

БЛОЧНЫЕ ШИФРЫ

Определение

- Блочный шифр можно рассматривать как функцию преобразования строки фиксированной длины в строку аналогичной длины с использованием некоторого ключа, а также соответствующую ей функцию расшифрования.
- Требования к функциям шифрования и расшифрования:
 - **однозначное восстановление** исходного сообщения
 - функции шифрования и расшифрования должны быть вычислительно **простыми** для **легальных** пользователей (знающих ключ)
 - должно быть **невозможно** найти открытый текст сообщения по шифртексту **без знания ключа**, кроме как **полным перебором** всех возможных ключей расшифрования

Виды блочных шифров

- Шифры, построенные на SP-сетях (Substitution-Permutation network, SP network, подстановочно-перестановочная сеть). Такие шифры основаны на обратимых преобразованиях с открытым текстом. При их разработке криптограф должен следить за тем, чтобы каждая из производимых операций была и криптографически надёжна, и обратима при знании ключа.
- Шифры, в той или иной степени построенные на ячейке Фейстеля. В данных шифрах используется конструкция под названием «ячейка Фейстеля», которая по методу построения уже обеспечивает обратимость операции шифрования легальным пользователем при знании ключа. Криптографу при разработке функции шифрования остаётся сосредоточиться на надёжности конструкции.

Особенности блочных шифров

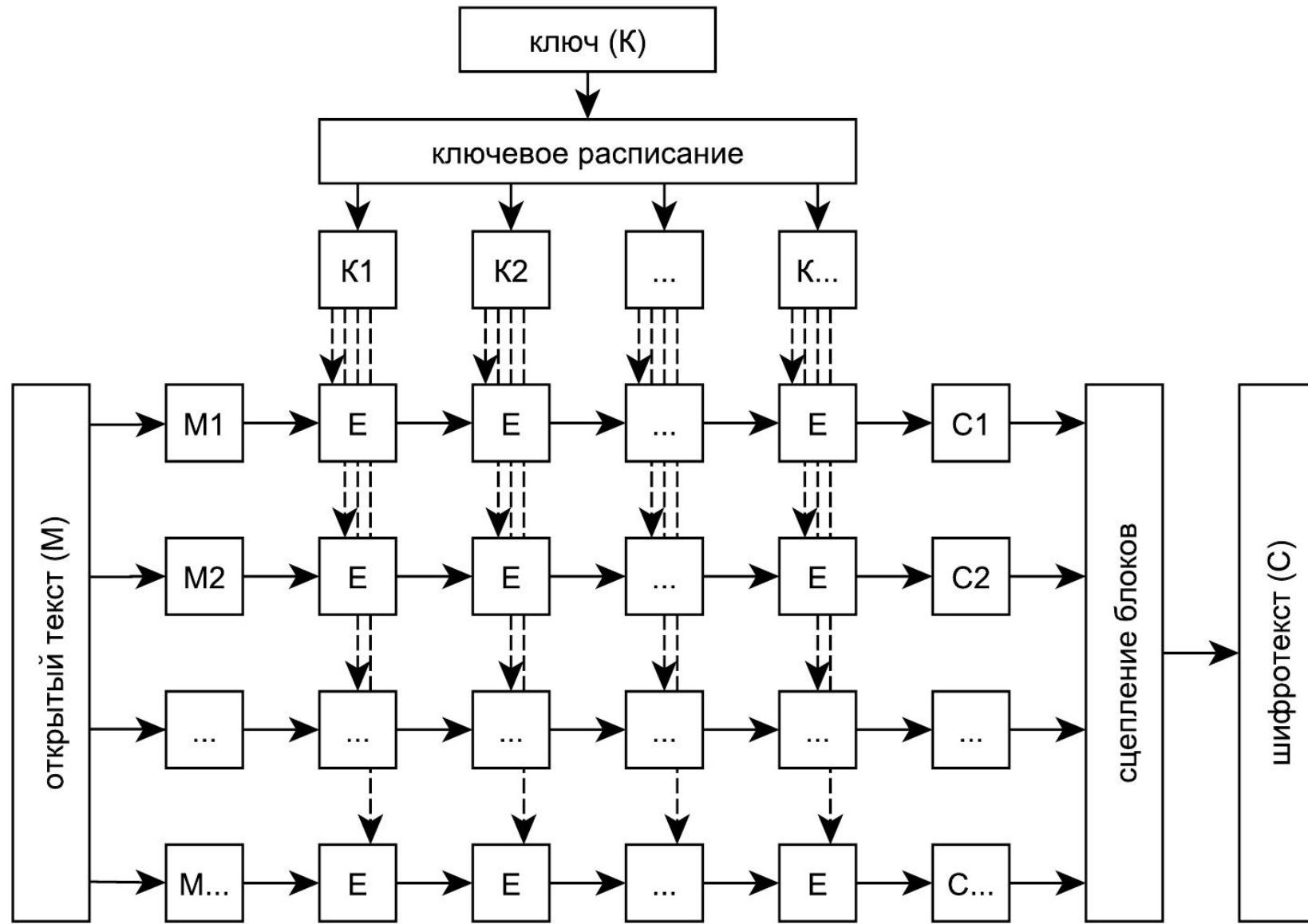
- Все современные блочные шифры являются **раундовыми**. То есть блок текста проходит через несколько одинаковых (или похожих) преобразований, называемых **раундами шифрования**. У функции шифрования также могут существовать начальный и завершающий раунды, отличающиеся от остальных (обычно - отсутствием некоторых преобразований, которые не имеют смысла для «крайних» раундов).

Особенности блочных шифров

- Аргументами каждого раунда являются результаты предыдущего раунда (для первого -- часть открытого текста) и **раундовый ключ**. Раундовые ключи получаются из оригинального ключа шифрования с помощью процедуры, получившей название алгоритма **ключевого расписания**. Функция ключевого расписания является важной частью блочного шифра. На потенциальной слабости этой функции основаны такие криптографические атаки, как атака на основе связанных ключей и атака скольжения.

Особенности блочных шифров

- После прохождения всех раундов шифрования зашифрованные блоки объединяются в шифртекст с помощью одного из режимов сцепления блоков. Простейшим примером режима сцепления блоков является режим электронной кодовой книги, когда блоки просто конкатенируются в шифртекст без дополнительной обработки.
- Также надёжные блочные шифры обладают **лавинным эффектом** (avalanche effect): изменение одного бита в блоке открытого текста или ключа приводит к полному изменению соответствующего блока шифртекста.



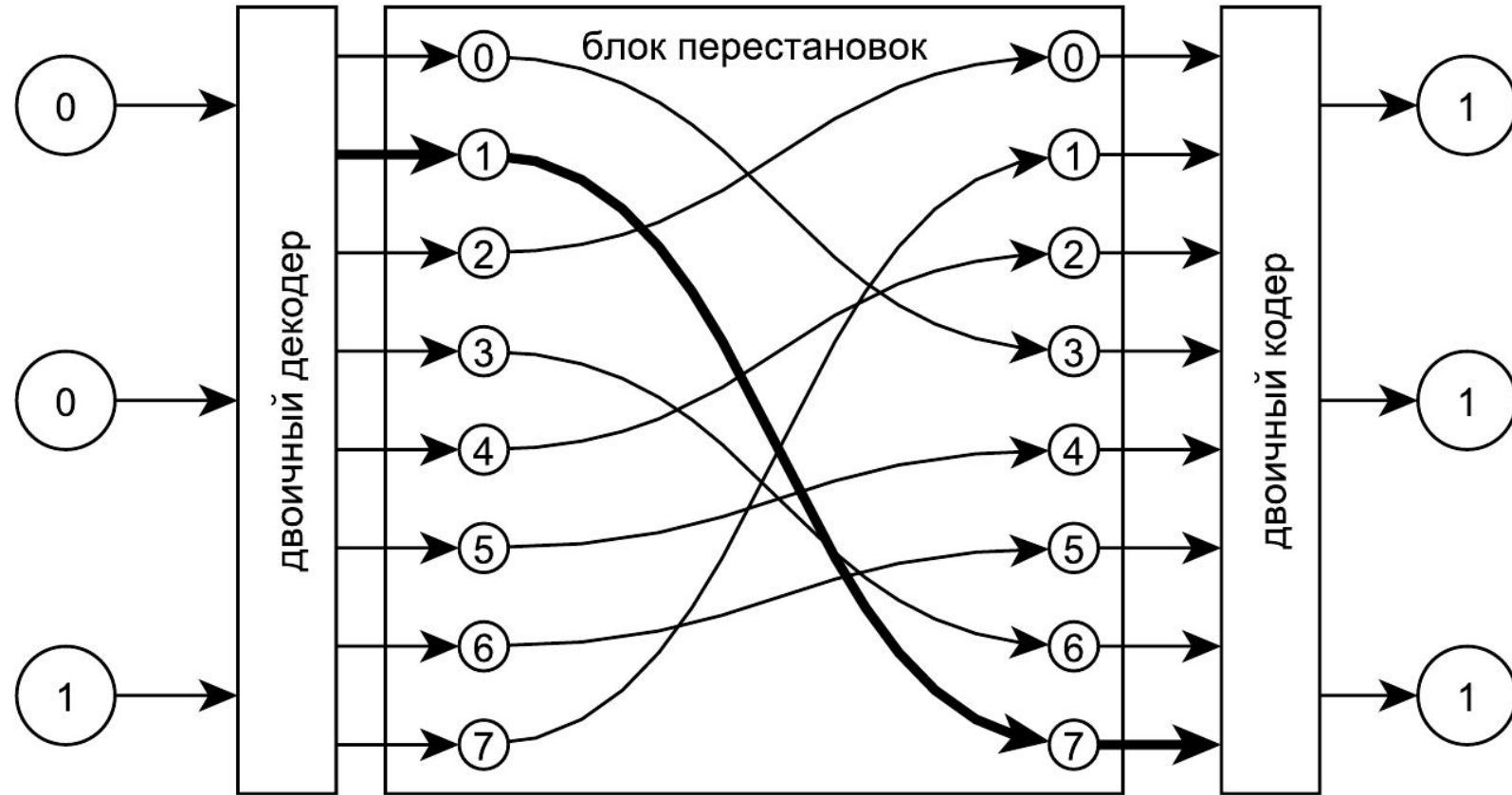
Базовые идеи

- В 1973 году в журнале Scientific American появилась статья сотрудника IBM (а ранее -- ВМС США) Хорста Фейстеля «Cryptography and Computer Privacy», описывающая проект функции шифрования «Люцифер», который можно считать прообразом современных блочных шифров. Развитием данной системы стал государственный стандарт США «Digital Encryption Standard» (DES) с 1979 по 2001 годы.

Базовые идеи

- Фейстель высказал идею, что идеальный шифр для блока размером в 128 бит должен включать в себя блок замен (substitution box, s-box, далее s-блок), который мог бы обработать сразу 128 бит входного блока данных. S-блок принимает на вход блок битов и даёт на выходе другой блок бит (возможно, даже другого размера) согласно некоторому словарю или результату вычисления нелинейной функции. К сожалению, физическая реализация действительно произвольного блока замен для входа в 128 бит потребовала бы 2^{128} внутренних соединений или словаря из 2^{128} 128-битовых значений, если реализовывать программным способом, что технологически невозможно. Зато если такой блок можно было бы создать, то он был бы очень хорош с криптографической точки зрения.

S-блок

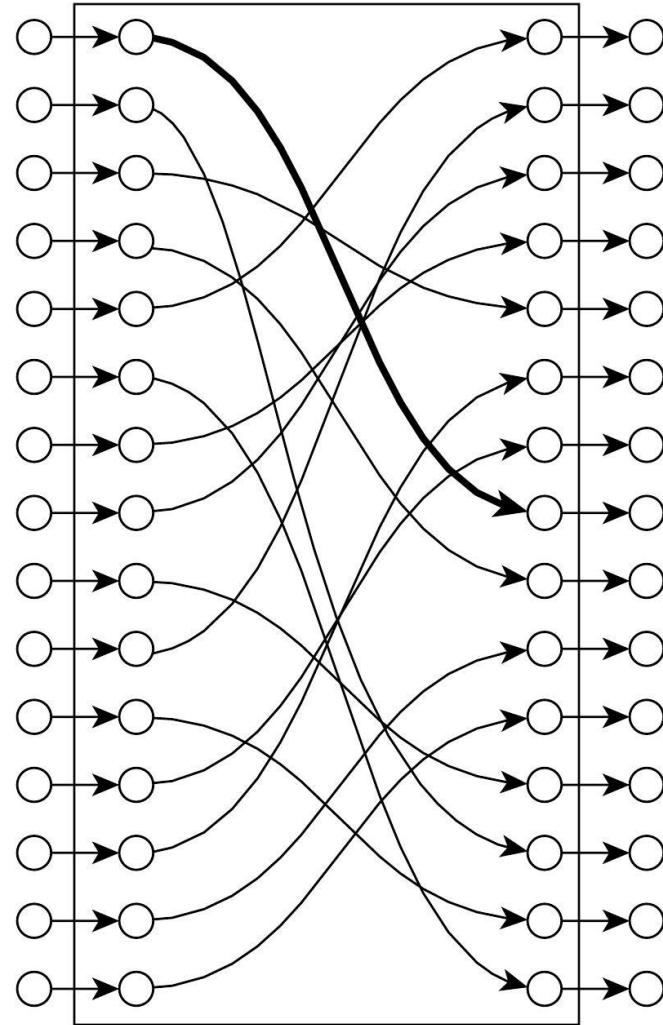


На вход поступают 3 бита информации, которые трактуются как двоичное представление номера одной из 8 линий внутреннего р-блока. На выходе номер активной сигнальной дорожки обратно преобразуется в 3-битовое представление

Базовые идеи

- С другой стороны, блок перестановок (permutation box, p-box, далее p-блок) может обрабатывать блоки битов любого размера. Однако какая-либо криптографическая стойкость у него отсутствует: он представляет собой тривиальное линейное преобразование своего входа. Криptoаналитику достаточно иметь N линейно независимых пар значений входа и выхода (где N - размер блока), чтобы получить полное представление о структуре p-блока.

P-блок

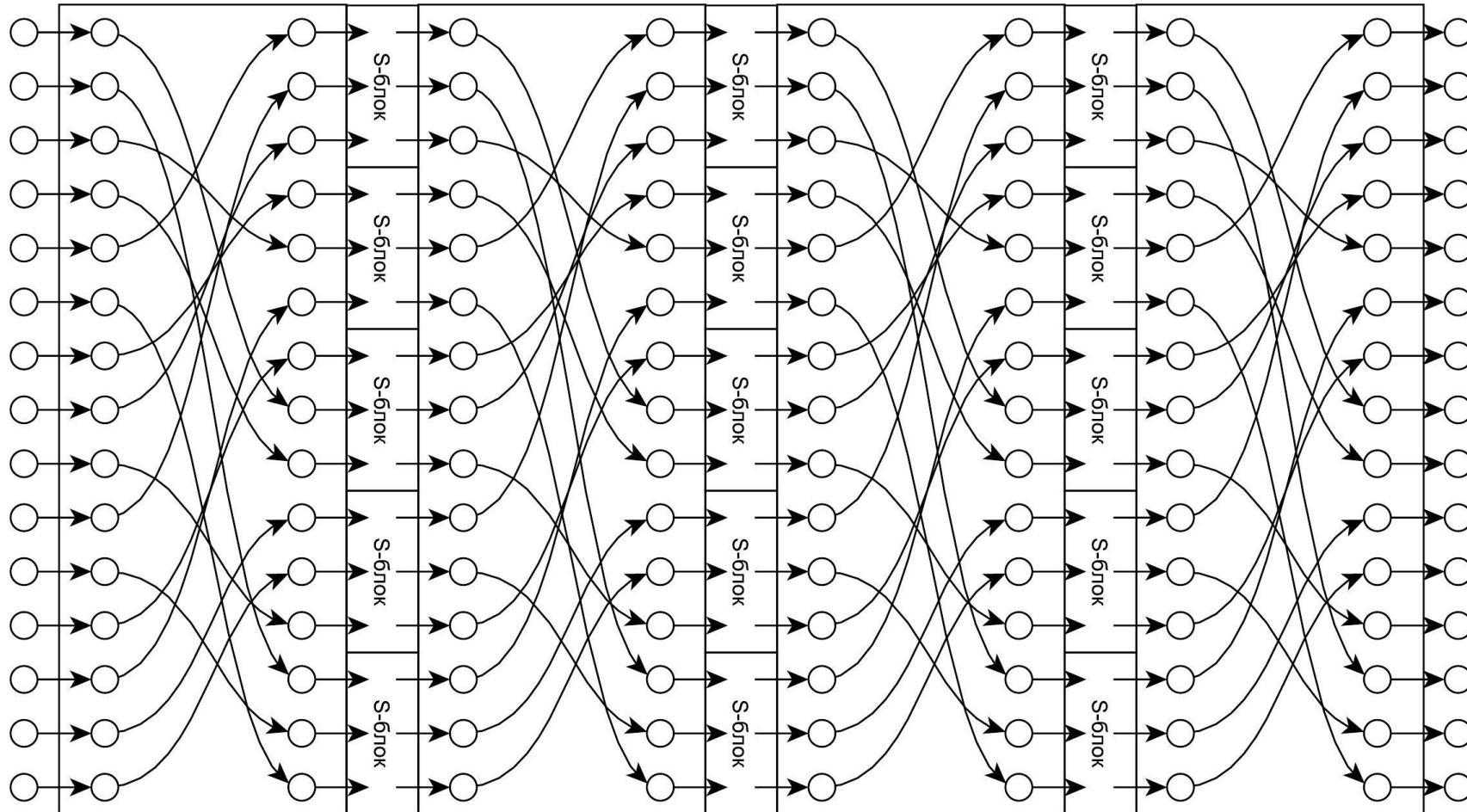


Все поступающие на вход биты не меняются, но перемешиваются внутри блока

Базовые идеи

- Идея Фейстеля состояла в том, чтобы комбинировать s- и p-блоки, позволяя на практике получить большой блок нелинейных преобразований (то есть один большой s-блок). При достаточном числе «слоёв» SP-сеть начинает обладать свойствами хорошего s-блока (сложностью криптографического анализа и выявления структуры), при этом оставаясь технологически простой в реализации.

SP-сеть

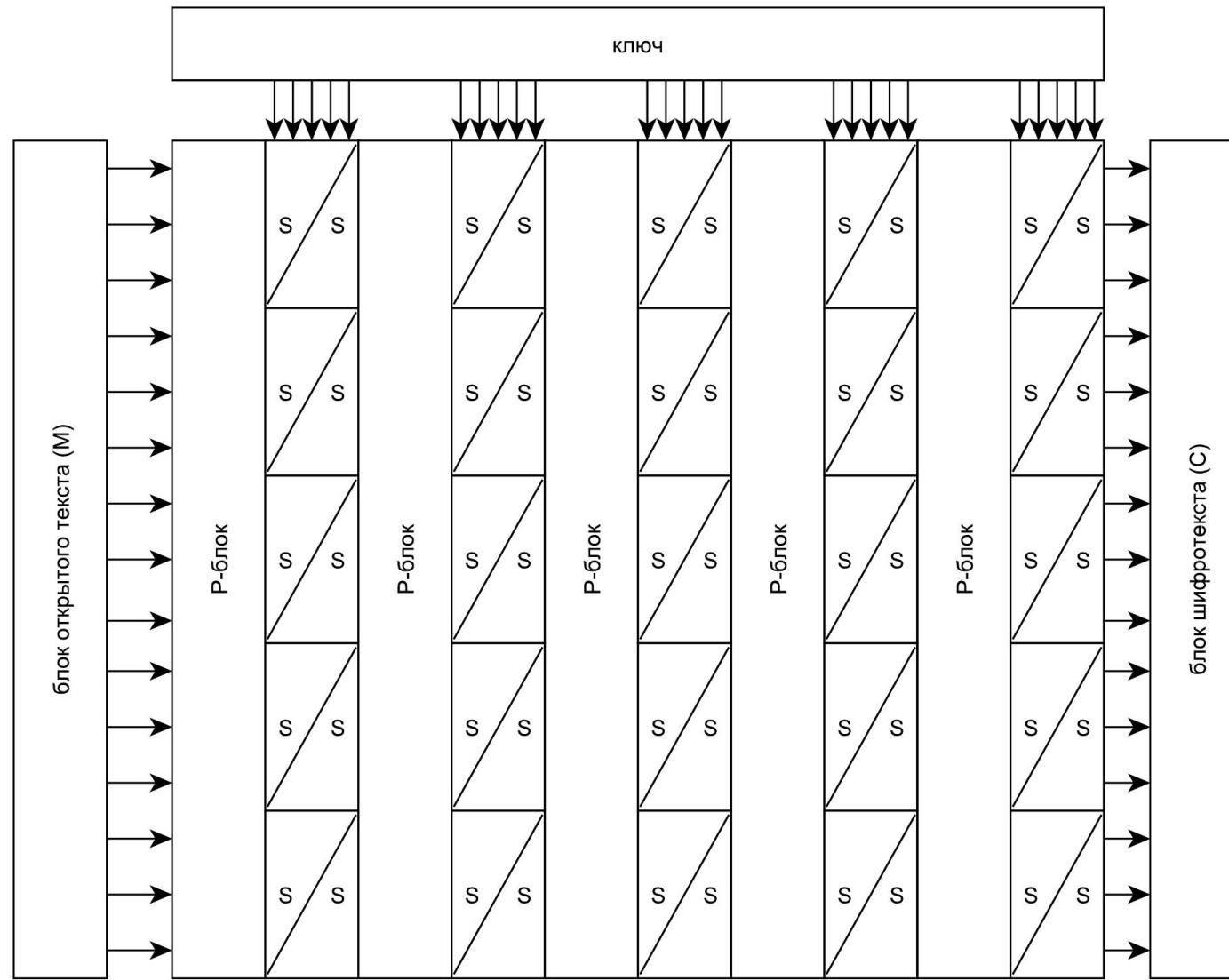


SP-сеть, состоящая из 4 р-блоков и 3 слоёв s-блоков, по 5 блоков в каждом слое

Базовые идеи

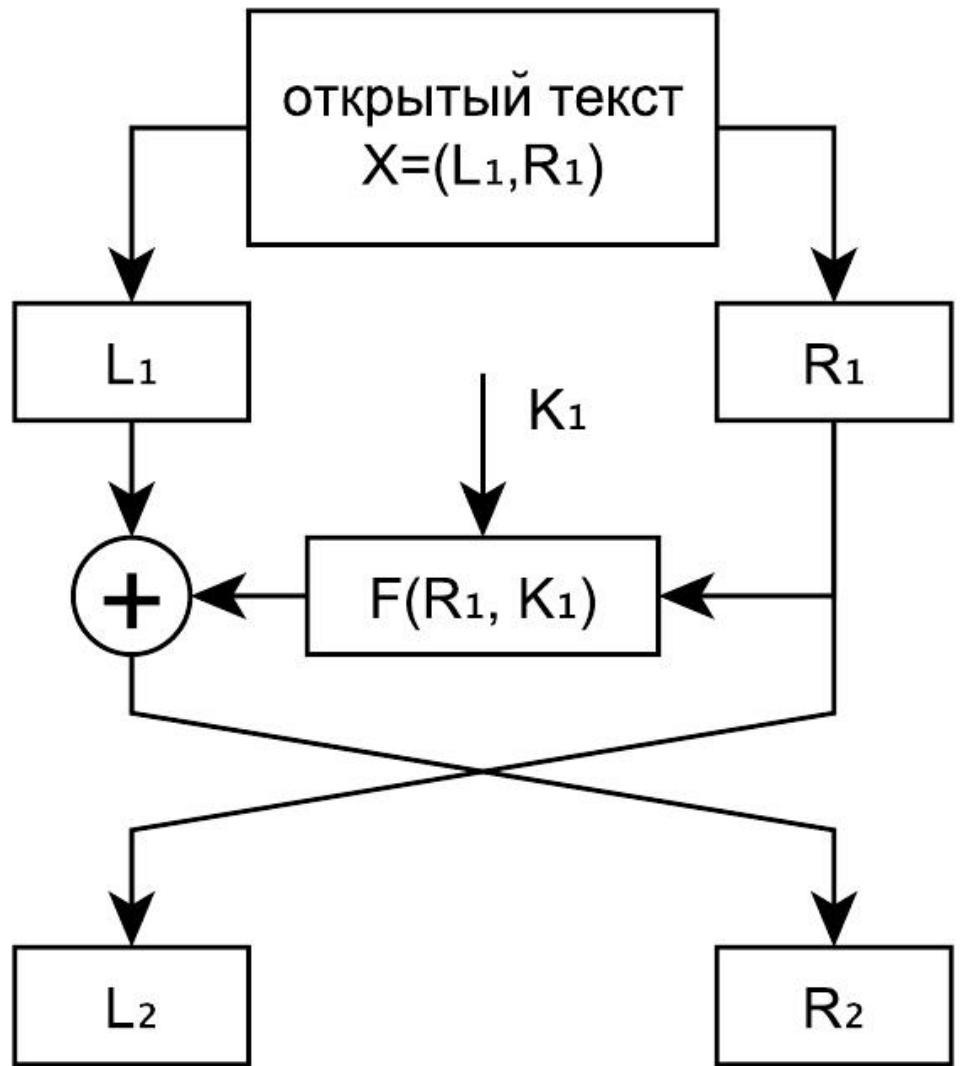
- Следующей составляющей будущего шифра стала возможность менять используемые s-блоки в зависимости от ключа. Вместо каждого из s-блоков в SP-сети Фейстель поместил модуль с двумя разными s-блоками. В зависимости от одного из битов ключа (своего для каждой пары блоков) использовался первый или второй s-блок. Результатом данного подхода стал первый вариант шифра в проекте «Люцифер»

Проект «Люцифер»



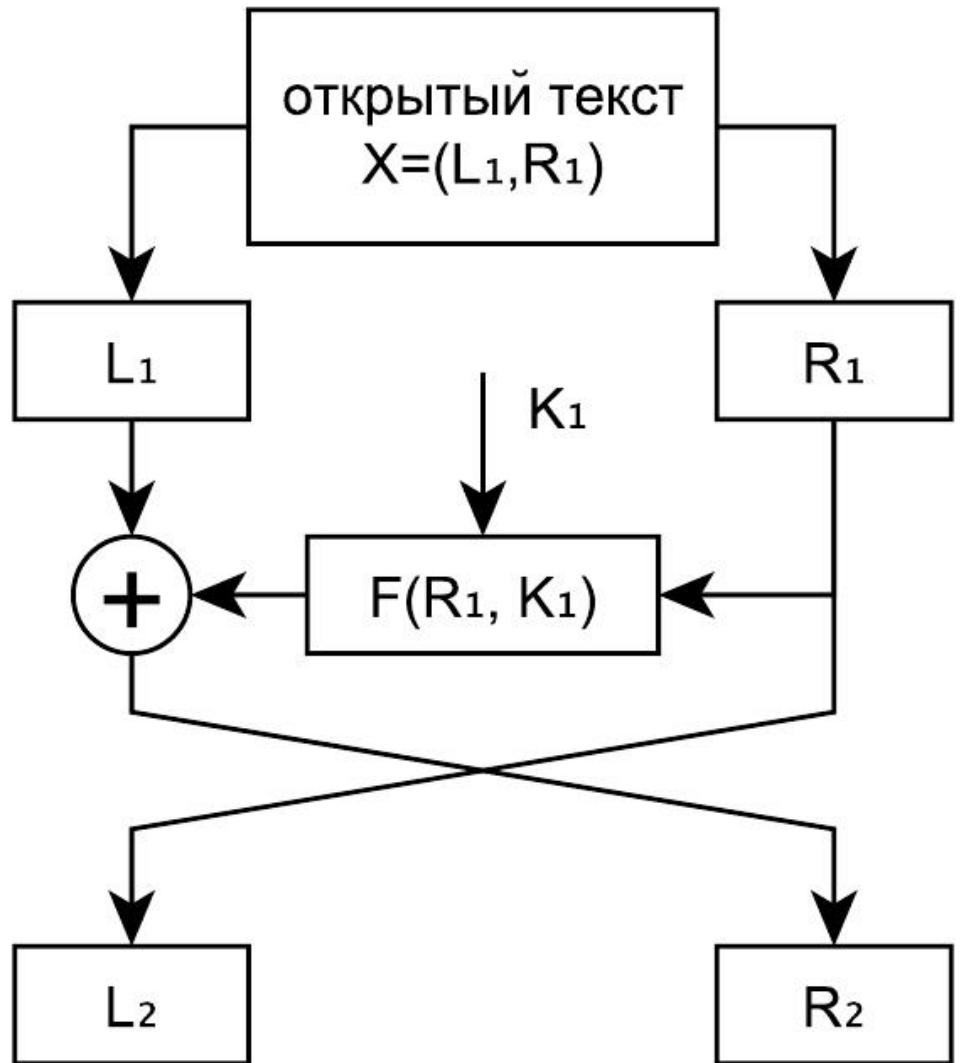
Общий вид (упрощённая схема) функции шифрования в одном из вариантов проекта «Люцифер». Входной блок (в проекте «Люцифер» его объём -- 128 бит) подавался на вход на несколько слоёв (в «Люцифере» слоёв 16) из р-блоков и пар s-блоков. S-блок в каждой паре выбирался в зависимости от значения соответствующего бита ключа

Ячейка Фейстеля



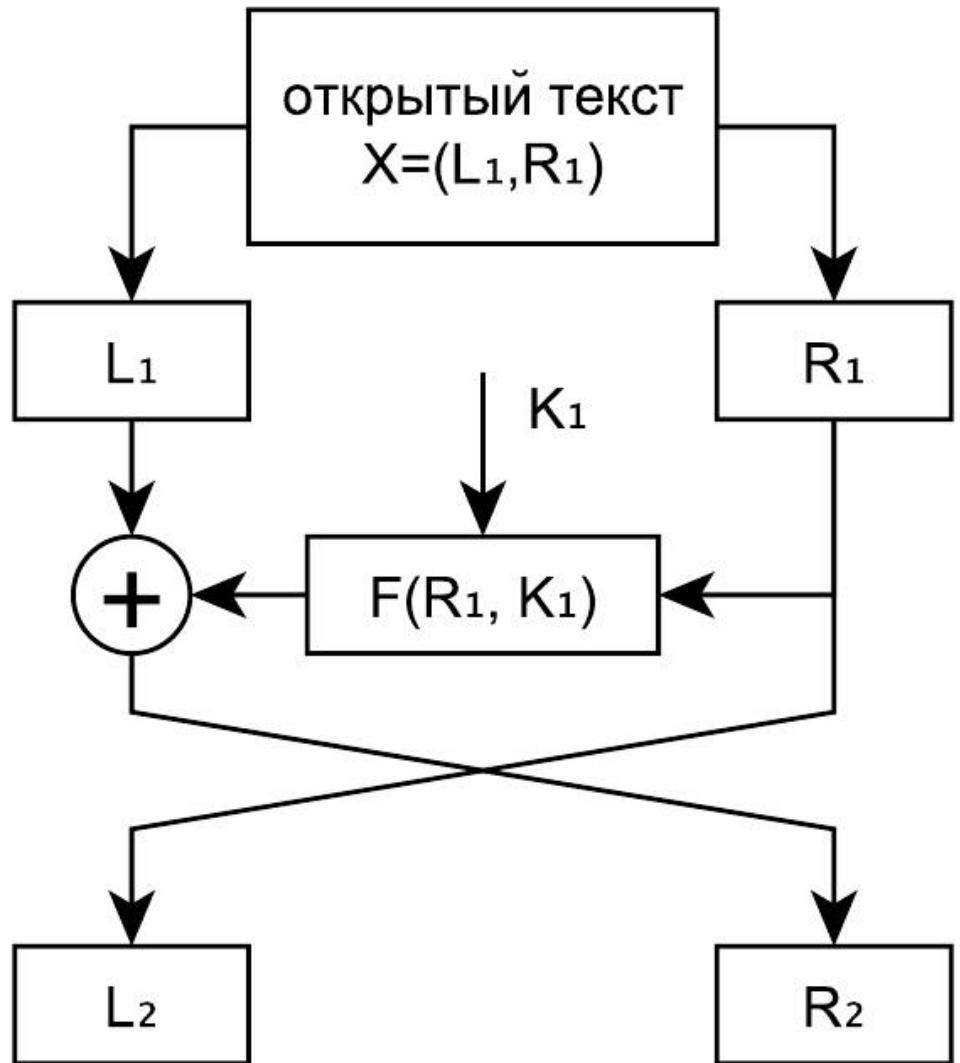
На рисунке изображён один раунд шифрования блочного шифра, использующего оригинальную ячейку Фейстеля. Каждый раунд шифрования принимает на вход блок с чётным количеством бит и делит его на две равные части L_k и R_k . Входным блоком для первого раунда является блок открытого текста. Правая часть R_k без изменений становится левой частью входного блока L_{k+1} следующего раунда шифрования.

Ячейка Фейстеля



Кроме того, правая часть подаётся на вход **функции Фейстеля** $F(R_k, K_k)$, аргументами которой являются половина блока данных и раундовый ключ (раундовые ключи получаются в результате работы алгоритма ключевого расписания). Результат работы функции Фейстеля складывается с помощью побитового сложения по модулю 2 с левой частью входного блока L_k . Полученная последовательность бит становится правой частью выходного блока раунда шифрования.

Ячейка Фейстеля



Таким образом, работа k-го раунда ячейки Фейстеля описывается следующими соотношениями:

$$\begin{aligned}L_{k+1} &= R_k \\R_{k+1} &= L_k \oplus F(R_k, K_k)\end{aligned}$$

Результатом шифрования является конкатенация последних выходных блоков L_n и R_n , где n -- число раундов шифрования.
Формулы расшифрования:

$$\begin{aligned}R_k &= L_{k+1} \\L_k &= R_{k+1} \oplus F(R_k, K_k)\end{aligned}$$

Шифр DES

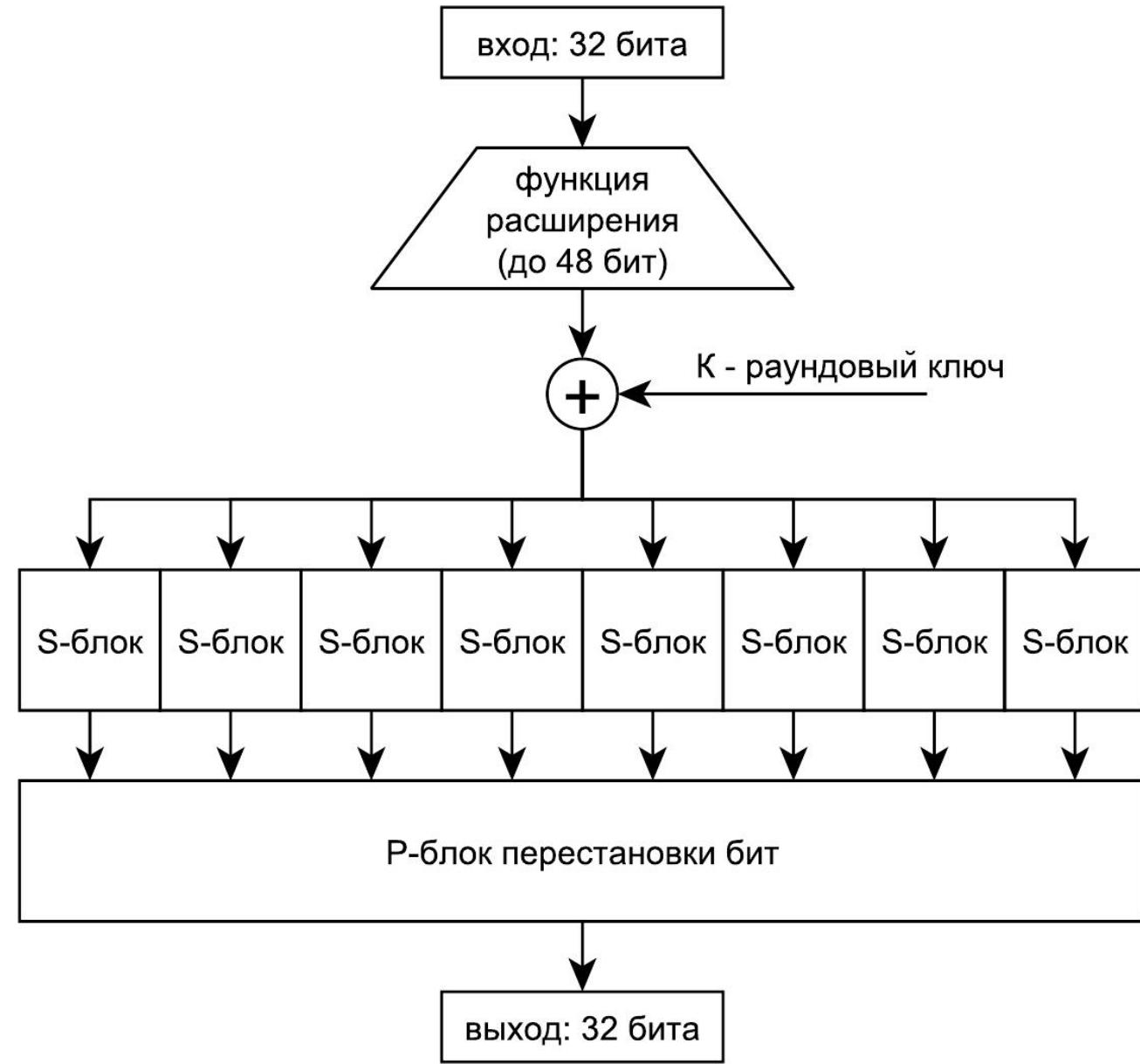
- Развитием проекта «Люцифер» стал государственный стандарт США, известный как DES (data encryption standard). Это первый из рассматриваемых нами блочных шифров, который имеет ярко выраженные раунды шифрования, отдельно выделенную функцию ключевого расписания и основан на классической ячейке Фейстеля. Поэтому для знакомства с шифром достаточно рассмотреть устройство функции Фейстеля как основного элемента, отличающего данный шифр от аналогичных.

Шифр DES

- В шифре DES открытый текст делится на блоки по 32 бита, и они обрабатываются в 16 раундах. Раундовые ключи генерируются из исходных 64 бит ключа (при этом значащими являются только 56 бит, а последние 8 бит используются для проверки корректности ввода ключа). На вход функции Фейстеля для шифра DES, схема которой приведена на рисунке, подаётся половина от размера входного блока -- 32 бита.

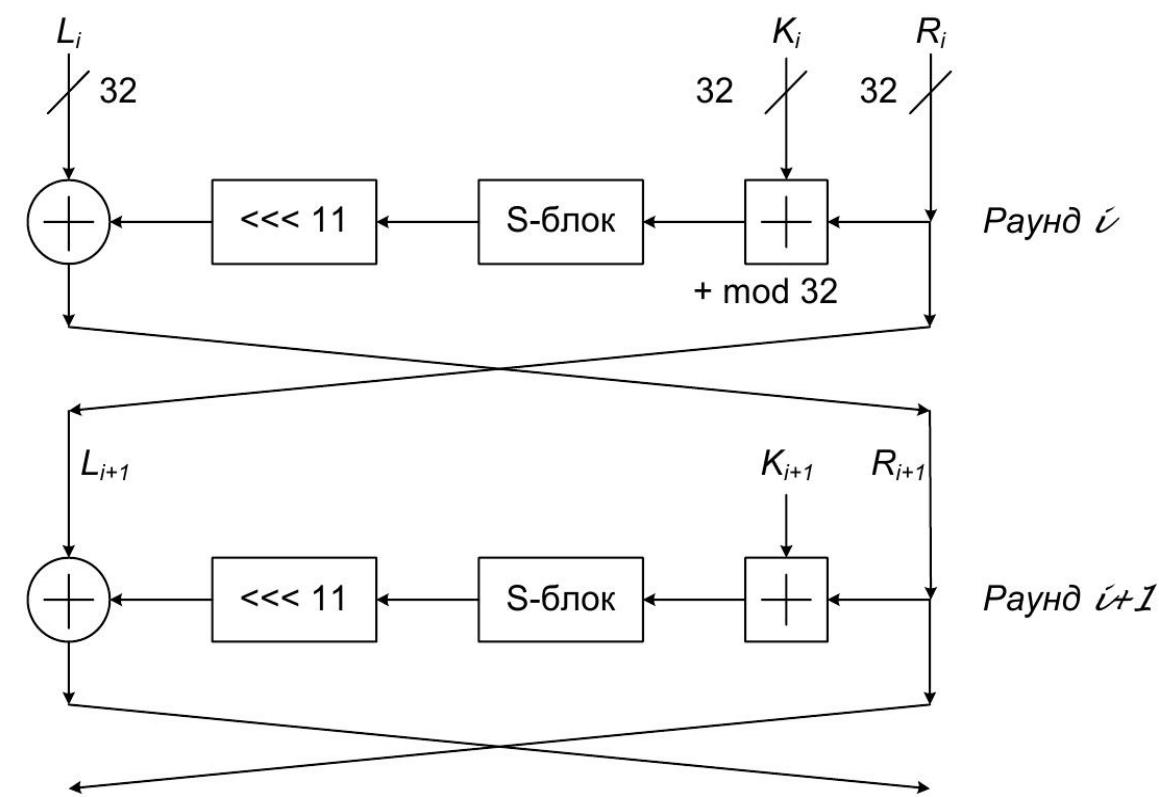
Шифр DES

Эти 32 бита проходят через функцию расширения, которая с помощью дублирования отдельных битов превращает их в 48 бит. Они суммируются побитово по модулю 2 с раундовым ключом. Результат подаётся на вход 8 S-блоков, которые работают как таблицы замен последовательности из 6 бит в 4 бита (каждый блок). На выходе S-блоков получаются 32 бита, которые попадают в P-блок перестановки бит. Результат работы P-блока является результатом функции Фейстеля для одного раунда шифра DES.



Шифр ГОСТ 28147-89

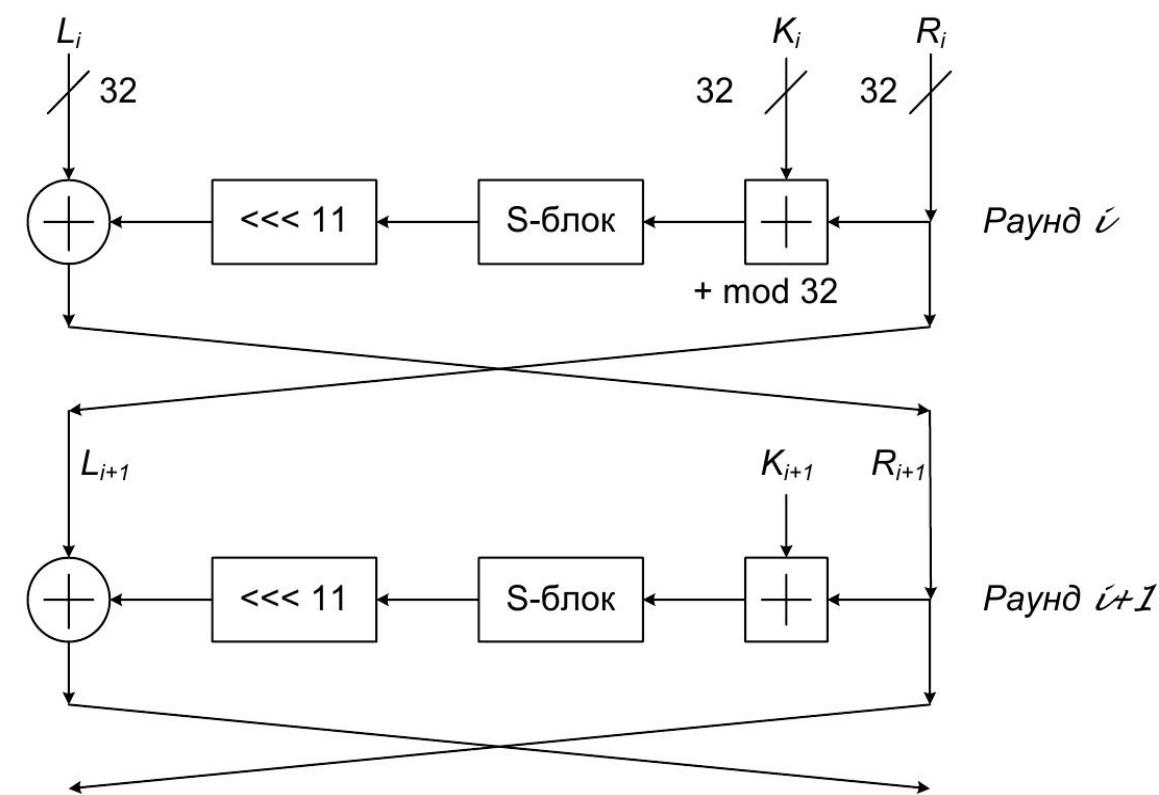
ГОСТ 28147-89 устанавливает единый алгоритм криптографических преобразований для систем обмена информацией в вычислительных сетях и определяет правила шифрования и расшифрования данных, а также выработки имитовставки. Основные параметры шифра таковы: размер блока составляет 64 бита, число раундов $m=32$, имеется 8 ключей по 32 бита каждый, так что общая длина ключа -- 256 бит. Основа алгоритма -- цепочка ячеек Фейстеля



Шифр ГОСТ 28147-89

Структурная схема алгоритма шифрования включает в себя:

- ключевое запоминающее устройство (КЗУ) на 256 бит, которое состоит из восьми 32-разрядных накопителей (X_0, X_1, \dots, X_7) и содержит сеансовые ключи шифрования одного раунда;
- 32-разрядный сумматор \boxplus по модулю 2^{32} ;
- сумматор \oplus по модулю 2;
- блок подстановки (S);
- регистр циклического сдвига на 11 шагов в сторону старшего разряда (R)



Шифр ГОСТ 28147-89

- Блок подстановки (S) состоит из 8 узлов замены -- s-блоков с памятью на 64 бита каждый. Поступающий на вход блока подстановки 32-разрядный вектор разбивается на 8 последовательных 4-разрядных векторов, каждый из которых преобразуется в 4-разрядный вектор соответствующим узлом замены. Узел замены представляет собой таблицу из 16 строк, содержащих по 4 бита в строке. Входной вектор определяет адрес строки в таблице, заполнение данной строки является выходным вектором. Затем 4-разрядные выходные векторы последовательно объединяются в 32-разрядный вектор. При перезаписи информации содержимое i -го разряда одного накопителя переписывается в i -й разряд другого накопителя. Ключ, определяющий заполнение КЗУ, и таблицы блока подстановки K являются секретными элементами.

Шифр ГОСТ 28147-89

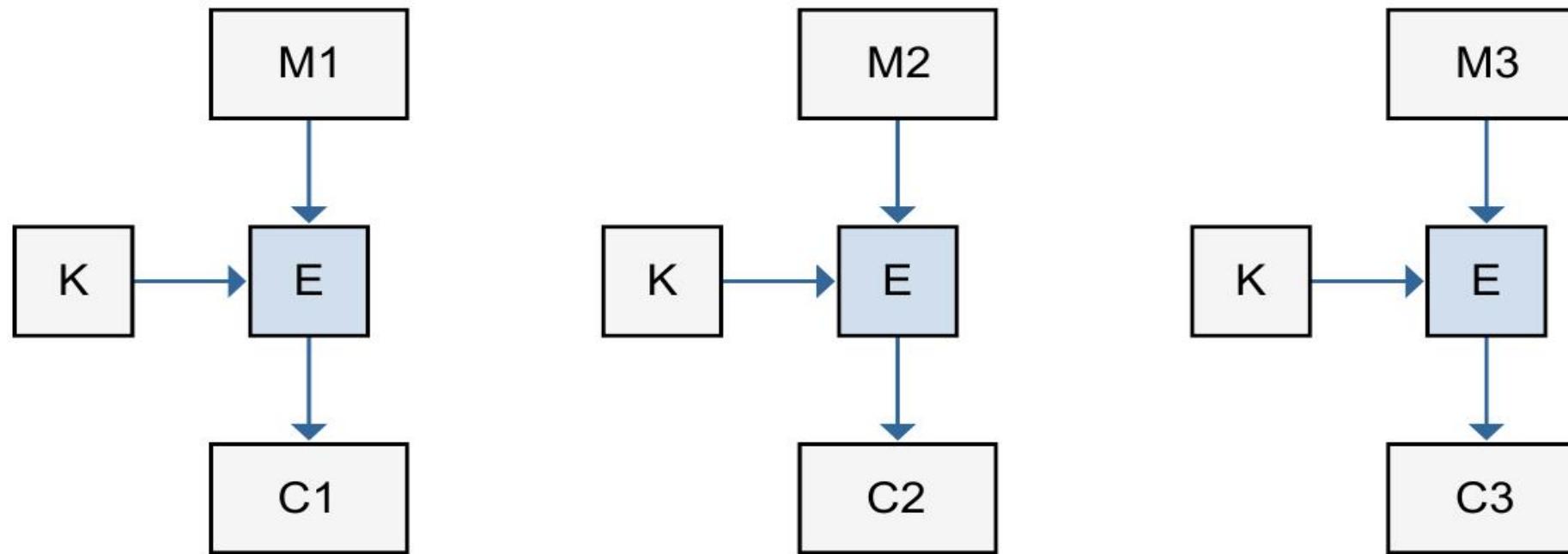
Алгоритм имеет четыре режима работы:

- простой замены,
- гаммирования,
- гаммирования с обратной связью,
- выработки имитовставки

Режим простой замены

- открытый текст в пакете разделён на блоки $M_1, M_2, \dots, M_{n-1}, M_n$
- В процессе шифрования каждому блоку M_j ставится в соответствие шифртекст C_j , определяемый с помощью ключа K : $C_j = E_K(M_j), j = 1 \dots n.$
- Если в открытом тексте есть одинаковые блоки, то в шифрованном тексте им также соответствуют одинаковые блоки. Это даёт дополнительную информацию для криptoаналитика, что является недостатком этого режима. Другой недостаток состоит в том, что криptoаналитик может подслушивать, перехватывать, переставлять, воспроизводить ранее записанные блоки, нарушая конфиденциальность и целостность информации. Поэтому при работе в режиме электронной кодовой книги нужно вводить аутентификацию сообщений.

Режим простой замены



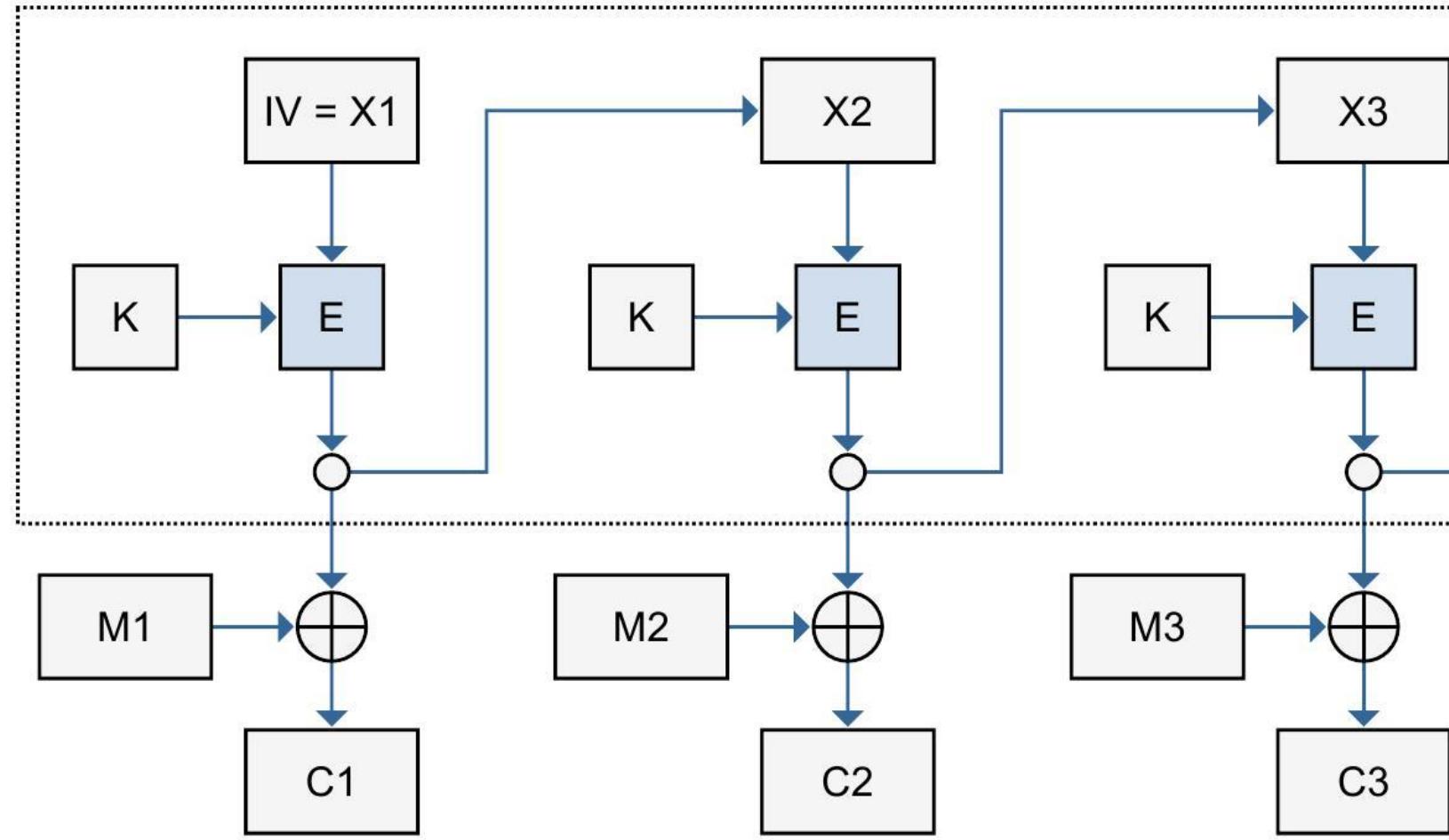
Гаммирование с обратной связью

В режиме обратной связи по выходу блоки открытого текста непосредственно на вход функции шифрования не поступают. Вместо этого функция шифрования генерирует псевдослучайный поток байтов (*гамму*), который суммируется побитово по модулю 2 с открытым текстом для получения шифртекста.

Шифрование осуществляют по правилу:

$$X_1 = IV, \quad Y_j = E_K(X_j), \quad C_j = Y_j \oplus M_j, \quad X_{j+1} = Y_j, \quad j = 1, \dots, n - 1$$

Гаммирование с обратной связью



Режим обратной связи по выходу. Пунктирной рамкой выделена область формирования гаммы, не зависящей от открытого текста.

Гаммирование с обратной связью

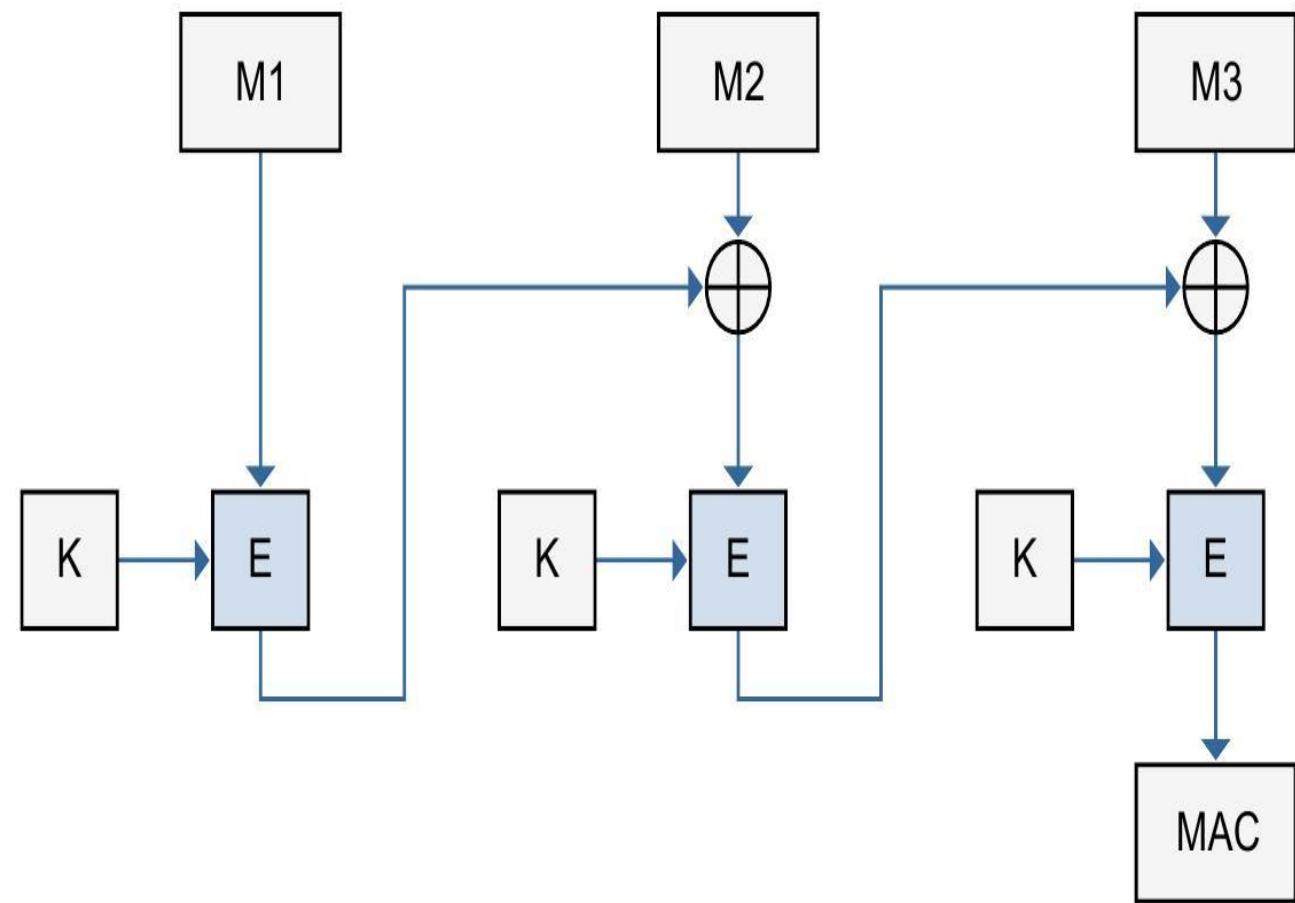
Входной блок X_j есть результат шифрования предыдущего блока X_{j-1} . Начальное значение $X_1 = IV$ известно отправителю и легальному получателю. На приёмной стороне расшифрование выполняют по правилу:

$$X_1 = IV, \quad Y_j = E_K(X_j), \quad M_j = Y_j \oplus C_j, \quad X_{j+1} = Y_j, \quad j = 1, \dots, n - 1$$

Вектор инициализации IV может быть выбран случайно и передан вместе с шифрованным текстом, либо вычислен на основе одноразовых меток. Здесь особенно важна уникальность вектора инициализации.

Режим имитовставки

Режим выработки имитовставки принципиально отличается от рассмотренных ранее режимов тем, что призван обеспечивать не конфиденциальность, а целостность. Результатом является блок данных фиксированного размера (в ГОСТ 28147-89 до 32 бит), длина которого не зависит от длины исходного сообщения.



Режим имитовставки

Входное сообщение как и ранее разбивается на блоки равной длины M_1, M_2, \dots, M_n . Последний блок, при необходимости, дополняется (ГОСТ 28147-89 -- нулями). Формула выработки имитовставки выглядит следующим образом:

$$\begin{aligned} X_1 &= M_1; Y_j = E_K(X_j), j = 1, 2, \dots, n; \\ X_j &= Y_{j-1} \oplus M_j, j = 2, \dots, n; MAC = Y_n \end{aligned}$$

В ГОСТ 28147-89 для режима выработки имитовставки функция шифрования использует 16 раундов вместо 32. Как уже было сказано, данный режим обеспечивает только целостность информации. Причём саму информацию необходимо передавать, и, возможно, шифровать отдельно. Режим не обеспечивает возможности параллельных вычислений для разных блоков открытого текста.