает 64-битовый зашифрованный текст и наоборот, получив 64 бита зашифрованного текста, он выдает 64 бита расшифрованного. В обоих случаях для шифрования и дешифрования применяется один и тот же 56-битовый ключ.



Структура DES

Процесс шифрования состоит из двух перестановок, которые называют начальной и финальной (конечной) перестановками, и 16 раундов Фейстеля. Каждый раунд использует различные сгенерированные 48-битовые ключи.

2) Начальная IP и конечная IP^{-1} перестановки

На вход каждой из них поступает 64 бита, которые затем переставляются в соответствии с заданными таблицами. Эти перестановки взаимно обратны. Другими словами, 58-й бит на входе начальной перестановке переходит в 1-ую позицию на выходе из нее. А финальная перестановка 1-ый входной бит переведет в 58-ую позицию на выходе.

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

бит 58 блока становится битом 1, бит 50 – битом 2 и т.д.

Обе перестановки не имеют никакого значения для криптографии. Причина, почему они включены в DES, не ясна и не указана проектировщиками DES.

Пример

начальная перестановка:

вход:

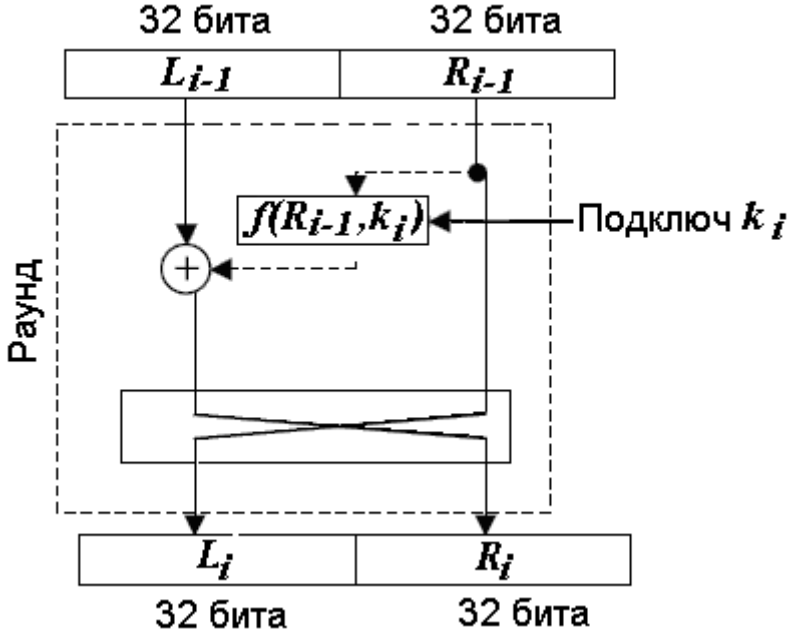
0000000000000010000000000000000000000000000000000000000000000001

выход:

0000000000000000000000010000000000000000000000000000000000000010

2) Раунды DES

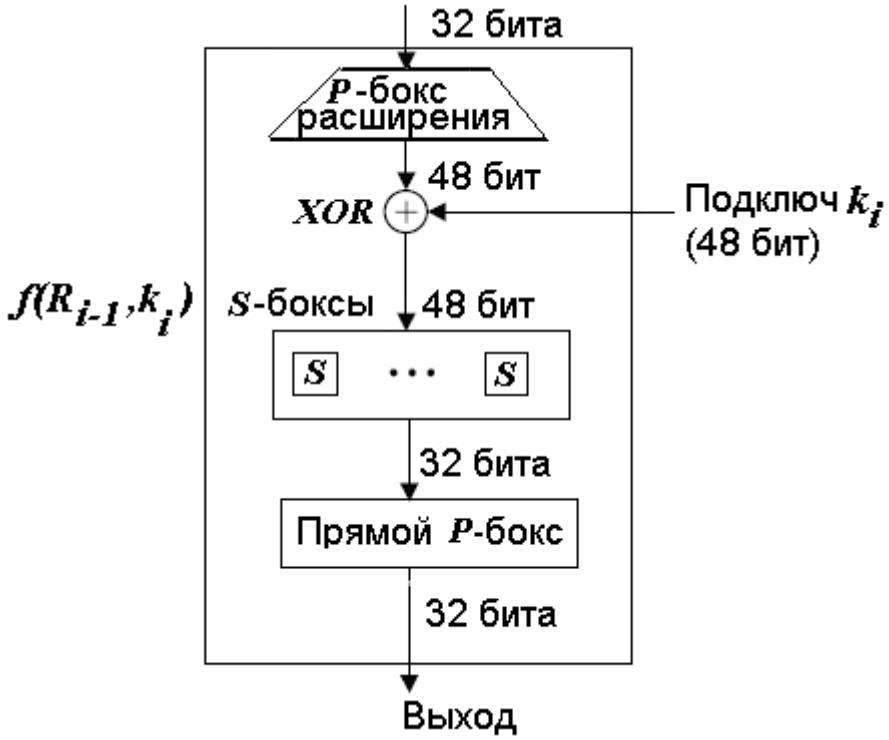
DES использует 16 раундов. Каждый раунд DES применяет шифр Фейстеля, как это показано на рисунке



Раунд принимает полублоки L_{i-1} и R_{i-1} от предыдущего раунда (или начального блока перестановки) и создает полублоки L_i и R_i для входа в следующий раунд (или конечный блок перестановки). Все необратимые элементы сосредоточены в функции $f(R_{i-1},k_i)$.

3)Функция DES

Функция DES с помощью 48-битового ключа зашифровывает 32 самых правых бит R_{i-1}, чтобы получить на выходе 32-битовый результат. Эта функция содержит, как это показано на 4 составляющие: операция XOR, P-бокс расширения, группу S-боксов и прямой бокс.



P-**бокс расширения** служит для расширения 32-битового блока R_{i-1} до 48 битов, чтобы согласовать его размеров с размерами подключа раунда. Блок R_{i-1} делится на 8 секций по 4 бита. Каждая секция расширяется до 6 бит. (*Для секции значения входных битов в позициях 1, 2, 3 и 4 присваиваются битам в позициях 2, 3, 4 и 5 соответственно на выходе. 1-ый выходной бит равен входному 4-му биту предыдущей секции; 6-ой бит выхода равен 1-му биту следующей секции. Если секции 1 и 8 рассматривать как соседние секции, то те же самые правила применяются к битам 1 и 32*).

Хотя отношения между входом и выходом могут быть определены математически, Р-бокс задают таблицей.

P-бокс расширения в i-м раунде					
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

(число выходов 48, диапазон значений – от 1 до 32. Некоторые входные биты порождают несколько выходных).

После расширения DES использует операцию XOR над расширенной частью правого полублока R_{i-1} и ключом раунда k_i.

После суммирования с битами ключа блок из 48 битов делится на 8 последовательных 6-битовых векторов $b_1, b_2, ..., b_8$, каждый из которых заменяется на 4-битовий вектор b'_j с помощью S-***боксов***.

S-боксы																	
		l – номер столбца															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
номер строки ***m***	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S_1
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
	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	S_2
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
	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	S_3
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
	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	S_4
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
	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	
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	S_5
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
	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	S_6
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
	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	S_7
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
	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	S_8
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
	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	

I вектор b_1 попадает в бокс S_1, II вектор b_2 – в бокс S_2. S-бокс – это таблица из 4 строк и 16-ти столбцов. Чтобы в S-боксе найти шифрообозначения вектора, надо:

- из 1-го и последнего битов вектора образовать двоичное число, перевести его в десятичную систему. Это будет номер строки m ($m = 0,1,2,3$);

- из 2-го, 3-го, 4-го и 5-го бита вектора образовать двоичное число, перевести его в десятичную систему. Это будет номер столбца l ($l = 0,...,15$);
- Число, стоящее в S-боксе на пересечении m-ой строки и l-го столбца, будет шифрообозначение b'_j вектора b_j;
- шифрообозначение b'_j перевести в двоичную систему. Ответ готов.

Пример. Найти шифрообозначение вектора $b_1 = 101011$

Р е ш е н и е: на вход S_1-бокса подано число 101011.

Номер строки - $11_2 = 3_{10}$;

Номер столбца - $0101_2 = 5_{10}$.

По таблице подстановки для S_1-бокса находим, на пересечении 3-ей строки и 5-го столбца число $6_{10} = 0110_2 \Rightarrow 0110$ – шифрообозначение для 101011.

ДЗ: Найти шифрообозначение, если на вход: а) S_1-бокса поступило 100011; б) S_8-бокса поступило 000000 (а)1100; б) 1101);

S-боксы – нелинейные и дают системе DES больше криптостойкости, нежели другие операции.

Принципи построения S-боксов:

• каждая строка S-бокса – некоторая перестановка чисел $\{0;1;2;...;15\}$;

• S-бокси не могут быть линейными или аффинными преобразованиями входных данных;

• изменение одного бита входных данных должно на выходе из S-бокса изменить хотя бы 2 бита;

• если на вход S-бокса поступил вектор $\bar{x}$, а потом вектор $\bar{y} = \bar{x} \oplus (0,0,1,1,0,0)$, то векторы $S(\bar{x})$ и $S(\bar{y})$ на выходе должны отличаться хотя бы 2 битами. Также $S(\bar{x}) \neq S(\bar{x} \oplus 11bc00)$, где b и c – любые биты

Таким образом, после S-боксов мы получаем 8 4-битовых векторов $b'_1, b'_2, ..., b'_8$, которые опять объединяют в 32-битовый блок. Далее биты блока перетасовываются в прямом P-боксе на основе заданной таблицы (правила пользования таблицей перестановки старые: например, 7-ой бит входа станет 2-ым битом выхода).

P-бокс прямой в i-м раунде							
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

После 16-го раунда DES правый и левый блоки уже не меняются местами, а объединяются в блок $R_{16}L_{16}$ и подвергаются финальной перестановке IP^{-1}.

4) Генерация ключей

DES создает 16 раундовых ключей k_i по 48 битов из ключа k шифра на 56 битов. Однако, чтобы задать ключ шифра надо среди 56 битов ключа дополнительно вписать 8 битов в позиции $8,16,...,64$ для проверки четности таким образом, чтобы каждый байт содержал нечетное число единиц. С помощью этой операции выявляют ошибки при обмене и хранении ключей.

Ключевое расписание состоит из этапов:

1). Перестановка сжатия для **удаления битов проверки –** из 64-битового ключа удаляют биты 8,16,24, 32,…,64 и переставляет остальные биты согласно таблице (в ходе перестановки сохраняется нумерация битов расширенного ключа).

Удаление проверочных битов ключа+перестановка														
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	

2) После перестановки 56 битов ключа делятся на два блока C_0 и D_0 по 28 бит каждый. Дале для генерации раундовых ключей из блоков C_0 и D_0 с помощью операции циклического сдвига влево на 1-2 бита строятся блоки C_i и D_i, $i = 1,2,...,16$. В раундах 1,2,9 и 16 смещение – на 1 бит, в других раундах — на 2 бита. После определения блоков C_i и D_i биты этих блоков объединяются в один ключ на 56 битов.

Левый циклический сдвиг															
Раунд	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Величина сдвига	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

3). Перестановка сжатия (P-бокс, таблица) изменяет 56 битов на 48 битов, которые образуют раундовый ключ.

Перестановка сжатия ключа							
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

При расшифровании – раундовые ключи теже, что и при зашифровании, но теперь они используются в обратном порядке !

Пример.

Шифрование

Открытый текст: 123456ABCD 132536 Ключ: AABB09182736CCDD			
После первоначальной перестановки: 14A7D67818CA18D После разбиения: L_0= 14A7D678; R_0 = 18CA18D			
Раунд	Левый полублок	Правый полублок	Подключ 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
После объединения: 19BA9212 CF26B472			
Зашифрованный текст: C0B7A8D05F3A829C (после конечной перестановки)			

Расшифрование

Ключ 1-го раунда расшифрования — такой же как ключ 16-го раунда зашифрования.

Зашифрованный текст: C0B7A8D05F3A829C			
После первоначальной перестановки: 19BA9212 CF26B472 После разбиения: L_0= 19BA9212 R_0 = CF26B472			
Раунд	Левая	Правая	Ключ раунда

Раунд 1	CF26B472	BD2DD2AB	181C5D75C66D
Раунд 2	BD2DD2AB	387CCDAA	3330C5D9A36D
.....	.....	.....	.....
Раунд 15	5A78E394	18CA182AD	4568581ABCCE
Раунд 16	19BA9212	18CA18AD	194CD072DE8C
После объединения: 14A7D67818CA18D			
Исходный текст: 123456ABCD 132536 (после конечной перестановки)			

§7. АНАЛИЗ S И P-БОКСОВ

S-боксы – единственное используемое в DES нелинейное преобразование, определяющее основные свойства алгоритма. Долгое время различные исследователи пытались раскрыть тайну создания S-боксов. Но только после появления дифференциального криптоанализа IBM раскрыла критерии проектирования S- и P-боксов.

Некоторые критерии проектирования S-боксов:

- в каждый S-бокс входит 6 битов, а выходит 4 (это самый большой размер, который мог быть реализован в одной микросхеме на 1974г.);
- S-боксы не могут быть линейными или аффинными преобразованиями;
- замена одного бита на входе в S-бокс должна вызывать на выходе изменение по крайней мере двух бит;
- если на вход S-бокса попал вектор x, а потом вектор $y = x \oplus (001100)$, то на выходе из бокса векторы $S(x)$ і $S(y)$ должны отличаться хотя бы двумя битами;
- если на вход S-бокса попал вектор x, а потом вектор $y = x \oplus (11bc00)$, где b, c – произвольные биты, то $S(x) \neq S(y)$;
- и др.
-

Некоторые критерии проектирования P-боксов:

- каждый выходной бит из S-бокса не может попасть в тот же S-бокс в следующем раунде;
- 4 бита от каждого S-бокса идут в 6 различных S-боксов (в следующем раунде);
- ни один из выходных битов S-бокса не идет в тот же самый S-бокс (в следующем раунде);
- и др.

Но, оказалось, что S-боксы DES не так уж случайны. Если описать вход-выход S-бокса как $S(x_1x_2x_3x_4x_5x_6) = y_1y_2y_3y_4$, то во всех

S-боксах наблюдается высокая корреляция между битом x_2 на входе и комбинацией $y_1 \oplus y_2 \oplus y_3 \oplus y_4$.

§8. ЛАВИННЫЙ ЭФФЕКТ

Лавинный эффект – проявление зависимости всех выходных битов шифротекста от каждого входного бита открытого текста (в криптографии такой анализ проводят для блочных шифров и хэш-функций). Лавинный эффект проявляется в зависимости всех выходных битов от каждого входного бита. Термин введен Фейстелем, хотя концептуальное понятие использовалось еще Шенноном. Если криптографический алгоритм не обладает лавинным эффектом в достаточной степени, противник может сделать предположение о входной информации, основываясь на выходной информации. Таким образом, достижение лавинного эффекта является важной целью при разработке криптографического алгоритма.

Криптоалгоритм удовлетворяет ***лавинному критерию***, если при изменении одного бита на входе алгоритма изменяется в среднем половина битов на выходе алгоритма. Если же при изменении одного бита на входе каждый бит на выходе изменяется с вероятностью ½, то криптоалгоритм удовлетворяет ***строгому лавинному критерию***.

В DES лавинный эффект проявляется уже на 4-5 раунде. Так, если зашифровать на одном ключе с помощью DES, 2 блока открытого текста, отличающиеся одним битом, то блоки шифротекстов будут отличаться на 29 бит, т.е. изменение открытого текста на 1,5% вызывает 45% изменений шифротекста.

Изменение битов по раундам:

Число измененных бит																
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

Доказано, что после того как текст зашифрован в 8 раундах, каждый бит шифрованного текста — функция каждого бита открытого текста и ключа. Но эксперименты показывают, что DES с менее чем 16 раундами более уязвимы к атакам на основе открытого текста, чем к атаке грубой силы, требующей использования 16 раундов.

§9. ПРОБЛЕМЫ КЛЮЧЕЙ DES

1) Самая серьезная проблема DES –***размер ключа*** (56 битов). Чтобы предпринять атаку грубой силы, надо проверить 2^{56} ключей.

- Если проверять 10^6 ключей/сек., то потребуется ≈ 2000 лет, чтобы выполнить атаку грубой силы на DES на одном процессоре.
- Если сделать компьютер с миллионом крипточипов, то все множество ключей проверится за 20 часов. Когда был введен DES,

стоимость такого компьютера была немного более миллиона долларов, но она быстро снизилась. Такой специальный компьютер был создан в 1998 году и нашел ключ за 56 часов.

○ С помощью компьютерных сетей моделируют параллельный поиск ключей, что также ускоряет вскрытие.

$\Rightarrow$ ключ 56 битов не обеспечивает достаточной безопасности. (решение проблемы – использование тройного DES(3DES) с двумя ключами (112 битов) или тройного DES с тремя ключами).

Ключевое расписание ключей для раундов допускает слабые ключи. При генерировании раундовых ключей 56-битный ключ шифра делится на две половины и каждая из них сдвигается независимо. Если ключ шифра состоит только из 0 или 1 или, если одна его половина из 0, а другая – из 1, то в этом случае раундовые ключи оказываются попарно одинаковыми, т.е. $k_1 = k_{16}$, $k_2 = k_{15}$ и т.д., и процедуры шифрования/дешифрования оказываются идентичными. Формально ***слабым ключом*** DES называется такой 56-битный ключ K, при котором $E_K(E_K(X)) = X$, где X – 64-битный блок открытого текста. Среди всех 2^{56} возможных ключей DES имеется 4 слабых.

Слабые ключи (в шестнадцатеричной системе исчисления)	
До удаления проверочных бит (64 бита)	Действующий ключ (56 бит)
0101 0101 0101 0101	0000000 0000000 = $[0]^{28}[0]^{28}$
1F1F 1F1F 1F1F 1F1F	0000000 FFFFFFF = $[0]^{28}[1]^{28}$
E0E0 E0E0 E0E0 E0E0	FFFFFFF 0000000 = $[1]^{28}[0]^{28}$
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF = $[1]^{28}[1]^{28}$

$[0]^{28}$ ($[1]^{28}$) обозначает вектор, состоящий из 28 нулей (единиц).

Для алгоритма DES кроме слабых ключей существуют еще ***полуслабые ключи*** – это пара 56-битных ключей K_1, K_2, при которой $DES_{K_1}(DES_{K_2}(X)) = X$. Полуслабые ключи дают одинаковый результат при шифровании текстов. Это тоже связано с генерацией раундовых ключей – вместо 16 различных ключей генерируются только 2 различных, которые затем используются 8 раз в алгоритмах. Таких ключей у DES 6 пар.

Пары полуслабых ключей	
До удаления проверочных бит	Действующие ключи (56 бит)
01FE 01FE 01FE 01FE $\leftrightarrow$ FE01 FE01 FE01 FE01	$[01]^{28}[01]^{28} \leftrightarrow [10]^{28}[10]^{28}$
1FE0 1FE0 1FE0 1FE0 $\leftrightarrow$ E0F1 E0F1 E0F1 E0F1	$[01]^{28}[01]^{28} \leftrightarrow 10]^{28}[10]^{28}$

01E0 01E0 01F1 01F1 ↔ E001 E001 F101 F101	$[01]^{28}[0]^{28} \leftrightarrow [10]^{28}[0]^{28}$
1FFE 1FFE 0EFE 0EFE ↔FE1F FE1F FE0E FE0E	$[01^{28}[1]^{28} \leftrightarrow [0]^{28}[1]^{28}$
O11F 011F 010E 010E ↔ 1F01 1F01 0E01 0E01	$[0]^{28}[01]^{28} \leftrightarrow [0]^{28}[10]^{28}$
E0FE E0FE F1FE F1FE ↔ FEE0 FEE0 FEF1 FEF1	$[1]^{28}[01]^{28} \leftrightarrow [1]^{28}[10]^{28}$

Чтобы проиллюстрировать идею, создадим раундовые ключи раунда от 1 пары:

Раунд 1	9153E54319BD	6EAC1ABCE642
Раунд 2	6EAC1ABCE642	9153E54319BD
Раунд 3	6EAC1ABCE642	9153E54319BD
Раунд 4	6EAC1ABCE642	9153E54319BD
Раунд 5	6EAC1ABCE642	9153E54319BD
Раунд 6	6EAC1ABCE642	9153E54319BD
Раунд 7	6EAC1ABCE642	9153E54319BD
Раунд 8	6EAC1ABCE642	9153E54319BD
Раунд 9	9153E54319BD	6EAC1ABCE642
Раунд 10	9153E54319BD	6EAC1ABCE642
Раунд 11	9153E54319BD	6EAC1ABCE642
Раунд 12	9153E54319BD	6EAC1ABCE642
Раунд 13	9153E54319BD	6EAC1ABCE642
Раунд 14	9153E54319BD	6EAC1ABCE642
Раунд 15	9153E54319BD	6EAC1ABCE642
Раунд 16	6EAC1ABCE642	9153E54319BD

Видно, что есть 8 одинаковых раундовых ключей. Кроме того, раундовый ключ первого раунда в первом множестве совпал с 16-м раундовым ключом во втором множестве; раундовый ключ второго раунда в первом множестве – с 15-м раундовым ключом во втором множестве и т. д. Это означает, что ключи инверсны друг другу: $E_{k2}(E_{k1}(P)) = P$.

Еще у DES существуют 48 ***возможно слабых ключей***, создающих только 4 различных раундовых ключа; другими словами, 16 ключей разделены на 4 группы, и каждая группа порождает 4 одинаковых раундовых ключей.

Оценим вероятность случайного выбора слабого, полуслабого или возможно слабого ключа. Общее число вышеупомянутых ключей: 4 + 12 + 48 = 64. Вероятность выбора одного из этих ключей исключительно мала: $64/2^{56}=8,8\cdot10^{-16}$. Поэтому слабость ключей не является существенным недостатком, поскольку в программной или аппаратной

реализации DES достаточно просто запретить использование проблемных ключей.

§10. СВОЙСТВО ДОПОЛНИТЕЛЬНОСТИ DES

Пусть черта означает инверсию битов ($\bar{0}=1, \bar{1}=0$). Найдем условие, при котором шифр Фейстеля обладает ***свойством дополнительности***: $y = E_k(x) \Rightarrow \bar{y} = E_{\bar{k}}(\bar{x})$. Пусть на вход шифра подаются открытый текст x и ключ K. Очевидно, $\bar{x} = x \oplus 1$, $\overline{K} = K \oplus 1$. Тогда $\bar{x} \oplus \overline{K} = x \oplus 1 \oplus K \oplus 1 = x \oplus K$. В шифре Фейстеля шифруемые блоки складывают по модулю 2 с ключом раунда $\Rightarrow E_k(x) = E_{\bar{k}}(\bar{x})$, т. е если в шифре использована операция XOR, то у шифра есть ***свойство дополнительности*** (говорят еще, ***комплементарность ключей***).

Это свойство присуще и алгоритму DES. Поскольку некоторые ключи можно получить из других ключей инверсией битов, то ключевое дополнение упрощает процесс криптоанализа, так как для полного перебора достаточно проверить только половину ключевого пространства, т.е. не 2^{56} , а 2^{55} возможных ключей.

§11. КРИПТОАНАЛИЗ DES

С самого начала использования алгоритма криптоаналитики всего мира прилагали множество усилий для взлома DES. Фактически DES дал невиданный доселе толчок развитию криптоанализа. Вышли сотни трудов, посвященных различным методам криптоанализа именно в приложении к алгоритму DES, а также деталям самого алгоритма и их влиянию на криптостойкость. Можно утверждать, что именно благодаря DES появились целые направления криптоанализа, такие как:

- ***линейный криптоанализ*** – анализ зависимостей между открытым текстом и шифротекстом;
- ***дифференциальный криптоанализ*** – анализ зависимостей между соотношениями двух или более открытых текстов и соответствующих им шифротекстов;
- криптоанализ на связанных ключах – поиск и анализ зависимостей между шифротекстами, полученными на искомом ключе и ключах, связанных предполагаемым соотношением с искомым ключом; однако DES оказался неуязвим к данному виду атак, так как по ключовому расписанию циклический сдвиг битов ключа выполняется на различное число позиций в разных раундах.

DES стойко выдержал 20 лет массового всемирного криптоанализа – десятилетия криптоанализа не привели к обнаружению серьезных

уязвимостей в алгоритме. Основными результатами усилий по взлому DES можно считать следующие:

1. Японский специалист Мицуру Мацуи (Mitsuru Matsui), изобретатель линейного криптоанализа, в 1993 году показал, что вычислить ключ шифра можно методом линейного криптоанализа при наличии у атакующего 2^{47} пар «открытый текст – шифротекст».

2. Криптологи из Израиля, изобретатели дифференциального криптоанализа Эли Бихам (Eli Biham) и Ади Шамир (Adi Shamir) в 1991 году представили атаку, в которой ключ шифрования вычислялся методом дифференциального криптоанализа при условии, что атакующий имеет 2^{47} специально выбранных пар «открытый текст – шифротекст».

В дальнейшем эти атаки были несколько усилены (например, атака линейным криптоанализом при наличии 2^{43} пар вместо 2^{47}), появлялись также новые виды атак на DES (например, атака, позволяющая вычислить ключ высокоточным облучением аппаратного шифратора и последующим анализом ошибок шифрования).

Однако, все эти атаки требуют наличия огромного количества пар «открытый текст – шифротекст», получение которых на практике является настолько трудоемкой операцией, что наиболее простой атакой на DES все еще можно считать полный перебор ключей.