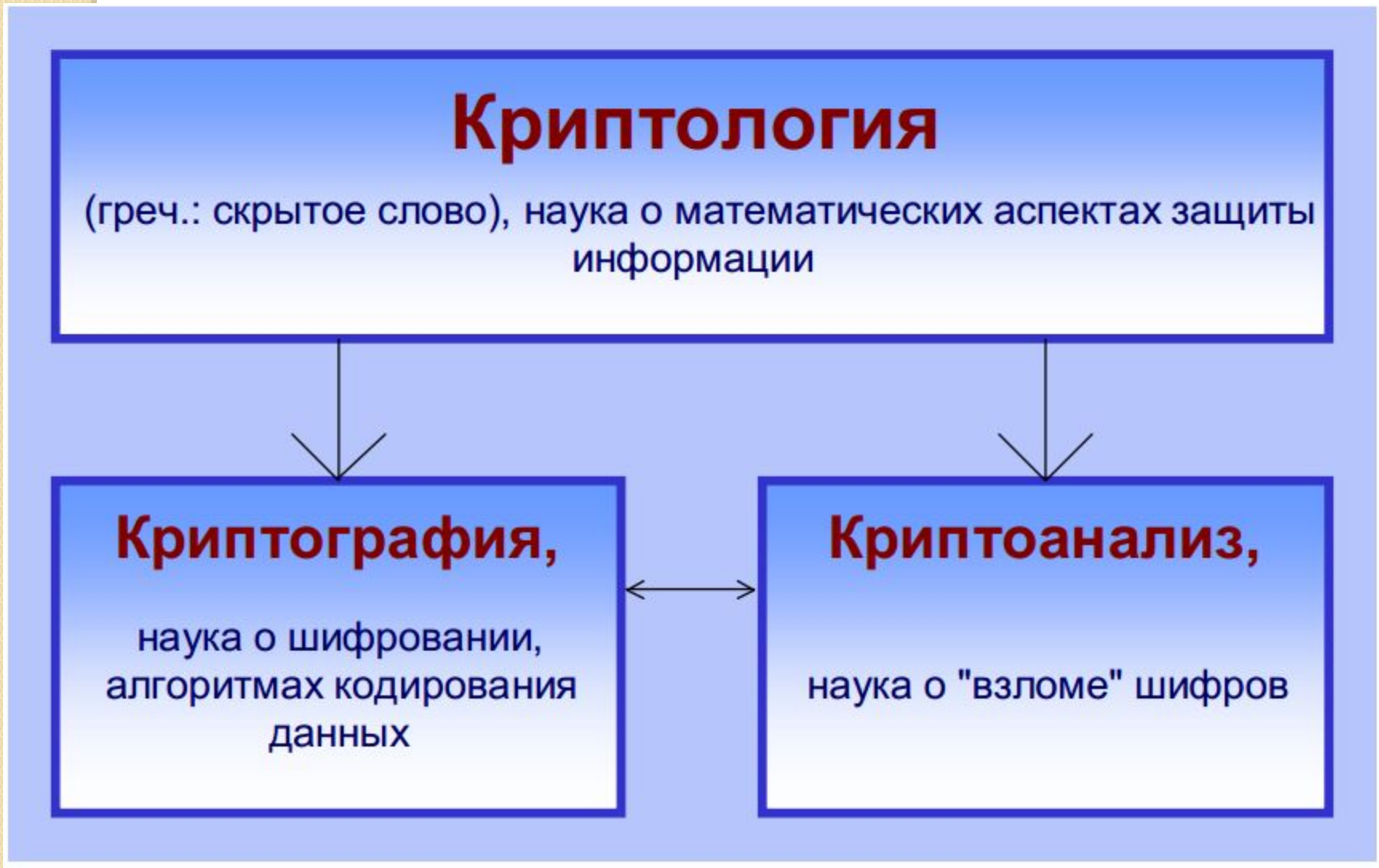
Криптографические методы защиты информации


Криптология
(греч.: скрытое слово), наука о математических аспектах защиты информации
Криптография,
наука о шифровании, алгоритмах кодирования данных
Криптоанализ,
наука о "взломе" шифров

- **Наука о том**
  - как сделать информацию конфиденциальной, избирательно доступной (шифрование)
  - как обеспечить целостность данных
  - как обеспечить аутентификацию (достоверную идентификацию)
    - субъекта: аутентичность информационного источника
    - объекта: пользователя, процесса
  - как обеспечить доказательность действия (неотказуемость)
  - как обеспечить контроль доступа (авторизацию)
- **Предмет науки:**
  - криптографические алгоритмы (математика)
  - криптографические протоколы (процессы с использованием криптографических алгоритмов)
- **Принцип (Август Керхоффс, 1835-1903):**
  - вся защита должна основываться *только* на качестве (длине, энтропии) ключа
  - алгоритмы должны быть тщательно выверены и публично доступны
- **Метод:**
  - для того, чтобы выполнить криптографическую операцию (за исключением, м.б., обеспечения целостности данных), нужно знать секретную информацию (ключ)
  - незнающий ключа должен «искать иголку в стоге сена» (а «стог» должен быть достаточно большим в математическим смысле)

- **В узком контексте сетевой безопасности основными задачами криптографии являются**
  - ***конфиденциальность*** **данных:**
    - **цель: сделать данные «нечитаемыми» для непосвященных**
    - **метод: шифрование**
  - ***целостность*** **и** ***имитостойкость*** **данных**
    - **цель: исключить возможность умышленного и неумышленного изменения (искажения) данных неуполномоченными лицами**
    - **метод: хэш, имитовставка, электронно-цифровая подпись**
  - ***аутентификация субъекта*** **– доказательство того, что субъект действия является именно тем, за кого себя выдает**
  - ***аутентификация источника данных*** **– доказательство того, что данные изданы определенным субъектом и являются подлинными (т.е. никем другим не искажены; в этом смысле – аутентификация источника данных автоматически обеспечивает их целостность)**
  - **обеспечение** ***неотказуемости*** **– невозможности для субъекта, выполнившего некоторое действие, впоследствии отказаться от факта выполнения этого действия**

Криптогра́фия

- Наука о математических методах обеспечения конфиденциальности и аутентичности информации.
- Изучает методы шифрования информации.
- *Традиционная криптография* - симметричные криптосистемы (шифрование и расшифровывание используя секретный ключ).

Современная криптография включает асимметричные криптосистемы (системы электронной цифровой подписи (ЭЦП), хеш-функции, управление ключами, получение скрытой информации, квантовую криптографию, легковесную (низкоресурсную) криптографию, стеганографию).

Основные исторические вехи

- **Древний мир**
  - жрецы и правители Египта, Греции, Рима
  - перестановочный шифр Цезаря
- **Средние века**
  - с X века: церковь
  - с XVI века: политики (Мария Стюарт)
- **Гражданская война в США** («Цилиндр Джефферсона», возможно, первая шифровальная машина)
- **1я мировая война**: Циммерман
- **2я мировая война**: Энигма, Элан Тьюринг
- **1948**: теория информации Клода Элвуда Шеннона
- **1974**: разработка первого криптостандарта (DES)
- **1976 (!)**: изобретение системы шифрования с открытым ключом (Витфилд Дифи и Мартин Хеллман)
- **1978**: система электронно-цифровой подписи (Роналд Райвест, Ади Шамир, Леонард Адлеман)
- **1980е**: первые открытые интернациональные исследования, создание международного центра разработки криптографических систем (International Association for Cryptographic Research, IACR, 1987)
- **с 1990х**: окончательная утрата занавеса секретности вокруг темы криптографии, либерализация законодательств распространения и применения криптосредств, повсеместное применение средств криптографии в программном обеспечении, коммуникациях, электронной коммерции

В истории криптографии выделяют три (четыре) этапа:

1. Наивная криптография.
2. Формальная криптография.
3. *Научная криптография.*
4. *Компьютерная криптография*.

ОСНОВНЫЕ ГРУППЫ МЕЖДУНАРОДНЫХ СТАНДАРТОВ МЕХАНИЗМОВ БЕЗОПАСНОСТИ

Стандарты механизмов шифрования
ISO/IEC 10116; ISO/IEC 18033

Стандарты цифровой подписи
ISO/IEC 9796; ISO/IEC 14888

Стандарты на функции хеширования
ISO/IEC 13888; ISO/IEC 9798

Стандарты на генераторы случайных и простых чисел
ISO/IEC 18031; ISO/IEC 18032

Стандарт на протоколы аутентификации
ISO/IEC 9797

Стандарты на управление и сертификацию ключей
ISO/IEC 11770; ISO 11166 и др.

Криптографические системы и примитивы

"Бесключевые"

"однонаправленные" функции (прямая функция считается легко, обратная, в идеале - бесконечно трудно)

хэш-функции

генераторы случайных чисел

математические примитивы для работы с большими простыми числами

Алгоритмы с секретным ключом,

(два или более объектов делят один секретный ключ)

симметричные шифры

имитовставки (message authentication code)

псевдо-случайные генераторы чисел

Алгоритмы с открытым ключом,

(секретный ключ знает только один объект, криптофункции выподняются при помощи пары "свой" секретный + "чужой" открытый ключ)

асимметричные шифры

электронно-цифровая подпись

криптографические протоколы

- Симметричные криптосистемы
- Ассиметричные криптосистемы
- Хеш-функции и цифровая подпись
- Криптографические протоколы
- Криптографический анализ

Симметричные криптосистемы

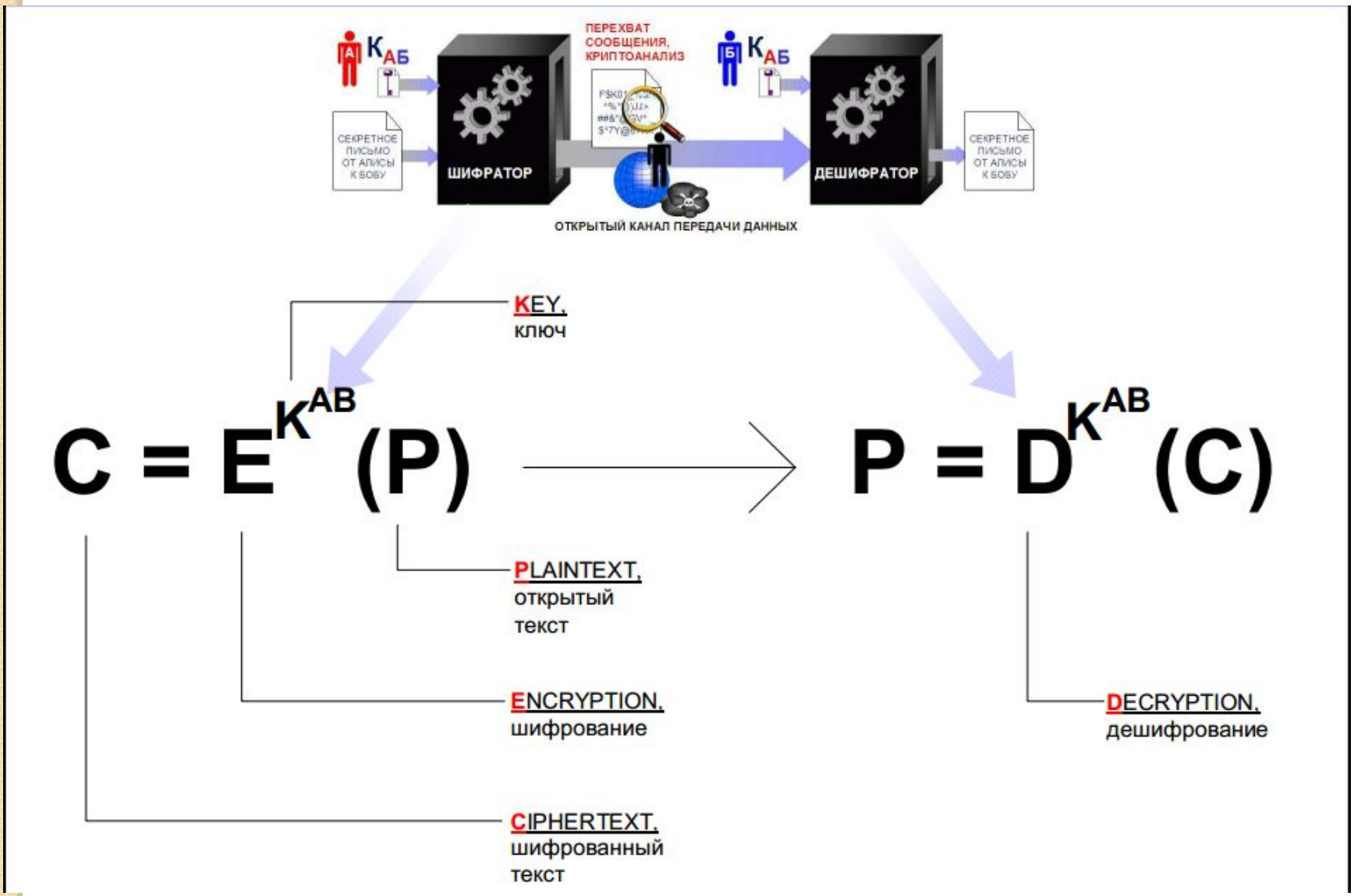
Симметричные криптосистемы

- криптосистемы, в которых для шифрования и расшифрования используется один и тот же общий секретный ключ, который необходимо предварительно распределить между обеими сторонами по защищенному каналу связи

Принцип Керкгоффса: секретность сообщения всецело зависит от знания ключа, т.е. предполагается, что метод шифрования/дешифрования известен злоумышленнику.


ПЕРЕХВАТ
СООБЩЕНИЯ,
КРИПТОАНАЛИЗ
А
К АБ
Б
К АБ
СЕКРЕТНОЕ
ПИСЬМО
ОТ АЛИСЫ
К БОБУ
ШИФРАТОР
ДЕШИФРАТОР
СЕКРЕТНОЕ
ПИСЬМО
ОТ АЛИСЫ
К БОБУ
ОТКРЫТЫЙ КАНАЛ ПЕРЕДАЧИ ДАННЫХ

ОБОБЩЕННАЯ СХЕМА СИММЕТРИЧНОЙ КРИПТОСИСТЕМЫ



Принципы впервые изложены в книге Керкгоффса «Военная криптография» (1883 год)

1. Система должна быть физически, если не математически, невскрываемой.
2. Нужно, чтобы не требовалось сохранение системы в тайне; попадание системы в руки врага не должно причинять неудобств.
3. Хранение и передача ключа должны быть осуществимы без помощи бумажных записей; корреспонденты должны располагать возможностью менять ключ по своему усмотрению.
4. Система должна быть пригодной для сообщения через телеграф.
5. Система должна быть легко переносимой, работа с ней не должна требовать участия нескольких лиц одновременно.
6. От системы требуется, учитывая возможные обстоятельства её применения, чтобы она была проста в использовании, не требовала значительного умственного напряжения или соблюдения большого количества правил.

- По мнению К. Шеннона, в практических шифрах необходимо использовать два общих принципа: *рассеивание* и *перемешивание*.

- *Рассеивание* представляет собой распространение влияния одного знака открытого текста на много знаков шифртекста, что позволяет скрыть статистические свойства открытого текста.

- *Перемешивание* предполагает использование шифрующих преобразований, которые усложняют восстановление взаимосвязи статистических свойств открытого и шифрованного текстов. Однако шифр должен не только затруднять раскрытие, но и обеспечивать легкость шифрования и расшифрования при известном пользователю секретном ключе.

ОСНОВНЫЕ ТИПЫ АЛГОРИТМОВ ШИФРОВАНИЯ

- *Поточный* – выполняет шифрование над каждым битом либо байтом исходного (открытого) текста с использованием гаммирования (наложения маски). Может быть легко реализован на основе блочного.

- *Блочный* – обрабатывает информацию блоками определённой длины (64, 128, 256 бит), применяя к блоку ключ в установленном порядке, как правило, несколькими циклами перемешивания и подстановки (раундами).

- **Блочные симметричные шифры:**
  - Обрабатывают информацию блоками определённой длины

- Каждый блок шифруется с использованием ключа.
  - Большинство симметричных шифров используют сложную комбинацию большого количества подстановок и перестановок.
  - Многие шифры исполняются в несколько (иногда до 80) проходов.

Алгебраические структуры

Группы

Группа (G) — набор элементов с *бинарной операцией* "•" обладает четырьмя свойствами (или удовлетворяет *аксиомам*), которые будут перечислены ниже.

Коммутативная группа, также называемая **абелева**, — группа, в которой оператор обладает теми же четырьмя свойствами для групп плюс дополнительным — коммутативностью. Эти пять свойств определены ниже

- **Замкнутость.** Если a и b — элементы G , то c = a • b — также элемент G . Это означает, что результат применения операции с любыми двумя элементами множества есть элемент этого множества.
- **Ассоциативность.** Если a , b и c — элементы G , то верно (a• b) • c = a• (b •c) Другими словами, не имеет значения, в каком порядке мы применяем операцию более чем с двумя элементами.
- **Коммутативность.** Для всех a и b в G мы имеем a • b = b • a . Обратите внимание, что это свойство должно быть верно только для коммутативной группы.
- **Существование нейтрального элемента.** Для всех элементов в G существует элемент e , который называется нейтральным элементом, такой, что e • a = a • e = a .
- **Существование инверсии.** Для каждого a в G существует элемент a' , называемый инверсией, такой, что a • a' = a' • a = e .

1. Замкнутость
2. Ассоциативность
3. Коммутативность
4. Существование нейтрального элемента
5. Существование инверсии

Замечание:
Третье свойство необходимо удовлетворять только для коммутативной группы

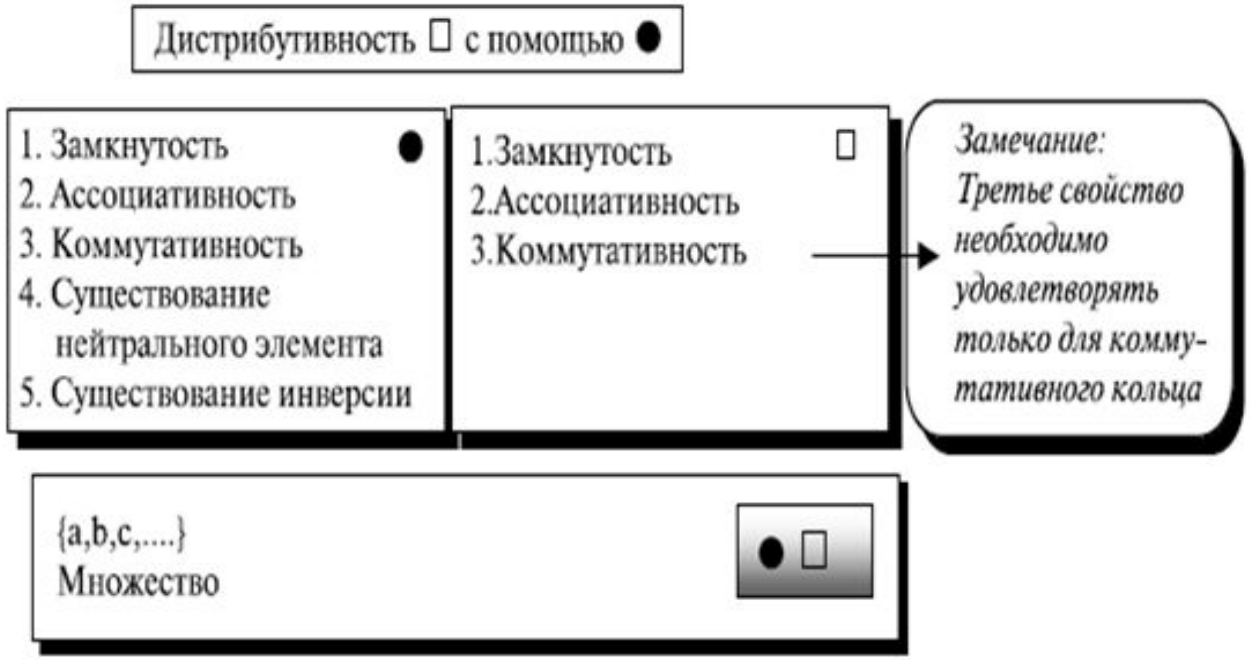
Группа

Кольцо

Кольцо, обозначенное как $R = \{...\}, \bullet, \perp$, является алгебраической структурой с двумя операциями. Первая операция должна удовлетворять всем пяти свойствам, требуемым для абелевой группы. Вторая операция должна удовлетворять только первым двум свойствам абелевой группы. Кроме того, вторая операция должна быть распределена с помощью первой.

Дистрибутивность означает, что для всех a, b и c элементов из R мы имеем $a \perp (b \bullet c) = (a \perp b) \bullet (a \perp c)$ и $(a \bullet b) \perp c = (a \perp c) \bullet (b \perp c)$. *Коммутативное* кольцо — кольцо, в котором *коммутативное* свойство удовлетворено и

Множество Z с двумя операциями — сложением и умножением — является коммутативным кольцом, которое обозначается $R = Z, +, \times$. Сложение удовлетворяет всем пяти свойствам; умножение удовлетворяет только трем свойствам.

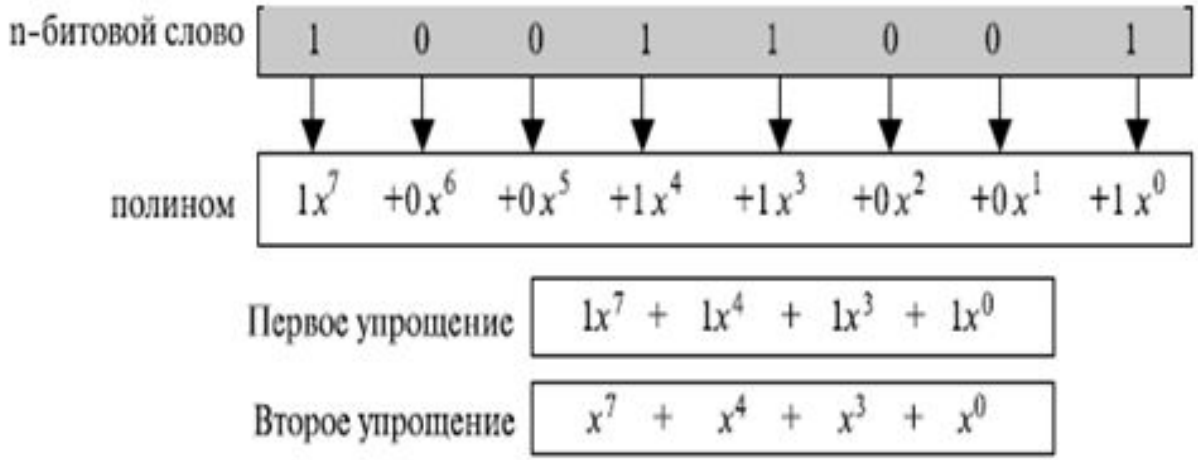


Кольцо

Поля GF(2n)

В криптографии мы часто должны использовать четыре *операции* (*сложение, вычитание, умножение* и *деление*). Другими словами, мы должны использовать поля. Однако когда мы работаем с компьютерами, положительные целые числа сохраняются в компьютере как `n` -битовые слова, в которых `n` является обычно `8, 16, 32, 64` , и так далее. Это означает, что *диапазон* целых чисел — `0` до $2^n - 1$. *Модуль* — 2^n . Так что возможны два варианта, если мы хотим использовать *поле*.

Использование полиномов для предоставления слова из `8` бит `(10011001)`



ОПЕРАТОРЫ, ИСПОЛЬЗУЕМЫЕ ДЛЯ ПОСТРОЕНИЯ БЛОЧНЫХ ШИФРОВ

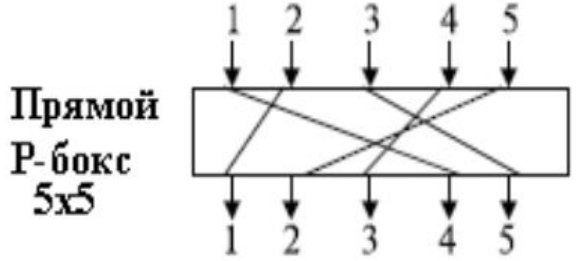
1. операторы перестановки, называемые P - блоками;
2. операторы подстановки, называемые S - блоками;
3. операция исключающего ИЛИ;
4. циклический сдвиг;
5. замена;
6. разбиение и объединение блока.

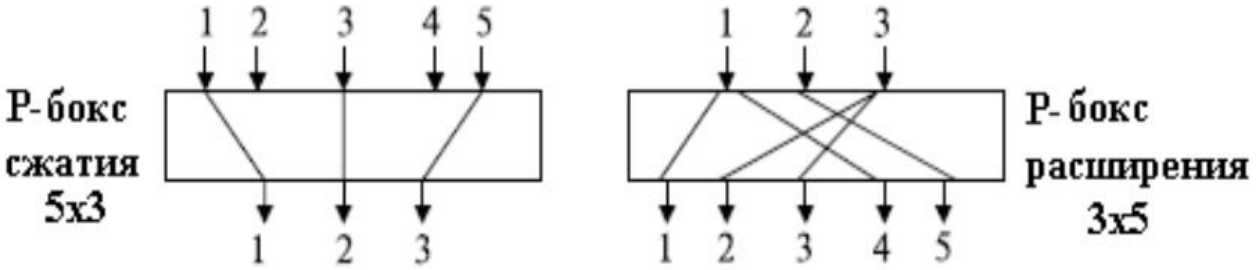
Компоненты блочного шифра

1. P-блок перестановки

P*-бокс** (блок перестановки***) подобен традиционному шифру перестановки (переставляемые символы – биты). Возможны 3 типа P -боксов:

- ***прямые* P-боксы** (простая перестановка символов, n входов и n выходов, всего возможно $n!$ отображений);
- ***P*-боксы расширения**
- ***P*-боксы сжатия**





- **2. S – блок подстановки**

***S*-бокс** (блок подстановки) – это миниатюрный шифр подстановки. На вход в S-бокс может подаваться n-битовый блок, а на выходе выйти уже m- битовый блок, где не всегда $m = n$. (говорят бокс с n входами и m выходами).

Обозначим в S-боксе с n входами и m выходами входы через $x_1, x_2, ..., x_n$, а выходы – через $y_1, y_2, ..., y_m$. Связь между входами и выходами задает система уравнений

$$\begin{cases} y_1 = f_1(x_1, x_2, ..., x_n), \\ y_2 = f_2(x_1, x_2, ..., x_n), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_m = f_m(x_1, x_2, ..., x_n) \end{cases}$$

S-боксы делятся на линейные и нелинейные. В ***линейном*** S-***боксе*** эту связь можно записать в виде линейных соотношений

$$\begin{cases} y_1 = a_{11}x_1 \oplus a_{12}x_2 \oplus \ldots \oplus a_{1n}x_n, \\ y_2 = a_{21}x_1 \oplus a_{22}x_2 \oplus \ldots \oplus a_{2n}x_n, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_m = a_{m1}x_1 \oplus a_{m2}x_2 \oplus \ldots \oplus a_{mn}x_n \end{cases}$$

В ***нелинейном*** S-***боксе*** линейные соотношения для каждого выхода задать нельзя.

Пример. В S-боксе с тремя входами и двумя выходами мы имеем

$$\begin{cases} y_1 = x_1 \oplus x_2 \oplus x_3, \\ y_2 = x_1 \end{cases}$$

S-бокс линеен, может быть представлен в матричном виде:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

В ***нелинейном*** S***-боксе*** линейные соотношения для каждого выхода задать нельзя.

Пример. В S-боксе с тремя входами и двумя выходами мы имеем

$$\begin{cases} y_1 = x_1 \oplus x_2 \oplus x_3, \\ y_2 = x_1 \end{cases}$$

S-бокс линеен, может быть представлен в матричном виде:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Пример. В S-боксе с тремя входами и двумя выходами мы имеем

$$\begin{cases} y_1 = x_1^3 \oplus x_2, \\ y_2 = x_1 \oplus x_1 x_2 \oplus x_3 \end{cases}$$

(умножение и сложение проводится в GF(2)). S-бокс нелинеен, так как нет линейных соотношений между входами и выходами.

3. Операция XOR (операция исключающее ИЛИ)

Операция XOR – побитовое сложение, сложения по модулю 2.

Свойства операции XOR:

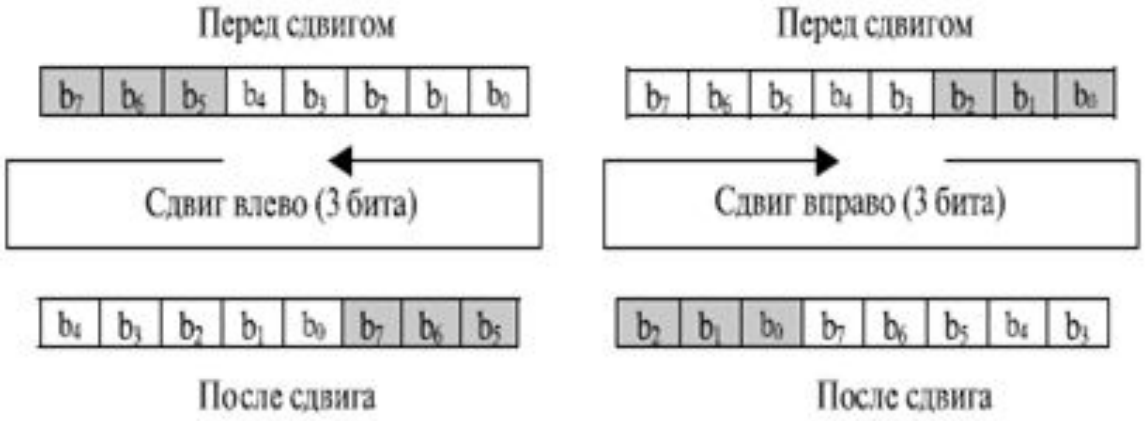
1. Замкнутость: если x, y – n-битовые слова, то $x \oplus y = z$, где z – n-битовое слово.
2. Ассоциативность: $x \oplus (y \oplus z) = (x \oplus y) \oplus z$.
3. Коммутативность: $x \oplus y = y \oplus x$.
4. Существование нулевого элемента с условием $x \oplus (00...0) = x$.
5. Существование инверсии – операция XOR слова с самим собой дает нулевой элемент $x \oplus x = (00...0)$.

Тесно связаны с XOR две операции:

операция дополнения – инвертирование каждого бита в блоке: 0 на 1, а 1 на 0. Если $\bar{x}$ – дополнение x, то $x \oplus \bar{x} = (11...1)$ и $x \oplus (11...1) = \bar{x}$.

операция инверсии, по которой из $y = x \oplus k$ следует $x = y \oplus k$, где k – ключ.

ЦИКЛИЧЕСКИЙ СДВИГ

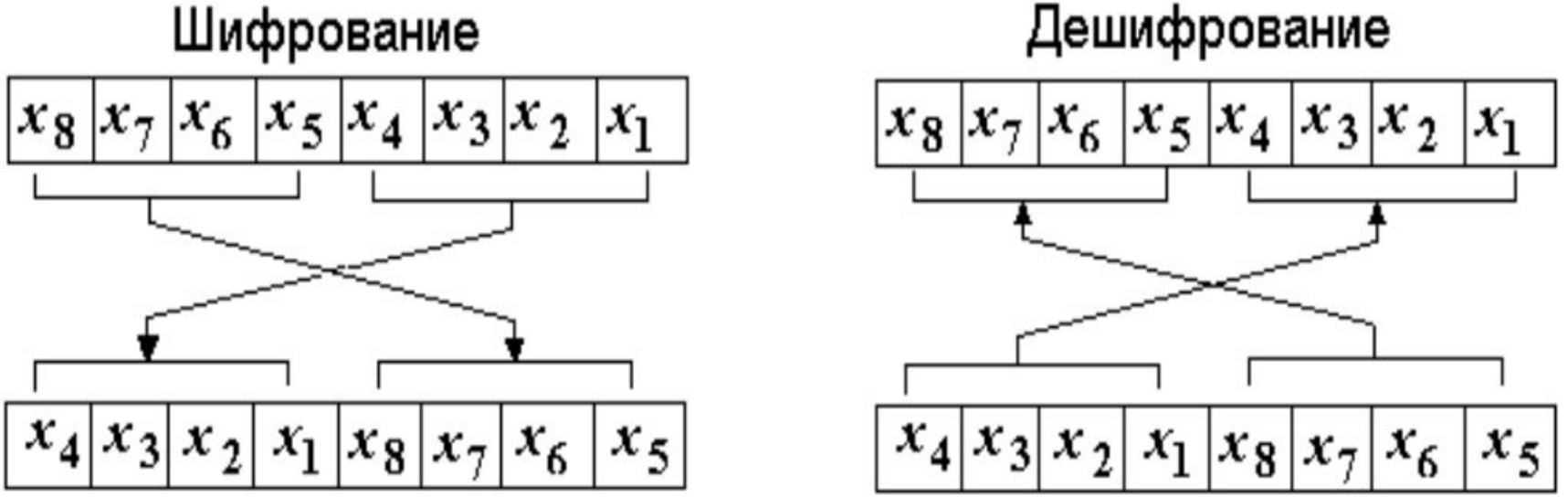


Свойства циклического сдвига:

- смещение по модулю n. Другими словами, если $k = 0$ или $k = n$, никакого смещения не происходит. Если $k > n$, то входная информация сдвигается на $mod\ k\ n$ бит.
- если смещение делается неоднократно, то вновь может появиться исходное n-битовое слово (сдвиг является групповой операцией).

Замена

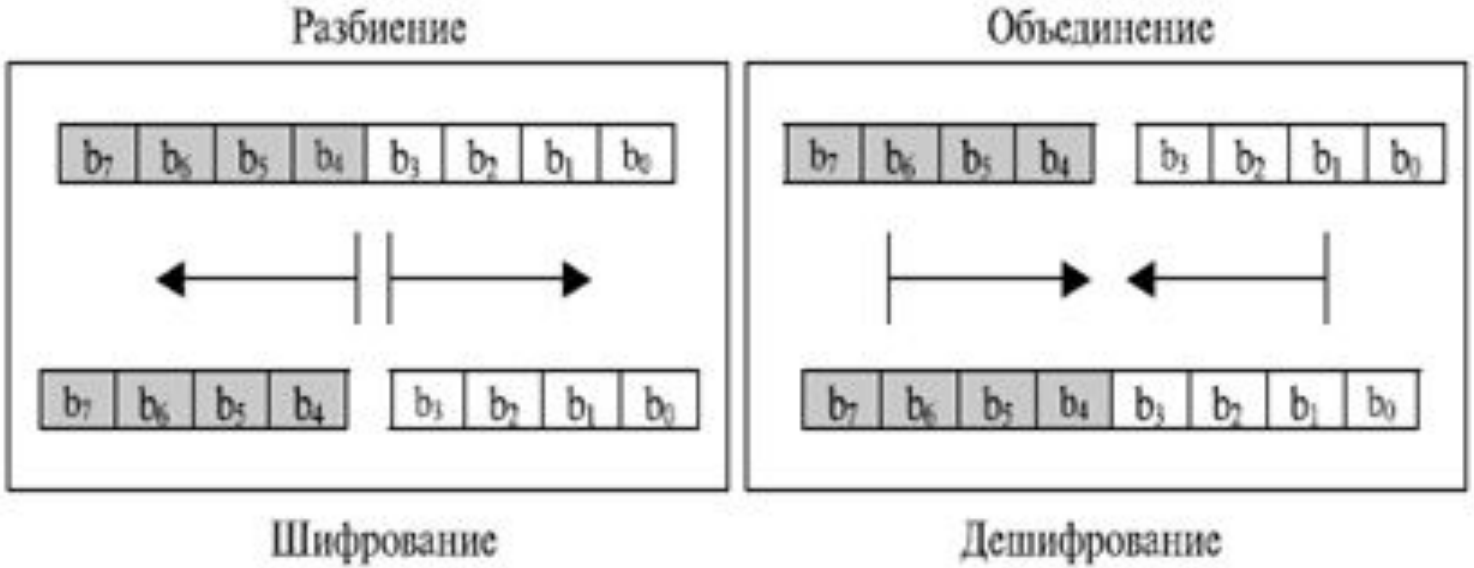
Замена – частный случай операции циклического сдвига на $k = n/2$ битов (здесь n – четное). Сдвиг влево на $n/2$ – то же самое, что сдвиг на $n/2$ вправо, $\Rightarrow$ операция является обратимой.



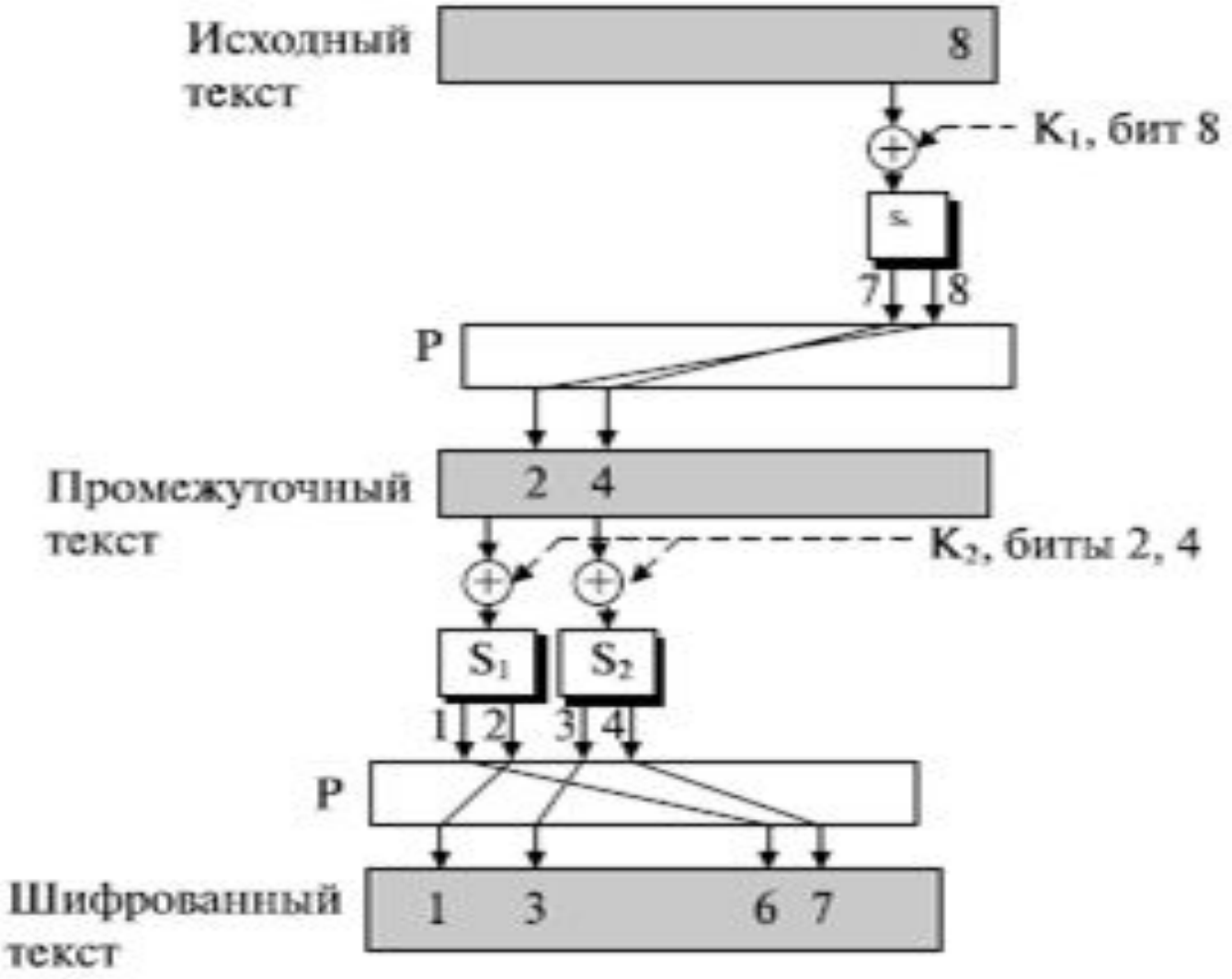
Операция замена в 8 битовом а слове

РАЗБИЕНИЕ И ОБЪЕДИНЕНИЕ

Разбиение разделяет n-битовое слово пополам, создавая два слова равной длины. ***Объединение*** связывает два слова равной длины, чтобы создать n-битовое слово. Эти две операции инверсны друг другу: если одна используется для шифрования, то другая – для дешифрования.



8. РАССЕИВАНИЕ И ПЕРЕМЕШИВАНИЕ



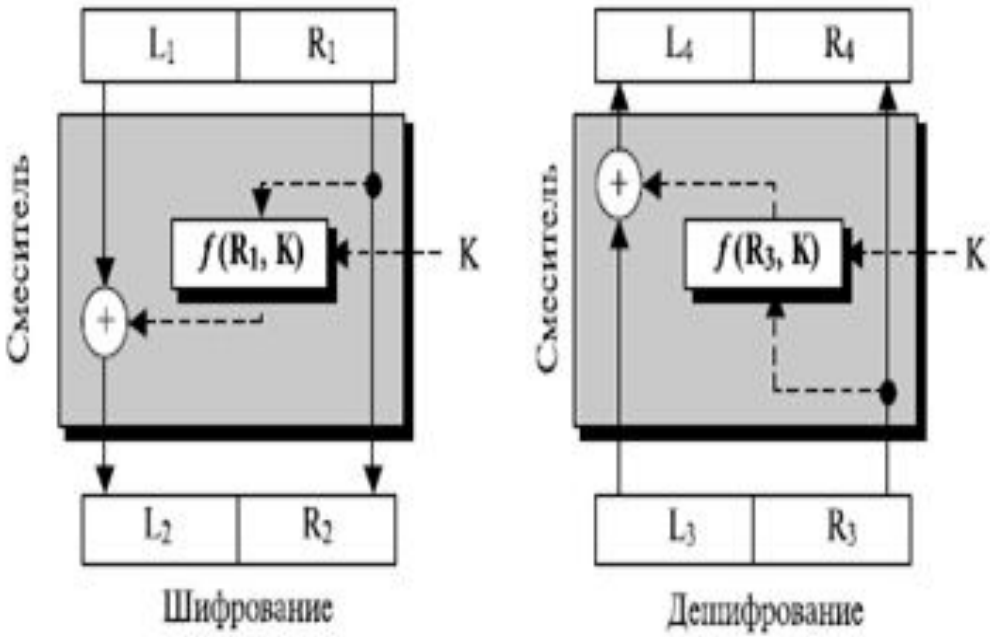
ПАРАМЕТРЫ АЛГОРИТМОВ СИММЕТРИЧНОГО ШИФРОВАНИЯ

- длина обрабатываемого блока
- длина ключа
- число раундов
- криптостойкость
- сложность аппаратной/программной реализации

Сеть Фейстеля (конструкция Фейстеля) — один из методов построения блочных шифров. Сеть представляет из себя определённую многократно повторяющуюся (итерированную) структуру, называющуюся ячейкой Фейстеля. При переходе от одной ячейки к другой меняется ключ, причём выбор ключа зависит от конкретного алгоритма.

Сеть Фейштеля

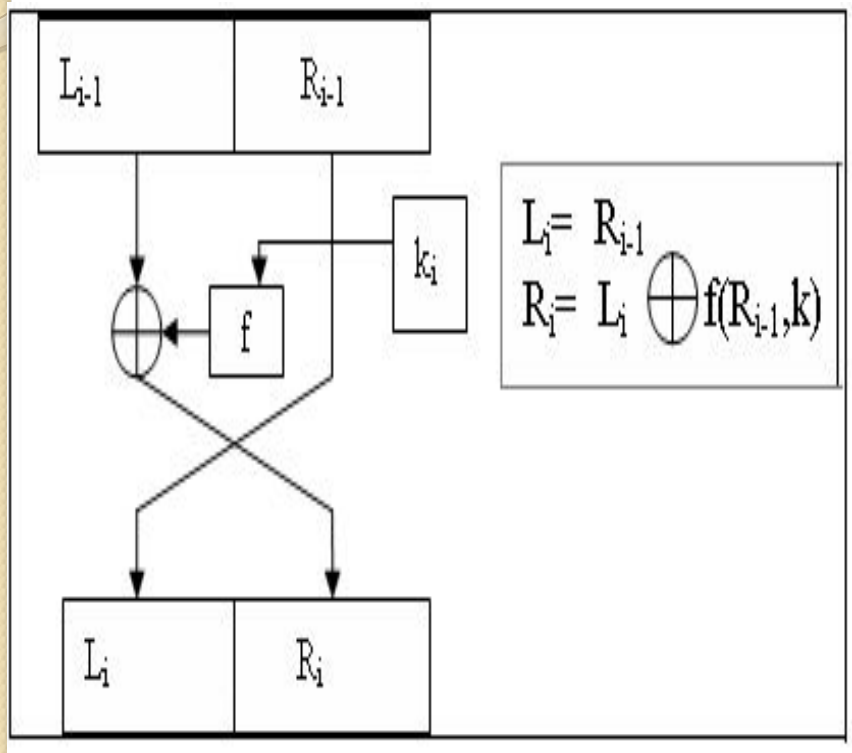
- Раунд сети Фейштеля

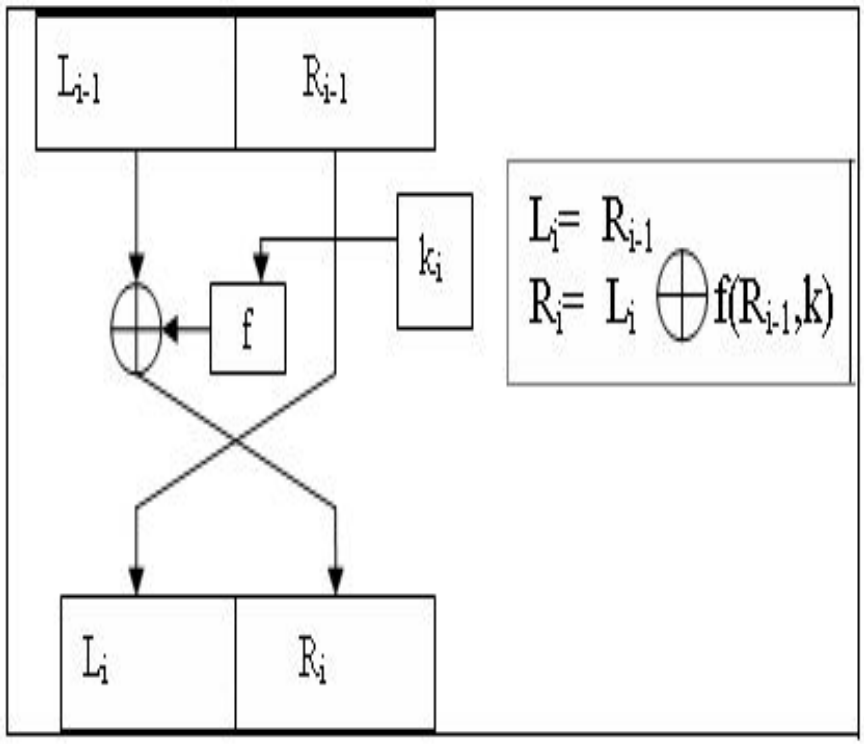


- Функция f () называется образующей.
- Каждый раунд состоит из вычисления функции f () для одной ветви и побитового выполнения операции XOR результата f () с другой ветвью.
- После этого ветви меняются местами

$$R_4 = R_3 = R_2 = R_1$$
$$L_4 = L_3 \oplus f(R_3, K) = L_2 \oplus f(R_2, K) = L_1 \oplus f(R_1, K) \oplus$$
$$f(R_1, K) = L_1$$

СТРУКТУРА СЕТИ ФЕЙСТЕЛЯ
(шифрование и расшифровывание)





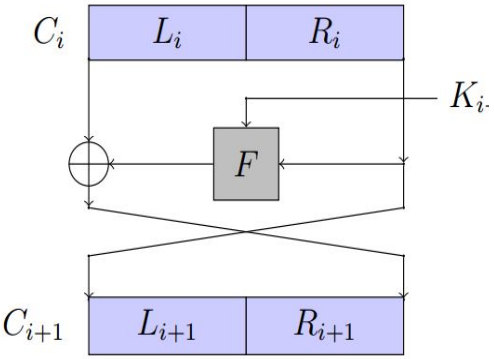


Схема Фейстеля

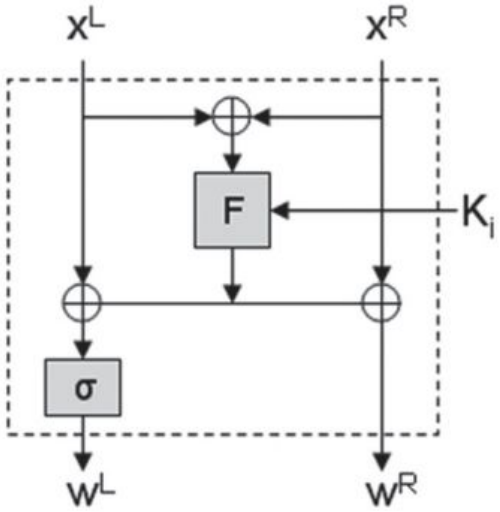
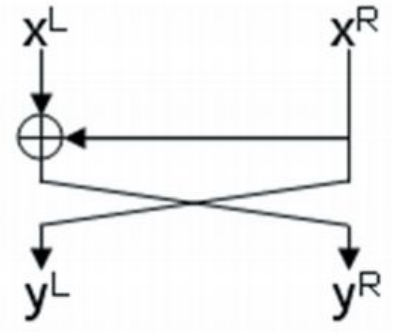


Схема Лей-Месси



Функция σ

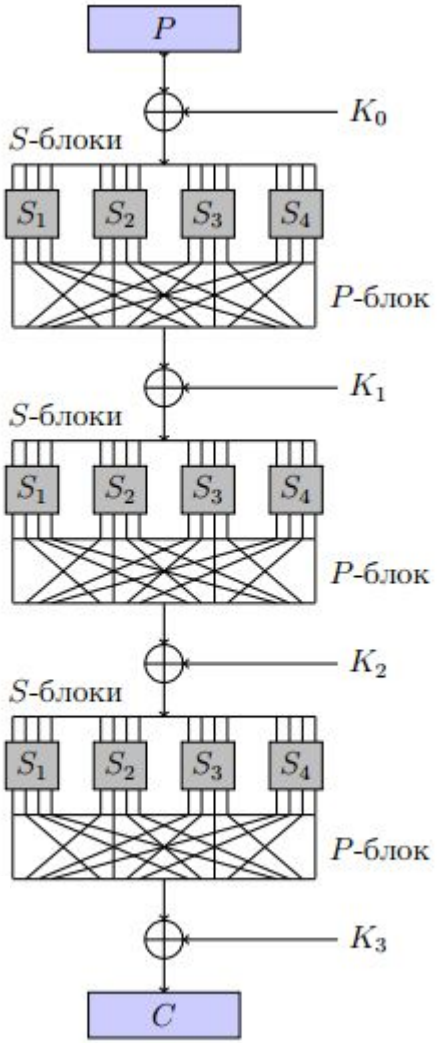
$$L_i = \sigma(L_{i-1} \oplus F(L_{i-1} \oplus R_{i-1}, K_{i-1})),$$

$$R_i = R_{i-1} \oplus F(L_{i-1} \oplus R_{i-1}, K_{i-1}), \text{ при } i \in 1...n-1.$$

На последнем раунде преобразование σ отсутствует:

$$L_n = L_{n-1} \oplus F(L_{n-1} \oplus R_{n-1}, K_{n-1}),$$

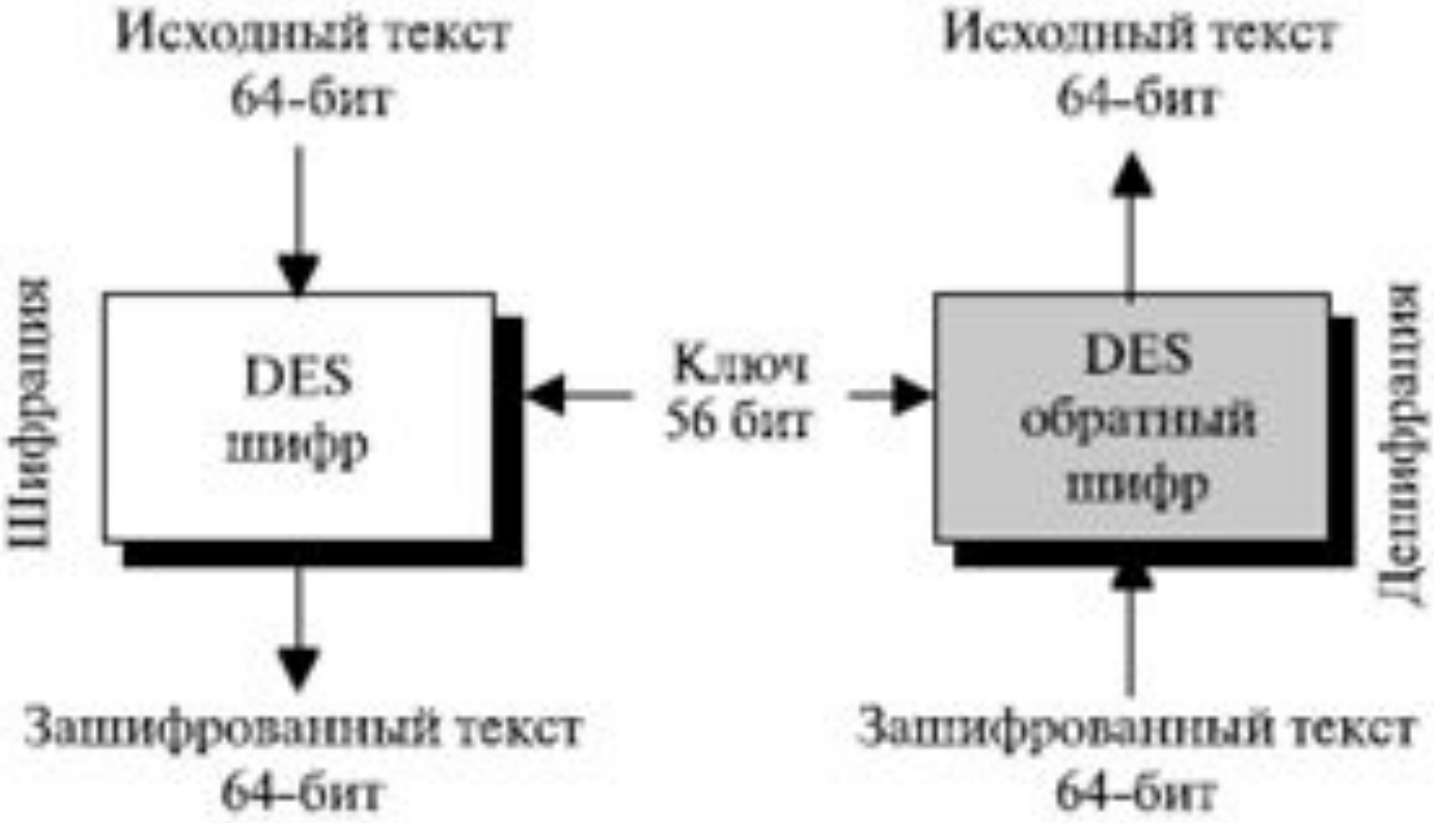
$$R_n = R_{n-1} \oplus F(L_{n-1} \oplus R_{n-1}, K_{n-1}).$$



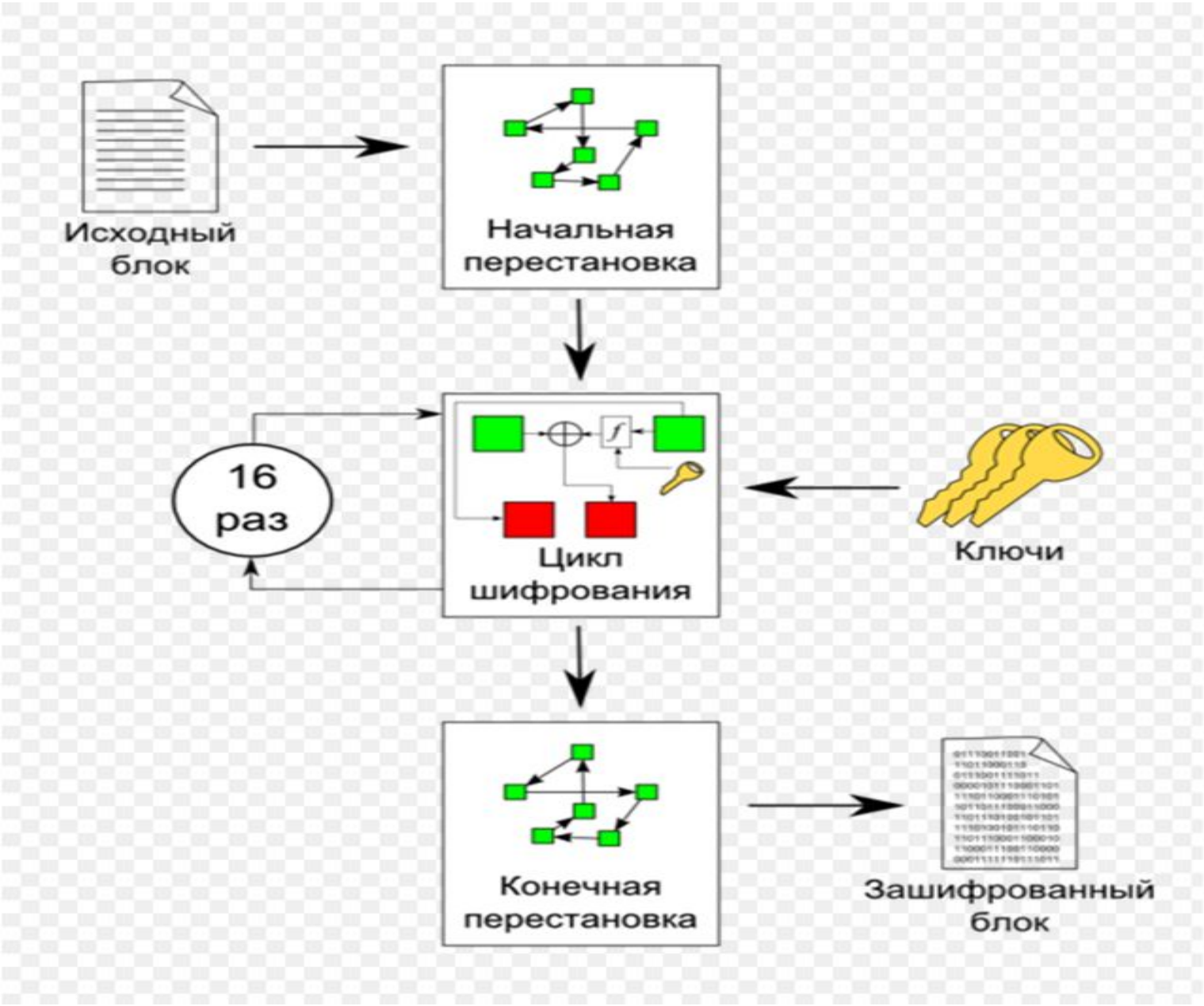
SP-сеть

DES (Data Encryption Standard) — симметричный алгоритм шифрования, в котором один ключ используется как для шифрования, так и для расшифровывания данных. DES разработан фирмой IBM и утвержден правительством США в 1977 году как официальный стандарт (FIPS 46-3). DES имеет блоки по 64 бит и 16 цикловую структуру сети Фейштеля, для шифрования использует ключ с длиной 56 бит. Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP-1) преобразований.

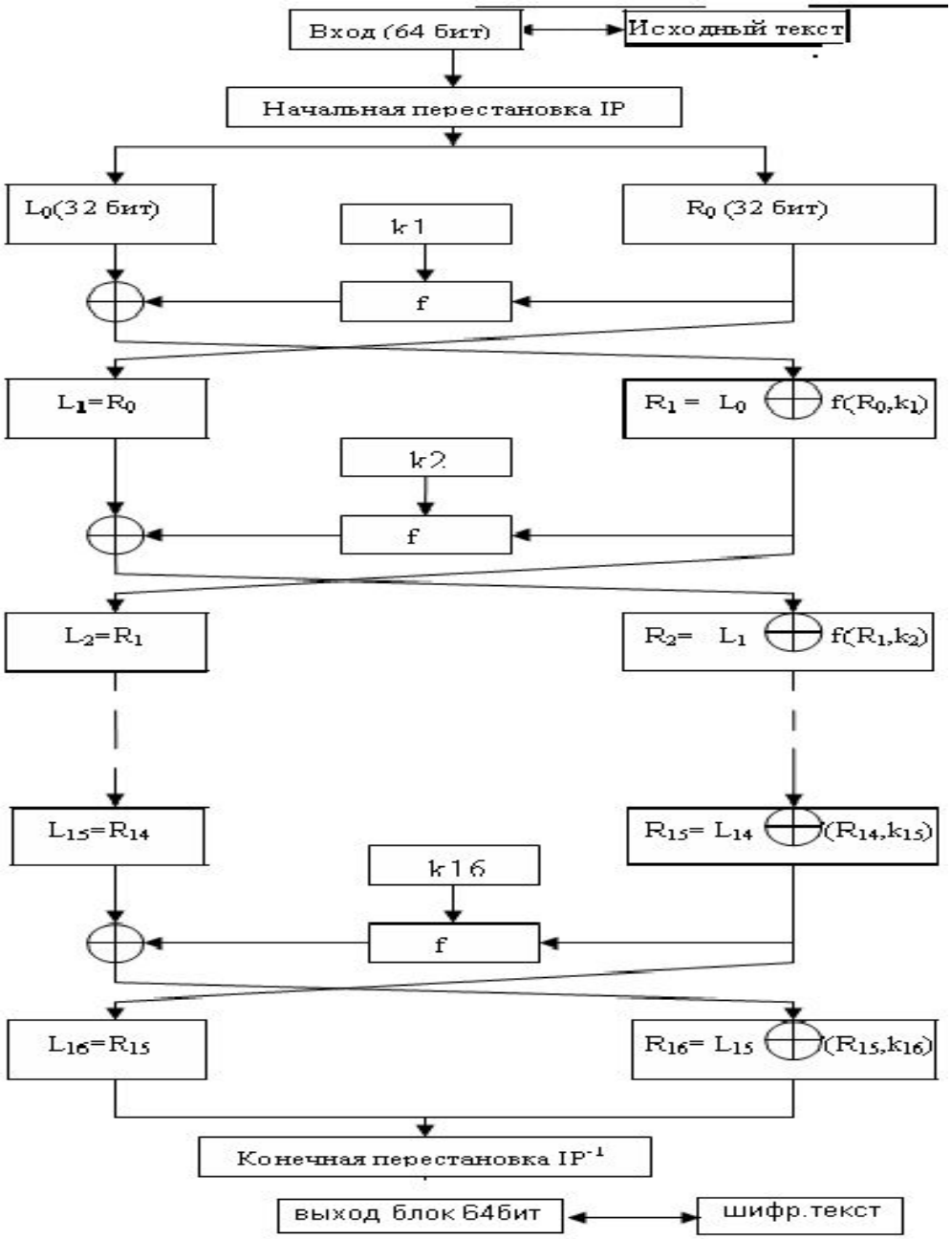
Стандарт шифрования DES



Обобщенная схема алгоритма DES

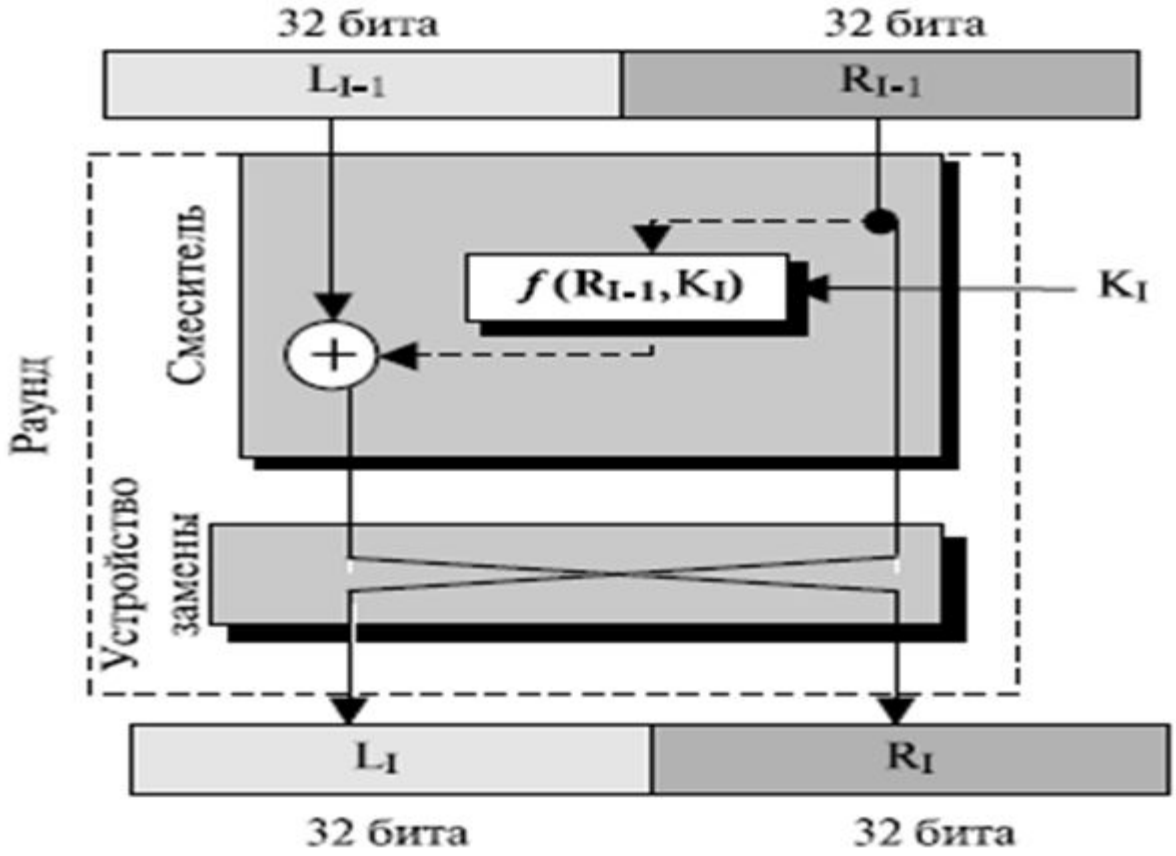


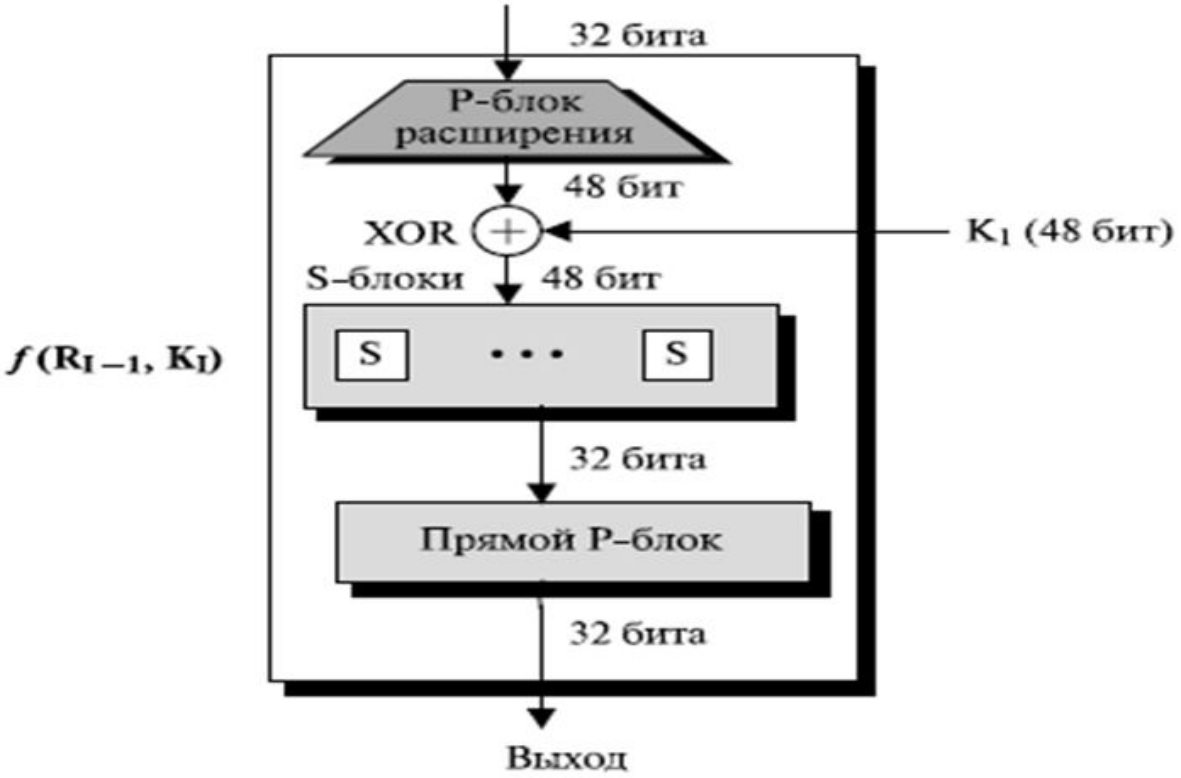
Алгоритм DES



Для вычисления значения функции f используются:

- функция E (расширение 32 бит до 48);
- функция S_1, S_2,..., S_8 (преобразование 6-битового числа в 4-битовое);
- функция P (перестановка битов в 32-битовой последовательности).




32 бита
P-блок расширения
48 бит
XOR
K_I (48 бит)
S–блоки
48 бит
$f(R_{I-1}, K_I)$
S
S
32 бита
Прямой P–блок
32 бита
Выход

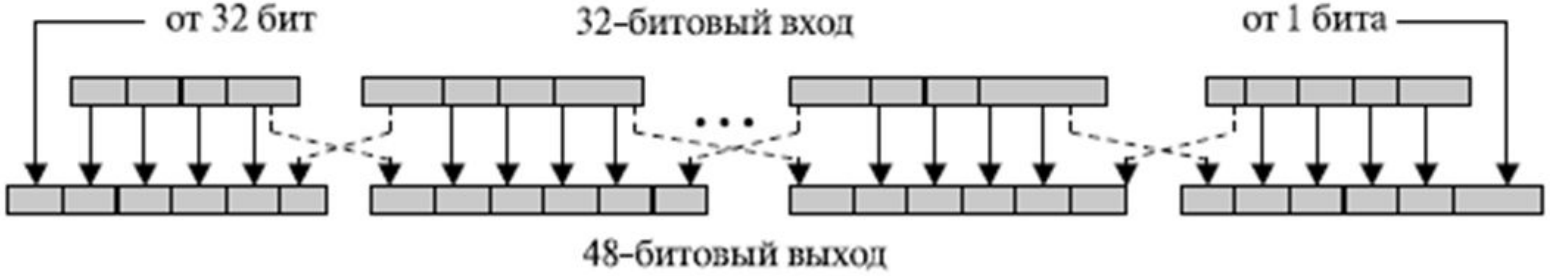
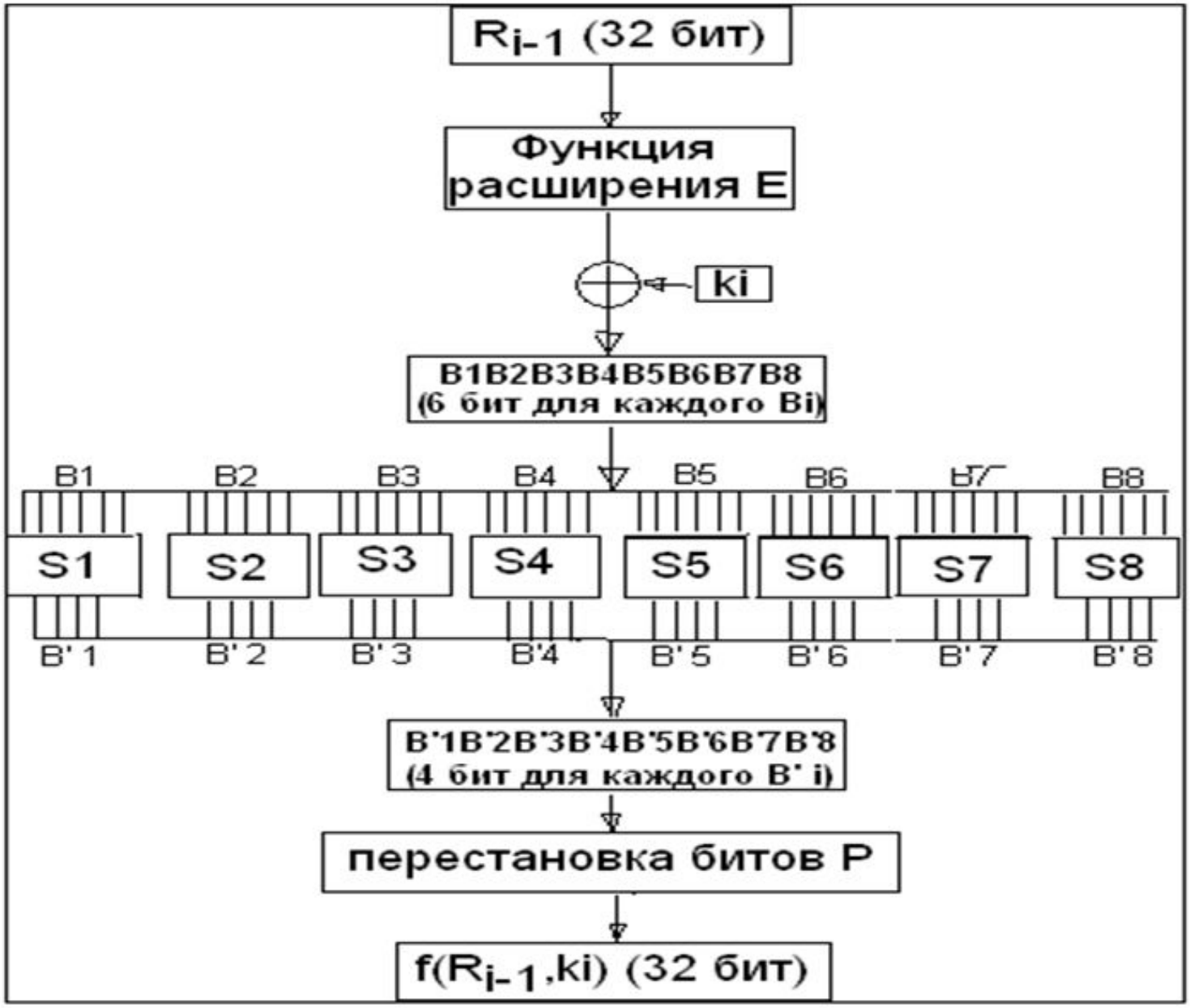
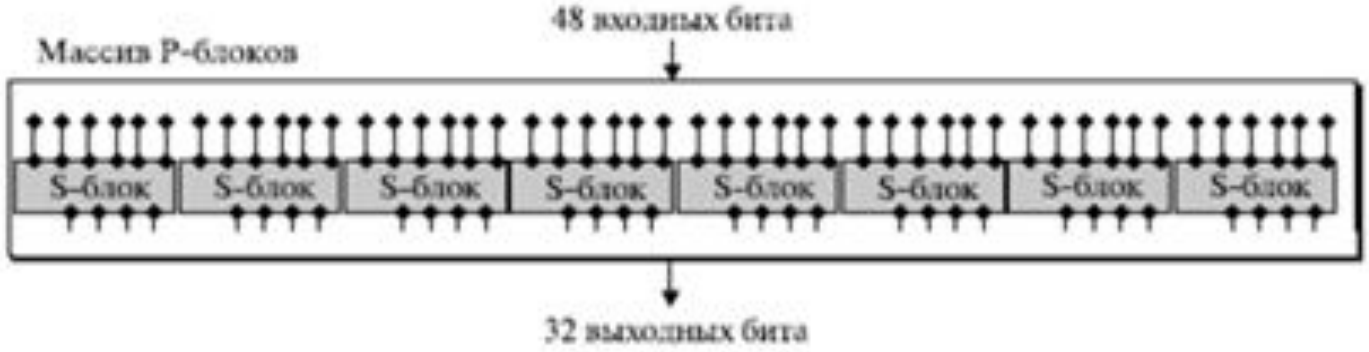
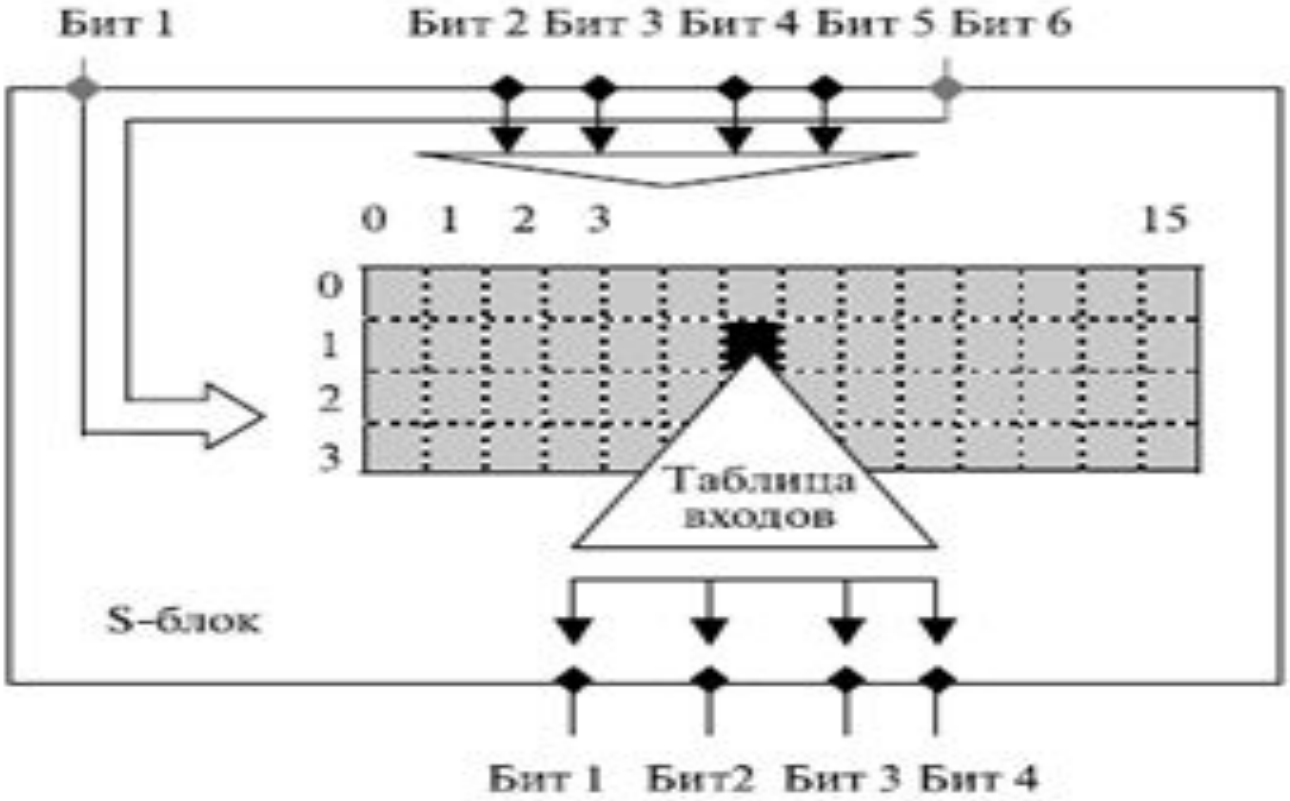

от 32 бит
32-битовый вход
от 1 бита
48-битовый выход

Схема вычисления функции шифрования f



Преобразование S-блока





Функция преобразования $S_1, S_2, \ldots, S_8$

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

S_1 S_2 S_3 S_4

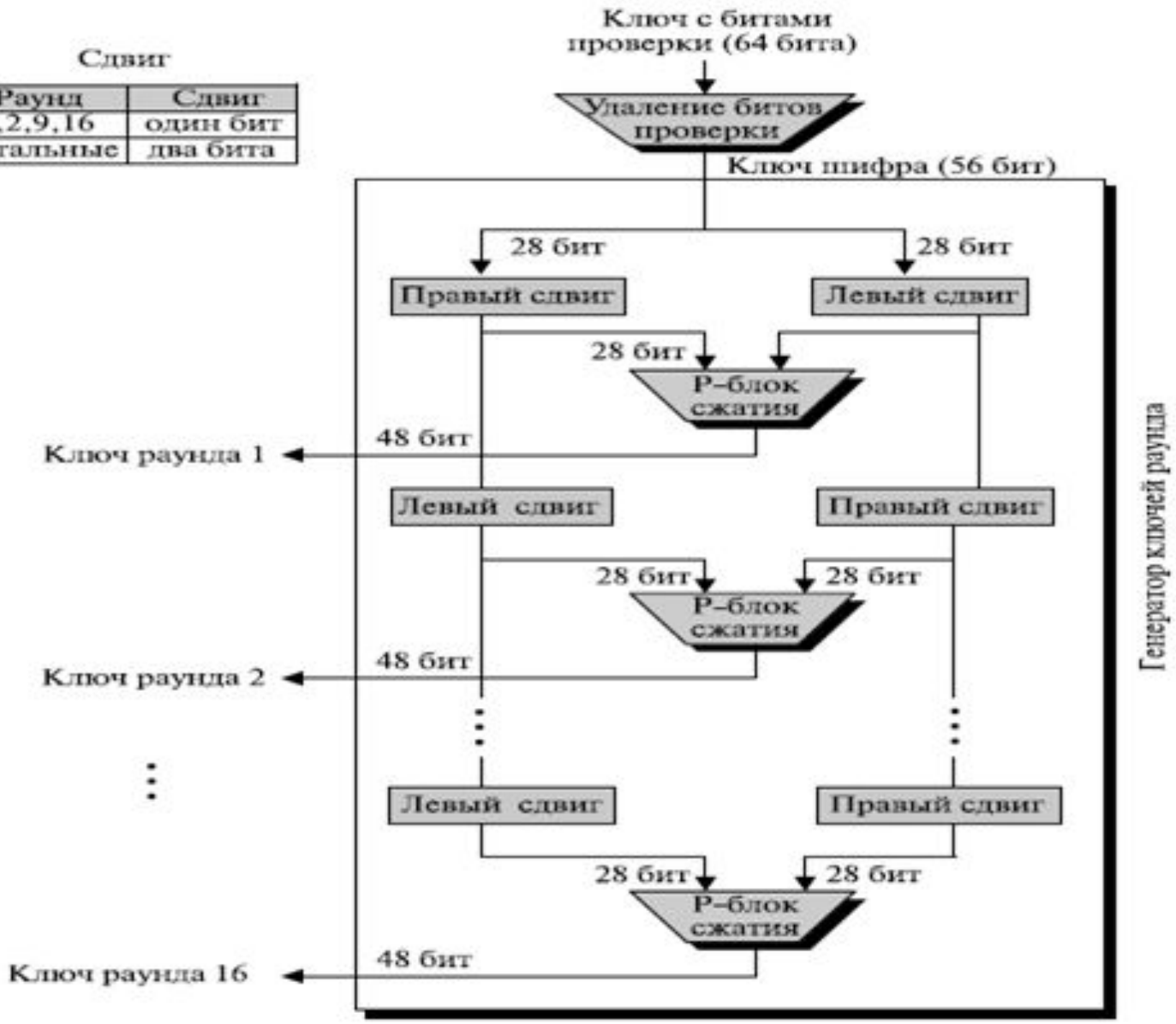
Функция P перестановки битов

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

Схема генерации ключа алгоритма DES

Сдвиг

Раунд	Сдвиг
1,2,9,16	один бит
Остальные	два бита



Функция G первоначальной подготовки ключа (переставленная выборка 1)

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 сдвигов Sj для вычисления ключа

Номер итерации	Количество S_i сдвигов влево, бит	Номер итерации	Количество S_i сдвигов влево, бит
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

Изменение длины ключа с 56 битов до 48 битов

Таблица сжатия ключа							
14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Трассировка данных в примере

Исходный текст: 123456ABCD 132536

После первоначальной перестановки: 14A7D67818CA18D После разбиения: L_0= 14A7D678 R_0 = 18CA18D

Раунд	Левая	Правая	Ключ 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

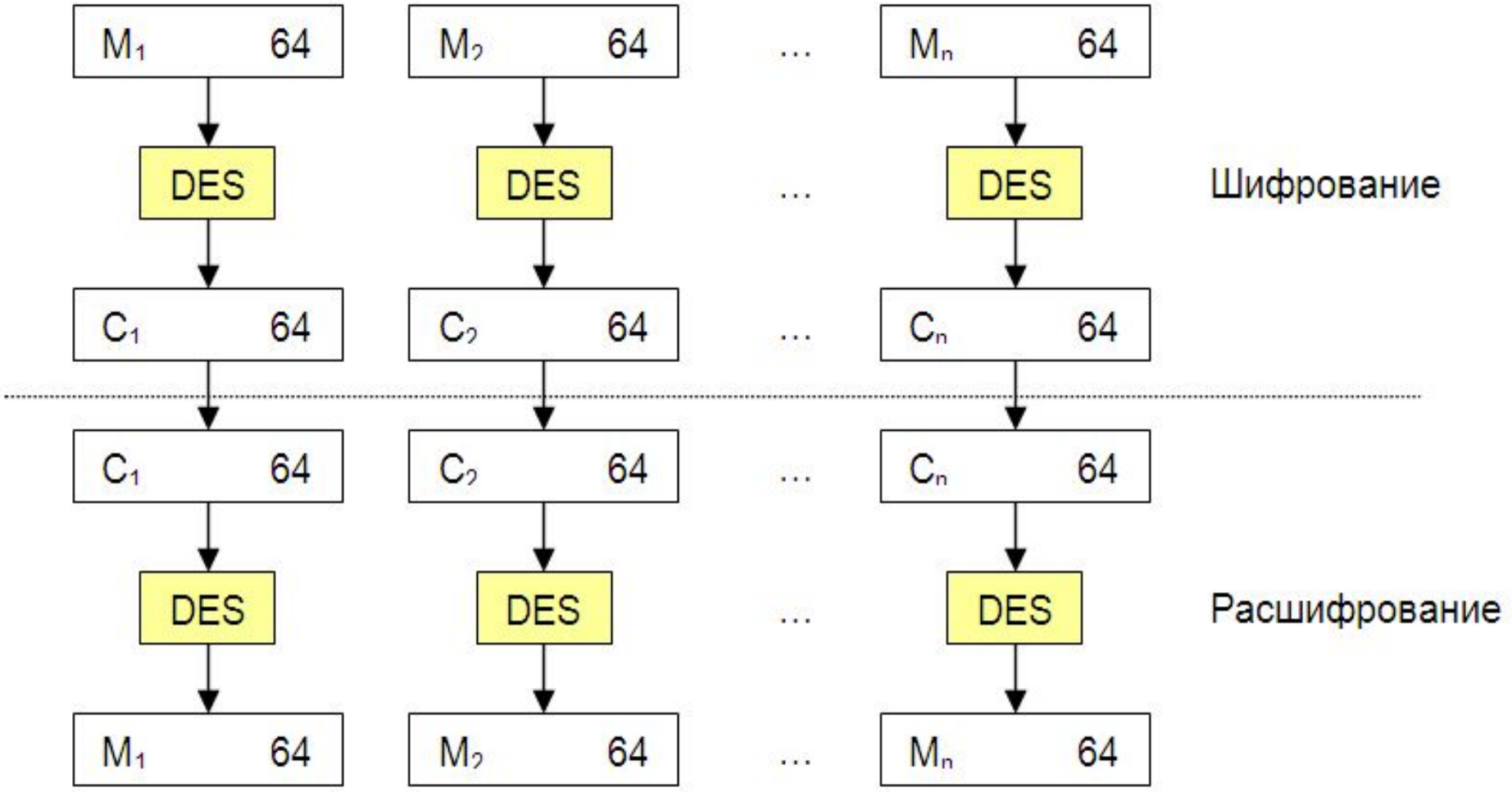
(после конечной перестановки)

ОСНОВНЫЕ РЕЖИМЫ РАБОТЫ АЛГОРИТМА DES

- электронная кодовая книга ECB (Electronic Code Book);
- сцепление блоков шифра CBC (Cipher Block Chaining);
- обратная связь по шифртексту CFB (Cipher Feed Back);
- обратная связь по выходу OFB (Output Feed Back).

ЭЛЕКТРОННАЯ КОДОВАЯ КНИГА ECB

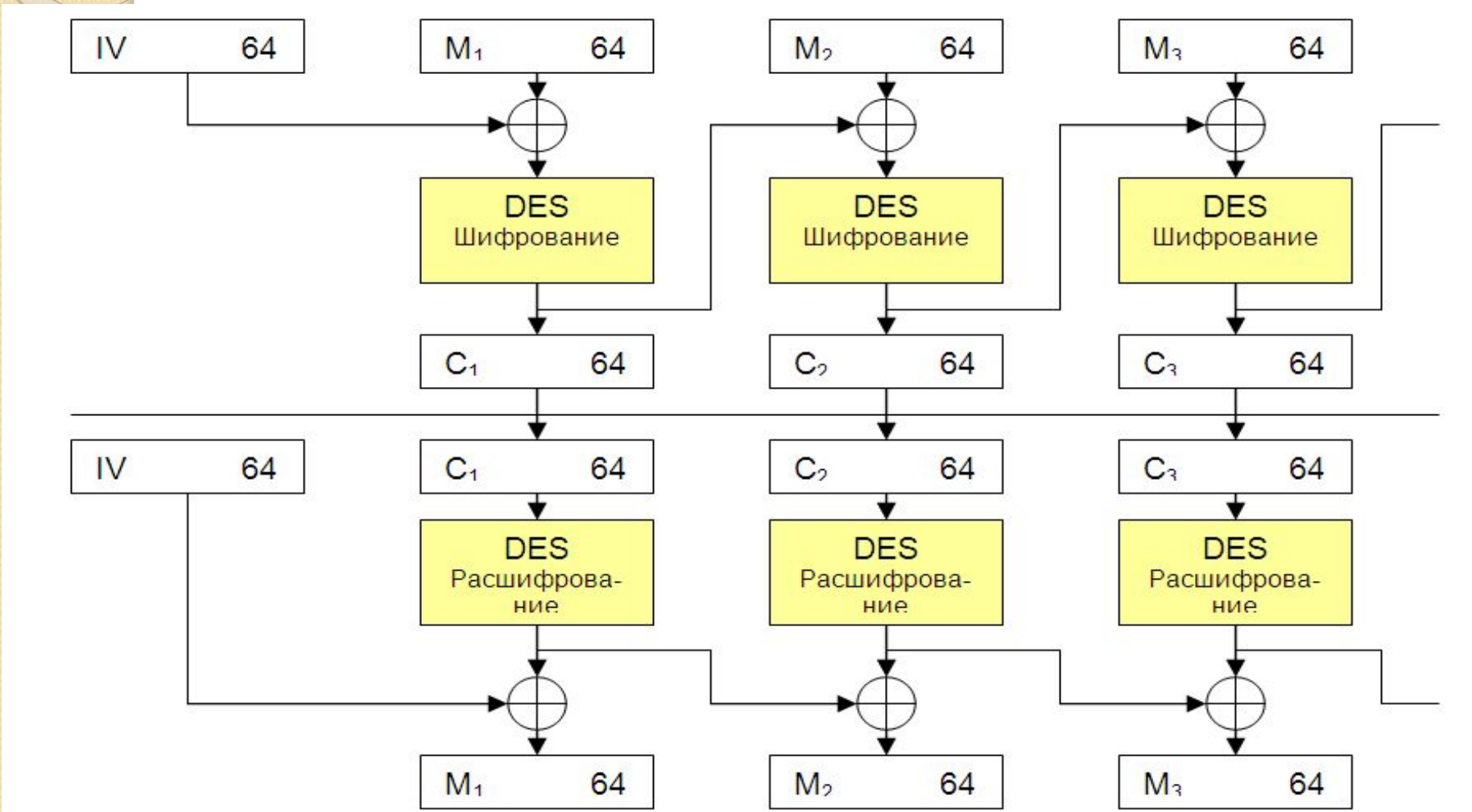
Шифруемый текст разбивается на блоки, при этом, каждый блок шифруется отдельно, не взаимодействуя с другими блоками.



СЦЕПЛЕНИЕ БЛОКОВ ШИФРА CBC

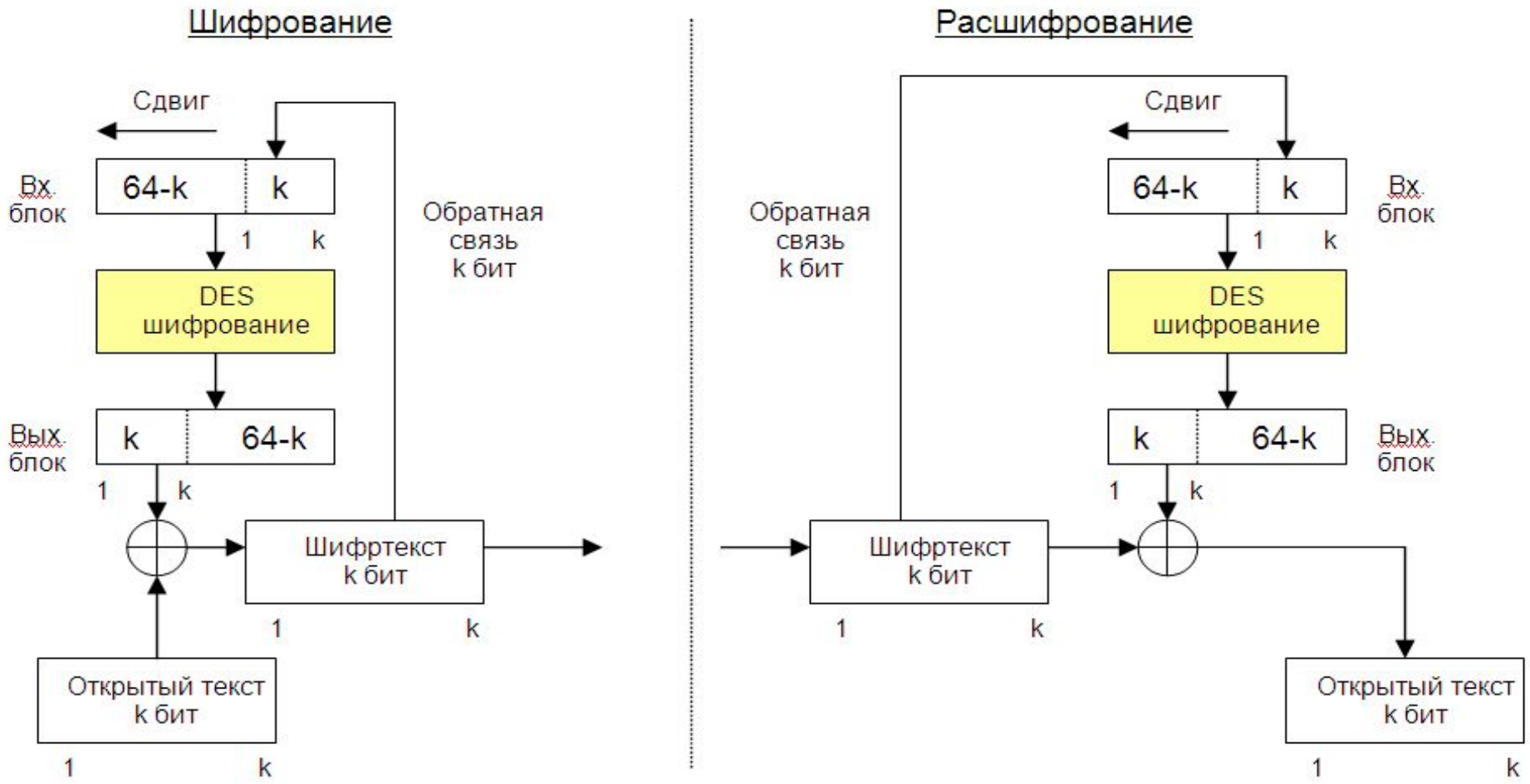
Блок C_i перед зашифровыванием складывается по модулю 2 со следующим блоком открытого текста M_{i+1}. Вектор C_0 — начальный вектор, он меняется и хранится в секрете.

$$C_i = DES_k(M_i \oplus C_{i-1})$$



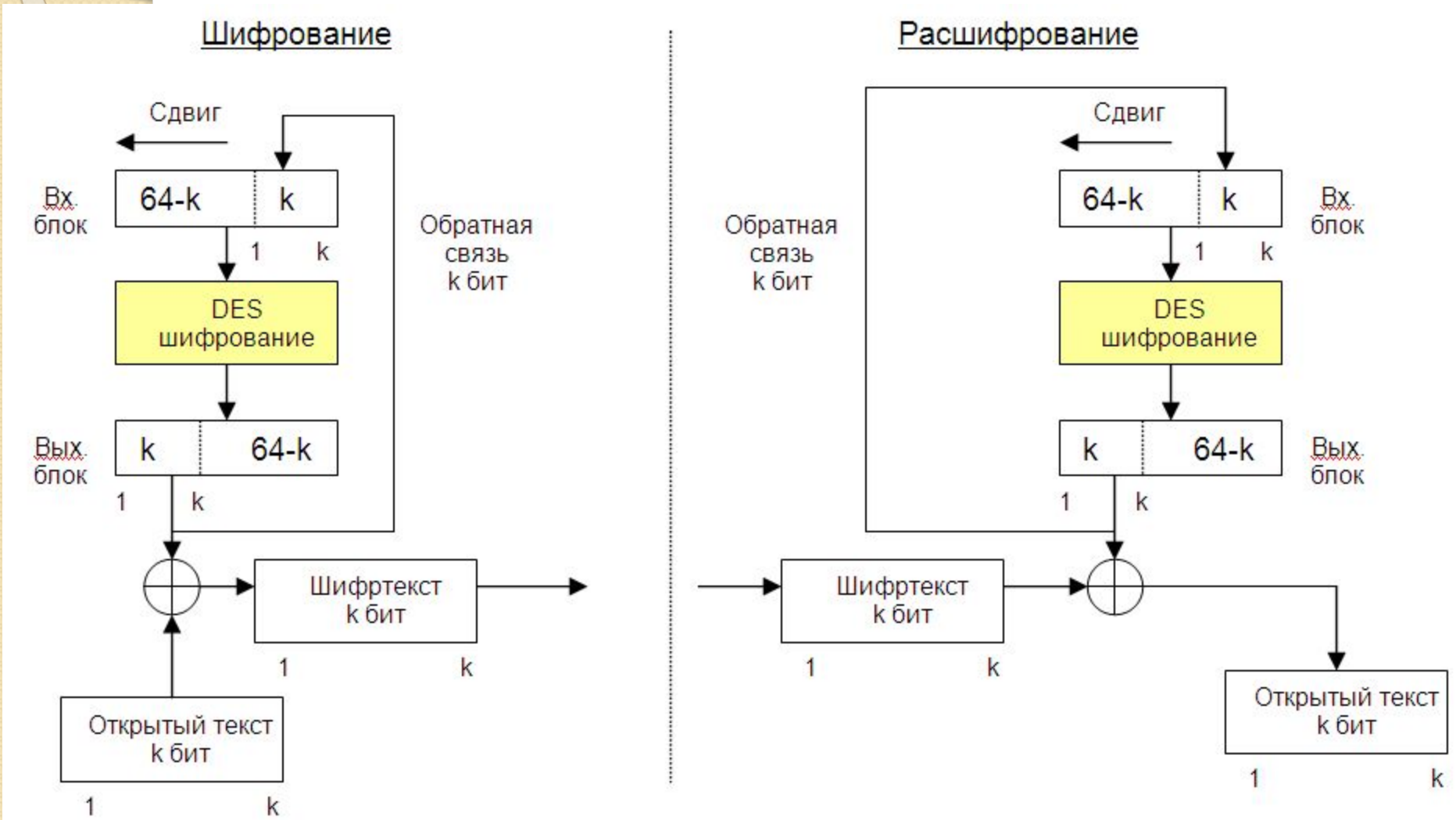
ОБРАТНАЯ СВЯЗЬ ПО ШИФРТЕКСТУ CFB

Вырабатывается блочная «гамма» $C_i = M_i \oplus P_{i-1}$.
Начальный вектор C_0 сохраняется в секрете,
где P_{i-1} – старшие k битов предыдущего зашифрованного текста



ОБРАТНАЯ СВЯЗЬ ПО ВЫХОДУ OFB

Вырабатывается блочная «гамма» $C_i = M_i \oplus P_i$
где P_i – старшие k битов операции $DES_k(C_{i-1})$, i>=1



КРИПТОСТОЙКОСТЬ DES

1. Небольшое количество ключей (всего 2^{56}) позволяет выполнить полный перебор за реальное время.
   В 1998 году компания The Electronic Foundation используя специальный компьютер DES-Cracker взломала шифр DES за 3 дня.
2. В DES существуют слабые и частично-слабые ключи.
   Слабыми ключами называется такие ключи k что $DES_k(DES_k(x)) = x$, x — блок 64 бит.
   Частично-слабые ключи — пары ключей (k_1,k_2) такие что $DES_{k1}(DES_{k2}(x)) = x$.
3. Известны 4 слабых ключа

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	$[1]^{28}$	$[0]^{28}$

ПОВЫШЕНИЕ КРИПТОСТОЙКОСТИ DES

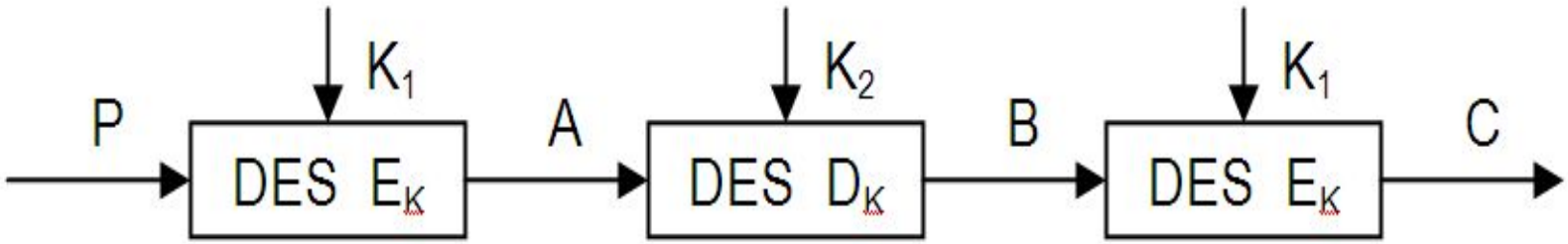
- 2DES и 3DES основаны на DES, но увеличивают длину ключей (2DES — 112 бит, 3DES — 168 бит).
- Схема 3DES имеет вид $DES(k_3,DES(k_2,DES(k_1,M)))$, где k_1,k_2,k_3 ключи для каждого шифра DES. Это вариант известен как в EEE так как три DES операции являются шифрованием. Существует 3 типа алгоритма 3DES:
  - DES-EEE3: Шифруется три раза с 3 разными ключами.
  - DES-EDE3: 3DES операции шифровка-расшифровка-шифровка с 3 разными ключами.
  - DES-EEE2 и DES-EDE2: Как и предыдущие, для исключением того, который первая и третья операции используют одинаковый ключ.

Известные атаки на DES

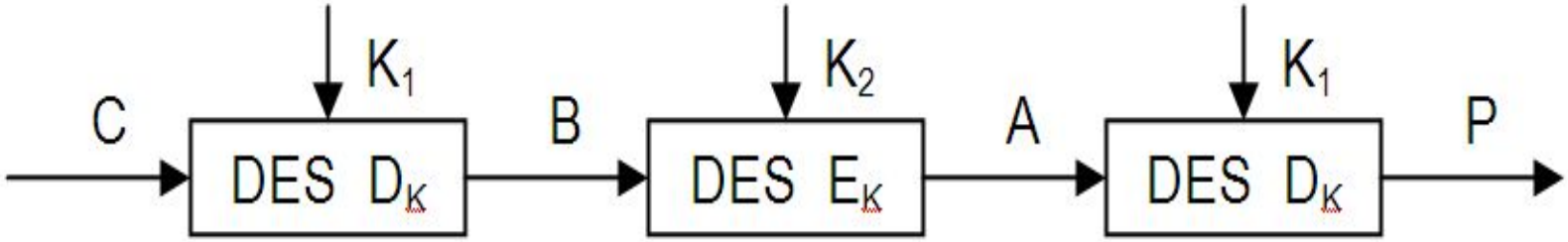
Методы атаки	Известные откр. тексты	Выбранные отк. тексты	Объём памяти	Количество операций
Полный поиск	1	-	Незначительный	2^{55}
Линейный Криптоанализ	$2^{43}(85\%)$	-	Для текста	2^{43}
Линейный Криптоанализ	$2^{38}(10\%)$	-	Для текста	2^{50}
Диффер. Криптоанализ	-	2^{47}	Для текста	2^{47}
Диффер. Криптоанализ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

Трехкратное применение блочного алгоритма с двумя ключами

$$C=E_{K1}(D_{K2}(E_{K1}(P))),$$
$$P=D_{K1}(E_{K2}(D_{K1}(C)))$$



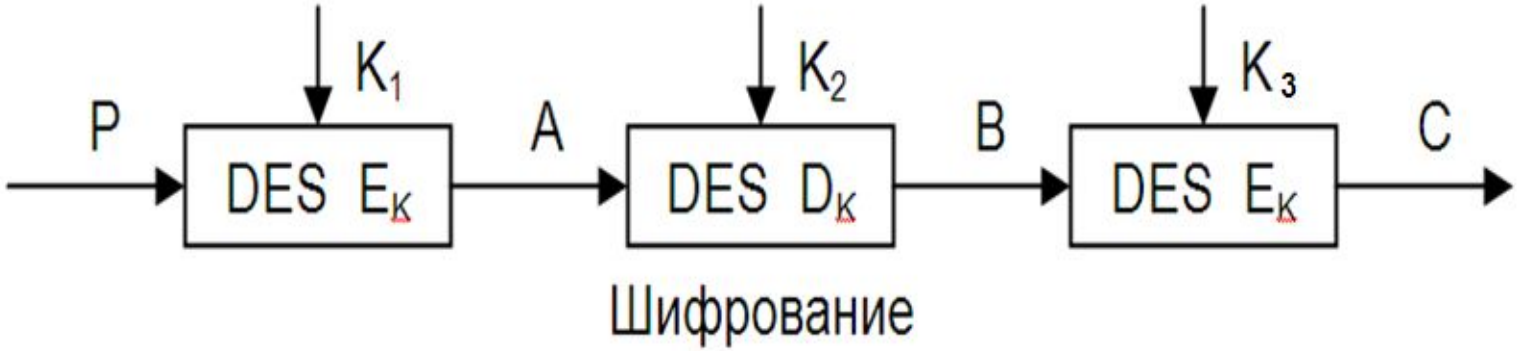
Шифрование



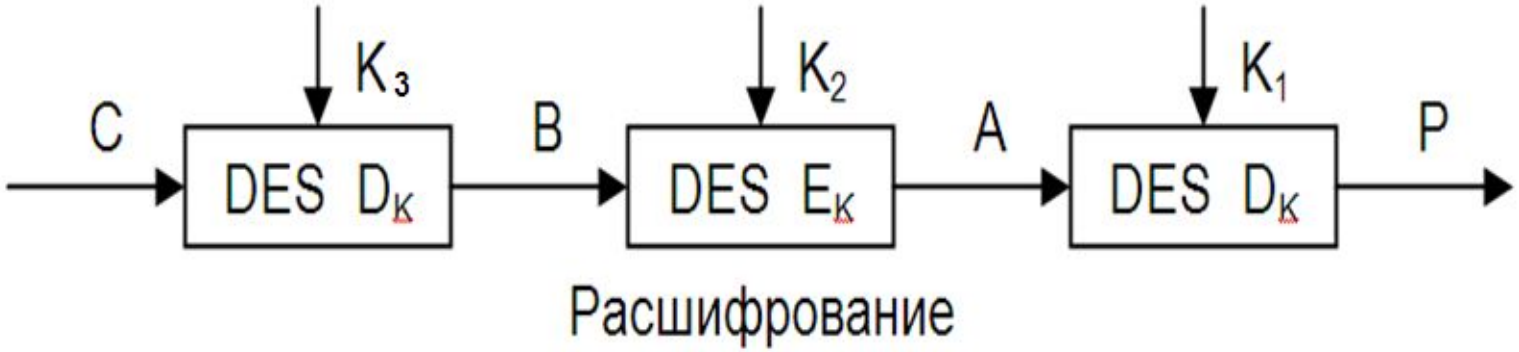
Расшифрование

Трехкратное применение блочного алгоритма с тремя ключами

$$C=E_{K3}(D_{K2}(E_{K1}(P))),$$
$$P=D_{K1}(E_{K2}(D_{K3}(C)))$$

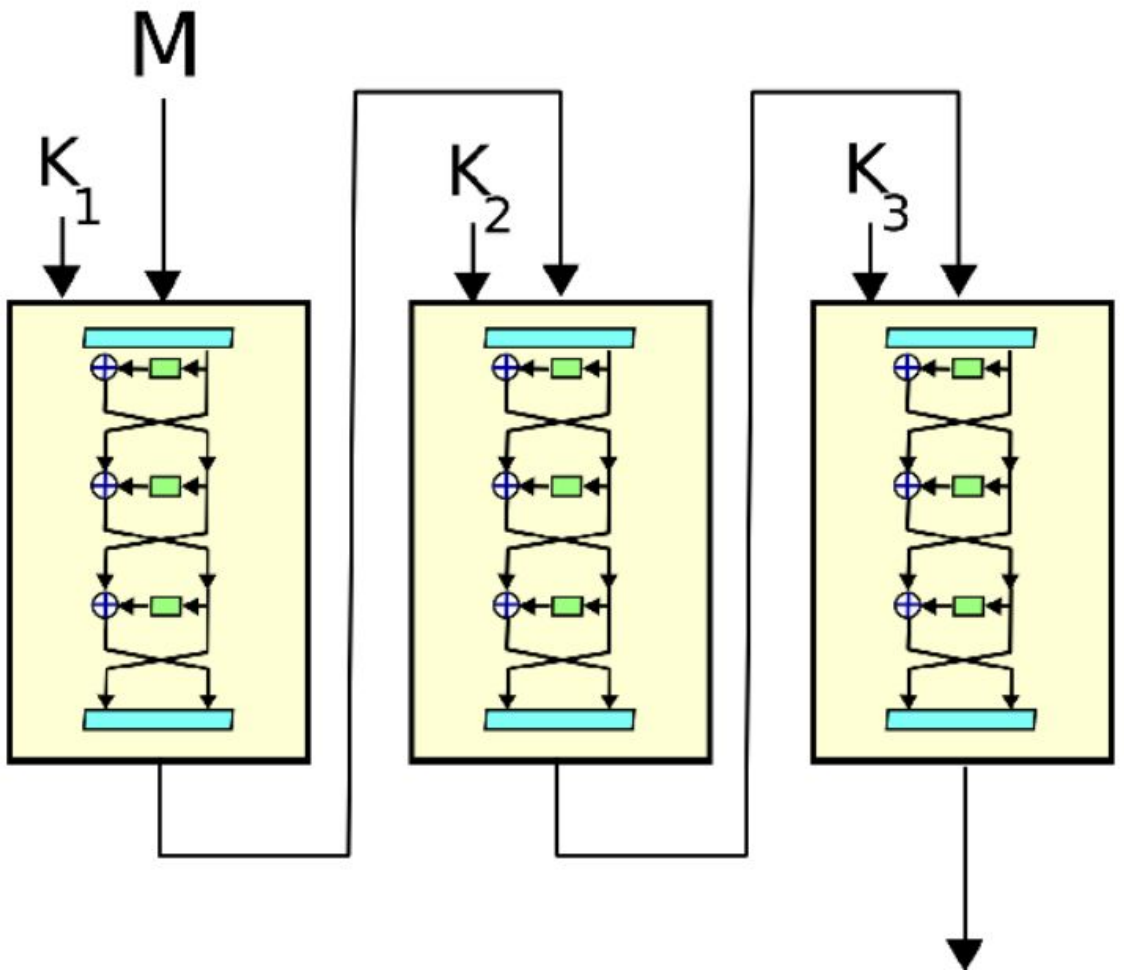


Шифрование



Расшифрование

Triple DES



- DES был национальным стандартом США в 1977—1980 гг., но в настоящее время DES используется (с ключом длины 56 бит) только для устаревших систем, чаще всего используют его более криптоустойчивый вид (3DES, DESX). 3DES является простой заменой DES, и сейчас он рассмотрен как стандарт. В ближайшее время DES и Triple DES будут заменены алгоритмом AES (Advanced Encryption Standard — Расширенный Стандарт Шифрования).

Симметричные алгоритмы

- DES (*Data Encryption Standard*) - стандарт шифрования данных в США до AES)
- 3DES (Triple-DES, тройной DES)
- IDEA (*International Data Encryption Algorithm*)
- ГОСТ 28147-89 — отечественный стандарт шифрования данных
- AES (*Advanced Encryption Standard*) - американский стандарт шифрования
- RC6 (Шифр Ривеста)
- Twofish
- SEED - корейский стандарт шифрования данных
- Camellia - сертифицированный для использовании в Японии шифр
- CAST (по инициалам разработчиков Carlisle Adams и Stafford Tavares)
- XTEA - наиболее простой в реализации алгоритм

ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ БЛОЧНЫХ ШИФРОВ

1. Сеть Фейстеля
2. Схема Лей-Месси
3. Подстановочно-перестановочная сеть (SP-сеть -Substitution-permutation network) ().

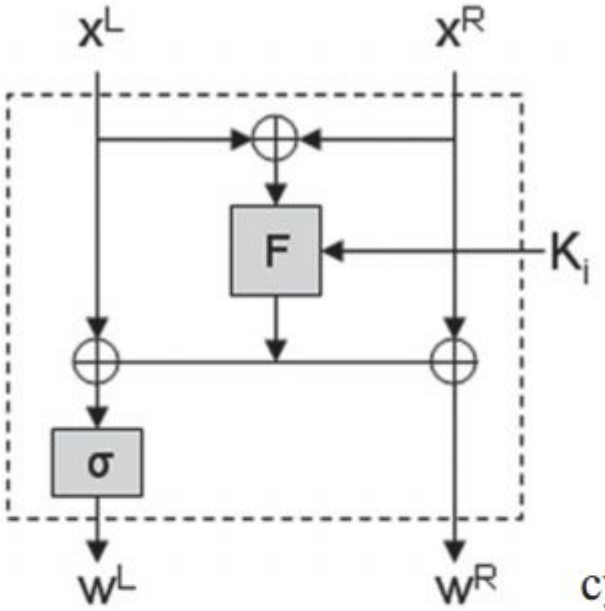
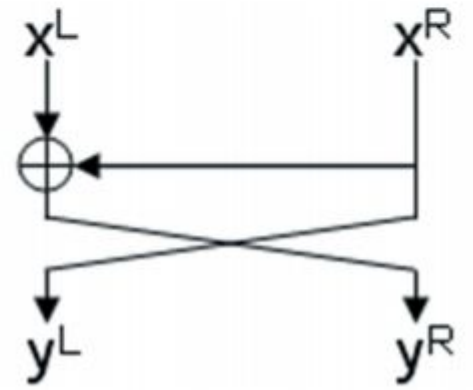


Схема Лей-Месси



Функция σ

$$L_i = \sigma(L_{i-1} \oplus F(L_{i-1} \oplus R_{i-1}, K_{i-1}))\,,$$

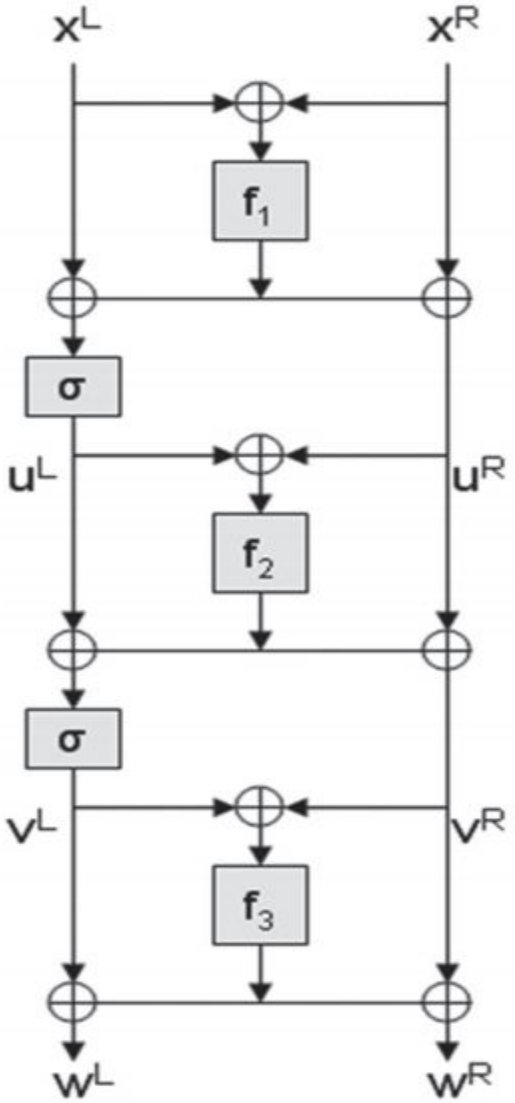
$$R_i = R_{i-1} \oplus F(L_{i-1} \oplus R_{i-1}, K_{i-1})\,,\ \text{при}\ i \in 1...n-1\,.$$

На последнем раунде преобразование σ отсутствует:

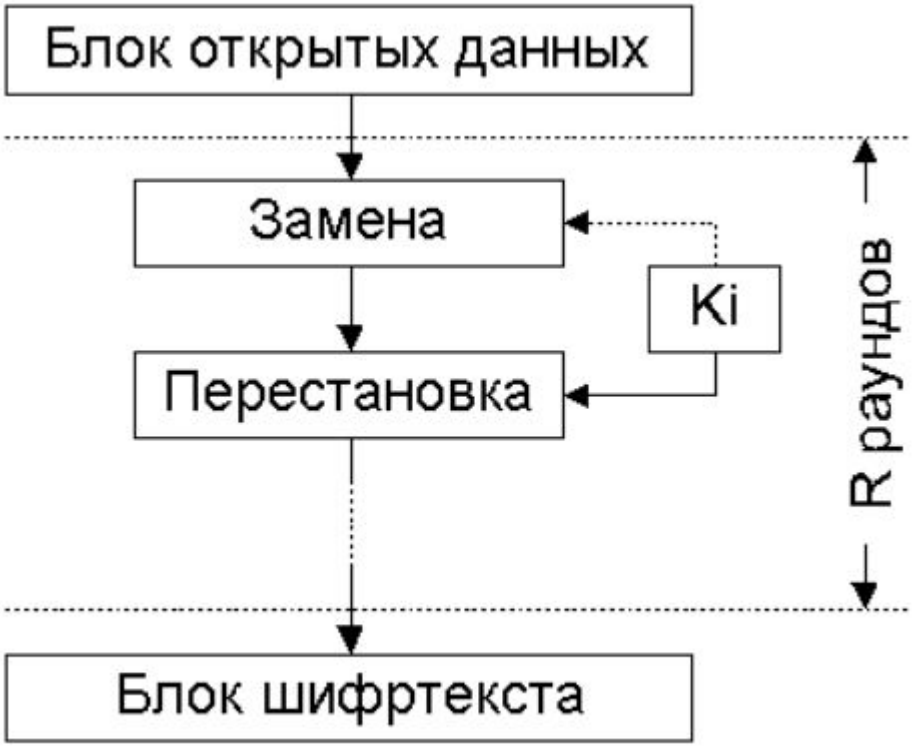
$$L_n = L_{n-1} \oplus F(L_{n-1} \oplus R_{n-1}, K_{n-1})\,,$$

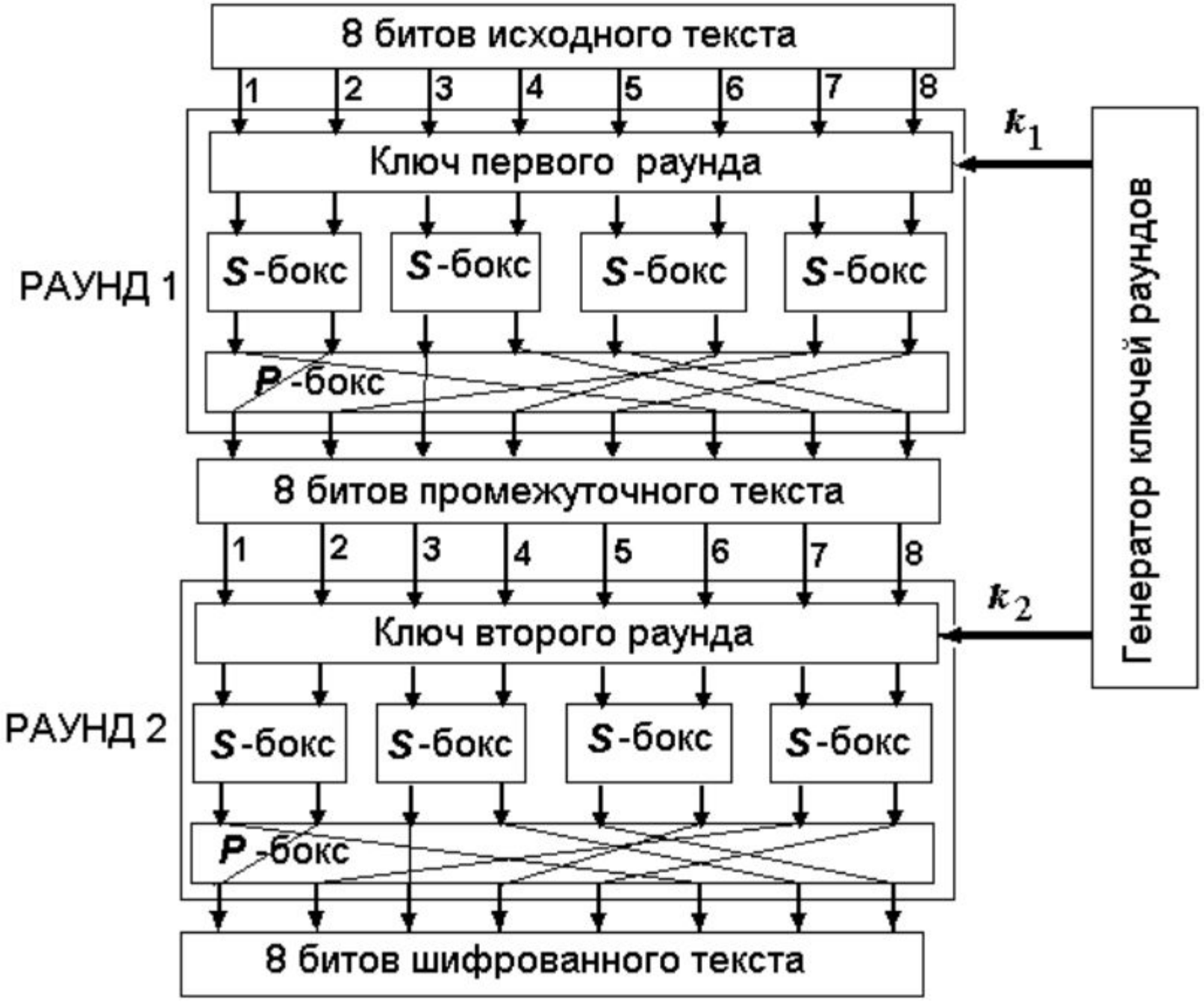
$$R_n = R_{n-1} \oplus F(L_{n-1} \oplus R_{n-1}, K_{n-1})\,.$$

3-х раундовая схема Лей-Месси



Подстановочно-перестановочная сеть (SP-сеть -Substitution-permutation network)




8 битов исходного текста
1
2
3
4
5
6
7
8
Ключ первого раунда
k_1
Генератор ключей раундов
РАУНД 1
S-бокс
S-бокс
S-бокс
S-бокс
P-бокс
8 битов промежуточного текста
1
2
3
4
5
6
7
8
Ключ второго раунда
k_2
РАУНД 2
S-бокс
S-бокс
S-бокс
S-бокс
P-бокс
8 битов шифрованного текста

ИСТОРИЯ AES

Advanced Encryption Standard (AES), известный как **Rijndael** — симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительством США по результатам конкурса AES.

Основные требования к новому стандарту:

- блочный шифр.
- длина блока, равная 128 битам.
- ключи длиной 128, 192 и 256 бит.

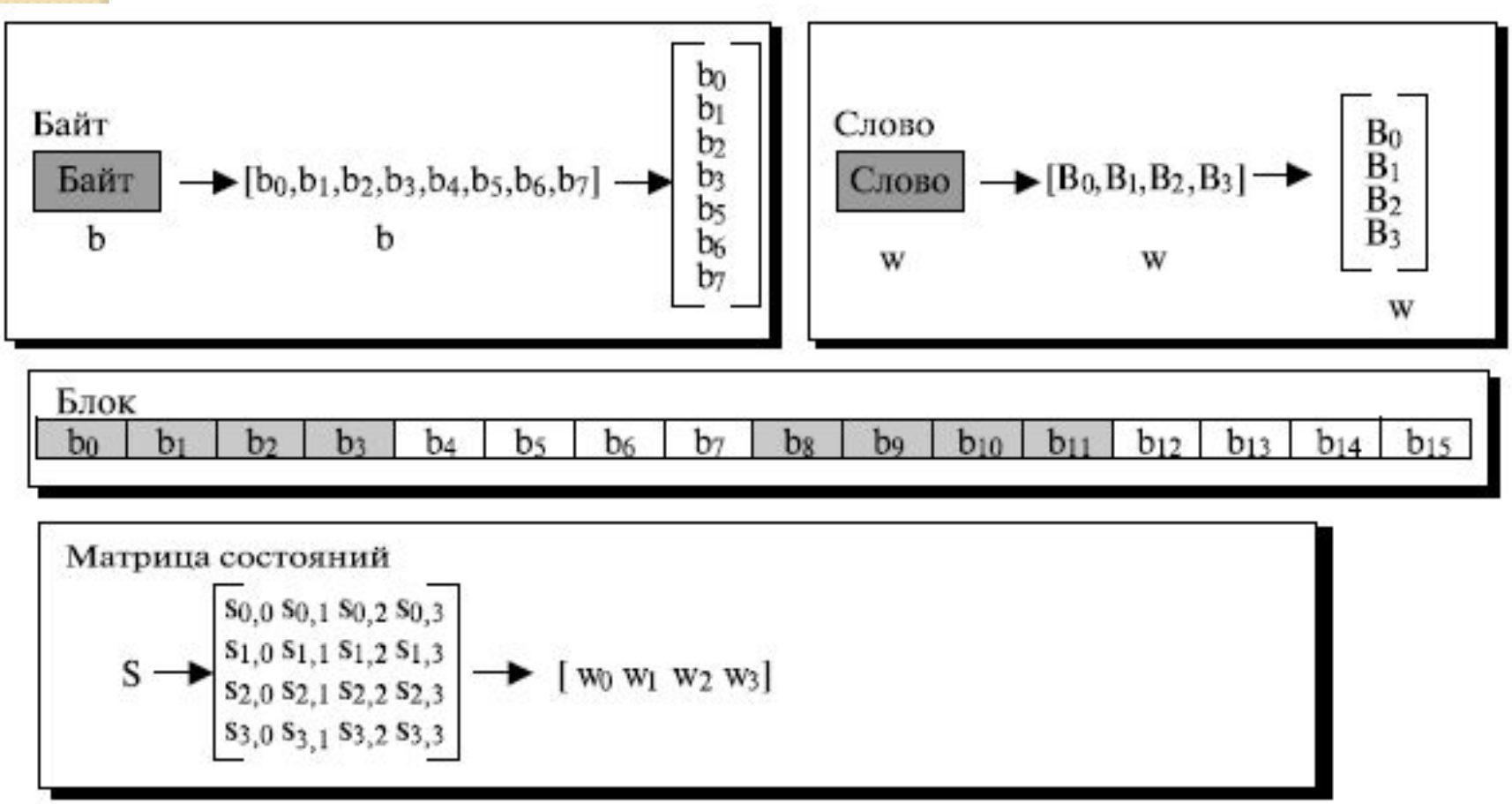
Йоан Даймен и Винсент Рэймен — создатели Rijndael

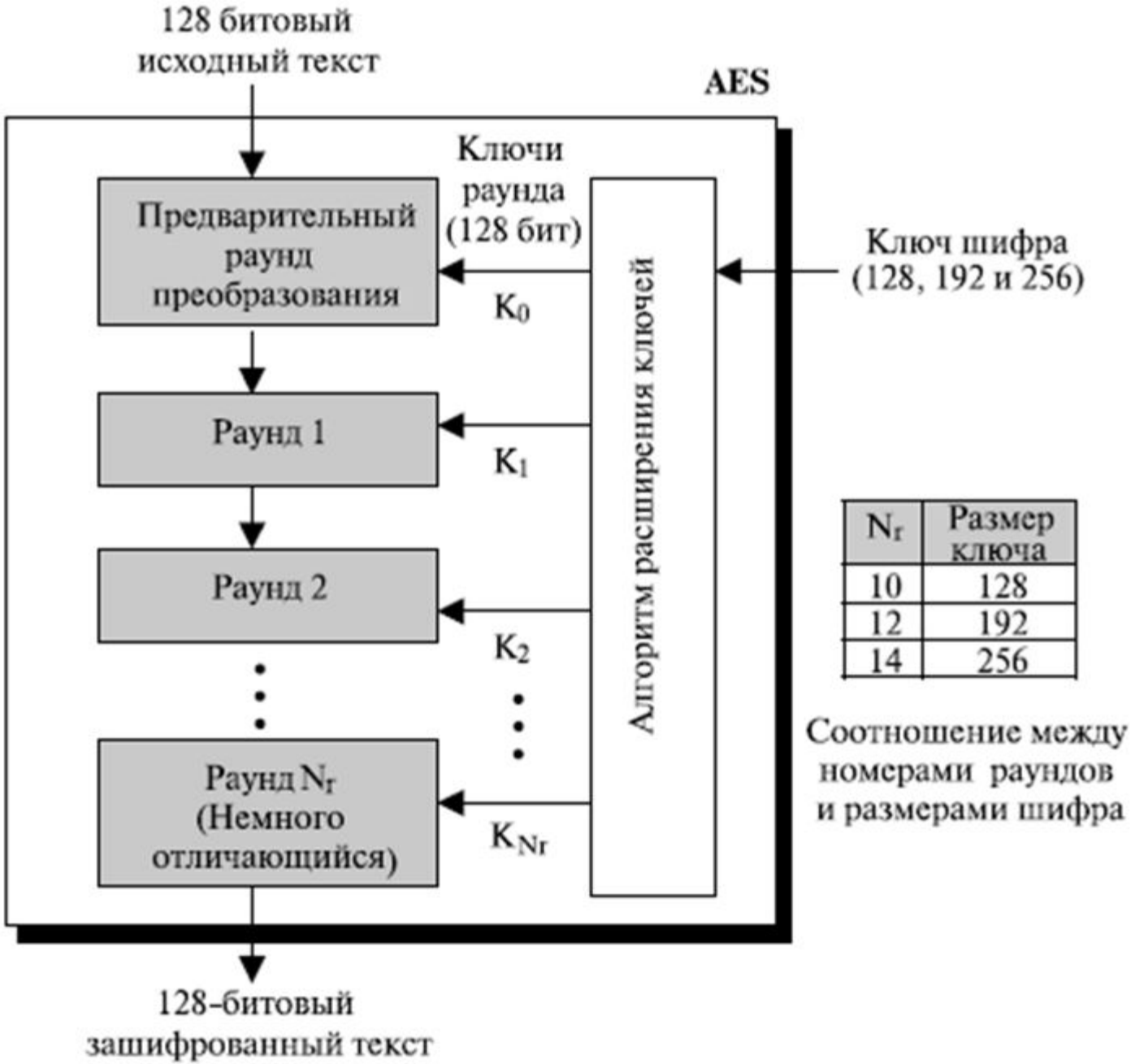
Дополнительные требования

1. Алгоритм должен быть стойким против криптоатак, известных на время проведения конкурса.
2. Структура алгоритма должна быть простой и обоснованной, что облегчит анализ алгоритма в рамках конкурса и дает некоторую гарантию отсутствия в алгоритме специально внедренных авторами «закладок», которые могли бы быть использованы авторами для его вскрытия.
3. Должны отсутствовать слабые и эквивалентные ключи (т.е. ключи, являющиеся различными, но приводящие к одному и тому же результату шифрования).
4. Скорость шифрования данных должна быть высокой на всех потенциальных аппаратных платформах – от 8-битных до 64-битных.
5. Структура алгоритма должна позволять распараллеливание операций в многопроцессорных системах и аппаратных реализациях.
6. Алгоритм должен предъявлять минимальные требования к оперативной и энергонезависимой памяти.
7. Не должно быть ограничений для использования алгоритма; в частности, алгоритм не должен ограничивать свое использование в различных стандартных режимах работы, в качестве основы для построения хэш-функций, генераторов псевдослучайных последовательностей.

Слово — группа из 32 битов, которая может быть обработана как единый объект. Это матрица из строки в четыре байта или столбец матрицы из четырех байтов.

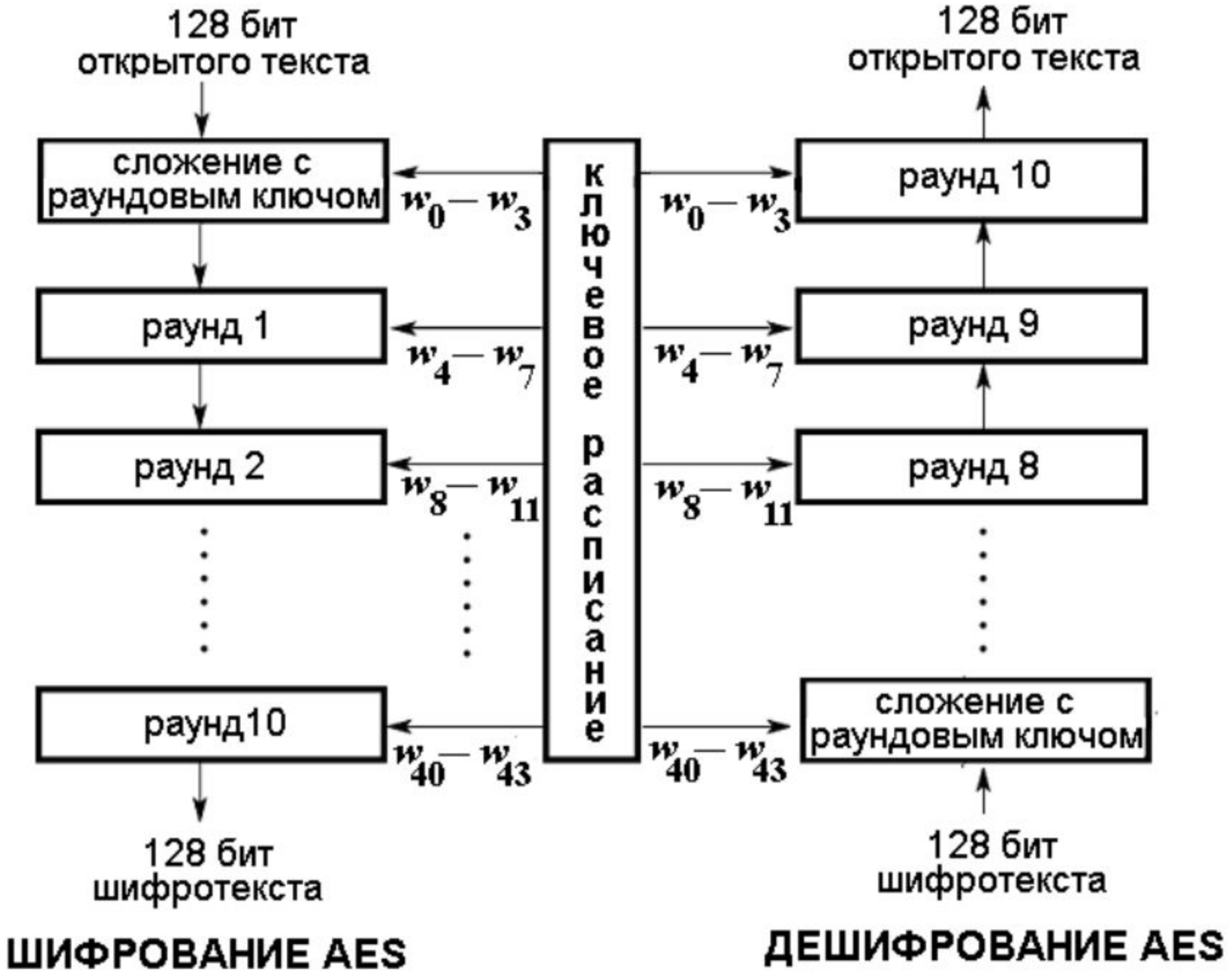
Матрица состояния (состояние) – промежуточный результат шифрования, который может быть представлен как прямоугольная матрица байт, имеющая 4 строки и *Nb* столбцов.





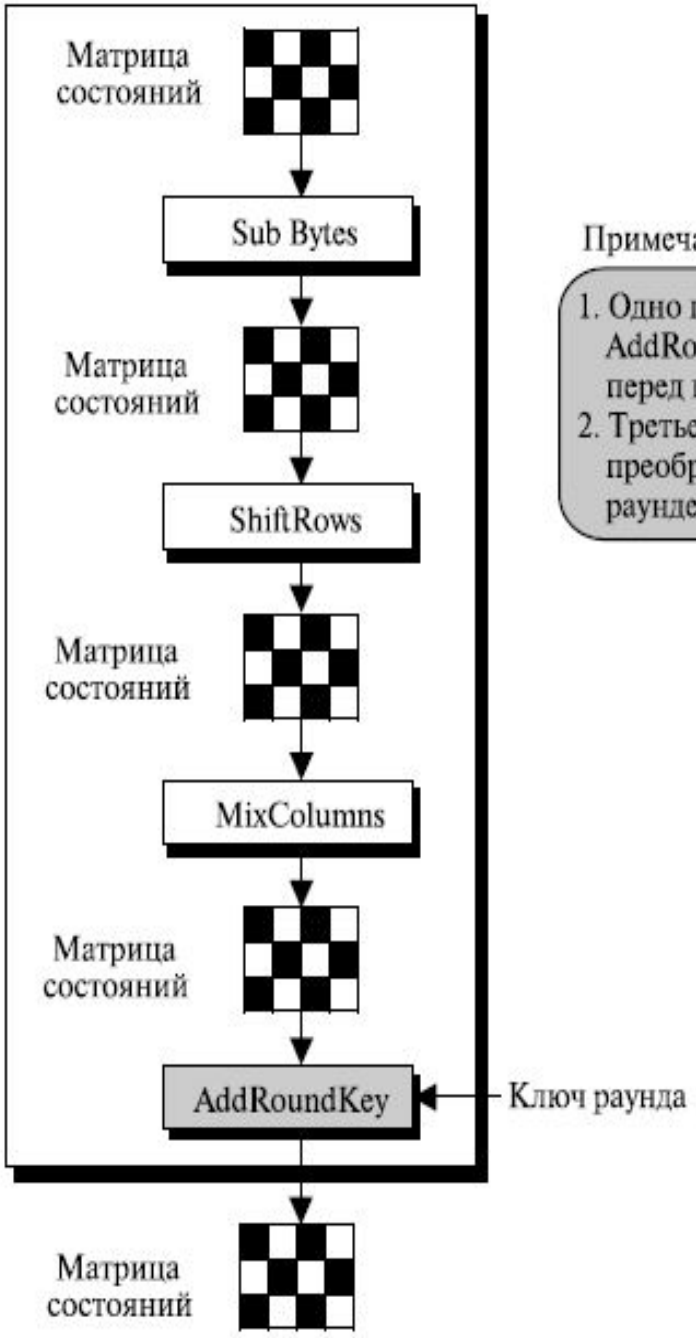
N_r	Размер ключа
10	128
12	192
14	256

Соотношение между номерами раундов и размерами шифра


128 бит
открытого текста
сложение с
раундовым ключом
раунд 1
раунд 2
раунд10
128 бит
шифротекста
ШИФРОВАНИЕ AES
$w_0 - w_3$
$w_4 - w_7$
$w_8 - w_{11}$
$w_{40} - w_{43}$
ключевое расписание
128 бит
открытого текста
раунд 10
раунд 9
раунд 8
сложение с
раундовым ключом
128 бит
шифротекста
ДЕШИФРОВАНИЕ AES

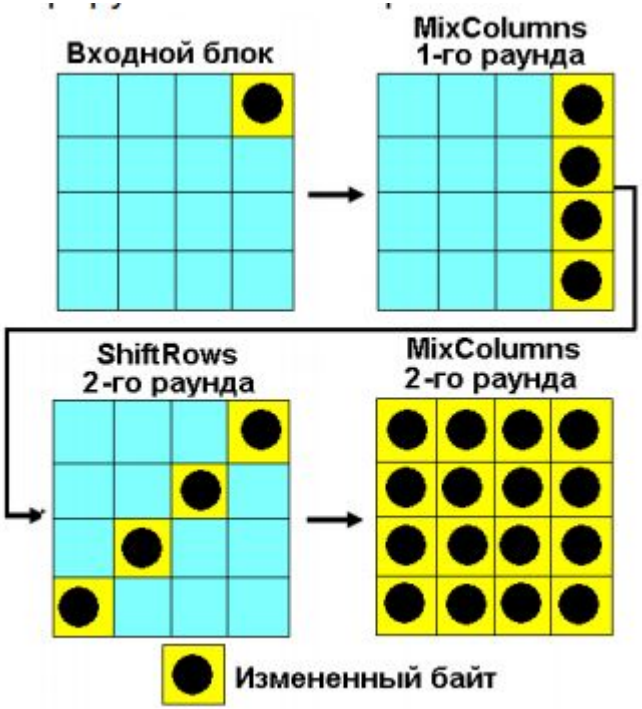
ЦИКЛ ШИФРОВАНИЕ AES СОСТОИТ ИЗ 4 ЭТАПОВ

1. **SubBytes()** трансформации при шифровании которые обрабатывают State используя нелинейную таблицу замещения байтов(S-box), применяя её независимо к каждому байту таблицы;
2. **ShiftRows()** трансформации при шифровании, которые обрабатывают таблицу, циклически смещая последние три строки на разные величины;
3. **MixColumns()** трансформация при шифровании которая берет все столбцы таблицы и смешивает их данные (независимо друг от друга), чтобы получить новые столбцы;
4. **AddRoundKey()** трансформация при шифровании и обратном шифровании, при которой к ключу раунда и таблице применяется операция XOR. Длина ключа раунда равна размеру таблицы (если число столбцов равно 4, то длина ключа равна 128 бит или 16 байт).



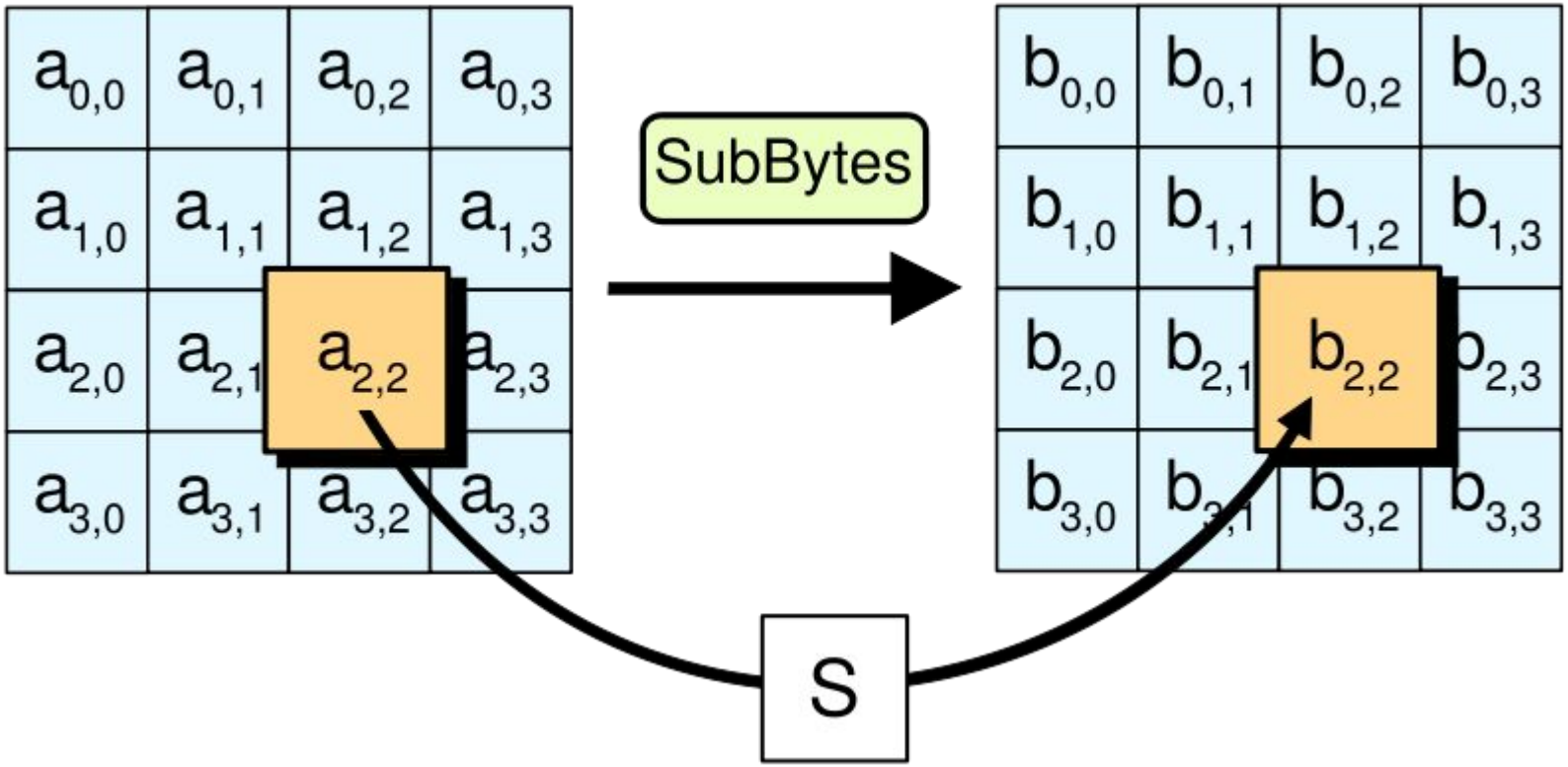
Примечание

1. Одно преобразование AddRoundKey применяется перед первым раундом
2. Третье MixColumn преобразование в последнем раунде отсутствует

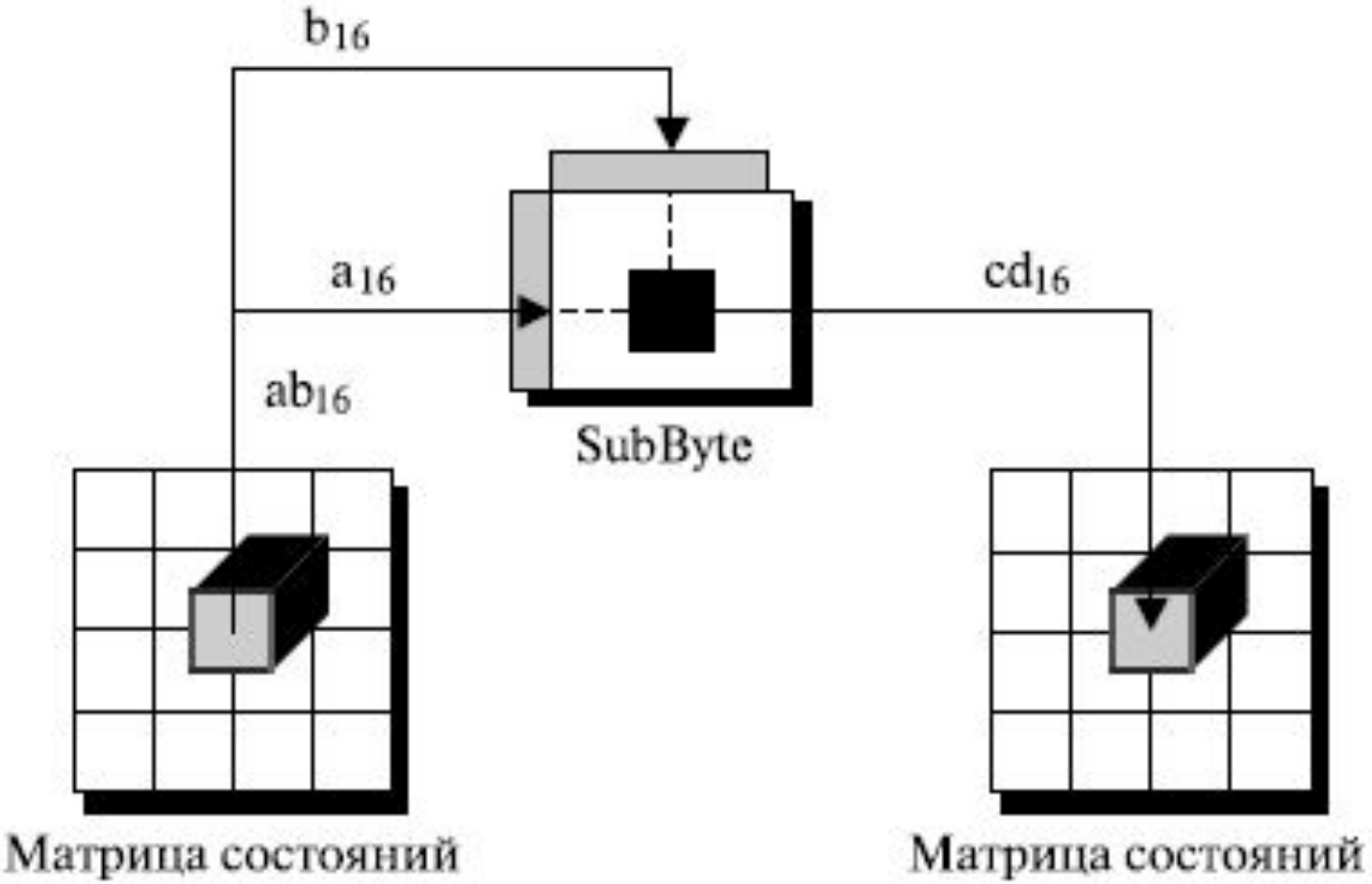


Алгоритм AES

1. **SubBytes()**



- Преобразование **SubBytes**



1. **Операция** SubBytes.

Операция выполняет нелинейную замену байтов, выполняемую независимо с каждым байтом матрицы *State*. Замена обратима и построена путем комбинации двух преобразований над входным байтом:

- нахождение обратного (инвертированного) элемента относительно умножения в поле $GF(2^8)$ (считается, что нулевой байт $\{00\}$ переходит сам в себя);
- выполнение некого аффинного преобразования: умножение инвертированного байта на многочлен $a(x) = x^4 + x^3 + x^2 + x + 1$ и суммирование с многочленом $b(x) = x^6 + x^5 + x + 1$ в поле $F_2[x]/x^8 + 1$.

Заметим, что $a^{-1}(x) = x^6 + x^3 + x$ и $a^{-1}(x)b(x) = x^2 + 1$

В матричной форме процедура SubBytes записывается как

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}^{-1} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix},$$

Например, байты $5A_{16}$ и $5B_{16}$ преобразовываются в BE_{16} и 39_{16},

Таблица преобразования SubBytes

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	FO	AD	D4	A2	AF	9C	A4	72	CO
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	AO	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	DO	EF	AA	FB	43	4D	33	85	45	F9	02	F7	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DE
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	CB	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	IF	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	OE	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	OD	BF	E6	42	68	41	99	2D	OF	BO	54	BB	16

- Преобразование ***InvSubBytes***

Таблица преобразования InvSubBytes

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DE	6E
A	47	E1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	D	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7E	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	CB	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Построение S-box состоит из двух шагов.

1. Производится взятие обратного числа в поле Галуа $GF(2^8)$.
2. К каждому байту b из которых состоит S-box применяется следующая операция:

$$b'_i = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

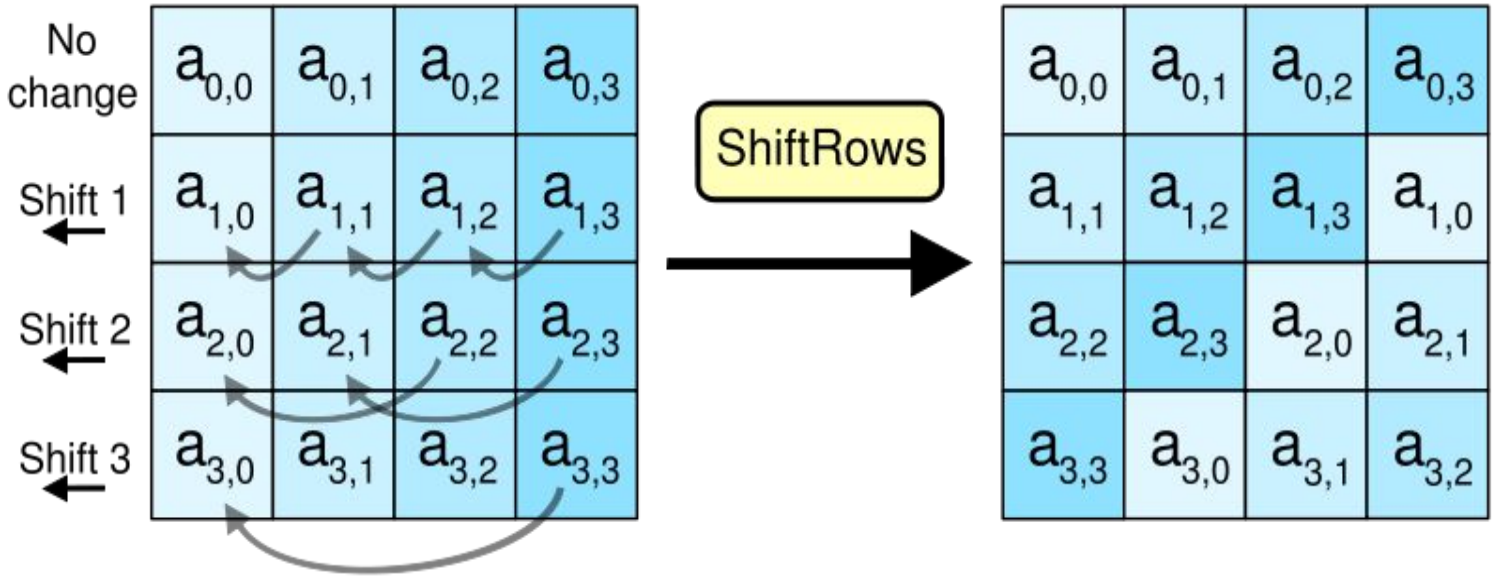
где $0 \leq i < 8$, и где b_i есть i-ый бит b, а c_i — i-ый бит константы $c = 63_{16} = 99_{10} = 01100011_2$. Таким образом, обеспечивается защита от атак, основанных на простых алгебраических свойствах.

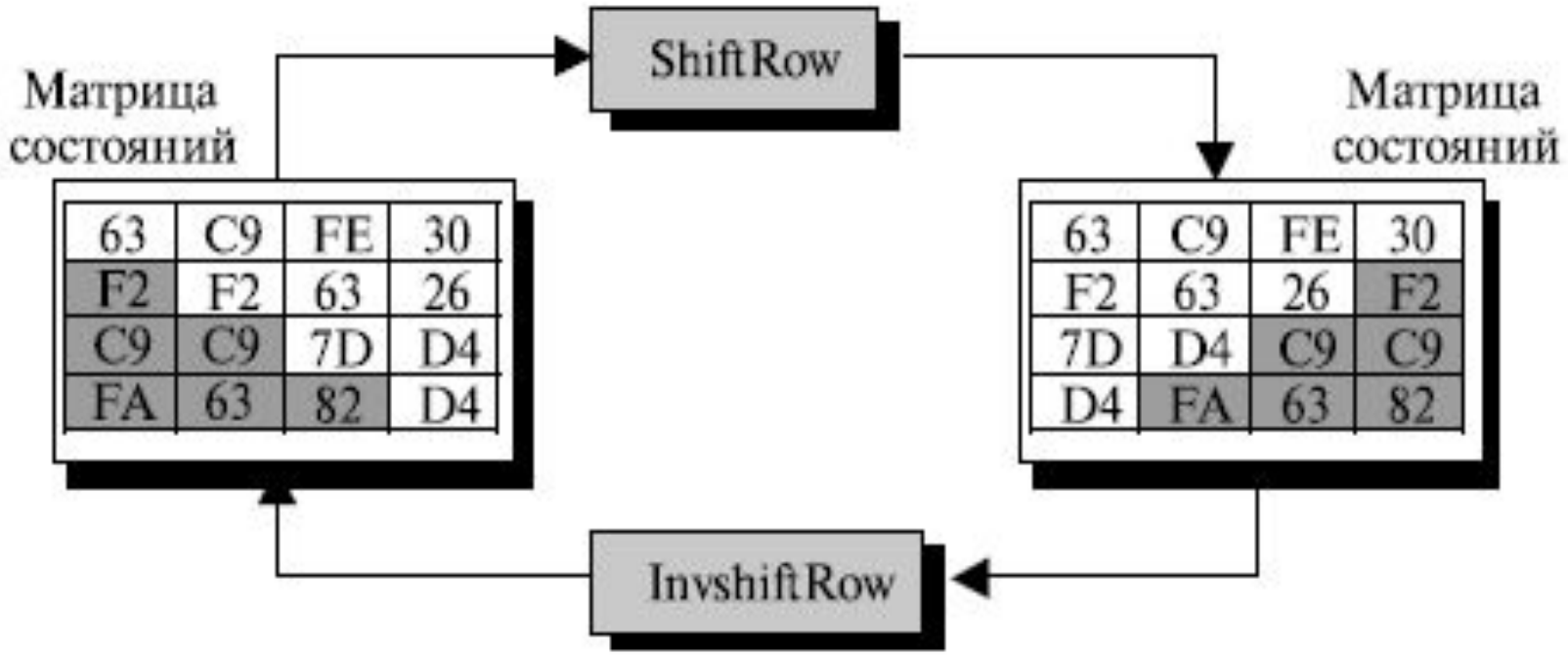
$$\begin{Vmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{Vmatrix} = \begin{Vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{Vmatrix} * \begin{Vmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{Vmatrix} + \begin{Vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{Vmatrix}$$

Алгоритм AES

2. В ShiftRow строки состояния циклически сдвигаются на различные значения. Нулевая строка не сдвигается. Строка 1 сдвигается на C_1 байтов, строка 2 на C_2 байтов, строка 3 на C_3 байтов. Величины C_1, C_2 и C_3 зависят от Nb − числа столбцов (32-х битных слов) .

Nb	C_1	C_2	C_3
4	***1***	***2***	***3***
6	**1**	**2**	**3**
8	**1**	**3**	**4**

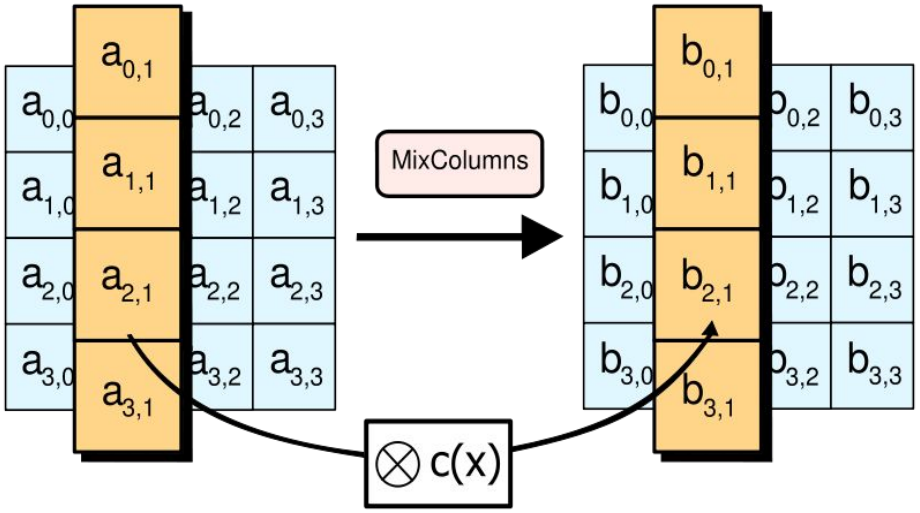




Пример преобразования ShiftRow

Алгоритм AES

3. *MixColumn*



3. **Операция** MixColumns.

С помощью этой операции выполняется перемешивание байтов в столбцах матрицы *State*. Каждый столбец этой матрицы принимается за многочлен над полем $GF(2^8)$ и умножается на фиксированный многочлен

$$c(x) = c_3x^3 + c_2x^2 + cx + c_0 = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

по модулю многочлена $x^4 + 1$ (напомним, все коэффициенты многочленов над полем $GF(2^8)$– байты). Как показано выше, такую операцию можно записать в матричном виде как

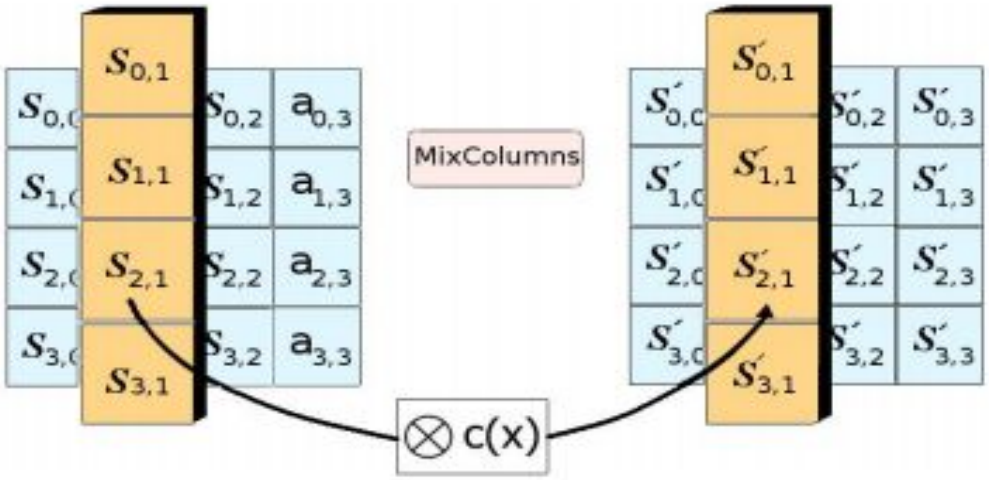
Операция ***MixColumns***. Выполняет умножение каждого столбца массива данных, который рассматривается как полином в конечном поле $GF(2^8)$, на фиксированный полином **a(x)**:

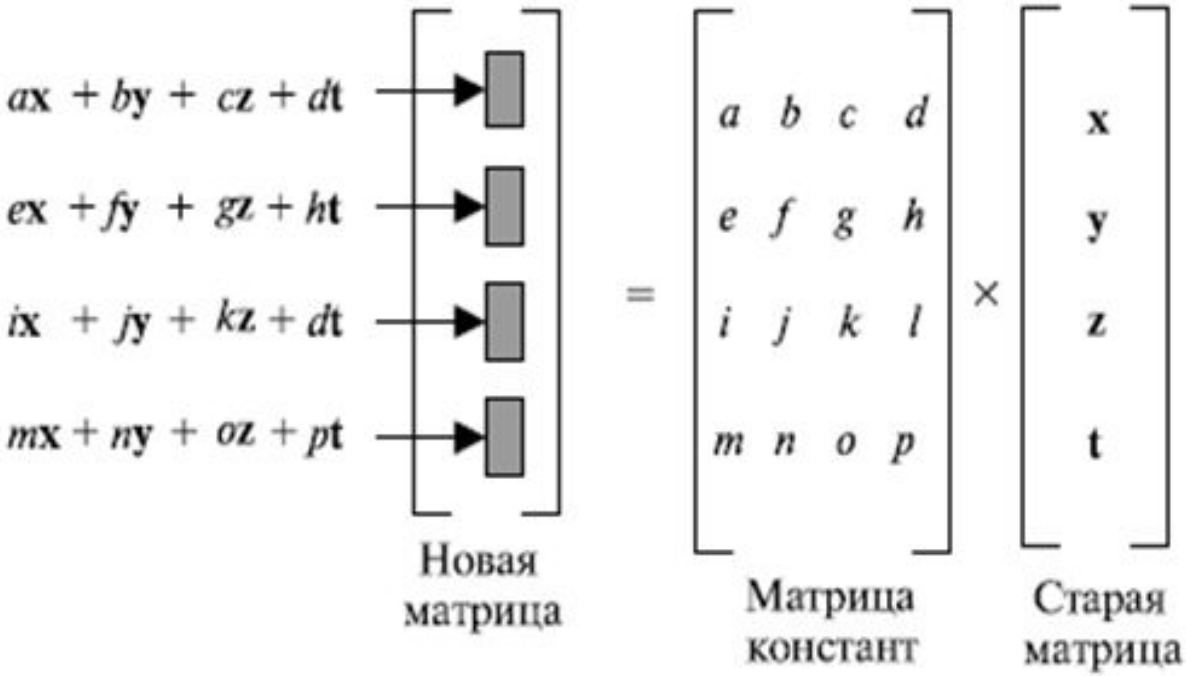
$$a(x) = 3x^3 + x^2 + x + 2.$$

Умножение выполняется по модулю

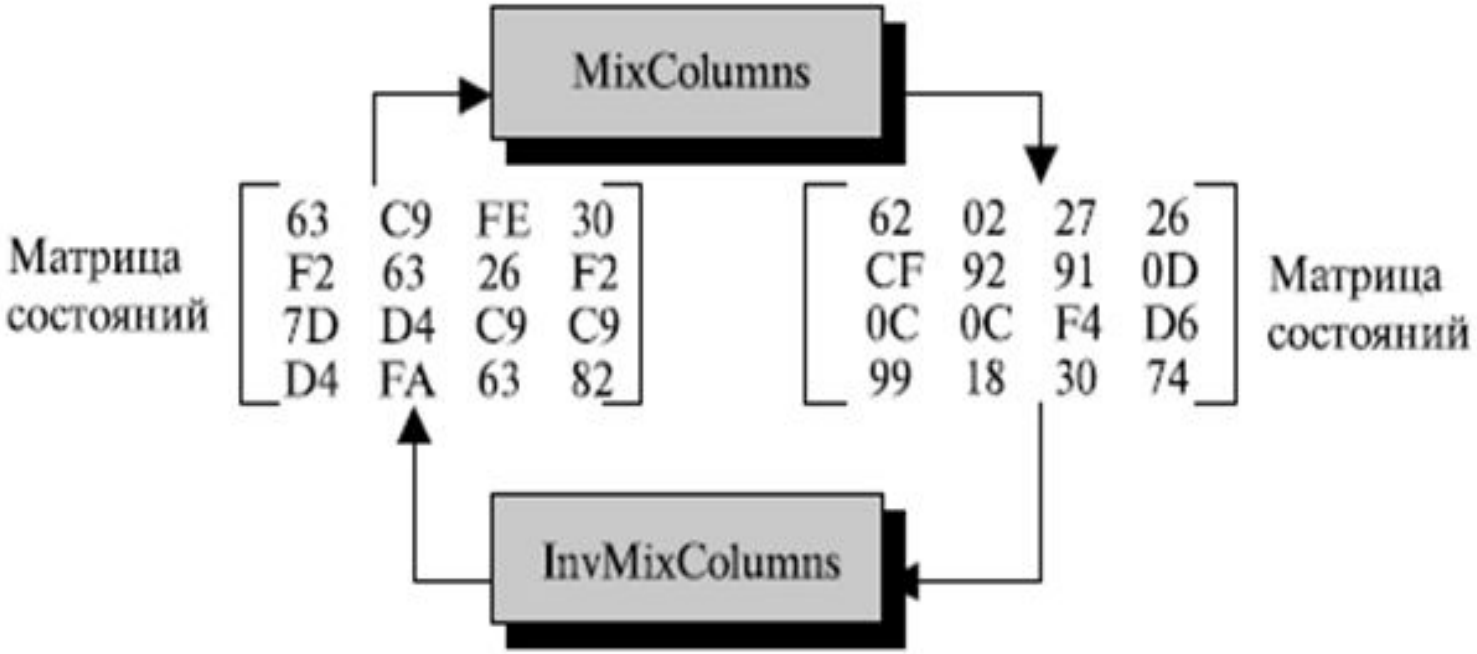
$$x^4 + 1.$$

$$\begin{pmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} s_0' \\ s_1' \\ s_2' \\ s_3' \end{pmatrix}.$$





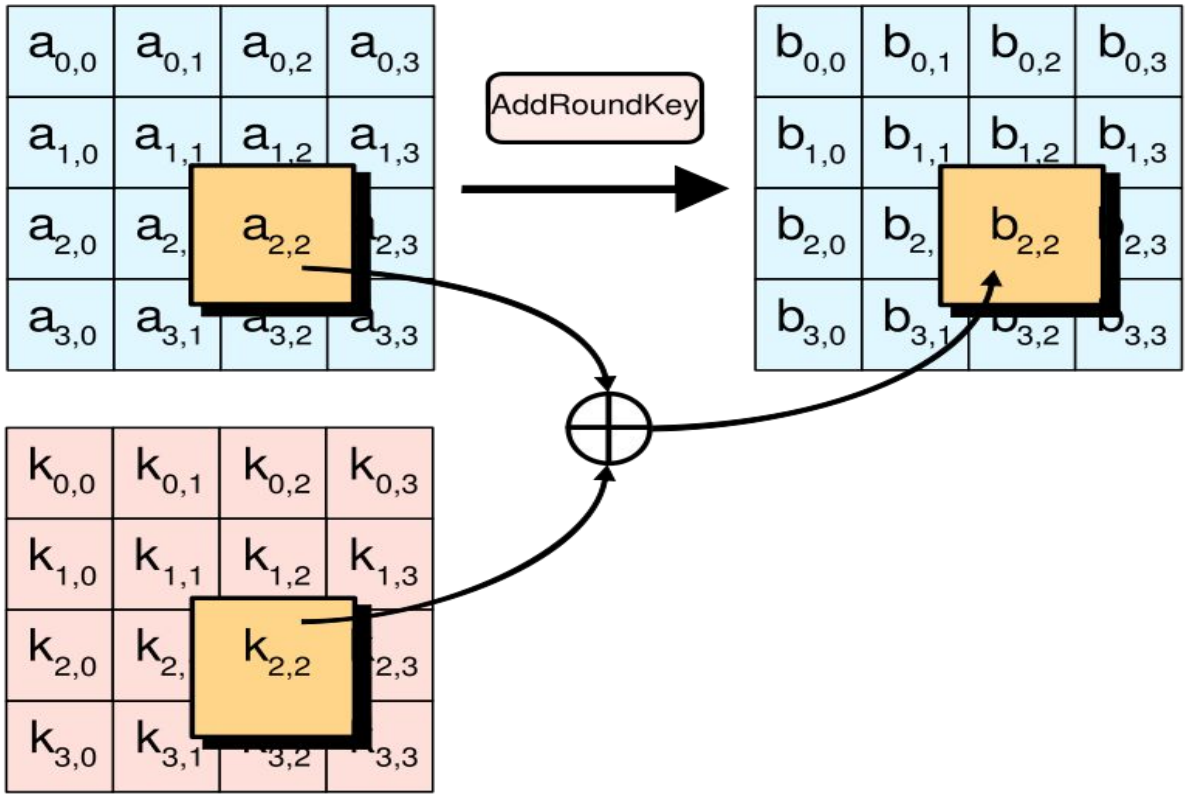
Смешивание байтов с использованием смешивающей матрицы



Преобразование MixColumns

Алгоритм AES

4. AddRoundKey

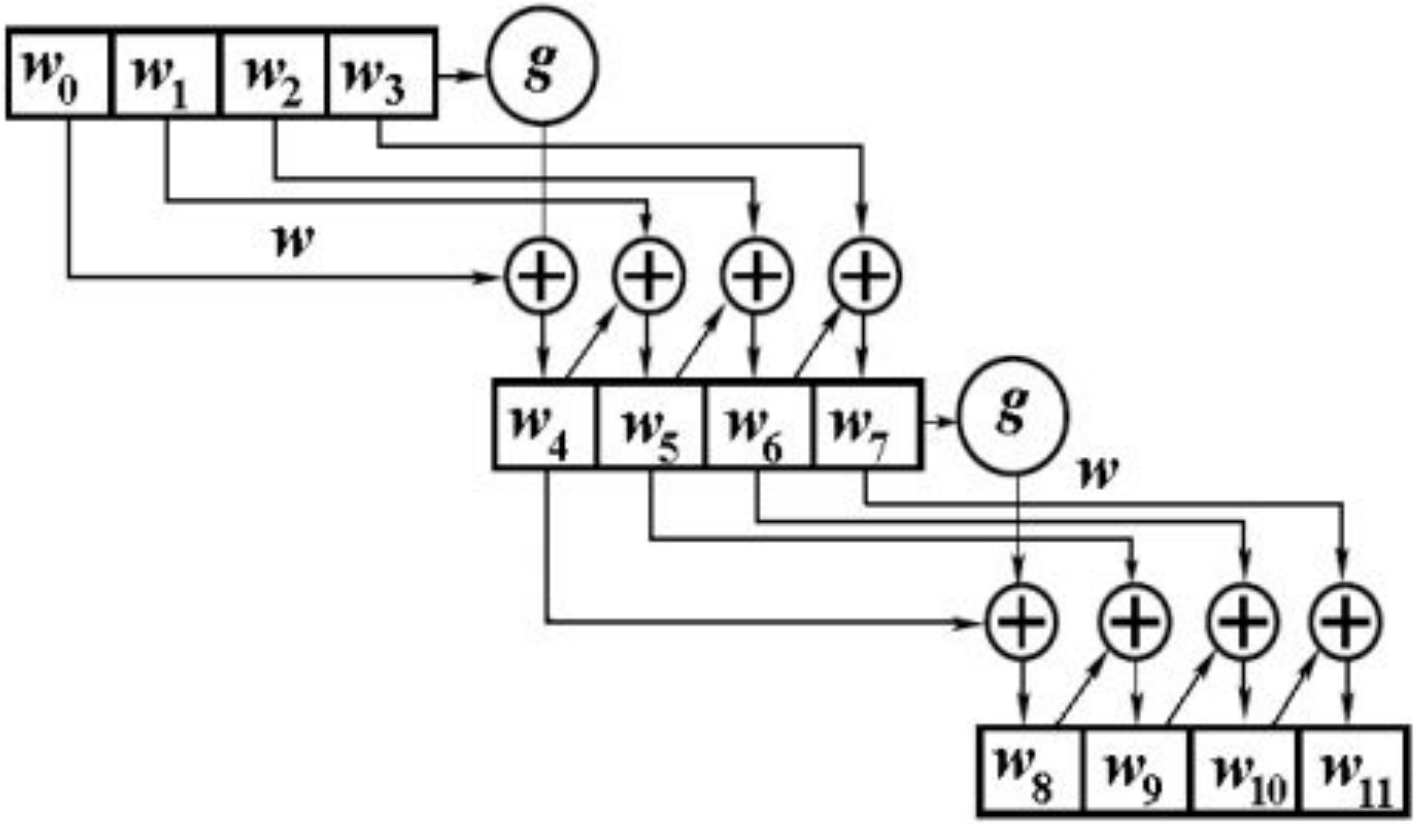


AES использует ключи шифрования трех фиксированных размеров: 128, 192, и 256 бит. В зависимости от размера ключа, вариант алгоритма AES обозначается: AES-128, AES-192 и AES-256

Операция ***AddRoundKey*** выполняет наложение на массив данных материала ключа. А именно, на **i**-й столбец массива данных (**i** = 0…3) побитовой логической операцией «исключающее или» (XOR) накладывается определенное слово расширенного ключа $\mathbf{W}_{4r+i}$, где **r** – номер текущего раунда алгоритма, начиная с 1.

Количество раундов алгоритма зависит от размера ключа.

Размер ключа, бит	Количество раундов
128	10
192	12
256	14


w_0 w_1 w_2 w_3
g
w
w_4 w_5 w_6 w_7
g
w
w_8 w_9 w_{10} w_{11}

Формирование ключей AES (KeyExpansion())

Алгоритм обработки ключа состоит из двух процедур:

1. **Расширение ключа**
2. **Выбор раундового ключа (ключа итерации)**

1. AES алгоритм, используя процедуру *KeyExpansion()* и подавая в её вход секретный криптографический ключ (*Cipher Key*), получает ключи для всех раундов. Всего она получает $N_b*(N_r+1)$ слов: изначально для алгоритма требуется набор из N_b слов, и каждому из N_r раундов требуется N_b ключевых набора данных.

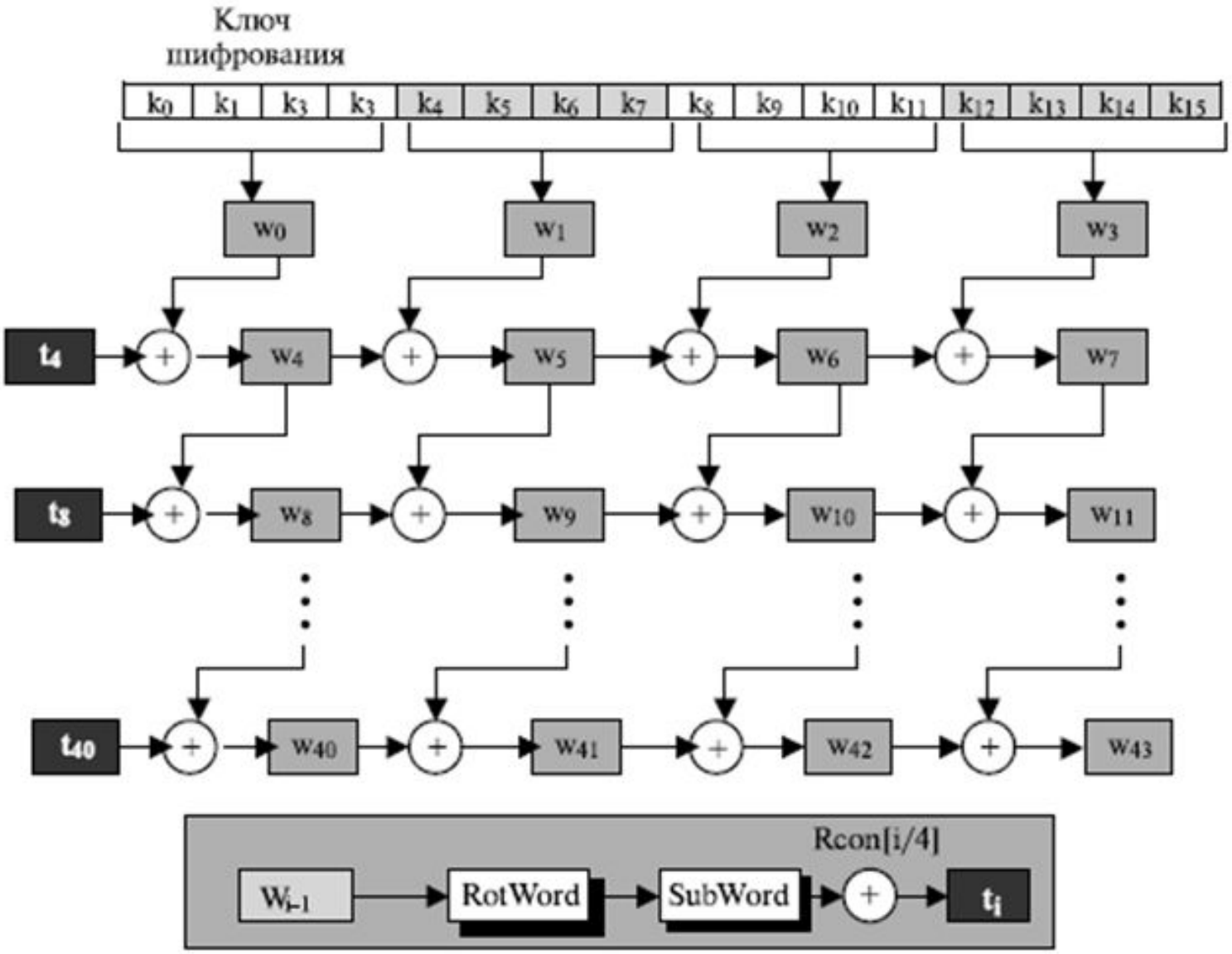
Количество раундов

N_k – число 32-битных слов, составляющих шифроключ.

N_b – число столбцов таблицы

N_r – число раундов

N_r	$N_b=4$	$N_b=6$	$N_b=8$
$N_k=4$	10	12	14
$N_k=6$	12	12	14
$N_k=8$	14	14	14



Вырабатываются t_i (временные) слова $i = 4N_r$

Расширение ключей в AES

ПРИМЕР. Начало зашифрования

Ключ шифра OF **15 71 C9 47 D9 E8 59 0C B7 AD DF AF 7F 67 98**

Раундовые ключи	Функция $g(w)$
w_0 =0F 15 71 C9	RotWord(w_3) = 7F 67 98 AF = x_1
w_1 =47 D9 E8 59	SubWord(x_1) = D2 85 46 79 = y_1
w_2 =0C B7 AD DF	R_{con} [1] = 01 00 00 00
w_3 =AF 7F 67 98	y_1 + R_{con} [1] = D3 85 46 79 = z_1
w_4 = w_4 + z_1 = DC 90 37 B0	RotWord(w_7) = 81 15 A7 38 = x_2
w_5 = w_4 + w_1 = 9B 49 DF E9	SubWord(x_4) = 0C 59 5C 07 = y_2
w_6 = w_5 + w_2 = 97 FE 72 3F	R_{con} [2] = 02 00 00 00
w_7 = w_6 + w_3 = 38 81 15 A7	y_2 + R_{con} [2] = 0E 59 5C 07 = z_2
w_8 = w_4 + z_2 = D2 C9 6B B7	RotWord(w_{11}) = FF D3 C6 E6 = x_3
w_9 = w_8 + w_5 = 49 80 B4 5E	SubWord(x_2) = 16 66 B4 8E = y_3
w_{10} = w_9 + w_6 = DE 7E C6 61	R_{con} [3] = 04 00 00 00
w_{11} = w_{10} + w_7 = E6 FF D3 C6	y_3 + R_{con} [3] = 12 66 B4 8E = z_3

1. Первые четыре слова (W_0, W_1, W_2, W_3) получены из ключа шифра. Ключ шифра представлен как массив из 16 байтов (k_0 до k_{15}). Первые четыре байта (k_0 до k_3) становятся W_0 ; следующие четыре байта (k_4 до k_7) становятся w_1 ; и так далее. Другими словами, последовательное соединение (конкатенация) слов в этой группе копирует ключ шифра.

2. Остальная часть слов (w_i) от $i = 4 - 43$ получается следующим образом:

a. Если $(i \bmod 4) \neq 0,\ w_i = w_{i-1} \oplus w_{i-4}$, это означает, что каждое слово получено из одного левого и одного верхнего.

b. Если $(i \bmod 4) = 0,\ w_i = t \oplus w_{i-4}$. Здесь t — временное слово, результат применения двух процессов, subword и rotword, со словом w_{i-1} и применения операции ИСКЛЮЧАЮЩЕЕ ИЛИ с константой раунда R_{con}. Другими словами, мы имеем

$$t = SubWord(Rotword\ (w_{i-1})) \oplus RCon_{i/4}.$$

RotWord

RotWord (rotate word) — процедура, подобная преобразованию ShiftRows, но применяется только к одной строке. Процедура принимает слово как массив из четырех байт и сдвигает каждый байт влево с конвертированием.

SubWord

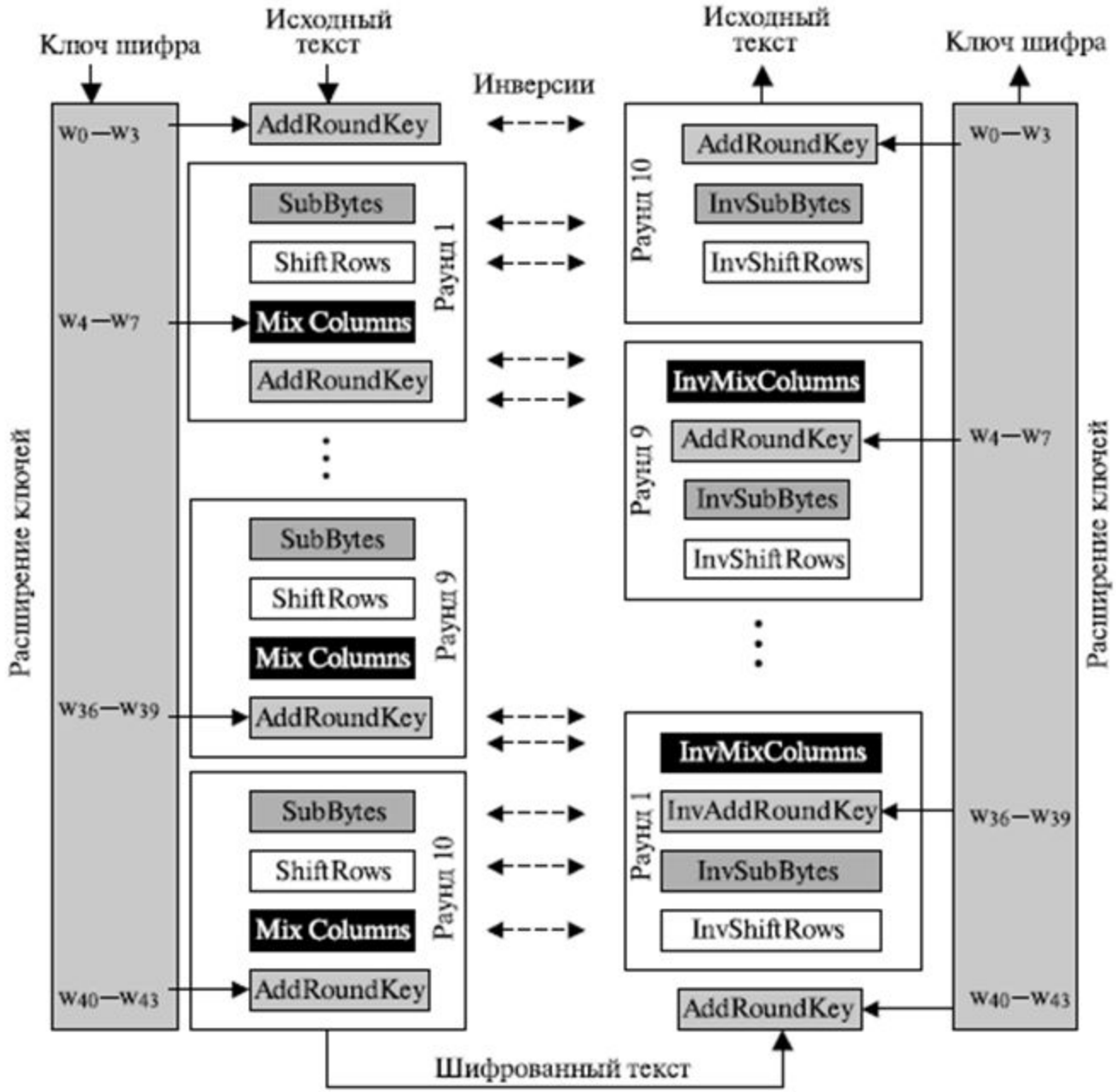
SubWord (substitute word) — процедура, подобная преобразованию SubBytes, но применяется только к одной строке. Процедура принимает каждый байт в слове и заменяет его другим.

RoundConstants

Каждая константа раунда R_{con} — это 4 -байтовое значение, в котором самые правые три байта являются всегда нулевыми.

Константы RCon

Раунд	Константа (RCon)	Раунд	Константа (RCon)
1	$(01\ 00\ 00\ 00)_{16}$	6	$(20\ 00\ 00\ 00)_{16}$
2	$(02\ 00\ 00\ 00)_{16}$	7	$(40\ 00\ 00\ 00)_{16}$
3	$(04\ 00\ 00\ 00)_{16}$	8	$(80\ 00\ 00\ 00)_{16}$
4	$(08\ 00\ 00\ 00)_{16}$	9	$(1B\ 00\ 00\ 00)_{16}$
5	$(10\ 00\ 00\ 00)_{16}$	10	$(36\ 00\ 00\ 00)_{16}$



Шифр и обратный шифр начального проекта

Расширение ключа в AES-192 и AES-256

Алгоритмы расширения ключей в *AES*-192 и *AES*-256 — очень похожи на алгоритм расширения ключа в *AES*-128

1. В *AES*-192 слова сгенерированы в группы по шесть вместо четырех.

a. Ключ шифра создает первые шесть слов (w_0 к w_5).

b. Если $i \bmod 6 \neq 0$, то $w_i \leftarrow w_{i-1} + w_{i-6}$; иначе $w_i \leftarrow t + w_{i-6}$.

1. В *AES*-256 слова сгенерированы в группы по восемь вместо четырех.

a. Ключ шифра создает первые восемь слов (w_0 до w_7).

b. Если $i \mod 8 \neq 0$, то $w_i \leftarrow w_{i-1} + w_{i-8}$; иначе, w_i <-t + w_{i-8}.

c. Если i mod 4=0, но $i \mod 8 \neq 0$, то w_i = Subword (w_{i-1}) + w_{i-8}.

ВАРИАНТЫ АЛГОРИТМА AES

На базе алгоритма *Rijndael*, лежащего в основе AES, реализованы альтернативные криптоалгоритмы.
Среди наиболее известных — участники конкурса Nessie:

- Anubis, автором которого является Винсент Рэймен
- усиленный вариант шифра — *Grand Cru* Йохана Борста.

Рабочие режимы

- электронная кодовая книга ECB (Electronic Code Book)
- сцепление блоков шифра CBC (Cipher Block Chaining)
- обратная связь по шифртексту CFB (Cipher Feed Back)
- обратная связь по выходу OFB (Output Feed Back)

применимы к любому симметричному алгоритму.

Достоинства симметричных систем

- скорость (по данным Брюса Шнайдера — на 3 порядка выше, чем ассиметричных);
- простота реализации (за счёт более простых операций);
- меньшая требуемая длина ключа для сопоставимой стойкости;
- изученность (за счёт больее длительного времени использования).

Недостатки симметричных систем

- **Сложность управления ключами в большой сети.** Означает квадратичное возрастание числа пар ключей, которые надо генерировать, передавать, хранить и уничтожать в сети. Для сети в 10 абонентов требуется 45 ключей, для 100 уже 4950, для 1000 — 499500 и т. д.
- **Сложность обмена ключами**. Для применения необходимо решить проблему надёжной передачи ключей каждому абоненту, так как нужен секретный канал для передачи каждого ключа обеим сторонам.
- **Невозможность** использования для подтверждения авторства, так как ключ известен каждой стороне.

Основні алгоритми симетричного блокового шифрування, що використовуються в Україні станом на 2015 р.

- ДСТУ ГОСТ 28147:2009
- AES (у складі операційних систем загального призначення)
- RC4 та ін. (іноземні реалізації засобів захисту Web-з'єднань відповідно до протоколів SSL/TLS)
- Triple DES (Національний банк України, іноземні реалізації засобів захисту мережевого трафіку IPsec)
- ДСТУ 7624:2014 ("Калина")

ГОСТ 28147-89

Алгоритм разработан в бывшем Главном Управлении КГБ СССР. Первоначально имел гриф секретности, затем гриф последовательно снижался и к моменту официального проведения алгоритма через Госстандарт СССР в 1989 году был снят. Алгоритм остался ДСП. В 1989 году стал официальным стандартом СССР, а позже федеральным стандартом Российской Федерации и основным шифром в Украине.

ГОСТ 28147-89 представляет собой блочный шифр с 256-битным ключом и 32 циклами преобразования, оперирующий 64-битными блоками. Основа алгоритма – сеть Фейштеля.

Алгоритм может работать в четырех режимах:

- простой замены;
- гаммирования;
- гаммирования с обратной связью;
- генерации имитовставок.

Основным является режим простой замены.

ДСТУ ГОСТ 28147:2009

Переваги

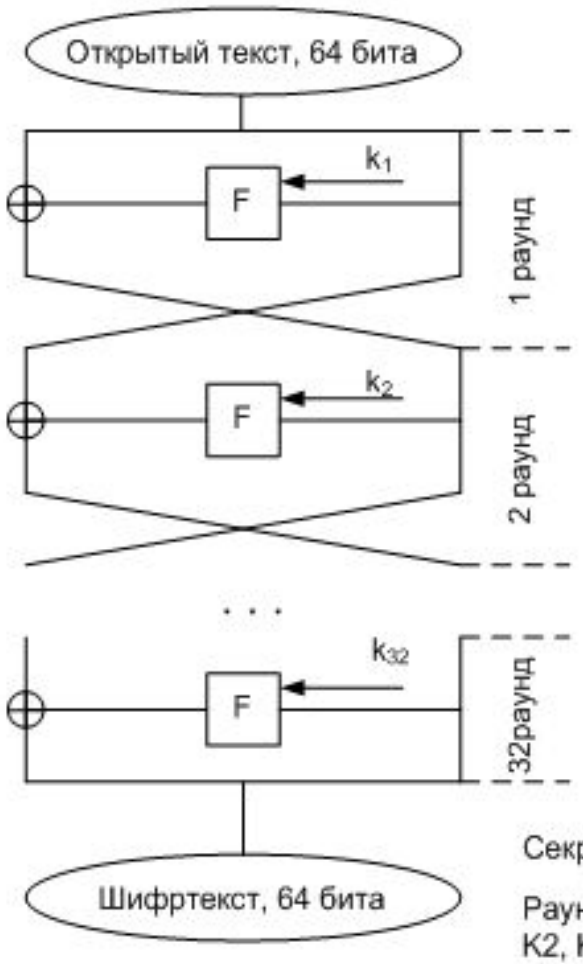
- відомий шифр, який добре досліджений міжнародною спільнотою більш ніж 20 років
- прийнятний рівень швидкодій (32-бітові платформи), достатньо зручний для апаратної реалізації, в т.ч. для малоресурсної (lightweight) криптографії
- вузли заміни (S-блоки) із гарними властивостями забезпечують практичну стійкість шифра

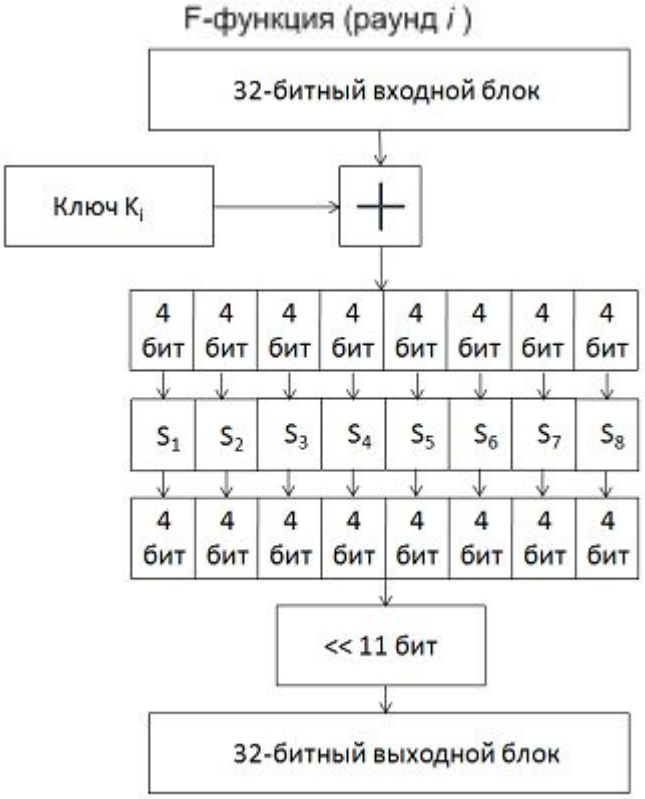
Недоліки

- наявність теоретичних атак із складністю, значно меншою повного перебора ключів
- великі класи слабких ключів
- використання вузлів заміни спеціального виду дозволяє зменшити рівень стійкості до реалізації практичних атак (виключно на основі шифртекстів) з використанням одного персонального комп'ютера
- швидкодія на сучасних системах суттєво нижча порівняно із іншими блоковими шифрами

ГОСТ 28147-89 виведений із дії у Білорусі (застосування дозволено тільки у режимі сумісності); у новому стандарті шифрування РФ використовується як додатковий 64-бітовий шифр; відмовлено у включенні цього шифру до міжнародного стандарту ISO/IEC 18033-3

При работе ГОСТ 28147-89 информация шифруется блоками по 64 бита, которые затем разбиваются на два субблока по 32 бита





Секретный ключ, 256 бит: Key={K1, K2, K3, K4, K5, K6, K7, K8}

Раундовый ключ, 32 бита: k_i={K1, K2, K3, K4, K5, K6, K7, K8, K1, K2, K3, K4, K5, K6, K7, K8, K1, K2, K3, K4, K5, K6, K7, K8, K8, K7, K6, K5, K4, K3, K2, K1}

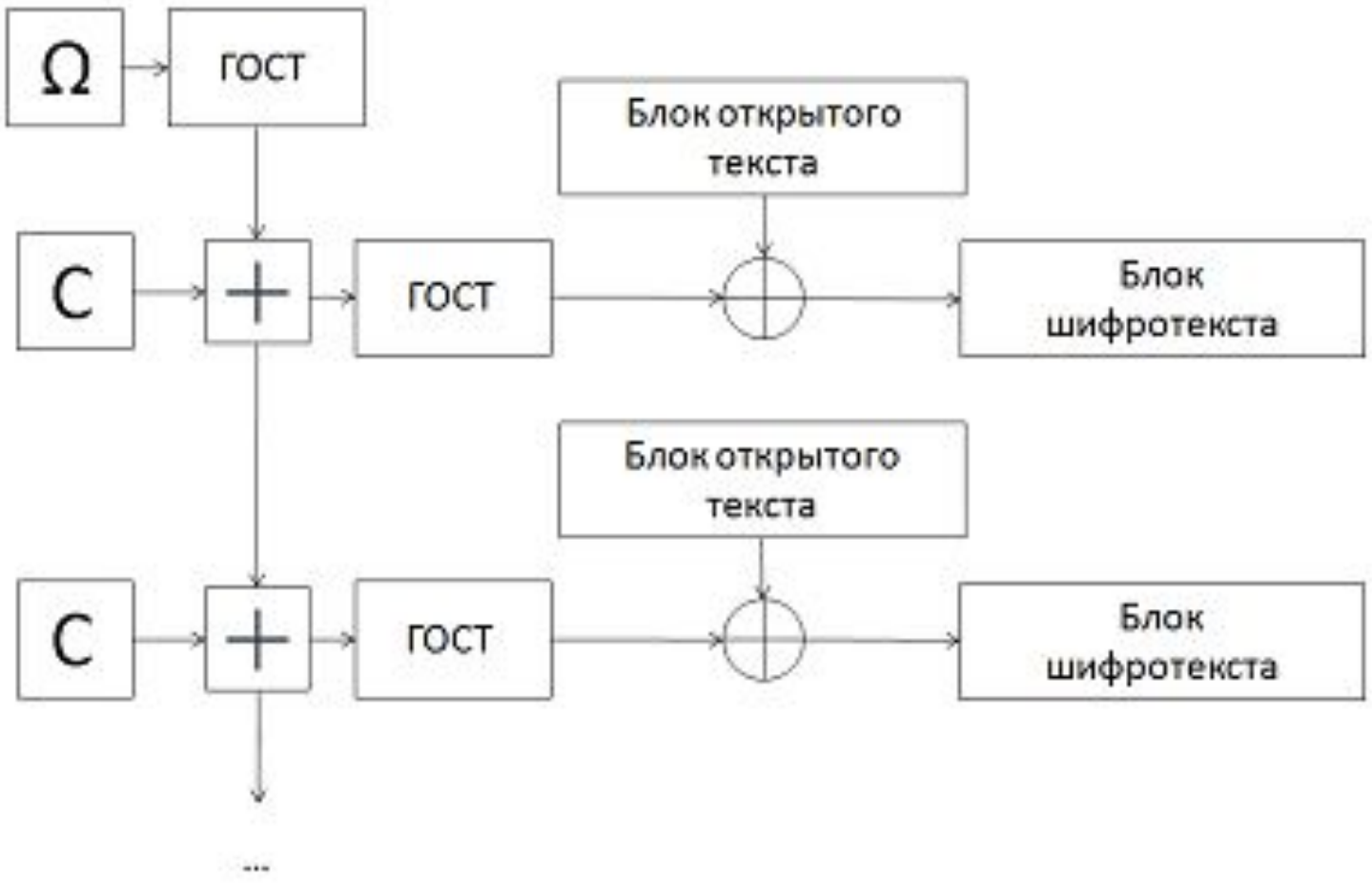
$$A_{i+1} = B_i \oplus f(A_i, K_i)$$ ($\oplus$ = двоичное «исключающее или»)

$$B_{i+1} = A_i$$

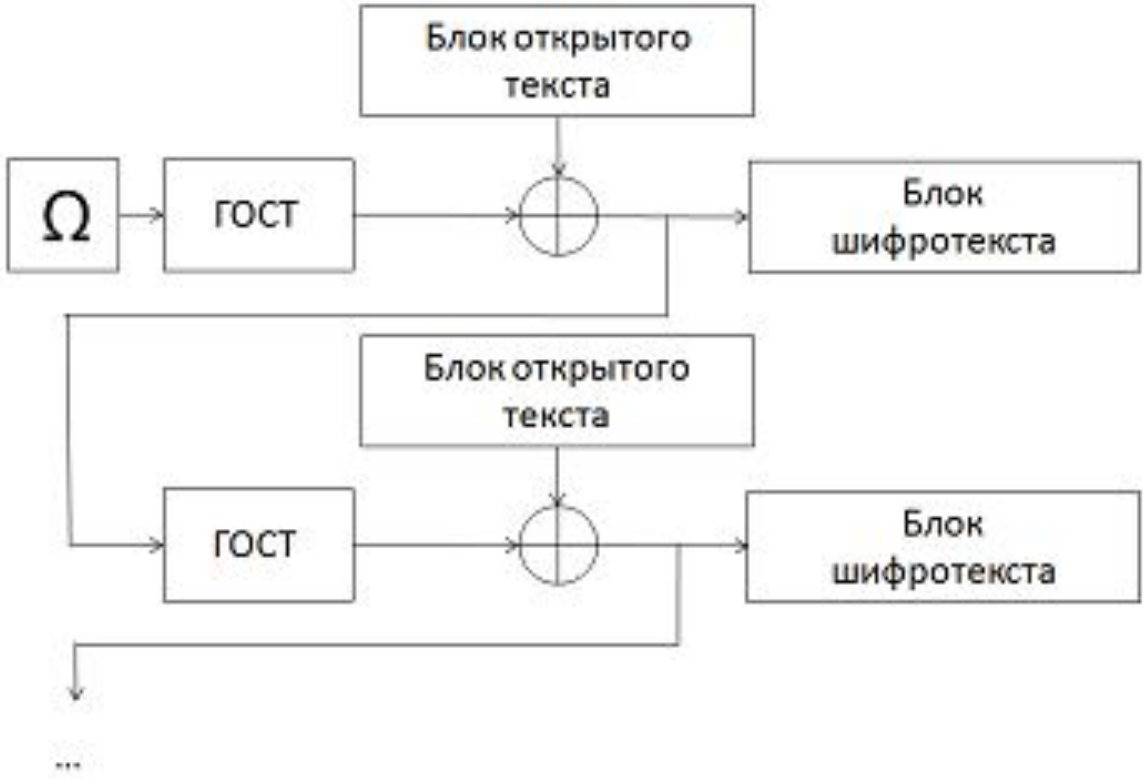
Блоки замены определенные документом RFC 4357

Номер S-блока	Значение															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	4	A	9	2	D	8	0	E	6	B	1	C	7	F	5	3
2	E	B	4	C	6	D	F	A	2	3	8	1	0	7	5	9
3	5	8	1	D	A	3	4	2	E	F	C	7	6	0	9	B
4	7	D	A	1	0	8	9	F	E	4	6	C	B	2	5	3
5	6	C	7	1	5	F	D	8	4	A	9	E	0	3	B	2
6	4	B	A	0	7	2	1	D	3	6	8	5	9	C	F	E
7	D	B	4	1	3	F	5	9	0	A	E	7	6	8	2	C
8	1	F	D	0	5	7	A	4	9	2	3	E	6	B	8	C

Режим гаммирования



Режим гаммирования с обратной связью



Режим выработки имитовставки

Advanced Encryption Standard (AES)

Переваги

- найбільш досліджений в світі (відкриті публікації) криптографічний алгоритм
- забезпечує високу практичну стійкість, включений до набору Suite-B АНБ США, дозволений для захисту ІзОД уряду США
- ефективний для реалізації на 32-бітових платформах
- наявність низки апаратних акселераторів (включаючи старші моделі процесорів загального призначення)

Недоліки

- відомі теоретичні атаки із складністю, меншою, ніж повний перебір
- не може в повній мірі використати можливості 64-бітових платформ
- деяка моральна застарілість (розроблений у 1997 р.): у конкурсі NIST SHA-3 перевага віддана рішенням із архітектурою, що значно відрізняється від AES
- відсутність довіри до іноземних апаратних реалізацій AES (в т.ч. набору інструкцій AES-NI процесорів x86 і x86_64) на основі даних Е.Сноудена
- світові лідери IT-індустрії почали поступово відмовляться від AES: наприклад, компанія Google у 2014 році впровадила на заміну алгоритм ChaCha20 для захисту каналів зв'язку мобільних пристроїв на базі операційної системи Android

Triple DES (3DES, TDEA)

Переваги

- відомий шифр, який добре досліджений міжнародною спільнотою більш ніж 30 років
- забезпечує припустиму практичну стійкість (2^{112})
- поширений у банківських системах, що імпортовані або орієнтовані на застарілі стандарти

Недоліки

- практична стійкість значно нижче теоретичної
- наявність класів слабких ключів
- швидкодія на сучасних системах суттєво нижча навіть порівняно із ДСТУ ГОСТ 2814:2009 і іншими блоковими шифрами

Заміни ГОСТ 28147-89 у країнах СНД

СТБ 34.101.31-2011 (Білорусь)

- блок 128 біт, ключ 128, 192 або 256 біт
- 8 циклів, які складаються з комбінації ланцюга Фейстеля та схеми Лей-Мессі
- один S-блок (8-біт-в-8) із гарними властивостями
- відсутність схеми розгортання ключів
- невідомі практичні атаки, ефективніші повного перебору
- швидший ніж ГОСТ 28147-89, але повільніший, ніж AES

“Кузнечик” (“Коник”, РФ)

- блок 128 біт, ключ 256 біт
- 9 циклів AES-подібного перетворення
- один S-блок (8-біт-в-8), нециркулянтна матриця лінійного перетворення: 16x16 над полем $GF(2^8)$
- схема розгортання ключів на базі циклового перетворення і ланцюга Фейстеля (конструкція CS-cipher)
- однаковий S-блок із новою функцією ґешування “Стрибог” (ГОСТ Р 34.11-2012), але різні матриці лінійного перетворення (ускладнена реалізація систем криптографічного захисту)
- великий розмір таблиць для оптимальної програмної реалізації
- швидкодія нижча за AES

Таблица 1. Сравнительные характеристики алгоритмов ГОСТ28147-89 и Rijndael.

Показатель	ГОСТ28147-89	Rijndael
Размер блока, бит	64	128, 192, 256[1]
Размер ключа, бит	256	128, 192, 256
Часть блока, шифруемая за один раунд, бит	32 (полблока)	128, 192, 256 (полный блок)
Размер раундового ключевого элемента, бит	32 (половина размера блока)	128, 192, 256 (равен размеру блока)
Структура раунда	Простая	Более сложная
Используемые на раунде операции	Только аддитивные операции, подстановки и сдвиги	Широкое использование операций над конечными полями
Эквивалентность прямого и обратного преобразований.	С точностью до порядка следования ключевых элементов	С точностью до вектора ключевых элементов, узла замен и прочих констант алгоритма

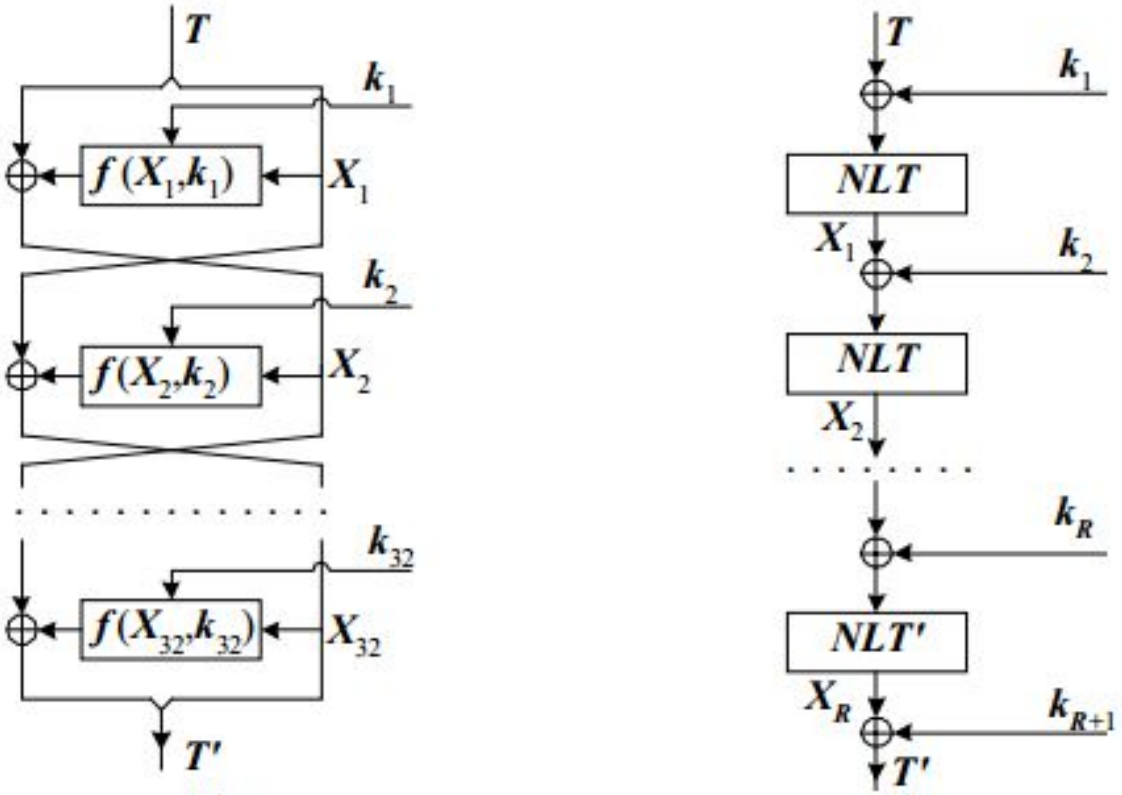
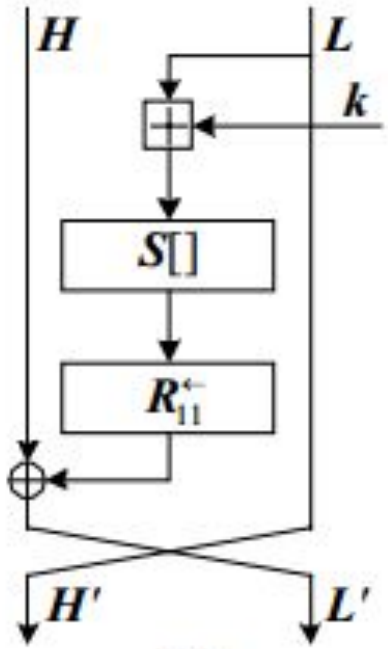
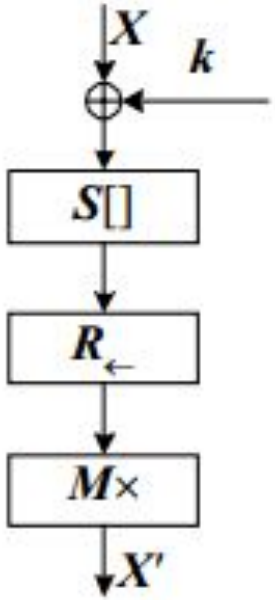


Схема преобразования данных при шифровании по алгоритмам ГОСТ28147-89 (а) и Rijndael (б) соответственно, где:

- T,T' – исходный и зашифрованный блоки соответственно;
- k_i – ключевой элемент раунда;
- X_i – состояние процесса шифрования после i-того раунда;
- $f(X,k)$ – функция шифрования алгоритма ГОСТ28147-89;
- NLT,NLT' – регулярное нелинейное преобразование и нелинейное преобразование последнего раунда алгоритма Rijndael соответственно;
- R – число раундов в алгоритме Rijndael (10, 12 или 14).



(а)



(б)

Схема преобразования данных для одного раунда шифрования по алгоритмам ГОСТ28147-89 (а) и Rijndael (б) соответственно, где:

$X(H,L)$, $X'(H',L')$ – преобразуемый блок (или его старшая и младшая части соответственно) на входе и на выходе раунда;

k – ключевой элемент раунда;

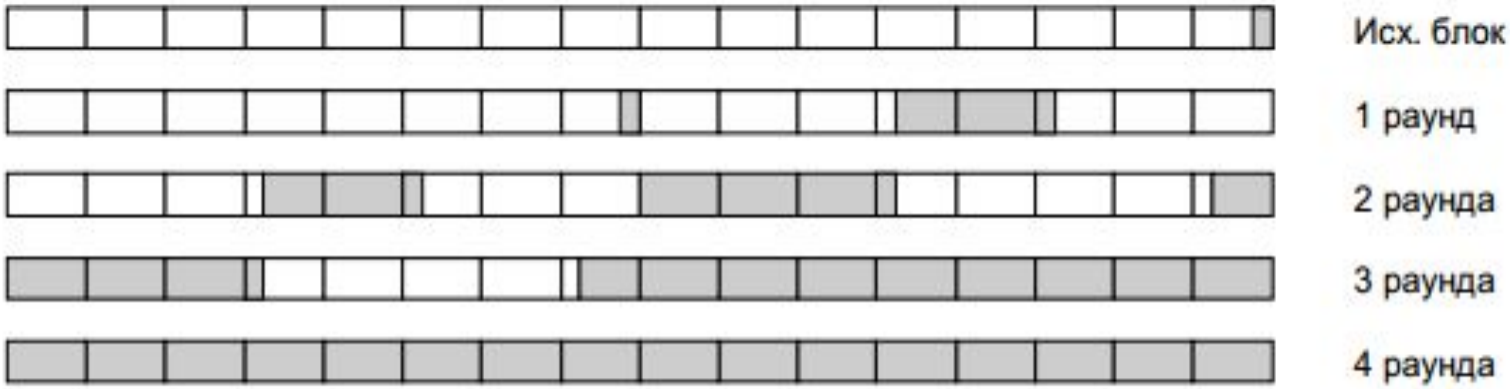
$S[]$ – функция подстановки, группами по 4 бита для ГОСТа и байтами для алгоритма Rijndael;

$R_{11}^{\leftarrow}$ – операция циклического сдвига (вращения) 32-битового слова на 11 бит в сторону старших разрядов;

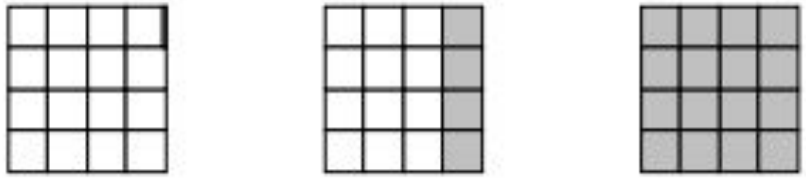
$R_{\leftarrow}$ – операция построчного вращения матрицы алгоритма Rijndael;

$M\times$ – умножение матрицы данных слева на матрицу M в алгоритме Rijndael.

Диффузионные характеристики



Диффузия изменения в исходных данных через несколько раундов шифрования алгоритма ГОСТ28147-89.



Диффузия изменения в исходных данных через несколько раундов шифрования алгоритма Rijndael для 128-битового блока.

- На конференции FSE 2011 (Fast Software Encryption) представлены алгебраические атаки на ГОСТ 28147-89, которые показали его слабость. В настоящее время известны несколько десятков алгебраических атак на этот стандарт.
- Это первый пример использования алгебраического криптоанализа для взлома стандартизованного государственного шифра.

МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ

НАКАЗ

від 30 грудня 2014 року N 1493

Про прийняття європейських стандартів як національних стандартів України та скасування національних стандартів України

Відповідно до Закону України "Про стандарти, технічні регламенти та процедури оцінки відповідності" та на виконання статті 56 і статті 57 Угоди про асоціацію між Україною, з однієї сторони, та Європейським союзом, Європейським співтовариством з атомної енергії і їхніми державами-членами, з іншої сторони, ратифікованої Законом України від 16.09.2014 N 1678-VII, наказую:

1. Прийняти європейські стандарти як національні стандарти України методом підтвердження, наведені в додатку 1, з набранням чинності з 01.01.2016.
2. Скасувати з 01.01.2016 національні стандарти України, наведені в додатку 2.
3. Департаменту технічного регулювання та метрології оприлюднити переліки національних стандартів України, прийнятих цим наказом, скасованих національних стандартів України на офіційному веб-сайті Мінекономрозвитку України.
4. Державному підприємству "Український науково-дослідний і навчальний центр проблем стандартизації, сертифікації та якості" опублікувати інформацію про прийняті цим наказом національні стандарти України, скасовані національні стандарти України та підтверджувальні повідомлення в наступному номері щомісячного інформаційного покажчика "Стандарти".

НАЦІОНАЛЬНИЙ СТАНДАРТ УКРАЇНИ

ДСТУ 7624:2014

Інформаційні технології

Криптографічний захист інформації

Алгоритм симетричного блокового перетворення

ОСНОВНІ ВЛАСТИВОСТІ

Київ 2015

Національний стандарт України ДСТУ 7624:2014

- симетричний блоковий шифр “Калина” (декілька варіантів розміру блоку і довжини ключа)
- режими роботи блокового шифру
- додаток: нелінійні таблиці заміни (S-блоки)
- додаток: алгоритм доповнення повідомлення для режимів роботи, що вимагають довжину, кратну розміру блоку
- додаток: приклади для перевірки, включаючи повідомлення із довжиною, не кратною розміру блоку і байту
- додаток: вимоги до реалізації

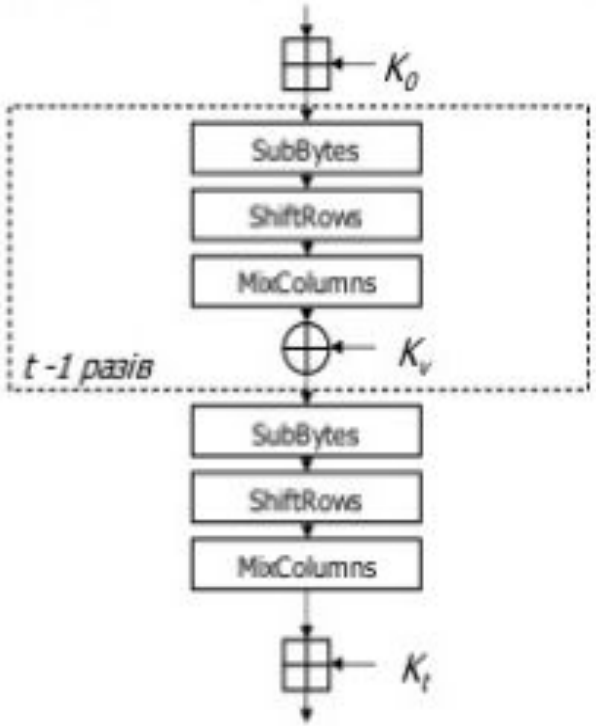
Вимоги до блокового шифру «Калина»

- високий рівень криптографічної стійкості із достатнім запасом на випадок появи нових атак протягом тривалого часу
- висока швидкодія програмної реалізації на сучасних та перспективних платформах
- компактність програмної і програмно-апаратної реалізації, можливість ефективної інтеграції декількох алгоритмів в одному засобі криптографічного захисту
- прозорість проектування, консервативний підхід щодо забезпечення стійкості
- вища (або порівняна) ефективність щодо найкращих світових рішень

Комбінації довжини ключа і розміру блоку шифру „Калина”

Розмір блоку	Довжина ключа
128	128, 256
256	256, 512
512	512

“Калина”: функція зашифрування



$$T_{l,k}^{(K)} = \eta_l^{(K_t)} \circ \psi_l \circ \tau_l \circ \pi_l' \circ$$
$$\circ \left(\prod_{r=1}^{t-1} \left(\kappa_l^{(K_r)} \circ \psi_l \circ \tau_l \circ \pi_l' \right) \right) \circ \eta_l^{(K_0)}$$

Ефективна програмна реалізація перетворення використовує один таблиць передобчислень (AES потребує два набору), виконавши оптимізацію для зашифрування, що дозволяє досягти вищої швидкодії як при зашифруванні, так і розшифруванні для більшості режимів (CTR, CFB, CMAC, OFB, GCM, GMAC, CCM).

Алгоритм шифрования «Калина»

Зашифрование	Расшифрование

void Kalina_Encrypt(byte in [8 * Nb], byte out [8 * Nb], byte subkey[Nr + 2][8 * Nb])
{
byte state[8, Nb] = in
XORRoundKey(state, subkey[0])
for(round = 1 to Nr/2 step 1)
{
Kalina_S_boxes(state)
ShiftRows(state)
MixColumns(state)
Add32RoundKey(state, subkey[2 * round - 1])
Kalina_S_boxes(state)
ShiftRows(state)
MixColumns(state)
XORRoundKey(state, subkey[2 * round])
}
Kalina_S_boxes(state)
Add32RoundKey(state, subkey[Nr + 1])
out = state
}
	void Kalina_Decrypt(byte in [8 * Nb], byte out [8 * Nb], byte subkey[Nr + 2][8 * Nb])
	{
	byte state[8, Nb] = in
	Sub32RoundKey(state, subkey[Nr + 1])
	Kalina_Inv_boxes(state)
	for(round = Nr/2 to 1 downstep 1)
	{
	XORRoundKey(state, subkey[2 * round])
	InvMixColumns(state)
	InvShiftRows(state)
	Kalina_Inv_boxes(state)
	Sub32RoundKey(state, subkey[2 * round - 1])
	InvMixColumns(state)
	InvShiftRows(state)
	Kalina_Inv_boxes(state)
	}
	XORRoundKey(state, subkey[0])
	out = state
	}
	}

Алгоритм разворачивания ключей ДСТУ 7624:2014

```
void Kalina_KeyExpansionKS( byte key[ 8 * Nk ], const Ci, byte KS[ 8 * Nk ])
{
      byte state[ Nk  ] = Ci

      XORRoundKey(state, key )
      Kalina_S_boxes( state )
      ShiftRows( state )
      MixColumns( state )
      Add32RoundKey(state, ~( key ) )
      Kalina_S_boxes( state )
      ShiftRows( state )
      MixColumns( state )
      XORRoundKey(state, key)

      KS = state
}
```

– XORRoundKey – сумма по модулю 2;

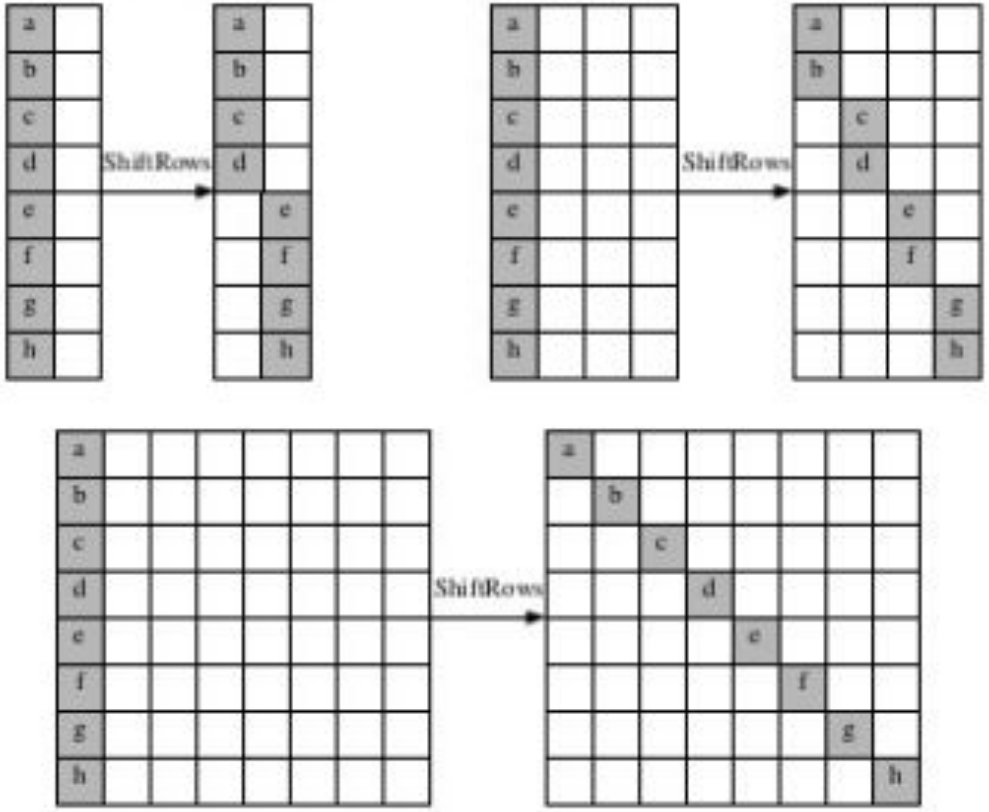
– Kalina_S_boxes – нелинейное отображение «байт в байт»;

– ShiftRows – циклический сдвиг строк состояния;

– MixColumns – умножение на фиксированную матрицу над полем $GF(2^8)$;

– Add32RoundKey – сложение по модулю 2^{32};

Калина: зсув рядків для різних значень розміру блоку



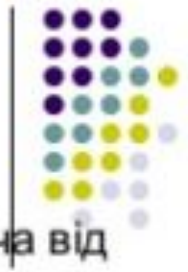
Блоковий шифр “Калина”: циклове перетворення

- чотири різних S-блоки із різних класів еквівалентності, із алгебраїчними властивостями, кращими ніж у AES, та нелінійністю вищою, ніж у СТБ 34.101.31-2011 та “Стрибог”/”Кузнечик” (**найкраще відоме у світі співвідношення** для захисту від різних атак)
  - можливість використання у якості довгосторокового ключа для спеціальних застосувань блокового перетворення
- один набір таблиць передобчислень, оптимізований для швидкодіючої реалізації як зашифрування, так і формування циклових ключів (для зашифрування і розшифрування більшість режимів роботи алгоритму вимагають виключно зашифрування режиму простої заміни, ECB)
- ефективність реалізації систем криптографічного захисту: основні елементи спільні для національних стандартів гешування і шифрування (блокового перетворення)

Функція зашифрування алгоритму “Калина”

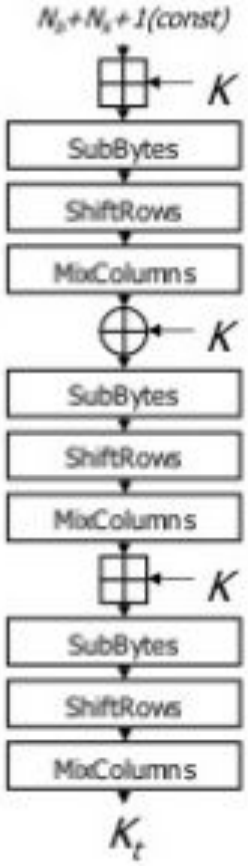
- прозора конструкція, консервативний підхід до проектування, використання відомої стратегії “широкого сліду” (wide trail design strategy), посилення попереднім та прикінцевим модульним (2^{64}) забілюванням
- наявність запасу стійкості на випадок появи нових атак протягом тривалого часу
- новий набір S блоків, які не можуть бути описані перевизначеною системою 2 го степеня (додатковий захист від алгебраїчних атак)
- орієнтація на 64 бітові платформи (додавання за модулем 2^{64}, МДВ матриця розміром 8x8)
- у більшості режимів роботи як для зашифрування, так і розшифрування повідомлень використовується лише пряме перетворення
- ефективна програмна і програмно апаратна реалізація

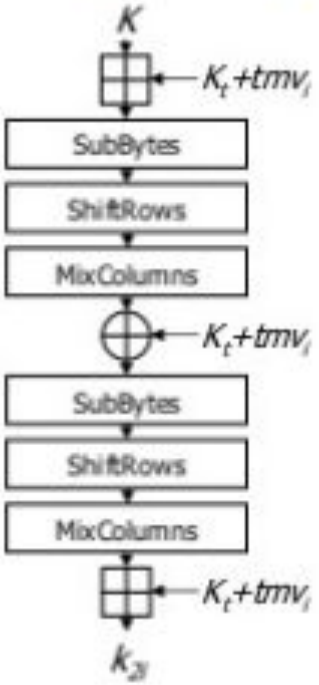
Вимоги до схеми формування циклових ключів алгоритму “Калина”



- нелінійна залежність кожного біта кожного циклового ключа від кожного біта ключа шифрування
- циклові ключі суттєво відрізняються і мають складну нелінійну залежність
- захист від відомих криптоаналітичних атак, що орієнтовані на схему розгортання ключів
- відсутність слабких ключів, при яких погіршуються криптографічні властивості або знижується стійкість перетворення
- висока обчислювальна складність відновлення ключа шифрування по одному або декільком цикловим ключам, що є доступними для криптоаналитика (додатковий захист від атак на реалізацію)
- обчислювальна складність формування всіх циклових ключів не перевищує складності зашифрування 2,5 блоків
- можливість формування циклових ключів у довільному порядку (однакова обчислювальна і просторова складність для зашифрування і розшифрування)
- простота програмної і програмно-апаратної реалізації

Схема формування циклових ключів алгоритму “Калина”





$$k_{2i+1} = k_{2i} <<< (2 \cdot N_b + 3)$$

tmv_0=0x01000100...0100

$tmv_{i+2} = tmv_i << 1$

Всі операції (за виключенням зсувів) взяті із функції зашифрування

Зсуви ефективно реалізуються операціями доступу до пам’яті (RAM)

$$\Theta^{(K)} = \psi_1 \circ \tau_1 \circ \pi_1' \circ \eta_1^{(K_w)} \circ \psi_1 \circ \tau_1 \circ \pi_1' \circ \kappa_1^{(K_w)} \circ \psi_1 \circ \tau_1 \circ \pi_1' \circ \eta_1^{(K_w)}$$

$$\Xi^{(K,K_\sigma,j)} = \eta_1^{(\varphi_j(K_\sigma))} \circ \psi_1 \circ \tau_1 \circ \pi_1' \circ \kappa_1^{(\varphi_j(K_\sigma))} \circ \psi_1 \circ \tau_1 \circ \pi_1' \circ \eta_1^{(\varphi_j(K_\sigma))}$$

Властивості схеми формування циклових ключів алгоритму “Калина”

- відповідність заданим вимогам
- всі операції взяті із функції зашифрування
- циклові ключі можуть бути сформовані як для зашифрування, так і розшифрування, з однаковою обчислювальною складністю
- ефективні засоби руйнування внутрішньої симетрії циклового перетворення
- мінімальна кількість констант, їхня прозорість
- обчислювальна складність формування всіх циклових ключів не перевищує складності зашифрування 2,5 блоків
- неін'єктивна залежність циклових ключів від ключа шифрування (додаткова стійкість до до низки методів криптографічного аналізу, спрямованого, в тому числі, і на апаратну або програмну реалізацію перетворення, без можливості зниження складності атак переборного типу)

Криптографічна стійкість блокового шифру “Калина”, блок 128 бітів

Метод криптоаналізу	Найменша кількість циклів, для якої шифр є стійким	Показники атак		
		Макс. кількість циклів	Обчисл. складність, екв. оп. шифрув.	Пам’ять, байтів
Диференційний	5	4	2^{55}	
Лінійний	5	3	$2^{52,8}$	
Усіч. диференц.	4	3		
Інтегральний	6	5	2^{97}	2^{33+4}
Нездійсн. дифер.	6	5	2^{62}	2^{66}
Бумеранг	5	4	2^{120}	

Криптографічна стійкість блокового шифру “Калина”, блок 256 бітів

Метод криптоаналізу	Найменша кількість циклів, для якої шифр є стійким	Показники атак		
		Макс. кількість циклів	Обчисл. складність, екв. оп. шифрув.	Пам’ять, байтів
Диференційний	7	6	2^{230}	
Лінійний	7	5	$2^{220,8}$	
Усіч. диференц.	4	3		
Інтегральний	7	6	2^{145}	$2^{64}+5$
Нездійсн. дифер.	6	5	2^{61}	2^{66}
Бумеранг	6	5	2^{220}	

Криптографічна стійкість блокового шифру “Калина”, блок 512 бітів

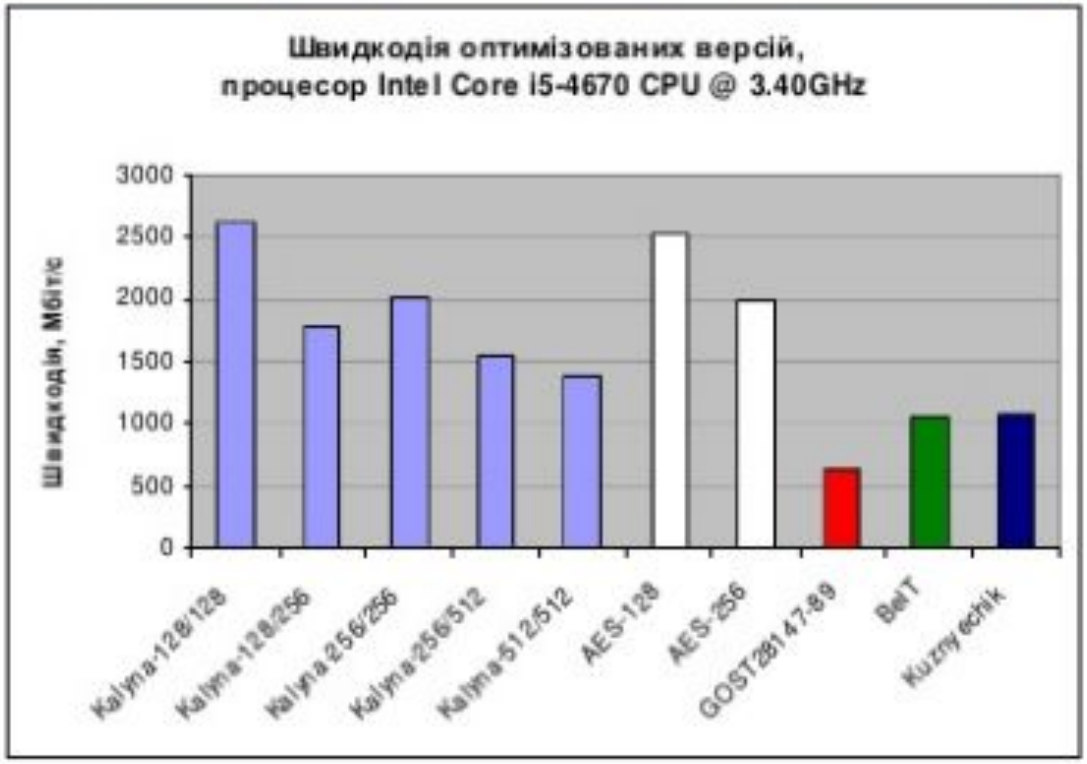
Метод криптоаналізу	Найменша кількість циклів, для якої шифр є стійким	Показники атак		
		Макс. кількість циклів	Обчисл. складність, екв. оп. шифрув.	Пам’ять, байтів
Диференційний	9	8	2^{490}	
Лінійний	9	7	$2^{470,4}$	
Усіч. диференц.	4	3		
Інтегральний	7	6	2^{137}	2^{64+5}
Нездійсн. дифер.	6	5	2^{60}	2^{66}
Бумеранг	7	6	2^{340}	

Криптографічна стійкість блокового шифру “Калина”

Стійкість забезпечується (наявність запасу):

- 128-битовий блок: 6 раундів (із 10 або 14, залежно від довжини ключа)
- 256-битовий блок: 7 раундів (із 14 або 18, залежно від довжини ключа)
- 512-битовий блок: 9 раундів (із 18)

Порівняння швидкодії блокового шифра “Калина” із іншими алгоритмами (Linux, найкраща оптимізація компілятора)



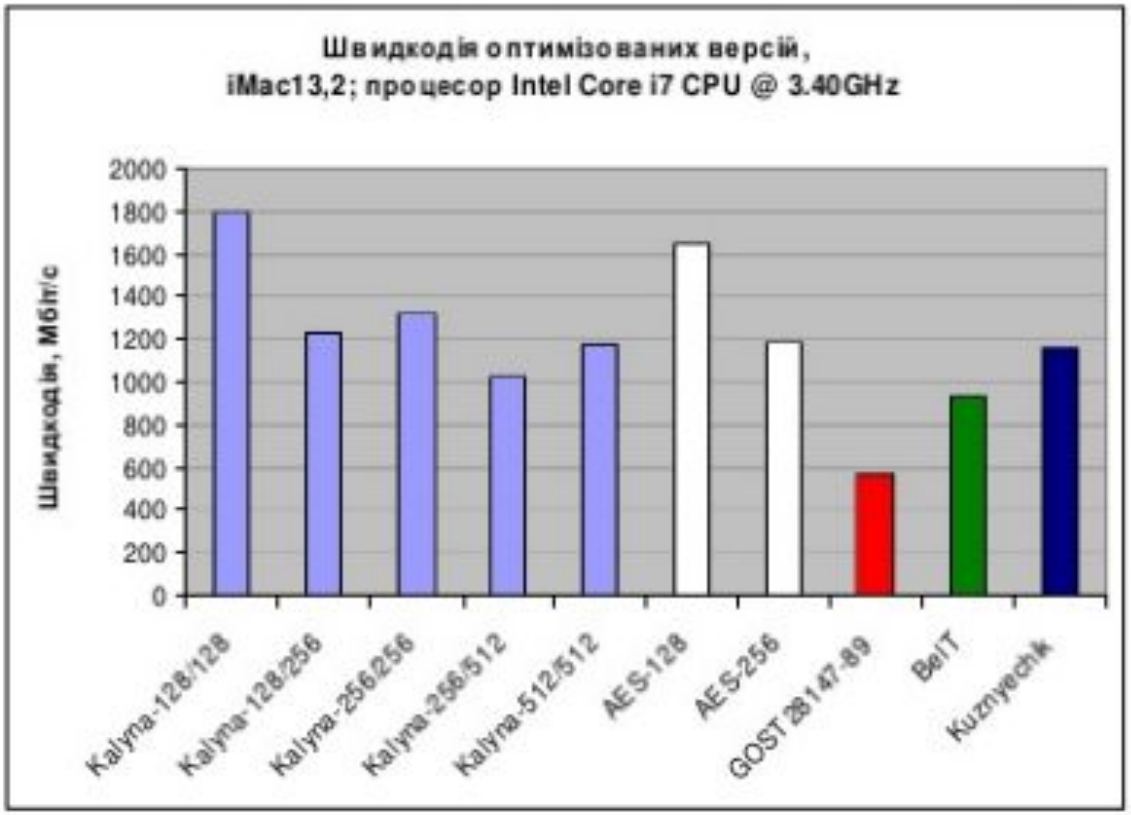
ОС: Linux 64 bit, компілятор gcc version 4.9.2 (Ubuntu 4.9.2-0ubuntu1~12.04, 30-Oct-2014)

https://github.com/Roman-Oliynykov/ciphers-speed/

Порівняння швидкодії блокового шифра “Калина” із іншими алгоритмами (Linux, найкраща оптимізація компілятора)

№ з/п	Блоковий шифр	Швидкодія, Мбіт/с
1	Kalyna-128/128	2611.77
2	Kalyna-128/256	1779.52
3	Kalyna-256/256	2017.97
4	Kalyna-256/512	1560.89
5	Kalyna-512/512	1386.46
6	AES-128	2525.89
7	AES-256	1993.53
8	GOST 28147-89	639.18
9	STB 34.101.31-2011(BelT)	1055.92
10	Kuznyechik	1081.08

Порівняння швидкодії блокового шифра “Калина” із іншими алгоритмами (iMac13.2, найкраща оптимізація компілятора)



https://github.com/Roman-Oliynykov/ciphers-speed/

Порівняння швидкодії блокового шифра “Калина” із іншими алгоритмами (iMac13.2, найкраща оптимізація компілятора)

№ з/п	Блоковий шифр	Швидкодія, Мбіт/с
1	Kalyna-128/128	1788.88
2	Kalyna-128/256	1236.32
3	Kalyna-256/256	1315.4
4	Kalyna-256/512	1023.76
5	Kalyna-512/512	1177.91
6	AES-128	1645.97
7	AES-256	1189.04
8	GOST 28147-89	569.475
9	STB 34.101.31-2011(BelT)	936.657
10	Kuznyechik	1158.67

Стандартні режими роботи блокового шифра: ISO/IEC 10116

- Electronic codebook (ECB)
- Cipher block chaining (CBC)
- Cipher feedback (CFB)
- Output feedback (OFB)
- Counter (CTR)

Недоліки стандартних режимів

- ECB: можливість компрометації повідомлення при наявності однакових блоків відкритого тексту
- CBC: вимоги до випадковості сінхропосилки, можливість компрометації повідомлення в специфічних умовах (які присутні в протоколах SSL/TLS); перешифрування є неефективним
- CFB: обов'язковий потрібен додатковий контроль цілісності повідомлення (модифікація шифртексту призводить до спрямованої модифікації відкритого тексту); перешифрування є неефективним

Недоліки стандартних режимів

- OFB: обов'язковий потрібен додатковий контроль цілісності повідомлення (модифікація шифртексту призводить до спрямованої модифікації відкритого тексту); перешифрування є неефективним
- CTR: найбільш зручний режим забезпечення конфіденційності, але обов'язковий потрібен додатковий контроль цілісності повідомлення (модифікація шифртексту призводить до спрямованої модифікації відкритого тексту), що накладає обмеження на застосування при прозорому шифруванні носіїв інформації

Режими роботи, визначені в стандарті

№ режиму	Назва режиму	Позначення	Послуга безпеки
1	Проста заміна (базове перетворення)	ECB	Конфіденційність
2	Гамування	CTR	Конфіденційність
3	Гамування зі зворотним зв'язком за шифртекстом	CFB	Конфіденційність
4	Вироблення імітовставки	CMAC	Цілісність
5	Зчеплення шифроблоків	CBC	Конфіденційність
6	Гамування зі зворотним зв'язком за шифргамою	OFB	Конфіденційність
7	Вибіркове гамування із прискореним виробленням імітовставки	GCM, GMAC	конфіденційність і цілісність (GCM), тільки цілісність (GMAC)
8	Вироблення імітовставки і гамування	CCM	цілісність і конфіденційність
9	Індексованої заміни	XTS	конфіденційність
10	Захисту ключових даних	KW	конфіденційність і цілісність

Режими роботи, визначені в стандарті

- 1-3,5,6: стандартні, відповідно до технічного завдання та ISO/IEC 10116:2006
- 4: контроль цілісності (вдосконалений алгоритм)
- режими 1-4 по призначенню і основними властивостям співпадають із тими, що визначені ДСТУ ГОСТ 28147:2009
- 7-10: додаткові режими, потрібні для сучасних засобів криптографічного захисту (шифрування IP-трафіку, носіїв інформації, ключових даних малого обсягу та ін.)

CMAC: вироблення імітовставки

- посилений варіант режиму виробки коду автентифікації повідомлення, що дозволяє захиститись від атаки збільшення довжини повідомлення (за рахунок доповнення останнього блоку)
- час виробки імітовставки дорівнює часу зашифрування повідомлення
- модифікований (зручний для розробників) варіант рекомендацій NIST
- подібний режим є в СТБ 34.101.31-2011 (орієнтований на 32-бітові процесори)

GCM, GMAC: вибіркове гамування із прискореним виробленням імітовставки

- одночасне забезпечення конфіденційності та цілісності (і найвищій рівень швидкодії в цих вимогах)
- захист мережевого трафіку при реалізації протоколів IPsec та ін.
- додаткові заходи захисту проти DoS атак ("відмова в обслуговуванні")
- є аналогічні режими NIST та СТБ 34.101.31-2011 (модифікації у національному стандарті спрямовані на більш зручну реалізацію)
- можливість використання нових інструкцій процесорів загального призначення для прискореного обчислення імітовставки
- особливості режиму враховані при розробці національного стандарту
  - ймовірність генерування т.з. "слабкого" ключа автентифікації (Markku Saarinen, "GCM, GHASH and Weak Key", 2011 р.) для довжин повідомлень (кількості блоків), визначеним Додатком Г, практично дорівнює ймовірності угадування зловмисником ключа блокого шифра;
  - складність пошуку т.з. "еквівалентних" ключів автентифікації на багато порядків вища, ніж складність традиційних колізійних атак (також врахованих Додатком Г стандарту)

CCM: вироблення імітовставки і гамування

- одночасне забезпечення конфіденційності та цілісності, заміна застосуванню двох режимів CMAC та CTR
- модифікований (зручний для розробників) варіант рекомендацій NIST, який використовується для захисту трафіку WiFi та WiMax

XTS: індексована заміна

- прозоре шифрування інформації, що зберегається на носіях із блоковим доступом (диски та ін.)
- є ефективним, коли неможливе забезпечення цілісності кожного блоку на носії (не дозволяє спрямовану модифікацію відкритого тексту через шифрований)
- швидкодія така ж, як і в CTR (але розшифрування трохи повільніше зашифрування із-за максимальної оптимізації швидкодії інших режимів)
- модифікований (зручний для розробників) варіант рекомендацій NIST та стандарту IEEE

KW: захист ключових даних

- дозволяє захищати блоки даних (особові ключі асиметричних систем і т.ін.) без використання синхропосилки
- забезпечує конфіденційність та цілісність
- швидкодія нижча, ніж у інших режимів відповідно до необхідності забезпечення залежності всіх блоків шифртексту від кожного блоку повідомлення (і навпаки)

Позначення параметрів перетворення і режимів роботи

Калина-*l*/*k*-режим-параметри режиму

- *l* –розмір блоку шифра
- *k* - довжина ключа

Калина-256/512-CCM-32,128

- розмір блоку 256 біт
- довжина ключа 512 біт
- режимі вироблення імітовставки і гамування
- довжина конфіденційної (та відкритої) частини повідомлення завжди менша 2^{32} байтів
- довжина імітовставки 128 біт

короткий запис, який однозначно визначає перетворення і всі його параметри

Використання позначень параметрів перетворення і режимів роботи

- кожен з режимів роботи визначає рекомендовані значення параметрів
- задача розробника технічного завдання
  - вибір параметрів перетворення
    - розмір блоку (128 біт - наприклад)
    - довжина ключа
      - персональні дані: 128 біт (приклад)
      - інформація, що є власністю держави: 256 біт (приклад)
  - вибір режиму роботи
    - електронний документ з ЦП: CTR (приклад)
    - мережевий трафік: GCM (приклад)
    - захищений диск: XTS (приклад)
  - взяти набір рекомендованих значень інших параметрів режиму (які можуть бути змінені при необхідності забезпечення сумісності із існуючими системами та ін.)
- у разі необхідності розробник може обрати інші обґрунтовані комбінації параметрів, і така реалізація також буде відповідати національному стандарту

Приклади для перевірки

- деталізована інформація для розробника
- формат little endian (процесори x86, x86_64, ARM та ін.)
- прості вхідні послідовності (00 01 02 …), які не потребують перенабору з паперового примірнику стандарту
- значення для кожної операції при зашифруванні, розшифруванні і формуванні циклових ключів для всіх комбінацій розміру блоку і довжини ключа
- режими роботи: приклади із повідомленнями, чия довжина не є кратною розміру блоку і, для деяких режимів, байту (бітова довжина)

Додаток (довідковий): вимоги до реалізації

- перелік загальних вимог щодо захисту від витоку побічними каналами (залежність часу шифрування, довжини повідомлення і ін. від конфіденційних параметрів)
- обмеження на сумарну довжину повідомлень, що захищаються з використанням одного ключа
- вимоги щодо синхропосилки
- захист від повторного прийому повідомлення

Рекомендації розробнику засобу КЗІ, який може не бути фахівцем у галузі криптоаналізу симетричних перетворень

Обмеження на сумарну довжину повідомлень, що захищаються з використанням одного ключа

Обмеження на кількість блоків, що захищаються на одному ключі			
Позначення режиму	Розмір блоку (l)		
	128	256	512
CTR, CFB, CBC, OFB, XTS	2^{60} (16 млн ТБ)	2^{124}	2^{251}
CMAC, CCM, GCM, GMAC, KW	2^{48} (64 ТБ)	2^{109}	2^{237}

Розрахунок зроблений на основі ймовірності успіху криптоаналітика 10^{-9} (рекомендації NIST: обмеження обчислюються для ймовірності 10^{-6} або 10^{-2}; доповіді щодо перспективного шифру РФ: ймовірність 1/2)

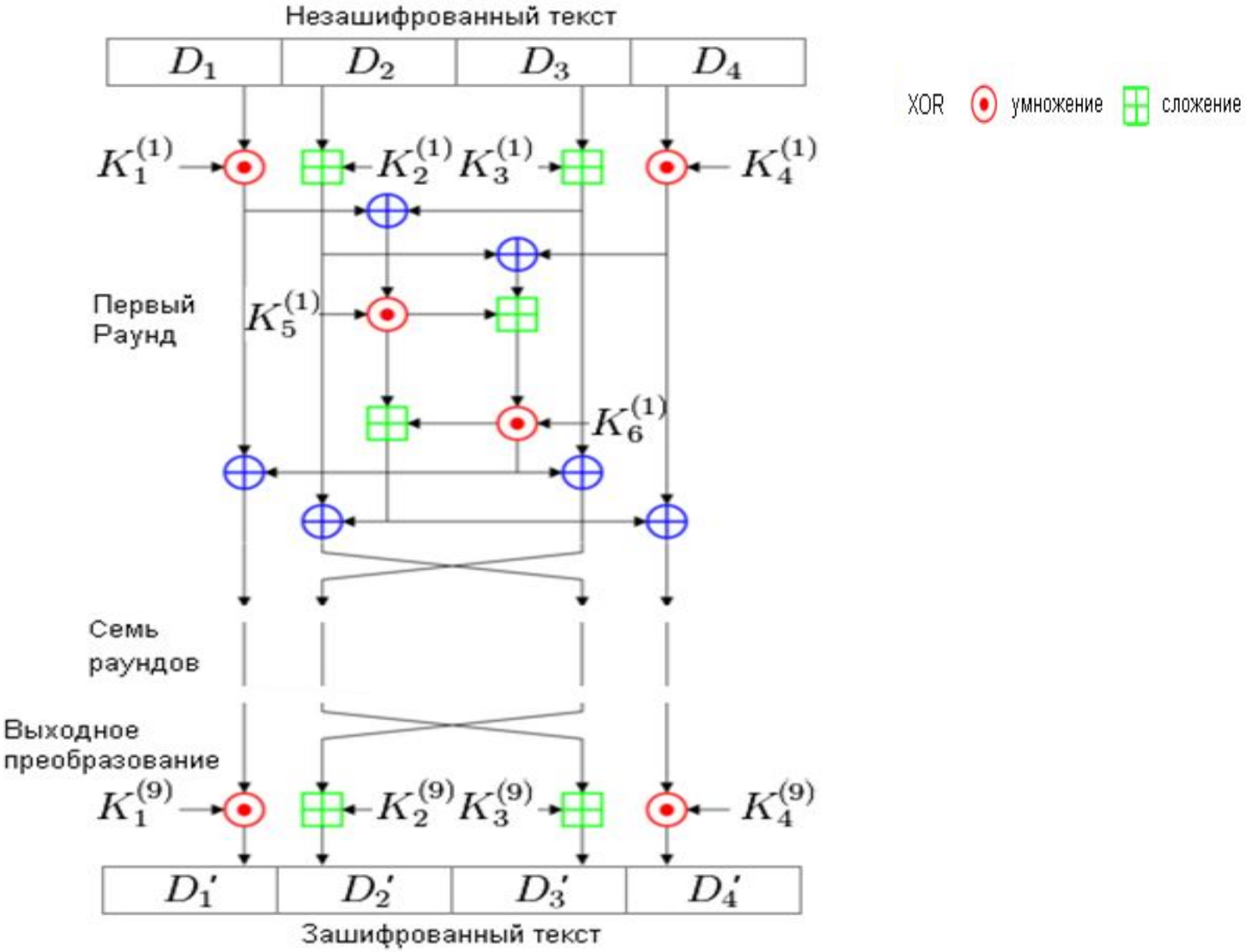
Необхідність враховувати властивості всієї системи при розробці засобів КЗІ: приклади атак 2013-2014 рр.

- режими роботи шифру
  - атака BEAST (режим CBC в протоколах SSL/TLS)
- атаки на реалізацію
  - атаки CRIME/BREACH
  - вразливість heartbleed в OpenSSL
  - часові (timing) атаки: промахи кешу процесора
  - виток побічними каналами
  - вразливості програмного забезпечення (buffer and heap overflows, etc.)
- безпека протоколів вищого рівня (наприклад, генерації ключа шифрування, DUAL_EC_DRBG з NIST SP 800-90A)

Новий національний стандарт симетричного блокового перетворення забезпечує

- високий і надвисокий рівень стійкості із запасом на випадок появи нових атак та вдосконалення криптоаналітичних комплексів протягом тривалого часу
- високу швидкодію програмної реалізації на сучасних та перспективних платформах
- вищу або порівняну ефективність щодо найкращих світових рішень
- наявність режимів роботи, необхідних для ефективної реалізації сучасних засобів криптографічного захисту
- можливість ефективної інтеграції двох національних алгоритмів в одному засобі криптографічного захисту
- зручність реалізації для розробників засобів криптографічного захисту

IDEA (***I**nternational **D**ata **E**ncryption **A**lgorithm*, международный алгоритм шифрования данных) — симметричный блочный алгоритм шифрования данных, запатентованный швейцарской фирмой Ascom



Математическое описание

- Блок открытого текста в 64 бит делится на четыре блока размером по 16 бит $(D_1^{(0)}, D_2^{(0)}, D_3^{(0)}, D_4^{(0)})$
- Для каждого раунда $(i = 1...8)$ вычисляются:

$$
\begin{aligned}
A^{(i)} &= D_1^{(i-1)} * K_1^{(i)} \\
B^{(i)} &= D_2^{(i-1)} + K_2^{(i)} \\
C^{(i)} &= D_3^{(i-1)} + K_3^{(i)} \\
D^{(i)} &= D_4^{(i-1)} * K_4^{(i)} \\
E^{(i)} &= A^{(i)} \oplus C^{(i)} \\
F^{(i)} &= B^{(i)} \oplus D^{(i)} \\
D_1^{(i)} &= A^{(i)} \oplus ((F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \\
D_2^{(i)} &= C^{(i)} \oplus ((F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \\
D_3^{(i)} &= B^{(i)} \oplus (E^{(i)} * K_5^{(i)} + (F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \\
D_4^{(i)} &= D^{(i)} \oplus (E^{(i)} * K_5^{(i)} + (F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)})
\end{aligned}
$$

Результатом выполнения восьми раундов есть следующие четыре блока $(D_1^{(8)}, D_2^{(8)}, D_3^{(8)}, D_4^{(8)})$

- Выходное преобразование $(i = 9)$:

$$
\begin{aligned}
D_1^{(9)} &= D_1^{(8)} * K_1^{(9)} \\
D_2^{(9)} &= D_3^{(8)} + K_2^{(9)} \\
D_3^{(9)} &= D_2^{(8)} + K_3^{(9)} \\
D_4^{(9)} &= D_4^{(8)} * K_4^{(9)}
\end{aligned}
$$

Результатом выполнения выходного преобразования является зашифрованный текст $(D_1^{(9)}, D_2^{(9)}, D_3^{(9)}, D_4^{(9)})$

Таблица подключей для каждого раунда	
Номер раунда	**Подключи**
1	$K_1^{(1)}\ K_2^{(1)}\ K_3^{(1)}\ K_4^{(1)}\ K_5^{(1)}\ K_6^{(1)}$
2	$K_1^{(2)}\ K_2^{(2)}\ K_3^{(2)}\ K_4^{(2)}\ K_5^{(2)}\ K_6^{(2)}$
3	$K_1^{(3)}\ K_2^{(3)}\ K_3^{(3)}\ K_4^{(3)}\ K_5^{(3)}\ K_6^{(3)}$
4	$K_1^{(4)}\ K_2^{(4)}\ K_3^{(4)}\ K_4^{(4)}\ K_5^{(4)}\ K_6^{(4)}$
5	$K_1^{(5)}\ K_2^{(5)}\ K_3^{(5)}\ K_4^{(5)}\ K_5^{(5)}\ K_6^{(5)}$
6	$K_1^{(6)}\ K_2^{(6)}\ K_3^{(6)}\ K_4^{(6)}\ K_5^{(6)}\ K_6^{(6)}$
7	$K_1^{(7)}\ K_2^{(7)}\ K_3^{(7)}\ K_4^{(7)}\ K_5^{(7)}\ K_6^{(7)}$
8	$K_1^{(8)}\ K_2^{(8)}\ K_3^{(8)}\ K_4^{(8)}\ K_5^{(8)}\ K_6^{(8)}$
выходное преобразование	$K_1^{(9)}\ K_2^{(9)}\ K_3^{(9)}\ K_4^{(9)}$

Таблица сравнения скоростей			
Блочный шифр	**Длина блока, бит**	**Число циклов**	**Скорость шифрования, Мбайт/с**
AES	128	226	216,0
DES	64	340	72,0
IDEA	64	590	41,2
3DES	64	158	154,4

Сравнение с некоторыми блочными шифрами в реализации PGP

Таблица сравнения основных параметров блочных шифров в реализации PGP			
Алгоритм	**Ключ, бит**	**Блок, бит**	**Примечания**
3 DES	168	64	Сеть Фейстеля. имеет пространство полуслабых и слабых ключей.
AES	256	128	Основан на операциях с таблицами массивов данных; принят в качестве гос. стандарта в США; обладает высокой криптостойкостью.
IDEA	128	64	Основан на смешении операций из разных алгебраических групп; имеет пространство слабых ключей; не все работы по криптоанализу были опубликованы.

Области применения IDEA

- шифрование аудио и видео данных для кабельного телевидения, видеоконференций, дистанционного обучения;
- защита коммерческой и финансовой информации, отражающей конъюнктурные колебания;
- линии связи через модем, роутер или ATM линию, GSM технологию;
- смарт-карты;
- общедоступный пакет конфиденциальной версии электронной почты PGP v2.0[3] и (опционально) в OpenPGP.