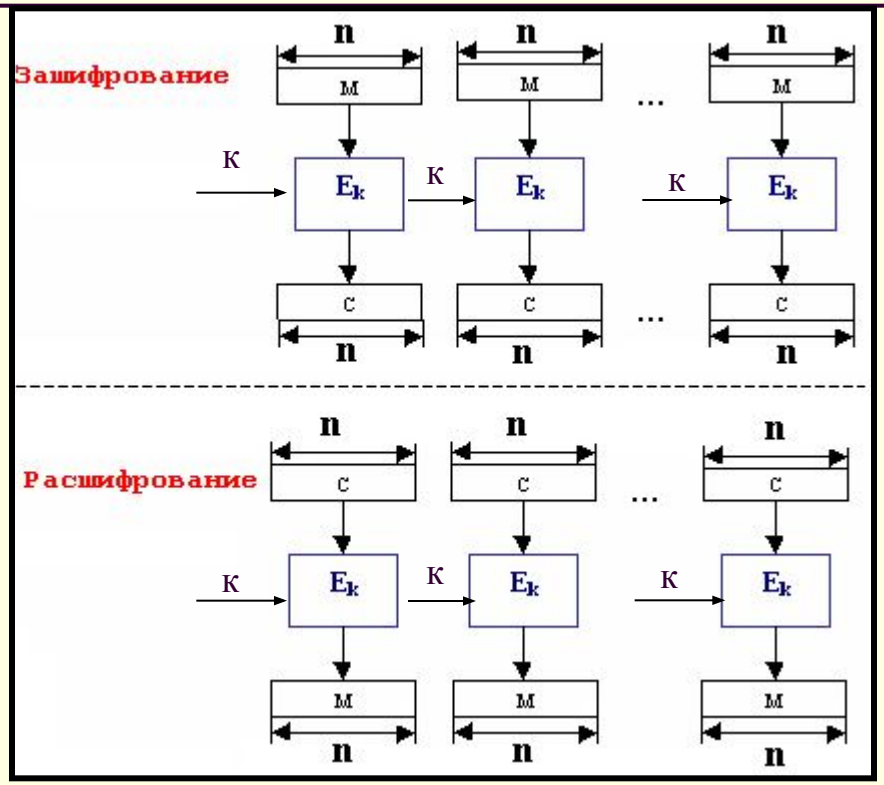
Тема : Блочные алгоритмы

Принципы блочного шифрования
Сравнение блочных и поточных шифров
Предпосылки создания шифра Фейстеля

Блочные шифры



Принцип работы блочного шифра

Блочными называются шифры, в которых логической единицей шифрования является некоторый блок открытого текста, после преобразования которого получается блок шифрованного текста такой же длины.

M – сообщение
C – зашифрованное сообщение
K – ключ шифрования
$E_к$ – функция шифрования с ключом *к*
D_k – функция дешифрования с ключом *к*
n – кол-во бит в блоке, обычно 64 бита

Процедура зашифрования $C= E_k(M)$
Процедура расшифрования $M= D_k(C)$
$D_k(E_k(M))= M$

Поточные шифры

- Поточными называются шифры, в которых поток цифровых данных шифруется последовательно бит за битом или байт за байтом.
- Высокая скорость работы поточных шифров определяет область их использования – закрытые данных, требующих оперативной доставки потребителю, например, аудио- или видеоинформация

Предпосылки создания шифра Фейстеля

Обратимое отображение
n =2

Открытый текст	Шифрованный текст
00	11
01	10
10	00
11	01

Необратимое отображение
n =2

Открытый текст	Шифрованный текст
00	11
01	10
10	01
11	01

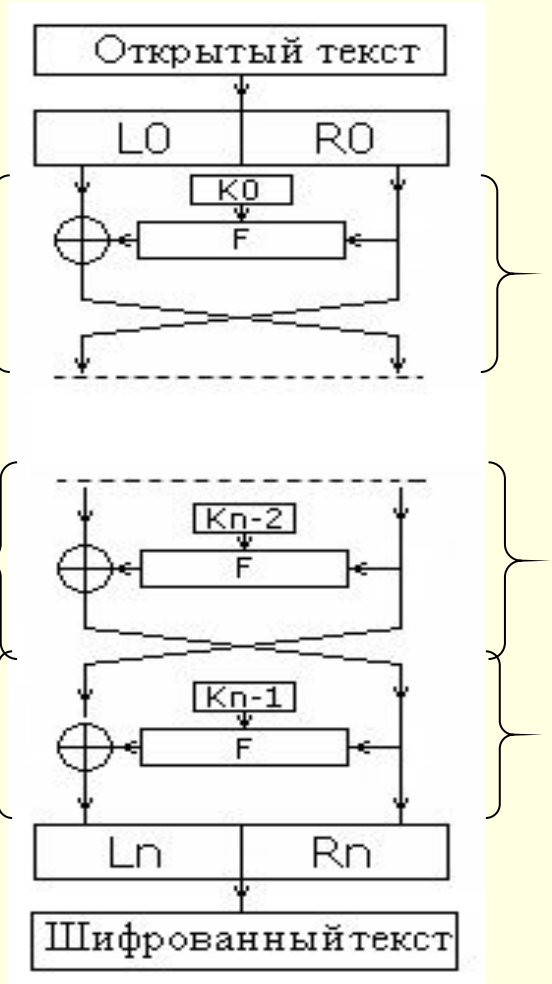
Число различных допустимых преобразований равно 2^n

Фейстель предложил аппроксимировать подстановочный шифр продукционными шифрами, которые строятся на применении операций подстановки и перестановки

Диффузия и конфузия

- Клод Шеннон в 1945 г. предложил идею разработать продукционный шифр в котором попеременно использовались бы функции *конфузии* и *диффузии.*
- Основная задача – воспрепятствовать попыткам криптоанализа, основанного на статистическом анализе сообщения.

Классическая схема Фейстеля



L_0 – левый подблок ОТ
R_0 – правый подблок ОТ
K_n – подключ раунда n
F – функция использующая в качестве исходных данных шифруемый текст и ключ K, зависящий от раунда

Для дешифрования применяется тот же алгоритм, но на вход подается шифрованный текст, а подключи используются в обратном порядке.

Практическая реализация схемы Фейстеля зависит от:

- Размер блока
- Размер ключа
- Число раундов обработки
- Алгоритм вычисления подключей
- Функция раунда
- Скорость выполнения программ шифрования/дешифрования
- Простота анализа

Принципы построения блочных шифров

$Y=E_k(X)$

$X=D_k(Y)$

- $N \rightarrow 2^{n-1}$
- $128 \rightarrow 10^{21}$лет
- Известные открытые и зашифрованные части блоков

Условия стойкого блочного алгоритма

- Функция $\mathbf{E}_{\boldsymbol{k}}\mathbf{(X)}$ должна быть обратимой.
- Не должно существовать иных методов прочтения сообщения **X** по известному блоку **Y**, кроме как полным перебором ключей ***k***.
- Не должно существовать иных методов определения, каким ключом ***k*** было произведено преобразование известного сообщения **X** в сообщение **Y**, кроме как полным перебором ключей.

Представление целых чисел

- 32 бита ->0… 4 294 967 295
- 16+16 -> 2 x 0…65 535
- 8+8+8+8 -> 4 x 0…256

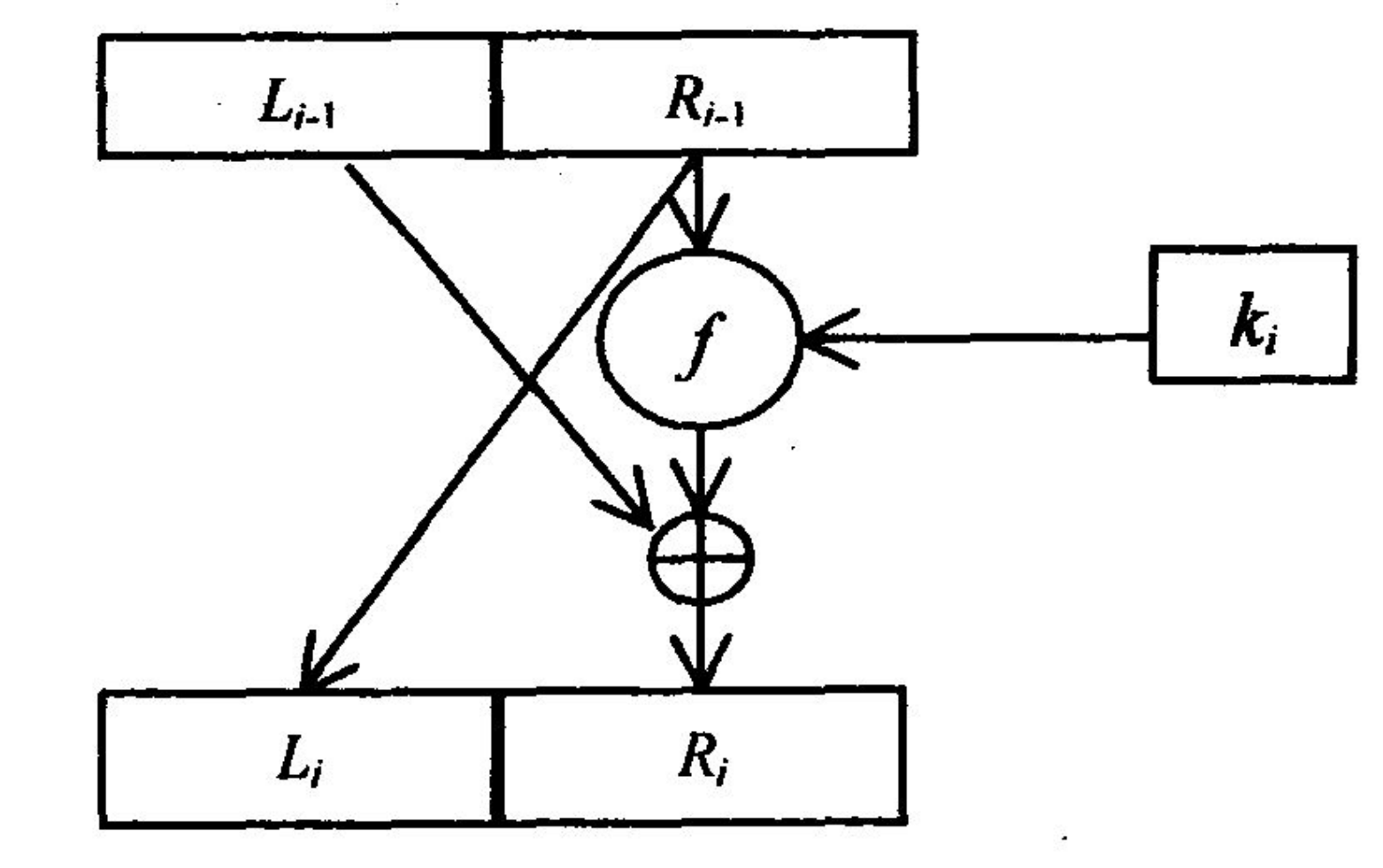
Биективные математические функции

- сложение (**X'=X+V**);
- исключающее ИЛИ (**X'=X XOR V**);
- умножение по модулю 2^{N+1}(**X'=(X*V) mod (2^{N+1})**);
- умножение по модулю 2^{N}(**X'=(X*V) mod (2^{N})**);

Битовые сдвиги

- арифметический сдвиг влево/вправо(**X'=X SHL/SHR V**);
- циклический сдвиг влево/вправо(**X'=X ROL/ROR V**);

Сеть Фейстеля (Feistel Network)



Сеть Фейстеля (Feistel Network)

$$\begin{cases} Y_1 = X_2, \\ Y_2 = X_1 \oplus f_i(X_2, k_i), \end{cases}$$

X – входной блок, разделённый на две половины X_1 и X_2,

(Y_1, Y_2) – результат зашифрования блока **X** на ключе k_i с помощью функции f_i.

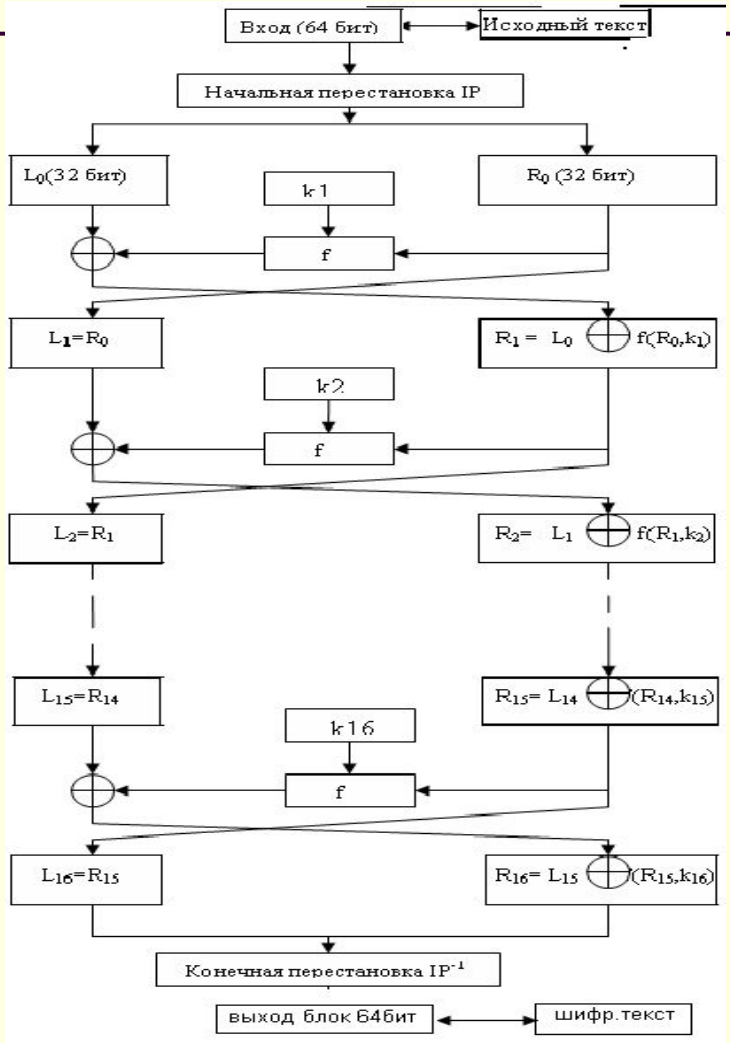
Сеть Фейстеля (Feistel Network)

$$\begin{cases} X_1 = Y_2 \oplus f_i(Y_1, k_i), \\ X_2 = Y_1. \end{cases}$$

Примеры блочных шифров

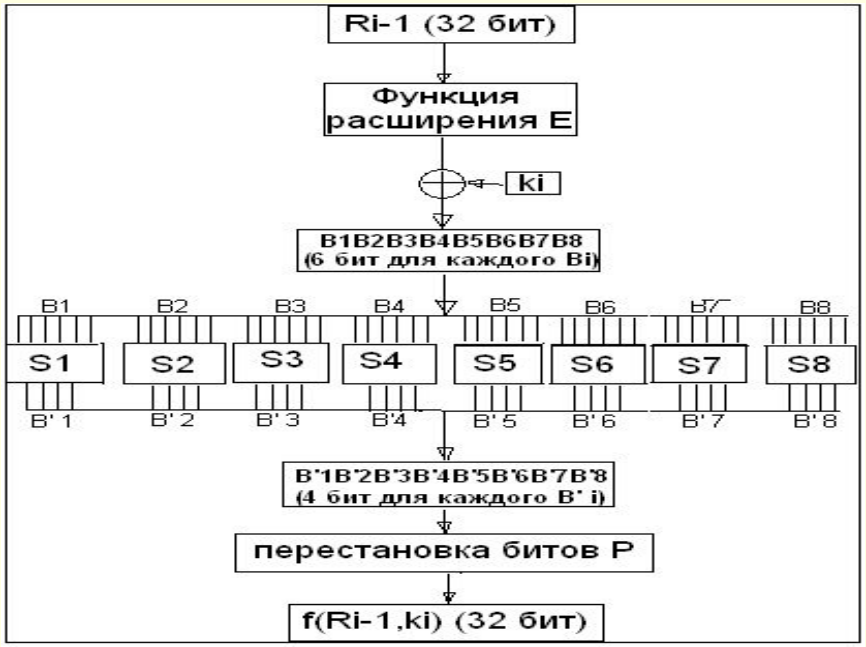
Название алгоритма	Автор	Размер блока	Длина ключа
IDEA	Xuejia Lia and James Massey	64 бита	128 бит
CAST128		64 бита	128 бит
BlowFish	Bruce Schneier	64 бита	128 – 448 бит
ГОСТ	НИИ ***	64 бита	256 бит
TwoFish	Bruce Schneier	128 бит	128 – 256 бит

Алгоритм DES (Data Encryption Standart)



Подробная схема шифрования алгоритма DES

Схема работы функции f



Определение S-матриц алгоритма DES

	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	

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

Лавинный эффект

- Высокая чувствительность результата к изменению начальных данных – любые малые изменения ОТ или ключа приводят к значительным изменениям в шифрованном тексте

Два блока ОТ отличающиеся друг от друга на 1 бит:

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Изменения в открытом тексте	
Раунд	*Число различающихся битов*
0	1
1	3
2	21
3	35
4	39
5	32
6	32
7	31
…	…
16	34

Слабые ключи DES

- Из-за небольшого числа возможных ключей (всего 2^{56}), появляется возможность их полного перебора на быстродействующей вычислительной технике за реальное время. В 1998 году The Electronic Foundation используя специальный компьютер DES-Cracker, удалось взломать DES за 3 дня.
- В алгоритме DES существуют слабые и частично-слабые ключи.
- Слабыми ключами называется ключи k такие что $DES_k(DES_k(x)) = x$, где x — блок 64 бит.
- Частично-слабые ключи — пары ключей (k_1,k_2) такие что $DES_{k1}(DES_{k2}(x)) = x$
- Известны 4 слабых ключа, они приведены в таблице 9. Для каждого слабого ключа существует 2^{32} «постоянные точки», то есть таких 64-битовых блоков x, в которых $DES_k(x) = x$

Слабые ключи
0101-0101-0101-0101
FEFE-FEFE-FEFE-FEFE
1F1F-1F1F-0E0E-0E0E
E0E0-E0E0-F1F1-F1F1

Пары частично-слабых ключей	
01FE-01FE-01FE-01FE –	FE01-FE01-FE01-FE01
1FE0-1FE0-1FE0-1FE0 –	E0F1-E0F1-E0F1-E0F1
01E0-01E0-01F1-01F1 –	E001-E001-F101-F101
FFE-1FFE-0EFE-0EFE –	FE1F-FE1F-FE0E-FE0E
O11F-011F-010E-010E –	1F01-1F01-0E01-0E01
E0FE-E0FE-F1FE-F1F –	FEE0-FEE0-FEF1-FEF1

Увеличение криптостойкости DES

Чтобы увеличивать криптостойкостьЧтобы увеличивать криптостойкость DES появляются несколько вариантов: double DESЧтобы увеличивать криптостойкость DES появляются несколько вариантов: double DES (2DESЧтобы увеличивать криптостойкость DES появляются несколько вариантов: double DES (2DES), triple DESЧтобы увеличивать криптостойкость DES появляются несколько вариантов: double DES (2DES), triple DES (3DESЧтобы увеличивать криптостойкость DES появляются несколько вариантов: double DES (2DES), triple DES (3DES), DESXЧтобы увеличивать криптостойкость DES появляются несколько вариантов: double DES (2DES), triple DES (3DES), DESX, G-DES.

Методы 2DESМетоды 2DES и 3DESМетоды 2DES и 3DES основаны на DES, но увеличивают длину ключей (2DES — 112 бит, 3DES — 168 бит) и поэтому увеличивается криптостойкость.

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. Это вариант известен как в EEE так как три DES операции являются шифрованиC = E_{k3} E_{k2}^{-1} (E_{k1}(P))па алгоритма 3DP = E_{k1}^{-1}(E_{k2}(E_{k3}^{-1}(C)))

DES-EEE3: Шифруется три раза с 3 разными ключами.

DES-EDE3: 3DES операции шифровка-расшифровка-шифровка с 3 разными ключами.

DES-EEE2DES-EEE2 и DES-EDE2DES-EEE2 и DES-EDE2: Как и предыдущие, за исключением того, что первая и третья операции используют одинаковый ключ.

Самый популярный тип при использовании 3DES — это DES-EDE3, для него алгоритм выглядит так: ЗашифрованиеСамый популярный тип при использовании 3DES — это DES-EDE3, для него алгоритм выглядит так: Зашифрование: Расшифрование: