Основы построения симметричных блочных алгоритмов криптографии

2015

Содержание

- Симметричные шифры
- Классификация симметричных шифров
- Принцип Кирхгофа
- Условия стойкости
- Блочные шифры
- Архитектура блочных шифров
  - Перестановки
  - Замены
  - Функциональные преобразования
- Пример составной системы
- Сети Файстеля

Содержание

Алгоритм DES

- Описание
- Общая схема
- Раундовая функция
- Выработка подключей

Режимы шифрования

- Режим ECB
- Режим CBC
- Режим CFB
- Режим OFB

Заключение

Литература

Вопросы? Комментарии?

Симметричные шифры

Одно-ключевые (*симметричные*) шифры – криптографические алгоритмы процессы за- и расшифровывание в которых отличаются лишь порядком выполнения и направлением некоторых простых шагов (в отличие асимметричных или алгоритмов с открытым ключом).

Особенности симметричных шифров:

- каждый из участников обмена может как зашифровать, так и расшифровать сообщение;
- необходима специальная служба для изготовления и доставки секретных ключей;
- позволяют защищать сообщения от *прочтения* и некоторых видов *модификации*.
- не позволяют подтверждать или опровергать *авторство* сообщений.

Классификация…

Требования, предъявляемые к практическим симметричным алгоритмам шифрования:

- шифр должен быть технически применим для закрытия массивов данных произвольного объема;
- шифр должен быть эффективно реализуем в виде устройства, имеющего ограниченный объем памяти.

Следовательно, криптоалгоритм, должен быть пошаговым – сообщение разбивается на блоки ограниченного размера, и за один шаг шифруется один блок:

$$P = (P_1, P_2, \ldots, P_n), |P_i| \le N, \text{ для } i \text{ от 1 до } n,$$

где N — максимальный размер блока.

Практически всегда размер блока полагают постоянным: $|P_1| = |P_2| = \ldots = |P_{n-1}| = N, |P_n| \le N,$

Классификация шифров

В *блочных шифрах* результат зашифрования очередного блока зависит только от него самого и не зависит от других блоков шифруемого массива данных:

$$C_i = E(P_i).$$

В результате зашифрования двух одинаковых блоков открытого текста одним алгоритмом получаются идентичные блоки шифротекста:

$$P_i = P_j \Rightarrow E(P_i) = E(P_j).$$

В *поточных* или *потоковых шифрах* результат зашифрования очередного блока зависит от него самого и, в общем случае, от всех предыдущих блоков массива данных:

$$C_i = E(P_1, P_2, \ldots, P_i).$$

Классификация шифров

К потоковым шифрам также относится важный частный случай, когда результат зашифрования очередного блока зависит этого блока и от его номера:

$$C_i = E(i, P_i).$$

Иногда потоковыми называют только такие шифры, в которых шифруемый за один шаг блок имеет размер один бит или один символ текста, а шифры с большим размером блока, формально относящиеся к потоковым, причисляют к блочным.

Принцип Кирхгофа

Шифр – параметризованный алгоритм, состоящий из *процедурной части*, и *параметров* — различных элементов данных, используемых в преобразованиях.

Раскрытие только процедурной части не должно приводить к увеличению вероятности успешного дешифрования сообщения злоумышленником выше допустимого предела.

Особого смысла хранить процедурную часть в секрете нет. В секрете держится некоторая часть параметров алгоритма, которая называется *ключом* шифра.

Принцип Кирхгофа

Следствия:

- разглашение конкретного шифра (алгоритма и ключа) не приводит к необходимости полной замены реализации всего алгоритма, достаточно заменить только скомпрометированный ключ;
- ключи можно отчуждать от остальных компонентов системы шифрования — хранить отдельно от реализации алгоритма в более надежном месте и загружать их в шифрователь только по мере необходимости и только на время выполнения шифрования.

Условия стойкости

Условия стойкости блочного шифра (по *К. Шеннону*) *:*

- *рассеивание* – один бит исходного текста должен влиять на несколько битов шифротекста, оптимально — на все биты в пределах одного блока. При шифровании двух блоков данных с минимальными отличиями между ними должны получаться совершенно непохожие друг на друга блоки шифротекста. *Аналогично* и для зависимости шифротекста от ключа — один бит ключа должен влиять на несколько битов шифротекста;
- *перемешивание* – шифр должен скрывать зависимости между символами исходного текста и шифротекста. Если шифр достаточно хорошо «перемешивает» биты исходного текста, то соответствующий шифротекст не содержит никаких статистических, и, тем более, функциональных закономерностей.

Условия стойкости

Если шифр обладает обоими указанными свойствами, то любые изменения в блоке открытых данных приведут к тому, что все биты в зашифрованном блоке данных с вероятностью ½ независимо друг от друга так же поменяют свои значения.

Такой шифр невозможно вскрыть способом, менее затратным с точки зрения количества необходимых операций, чем полный перебор по множеству возможных значений ключа.

Блочные шифры

Если известен закон распределения блоков открытого текста, то, проанализировав статистику блоков шифротекста, можно установить соответствие между ними.

Для того чтобы исключить подобную возможность, размер блока должен быть выбран достаточно большим.

Для большинства современных шифров блок имеет размер 64 бита, для нее исчерпывающий анализ практически исключен из-за невозможности набрать соответствующую статистику шифротекстов.

Архитектура...

Шифр обычно составляют из более простых шифрующих преобразований. *Простое шифрующие преобразование* – преобразование, которое реализуется аппаратно относительно несложной логической схемой или программно несколькими компьютерными командами.

Основные шифрующие преобразования:

- *перестановка* (*permutation*) – перестановка структурных элементов шифруемого блока данных (битов, символов, цифр);
- *замена, подстановка* (*substitution*) – замена группы элементов шифруемого блока на другую группу по индексной таблице;
- *функциональное преобразование* (*function*) – различные сдвиги, логические и арифметические операций.

Перестановки

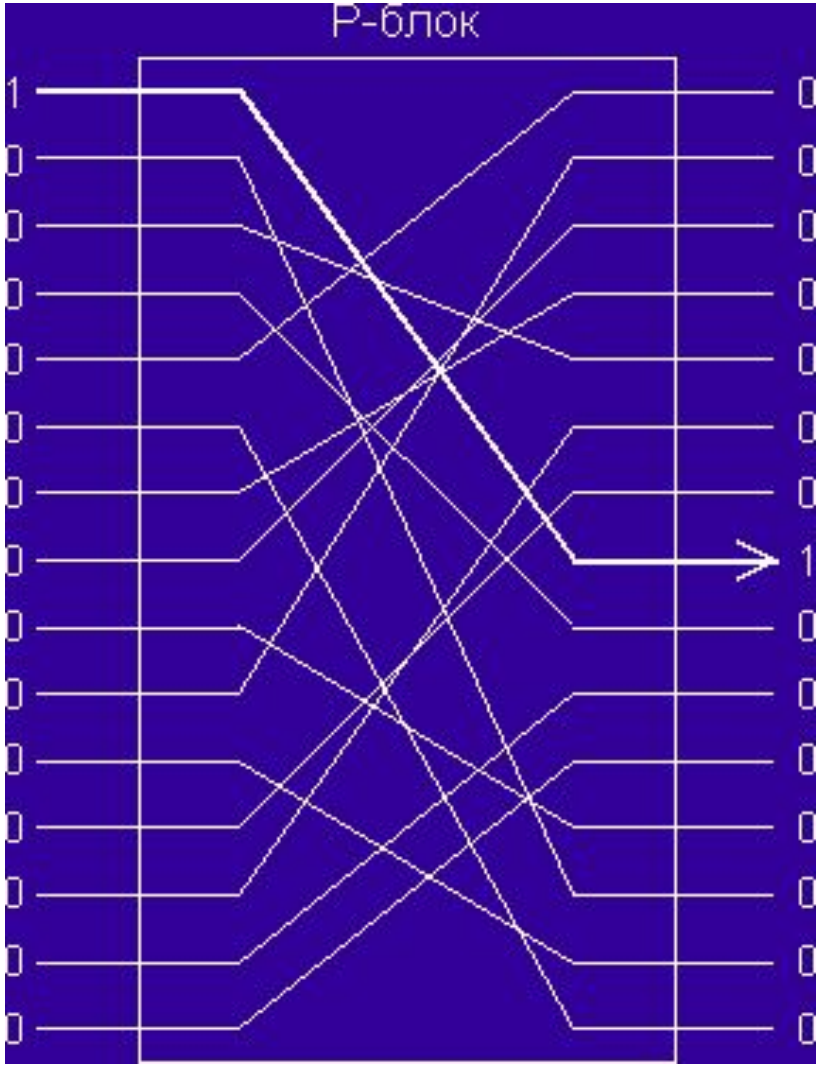
Перестановку можно представить как устройство с n входами и выходами. Имеется $n!$ возможных вариантов коммутации проводов внутри этого устройства (*ключей* блока перестановок).

Свойства блока перестановок:

- легко может быть аппаратно построен для достаточно больших n;
- очень неэффективно реализуются программно на процессорах общего назначения;
- путем использования набора специально сконструированных сообщений (содержащих одну единственную единицу в n-1 различных позициях) можно целиком определить ключ такой системы всего за n-1 операцию.

Перестановки

Пример блока перестановок:



Замены

Замена (подстановка) может быть представлена устройство с n входами и выходами. Устройство содержит мультиплексор и демультиплексор, а также 2^n внутренних соединений их выводов, которые могут быть выполнены $2^n!$ различными способами (*ключ* блока подстановок).

Свойства блока подстановок:

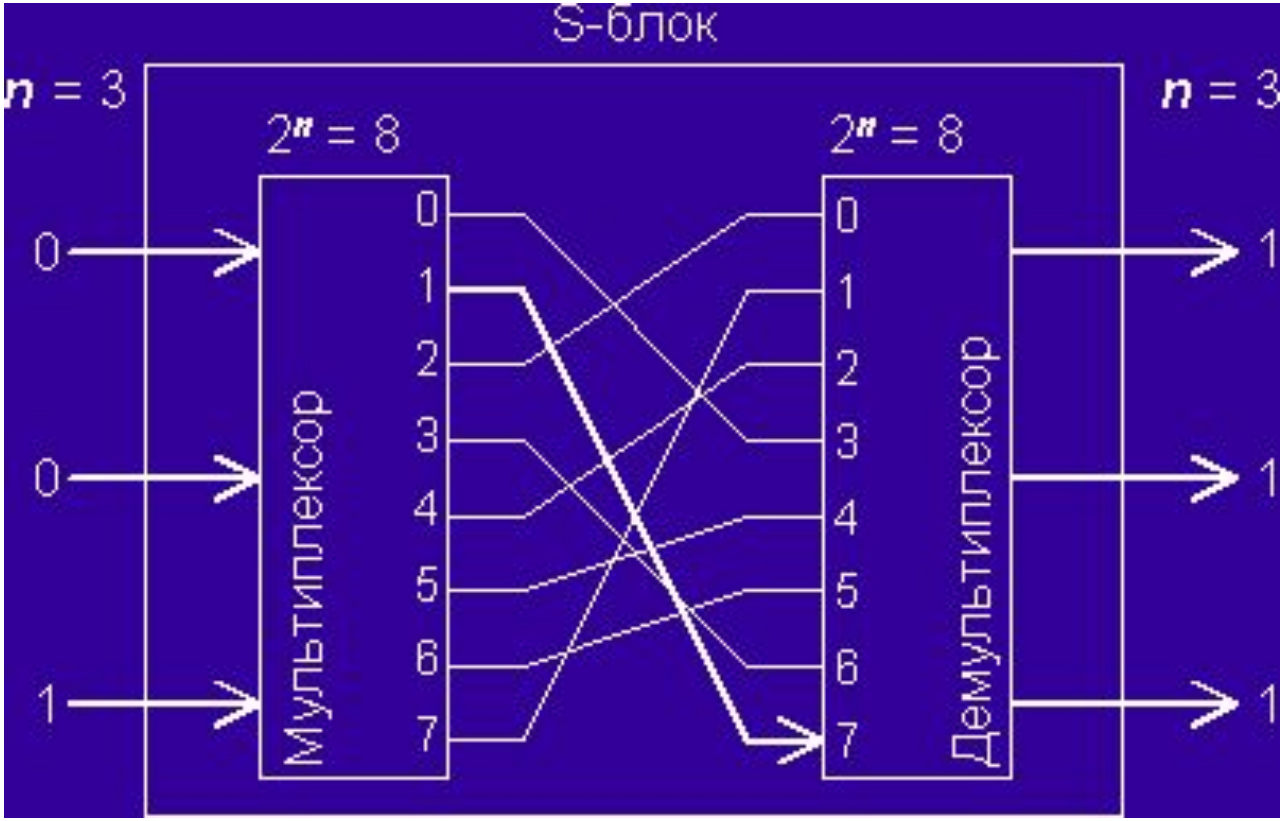
- включает любые линейные и нелинейные преобразования, может заменить любой входной блок цифр на любой выходной блок;
- аппаратно реализуется с помощью запоминающих устройств, программно – индексированным чтением из оперативной памяти, размером (в битах):

$$V = 2^n n;$$

Замены

- путем перебора $2^n - 1$ наборов сообщений можно целиком определить ключ такой системы за $2^n - 1$ операцию. Обычно n – мало и такой анализ реально осуществим.

Пример блока подстановок:



Функциональные…

Функциональные преобразования — унарные и бинарные логические и арифметические операции, реализуемые аппаратно логическими схемами, программно — одной-двумя процессорными командами.

Обычно используют операции, которые имеются в наборах команд универсальных процессоров и реализованы аппаратно в виде микросхем (инверсия, побитовые «и», «или», «исключающее или», изменение знака, сложение, вычитание, умножение, деление по модулю некоторого числа, циклические сдвиги).

Пример составной системы

Составная шифрующая система объединяет S-блоки и P-блоки в единую конструкцию.

P-блоки имеют большое количество входов, а S-блоки – небольшое, легко реализуемое количество входов.

P-блоки тасуют биты, обеспечивая *рассеивание*,

S-блоки выполняют нелинейные преобразования и обеспечивают *перемешивание*.

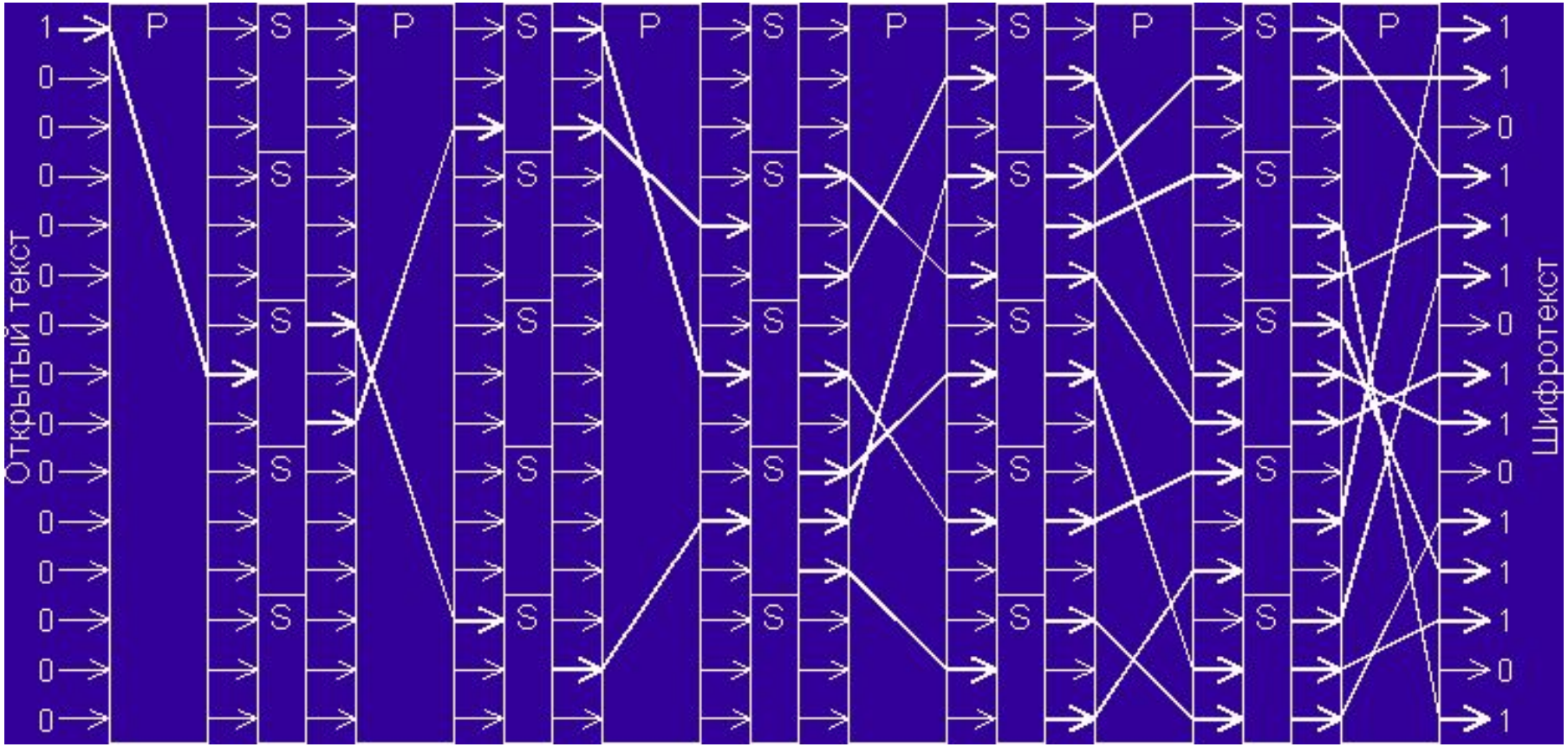
Поскольку S-блоки нелинейны, они потенциально увеличивают число единиц, тогда как P-блоки просто двигают единицы с места на место. Результатом может быть непредсказуемая лавина единиц.

Пример составной системы

Сложность вскрытия составной шифрующей системы, составленной из относительно простых блоков преобразований, будет намного выше, если структура этой системы отлична от линейной и содержит *ветвления*, *циклы* или *обратные связи*.

Компактность реализации алгоритма приводит к идеологии его построения из одинаковых групп преобразований, повторенных нужное число раз (*итеративность*).

Пример составной системы



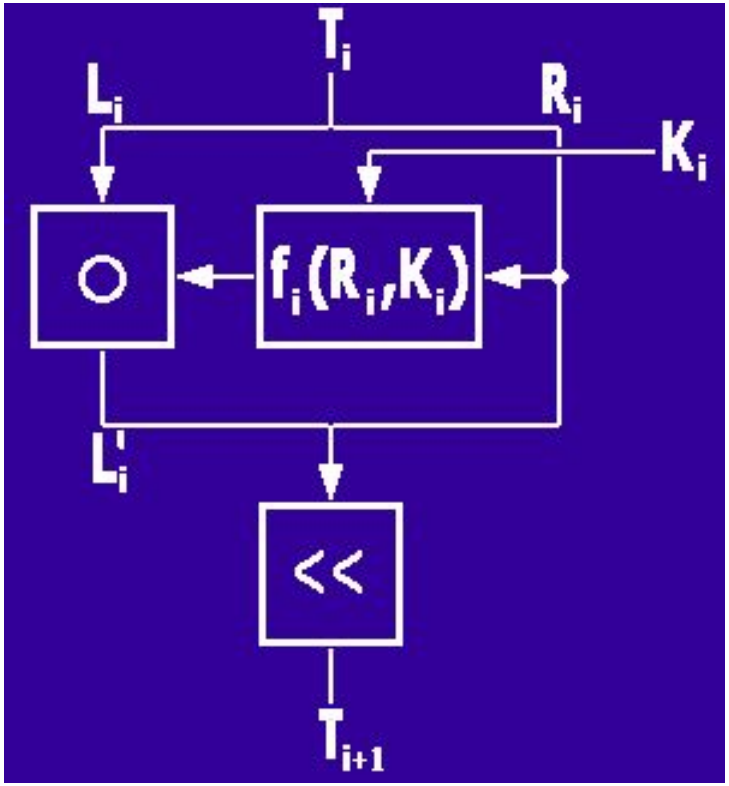
Сети Файстеля

В *сети Файстеля* шифрование блока данных осуществляется путем поочередного преобразования двух подблоков данных с использованием некоторой простой процедуры шифрования, называемой *раундом шифрования* или *раундовой функцией шифрования* f_i.

Для любой (необязательно обратимой) функции f_i расшифрование осуществляется путем выполнения тех же процедур преобразования, но с использованием *подключей* K_i в обратном порядке.

Подключи K_i генерируются из ключа K специальными алгоритмами, разрабатываемыми вместе с шифром.

Сети Файстеля



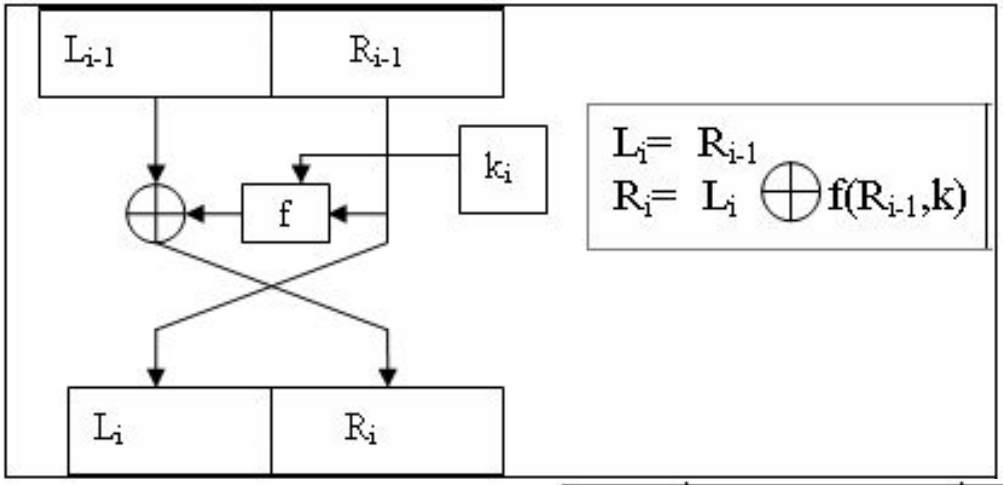
L_i, R_i – левая и правая части входного блока T_i;

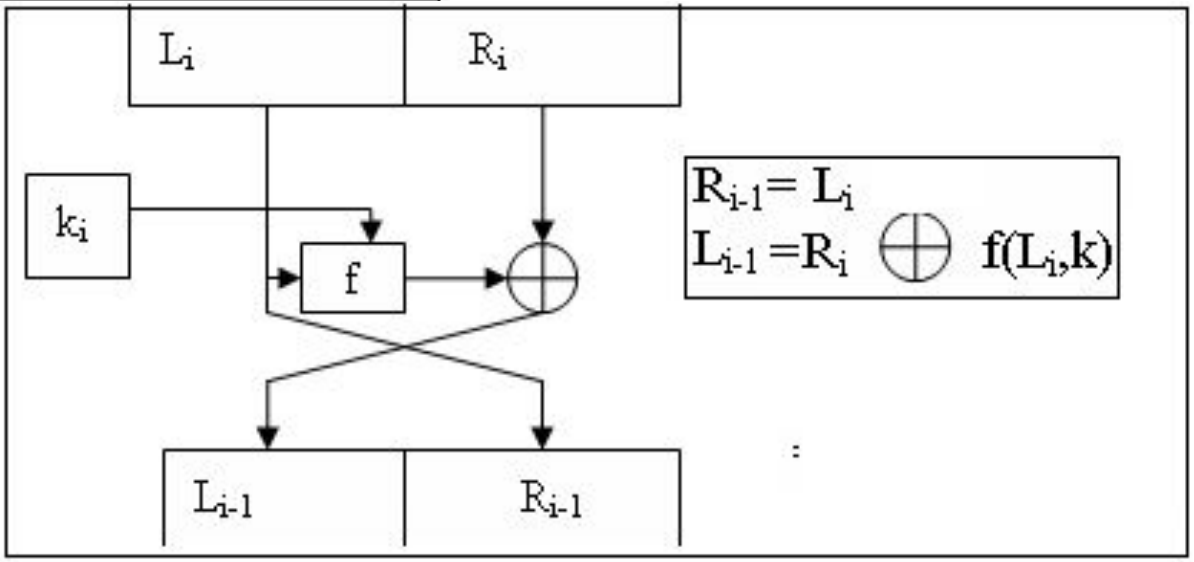
$\circ$ – любая обратимая бинарная операция;

K_i – подключ i-го этапа;

$f_i(L_i, K_i)$ – i-я шифрующая функция;

<< – циклический сдвиг.


L_{i-1}
R_{i-1}
k_i
f
L_i
R_i
$L_i = R_{i-1}$
$R_i = L_i \oplus f(R_{i-1},k)$


L_i
R_i
k_i
f
L_{i-1}
R_{i-1}
$R_{i-1} = L_i$
$L_{i-1} = R_i \oplus f(L_i,k)$

Сети Файстеля

Если последовательность раундовых функций палиндромиальна (т.е. если $f_1 = f_n$, $f_2 = f_{n-1}$, $f_{\lfloor n/2 \rfloor} = f_{n+1-\lfloor n/2 \rfloor}$ и, в частности, если все f_i – одинаковы), то за- и расшифрование различаются только порядком использования подключей. Использование одинаковых функций шифрования позволяет достигнуть *итеративности*.

Размер левой и правой частей блока может изменяться от раунда к раунду, но обычно эти величины постоянны. Если они равны друг другу, то такая схема называется *сбалансированной*, а если нет — то *несбалансированной* сетью Файстеля.

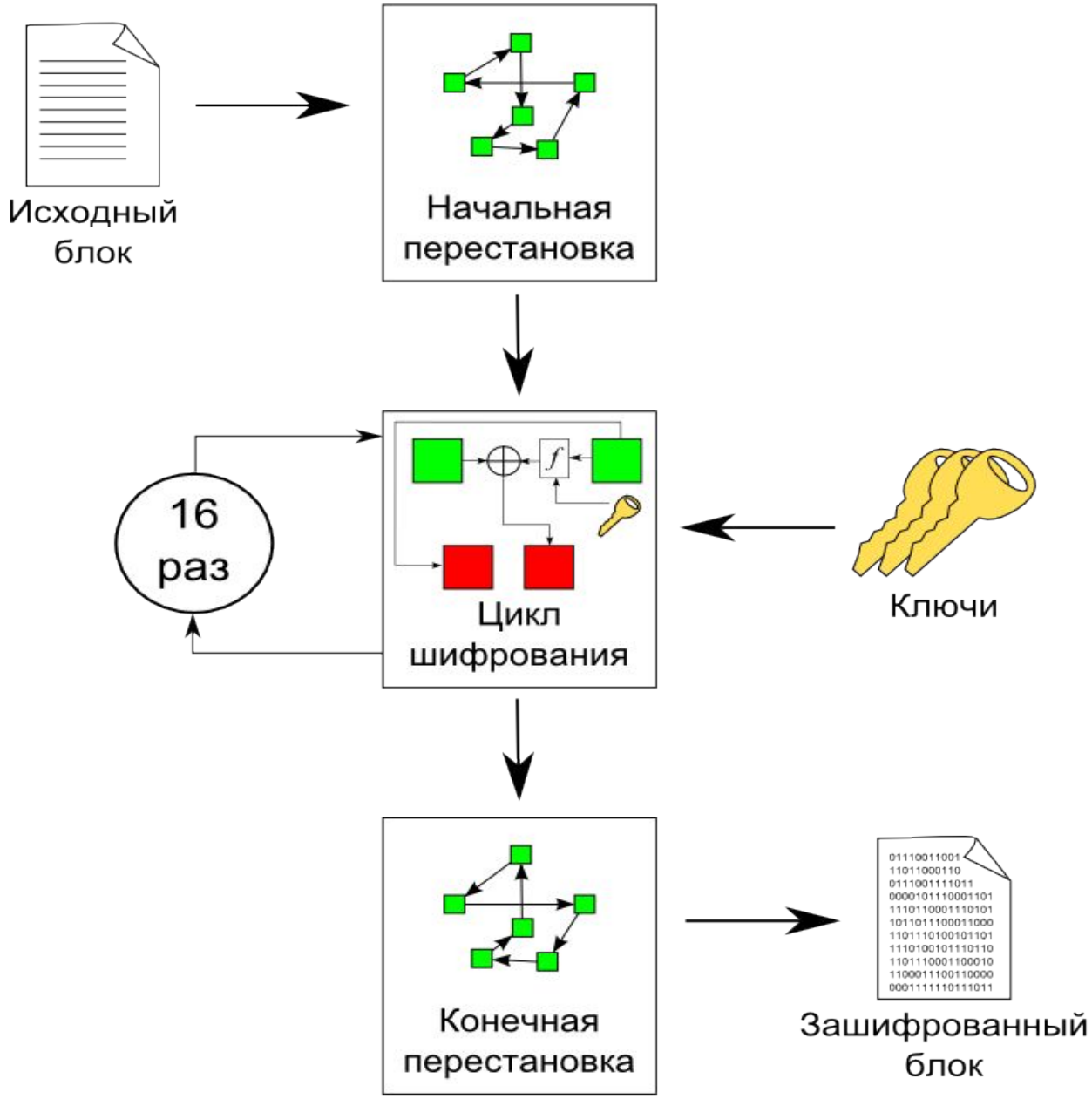
Обычно «°» – операция побитового исключающего ИЛИ, если используется другая подходящая бинарная операция, то сеть Файстеля называется *обобщенной*.

Алгоритм DES
Описание

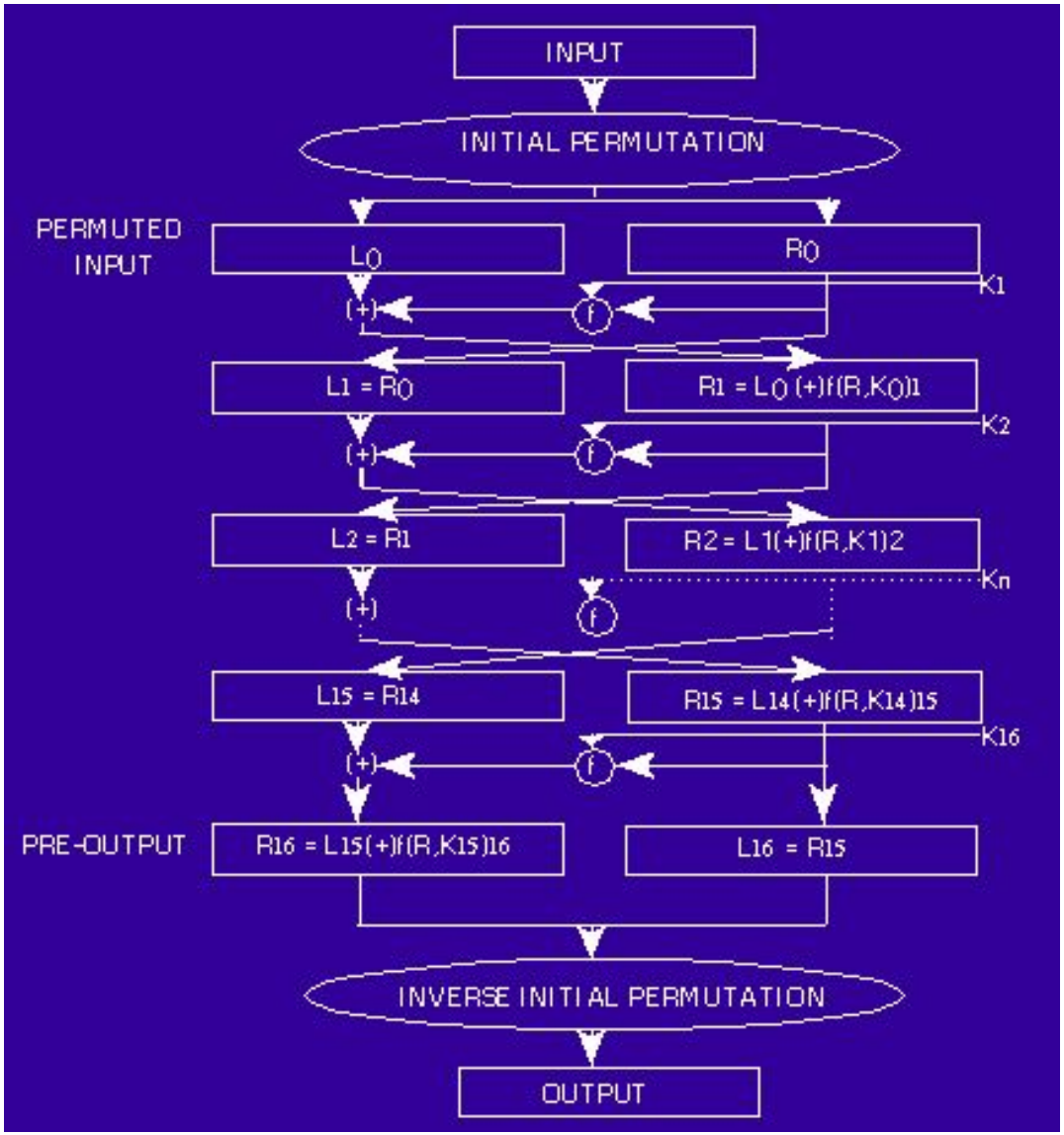
Data Encryption Standard (DES), Data Encryption Algorithm (DEA, DEA-1) – стандарт шифрования данных был принят в качестве криптографического стандарта NIST, ANSI, ISO.

Основные параметры:

- симметричный блочный шифр, размер блока 64 бит;
- длина ключа – 56 бит (обычно используется 64-битное число, но каждый восьмой бит используется для проверки четности и игнорируется);
- представляет собой 16 раундовую сбалансированную сеть Файстеля, плюс две перестановки не влияющих на криптостойкость;
- алгоритм может быть легко реализован аппаратно, программная реализация несколько более сложна.


Исходный блок
Начальная перестановка
16 раз
Цикл шифрования
Ключи
Конечная перестановка
Зашифрованный блок

Общая схема


INPUT
INITIAL PERMUTATION
PERMUTED INPUT
L0
R0
K1
(+)
f
L1 = R0
R1 = L0 (+)f(R,K0)1
K2
L2 = R1
R2 = L1(+)f(R,K1)2
Kn
L15 = R14
R15 = L14(+)f(R,K14)15
K16
PRE-OUTPUT
R16 = L15(+)f(R,K15)16
L16 = R15
INVERSE INITIAL PERMUTATION
OUTPUT

Алгоритм DES

CRYPTO 1: DES-алгоритм шифрования

Шаг 0: Задать 64-битовую последовательность *input*

Шаг 1: $L_0R_0 = IP(input)$

Шаг 2: Повторять шаг 2 для значений n от 1 до 16:

$$L_n = R_{n-1};$$

$$R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$$

Шаг 3: $output = IP^{-1}(R_{16}L_{16})$

Шаг 4: Останов. *Output* - шифрованный 64-битовый блок

Таблица 1. Начальная перестановка IP

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

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

S_8

Таблица 3. Преобразования , i=1...8

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

Таблица 4. Перестановка P

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

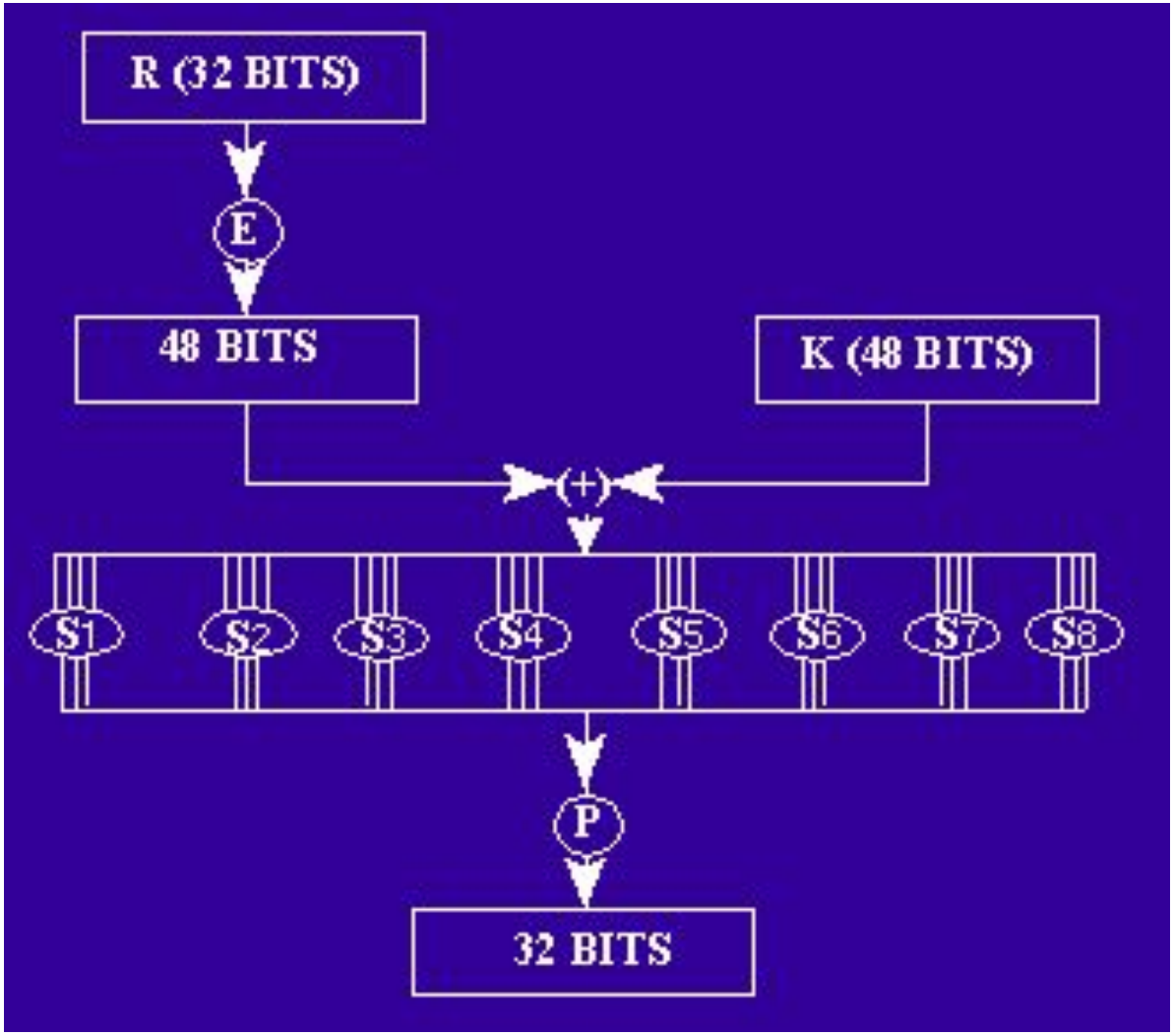
Таблица 5.

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

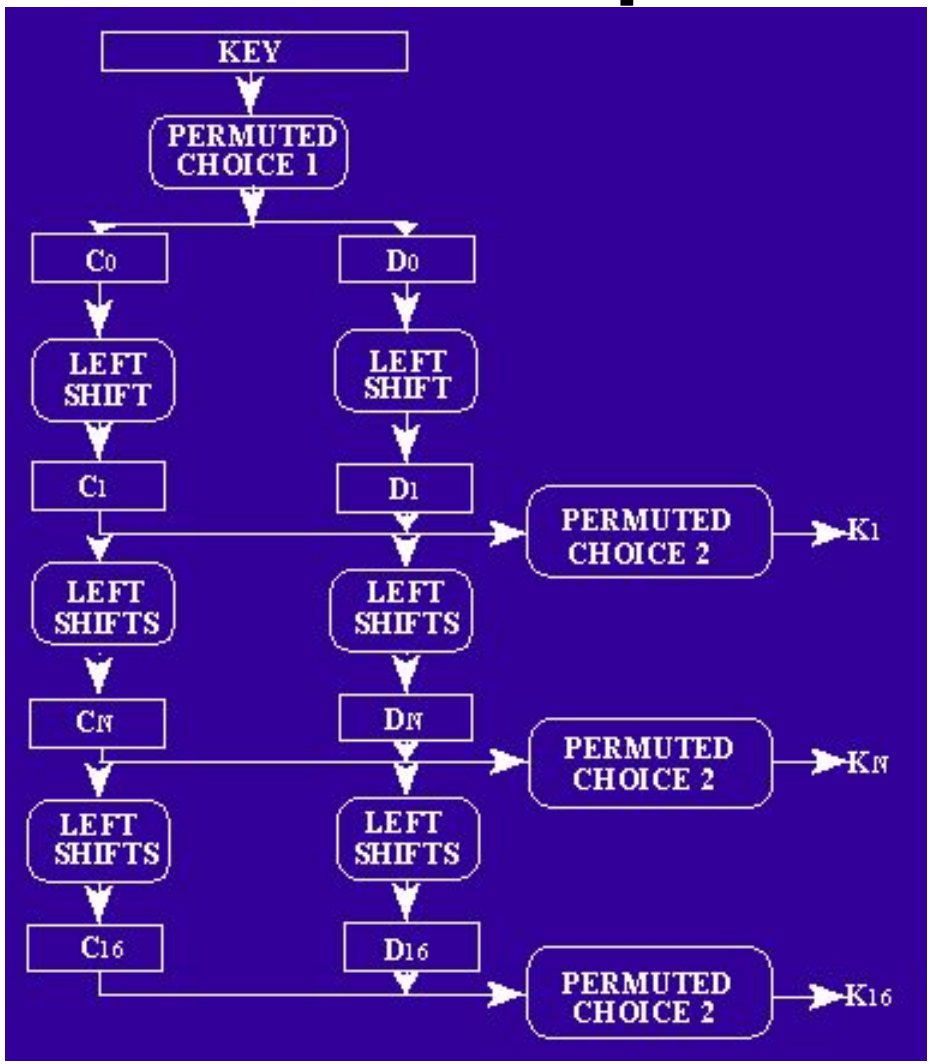
Таблица 6.

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

Алгоритм DES
Раундовая функция



Алгоритм DES
Выработка подключей



Режимы шифрования

Режим шифрования (*криптографический режим*) обычно объединяет базовый шифр, какую-либо обратную связь и ряд простых операций. Безопасность результирующего алгоритма определяется используемым шифром, а не режимом.

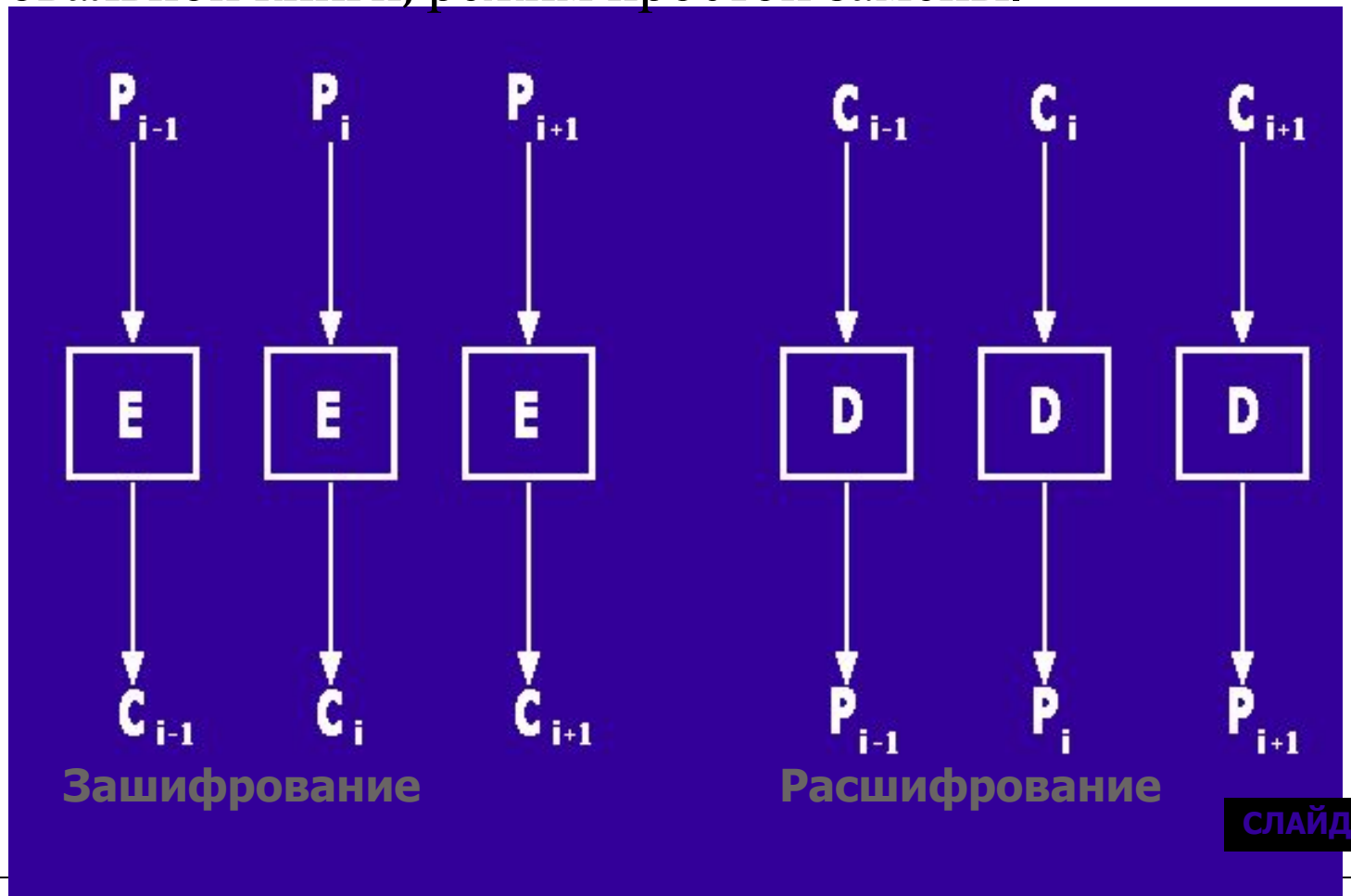
Различные криптографические режимы применяются с целью:

- скрыть структуру открытого текста;
- затруднить манипулирование открытым текстом;
- обеспечить возможность шифрования нескольких сообщений одним ключом;
- обеспечить устойчивость к сбоям, добавлению или потере битов.

Режимы шифрования

Режим ECB

ECB (Electronic Codebook) – режим электронной шифровальной книги, режим простой замены.



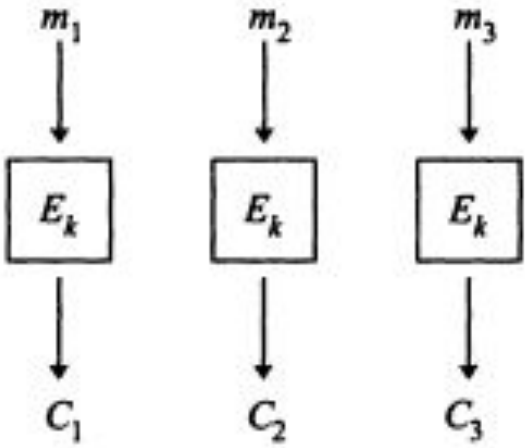


Рис. 5.8. Шифрование в режиме *ECB*

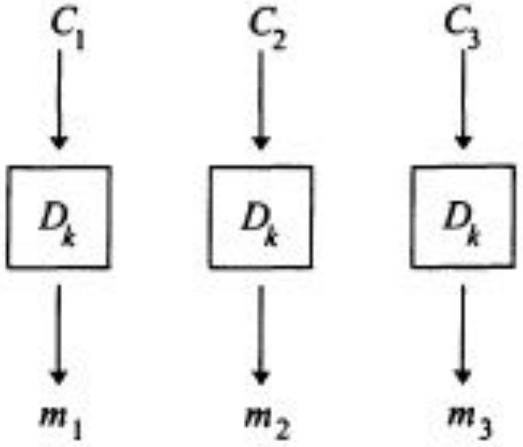


Рис. 5.9. Расшифровывание в режиме *ECB*

Режимы шифрования

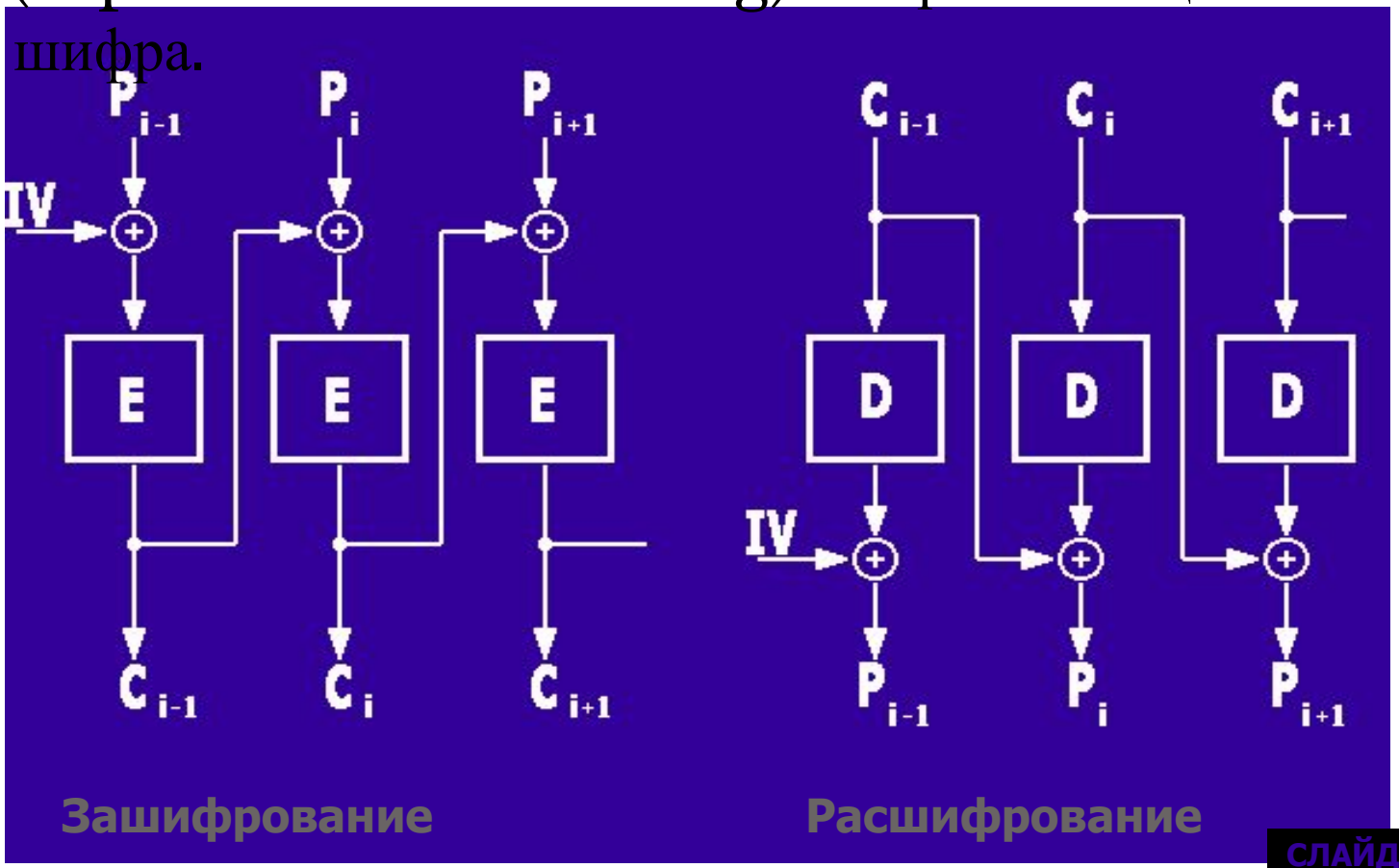
Режим ECB

- Размер сообщения должен быть кратен размеру блока.
- Возможно шифрование файлов с произвольным доступов (например, записей баз данных).
- Ошибка в одном бите шифротекста приводит к неверному расшифрованию соответствующего блока открытого текста.
- При потере (добавлении) битов весь последующий текст становится нечитаемым.
- Возможны удаление, повтор, подмена блоков без знания ключа и даже алгоритма.
- Рекомендуется только для шифрования случайных и небольших по размеру данных, например ключей.

Режимы шифрования

Режим CBC

CBC (Cipher block chaining) – режим сцепление блоков шифра.



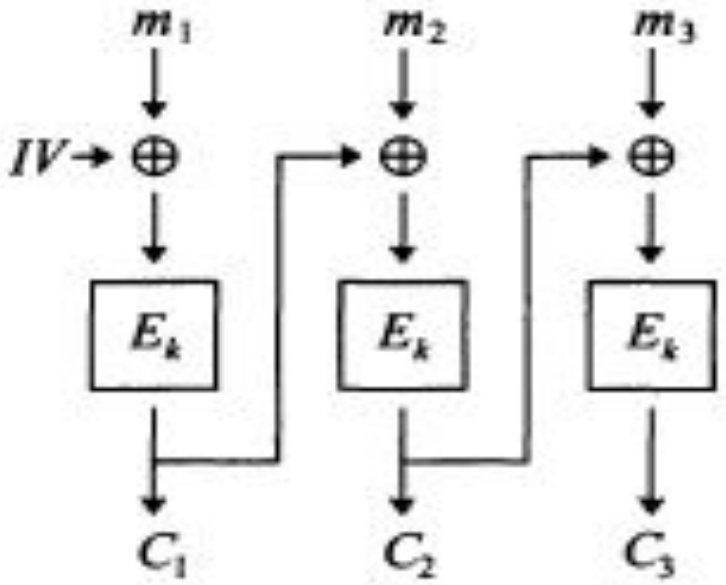


Рис. 5.10. Шифрование в режиме *CBC*

. Шифрование осуществляется согласно формулам

$$C_1 = E_k(m_1 \oplus IV), \qquad C_i = E_k(m_i \oplus C_{i-1}) \text{ при } i > 1,$$

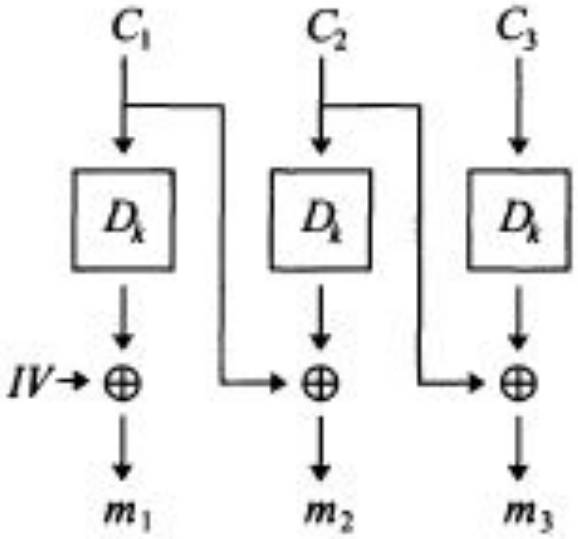


Рис. 5.11. Расшифрование в режиме *CBC*

$$m_1 = D_k(C_1) \oplus IV, \qquad m_i = D_k(C_i) \oplus C_{i-1} \text{ при } i > 1.$$

Режимы шифрования

Режим CBC

- Размер сообщения должен быть кратен размеру блока.
- Шифрование каждого блока зависит от всех предыдущих блоков.
- Два одинаково начинающихся сообщения будут одинаково зашифрованы до первого различия, против этого применяют *вектор инициализации* (*initialization vector*, *IV*) , который не обязательно хранить в секрете.
- Ошибка в одном бите широтекста текста портит этот блок сообщения и один бит следующего блока.
- Потеря (добавление) бита полностью искажают весь открытый текст, начиная с этого блока.
- Возможна подмена последних битов сообщения.
- Отлично подходит для шифрования файлов.

Режимы шифрования

Режим CFB

CFB (Cipher-feedback) – режим обратной связи по шифру, режим гаммирования с обратной связью (с зацеплением блоков).

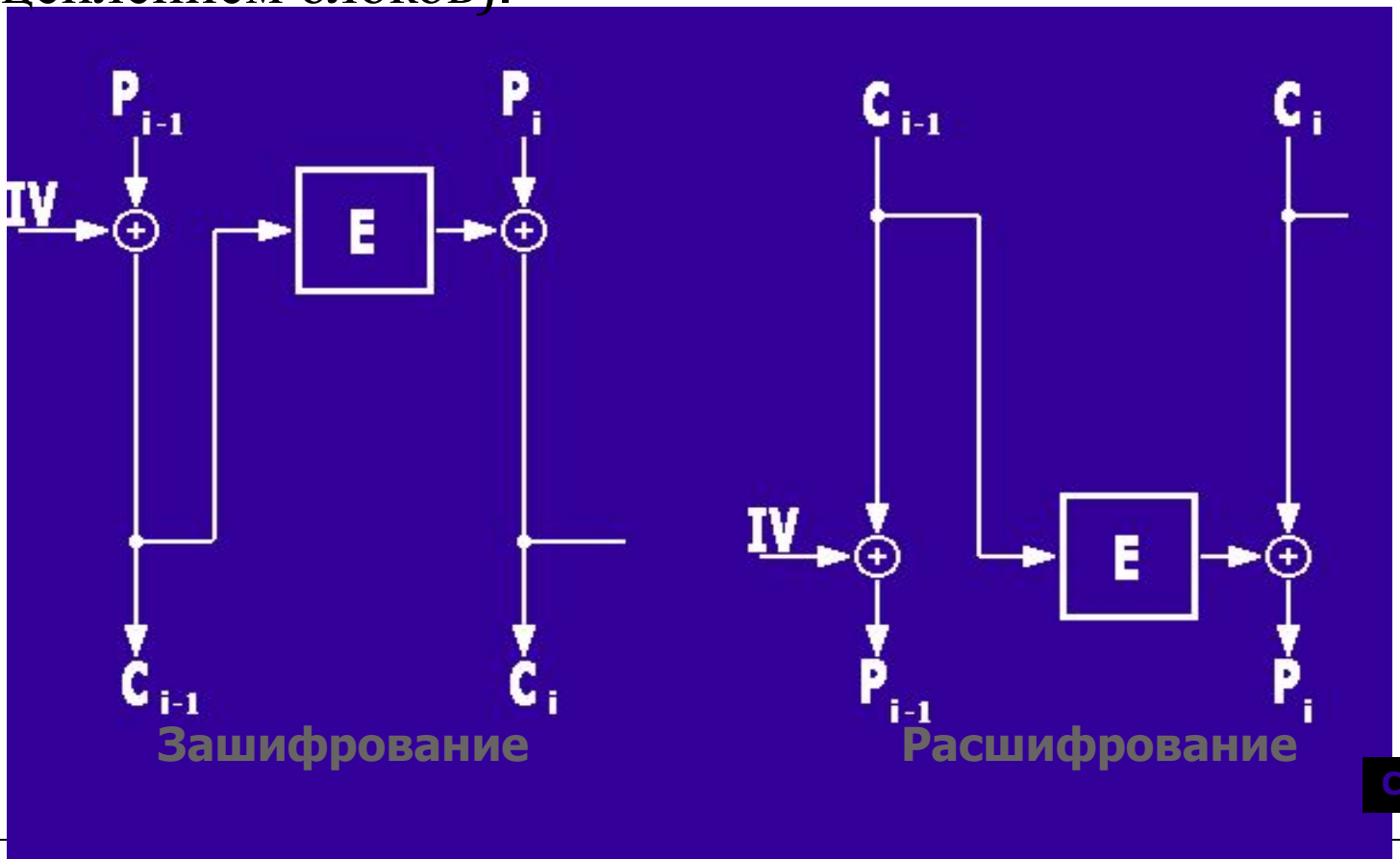


Рис. 5.14. Шифрование в режиме *CFB*

Режимы шифрования

Режим CFB

- Единица зашифрованных данных может быть меньше размера блока шифра.
- Шифрование каждого блока зависит от всех предыдущих блоков.
- Вектор инициализации обязательно должен быть уникален.
- Ошибка в бите шифротекста влияет на текущий и определенное число следующих блоков открытого текста, затем ошибка самоустраняется.
- Шифр самовосстанавливается после потери (добавления) бита шифротекста.

Режимы шифрования

Режим CFB

- Возможна подмена последних битов сообщения.
- Рекомендуется использовать для шифрования разреженного потока данных, например ввода с терминала.

Режимы шифрования

Режим OFB

OFB (Output-feedback) – режим выходной обратной связи, режим гаммирования.

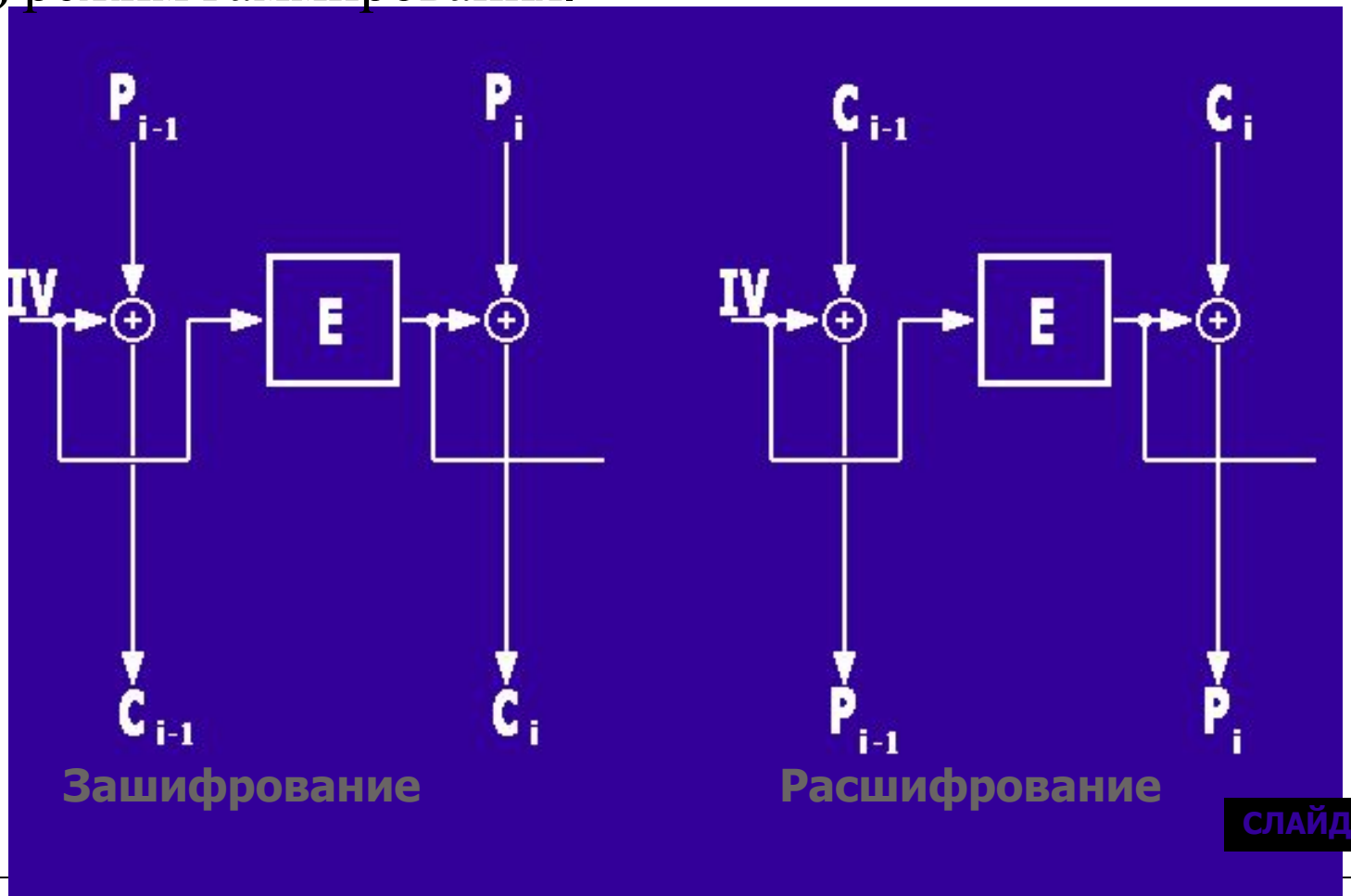


Рис. 5.12. Шифрование в режиме *OFB*

$$y_i = E_k(x_i),$$

$$e_i = j \text{ крайних слева битов блока } y_i,$$

$$C_i = m_i \oplus e_i,$$

$$x_{i+1} = y_i.$$

Рис. 5.13. Расшифрование в режиме *OFB*

$y_0 = IV$, $\qquad z_i = E_k(y_{i-1})$

$e_i = j$ крайних слева битов блока z_i, $\qquad y_i = m_i \oplus e_i$.

Режимы шифрования

Режим OFB

- Единица зашифрованных данных может быть меньше размера блока шифра (что не рекомендуется).
- Вектор инициализации обязательно должен быть уникален.
- Нет распространения ошибок – неправильный бит шифротекста приведет к одному неправильному биту открытого текста.
- Потеря добавление бита шифротекста портит весь открытый текст с этого места.
- Отлично подходит для шифрования аудио или видео потоков.

Шифр ГОСТ 28147-89

$+$ сложение слов по модулю 2^{32};

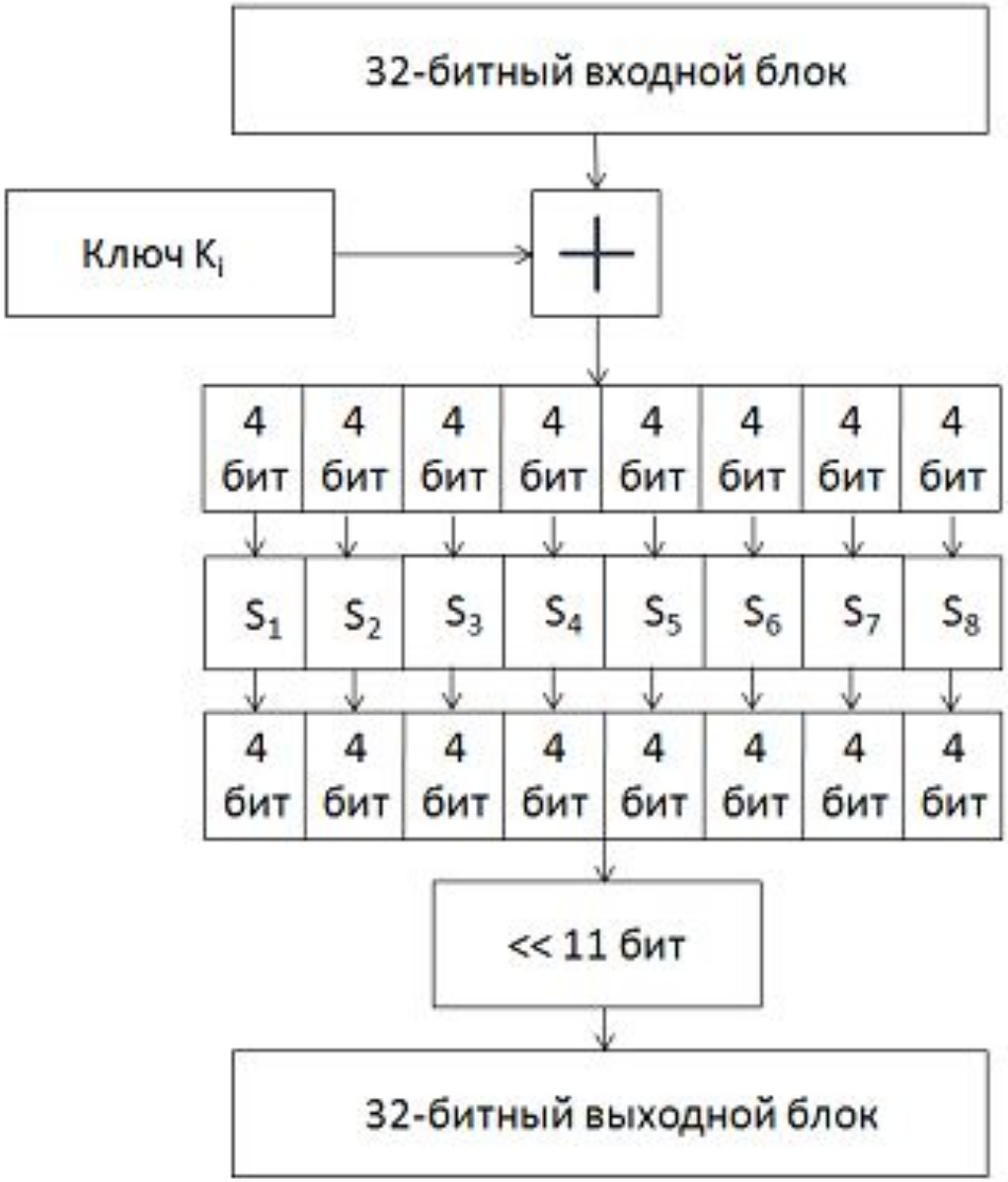
$\hookleftarrow$ циклический сдвиг слова влево на указанное число бит;

$\oplus$ побитовое «исключающее или» двух слов, т.е. побитовое сложение по модулю 2.

Алгоритм 8.1. Базовый цикл шифра ГОСТ 28147-89

ВХОД: Блок L, R, раундовый ключ W.
ВЫХОД: Преобразованный блок L, R.

1. FOR $i = 0, 1, \ldots, 31$ DO
2. $k \leftarrow R + W_i, \quad k = (k_7 \cdots k_0)_{16}$;
3. FOR $j = 0, 1, \ldots, 7$ DO
4. $k_j \leftarrow S_j[k_j]$;
5. $L \leftarrow L \oplus (k \hookleftarrow 11)$;
6. $L \longleftrightarrow R$;
7. RETURN $L.R$.


32-битный входной блок
Ключ K_i
+
4 бит
4 бит
4 бит
4 бит
4 бит
4 бит
4 бит
4 бит
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8
4 бит
4 бит
4 бит
4 бит
4 бит
4 бит
4 бит
4 бит
<< 11 бит
32-битный выходной блок

32 цикла

с подключами

$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7$

$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7$

$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7$

$K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0$

Заключение

- Все современные надежные шифры являются *составными*, то есть строятся из большого числа относительно несложных шифрующих преобразований так, чтобы в полной мере обеспечить наличие свойств *перемешивания* и *рассеивания*.
- В качестве «строительных элементов» шифров используются битовые *перестановки*, *замены* (*подстановки*) в битовых группах, *арифметические и логические операции*. При этом наибольшее перемешивание и рассеивание каскада из шифрующих преобразований достигается, если смежные операции в цепочке как можно сильней отличаются друг от друга.

Заключение

- Наиболее простой и популярный способ создать шифрующие структуры называются *сетями Файстеля*.
- Для использование на раундах шифрования обычно требуется больше *ключевой информации*, чем содержится в *ключе* шифрования. Для выработки нужного объема ключевой информации в раундах используют различные схемы, от самых простых – повторного использования одних и тех же фрагментов ключа, до наиболее сложных – выработки ключевых элементов с использованием тех же самых шифрующих преобразований, что используются при шифровании.
- Для того, чтобы скрыть структуру открытого текста, затруднить манипулирование им, обеспечить устойчивость шифра к сбоям, добавлению или потере битов служат различные *криптографические режимы*.

Литература

- *Шнайер Б.* Прикладная криптография. [http://ssl.stu.neva.ru/psw/crypto/appl_rus/]
- *Menezes A., van Oorschot P., Vanstone S.* Handbook of Applied Cryptography. [http://www.cacr.math.uwaterloo.ca/hac/]
- FIPS PUB 46 – Data Encryption Standard (DES). [http://www.itl.nist.gov/fipspubs/fip46-2.htm]
- FIPS PUB 74 – Guidelines for Implementing and Using the NBS Data. [http://www.itl.nist.gov/fipspubs/fip74.htm]
- FIPS PUB 81 – Des Modes of Operation. [http://www.itl.nist.gov/fipspubs/fip81.htm]

Литература

- *Файстель Х.* Криптография и компьютерная безопасность. [http://www.enlight.ru/crypto/articles/feistel/feist_0.htm]
- *Шеннон К.* Теория связи в секретных системах [http://www.enlight.ru/crypto/articles/shannon/shann_i.htm]
- *Винокуров А.* Криптография, ее истоки и место в современном обществе. [http://www.enlight.ru/ib/tech/crypto/index.htm]

Вопросы? Комментарии?