СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

Блоковые шифры 2

История криптоанализа DES

- 1977: Диффи-Хелман специализированная машина для поиска ключа DES (20 млн. $)
- 1990: Гарон и Отербридж – поиск ключа за 9 дней (1 млн. $)
- 1993: Вайнер – поиск ключа 3.5 часа (1 млн. $)
- 1997: DES-I конкурс: распределенные вычисления в Internet ------ **3months**
- 1998: Electronic Frontier Foundation (EFF: DeepCrack) ------ **3 days** (250K$)
- 1999: DES-III конкурс ------ **22 hours**
- 2006: COPACOBANA (120 FPGAs) **------ 7 days** (10K$)
- ⇒ НЕЛЬЗЯ использовать ключи 56 бит!!!
- (128---bit key $\Rightarrow 2^{72}$ days)

Deep Crack(EFF)

- 1856 специализированных ASIC DES чипов(Deep Crack or AWT-4500)
  - 29 плат по 64 чипа на каждой
- Может проверять 90 млрд. ключей в секунду.
  - Может найти ключ за 9 дней.
- 15 июля 1998, Deep Crack расшифровал сообщение, зашифрованное DES всего за 56 часов работы, получив приз $10,000.
- Это привело к отказу от DES.

История стандартизации AES

- 1997: NIST объявляет конкурс на разработку нового стандарта шифрования
- 1998: Предложено 15 кандидатов. 5 из них вскрыто.
- 1999: NIST выбирает 5 финалистов
- 2000: NIST объявляет победителя – Rijndael становится новым стандартом AES
- Размеры ключа: 128, 192, 256 бит. Блок: 128 бит.

Подстановочно-перестановочная сеть

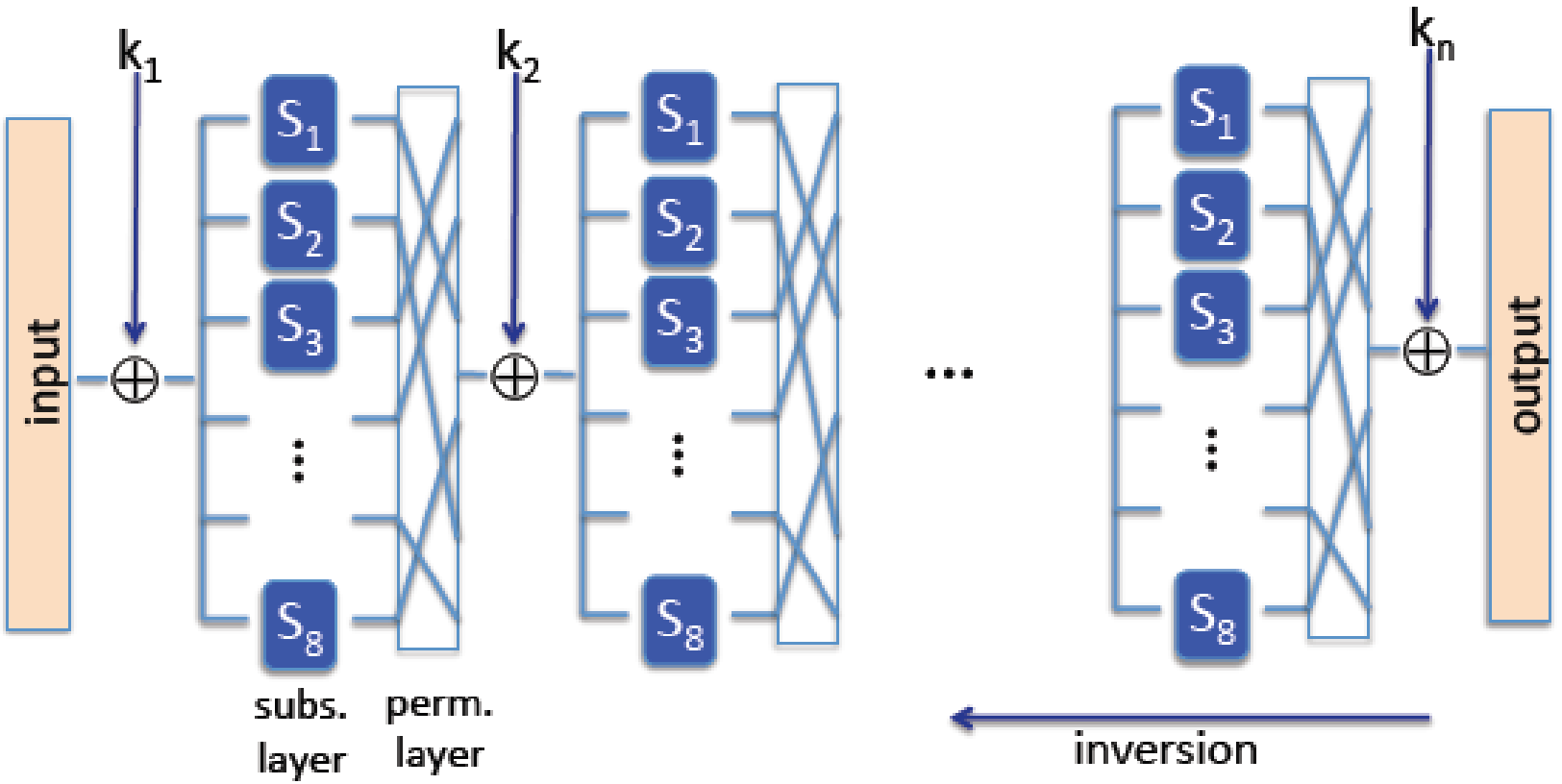
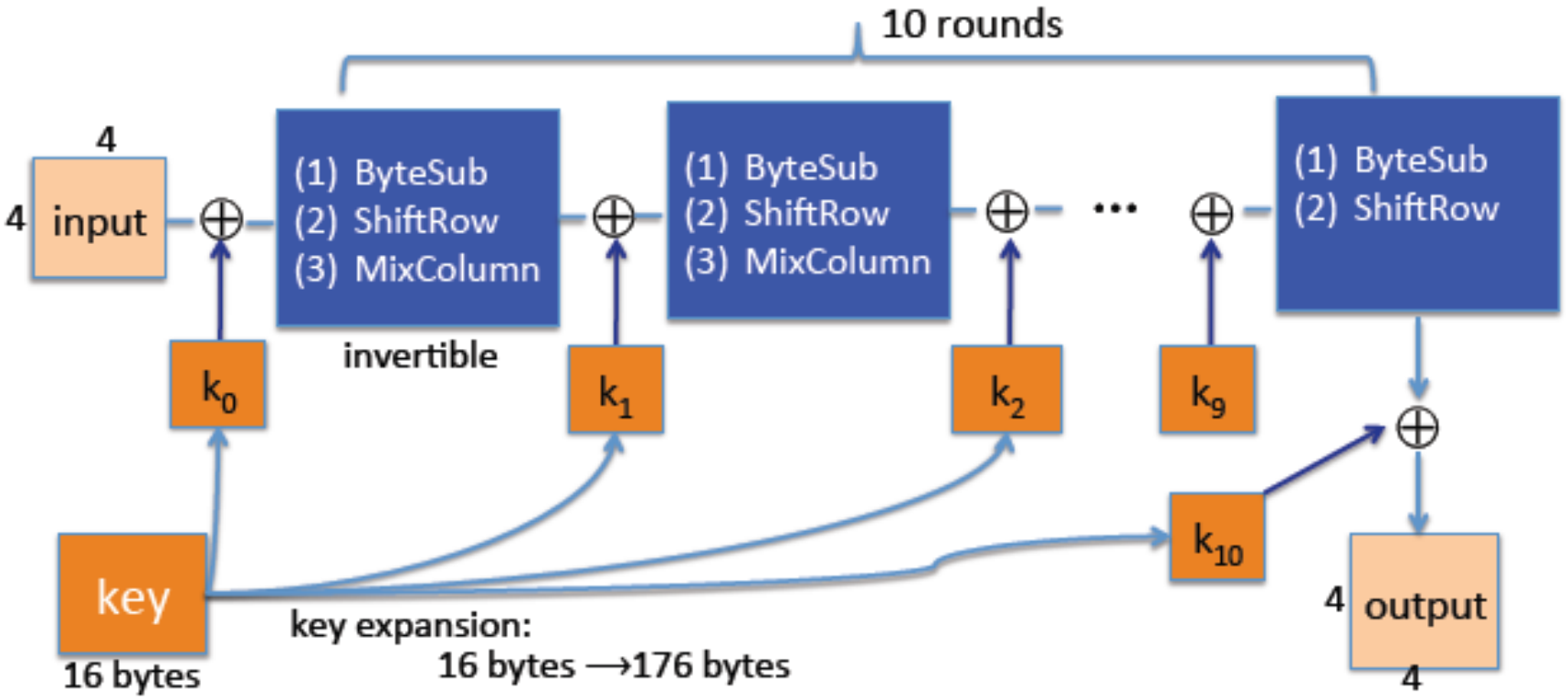
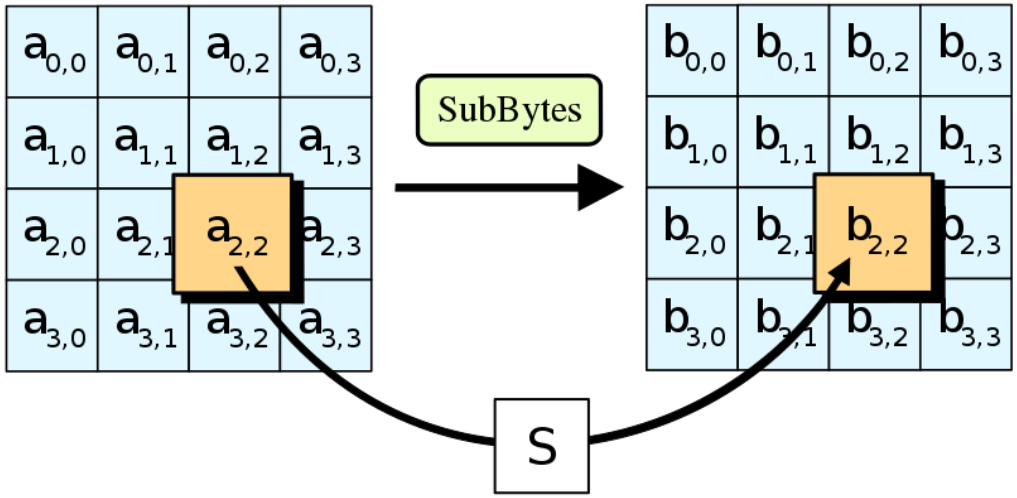


Схема работы AES



SubByte

- SubByte: Побайтная подстановка (256 байт->256 байт):
  - Табличная

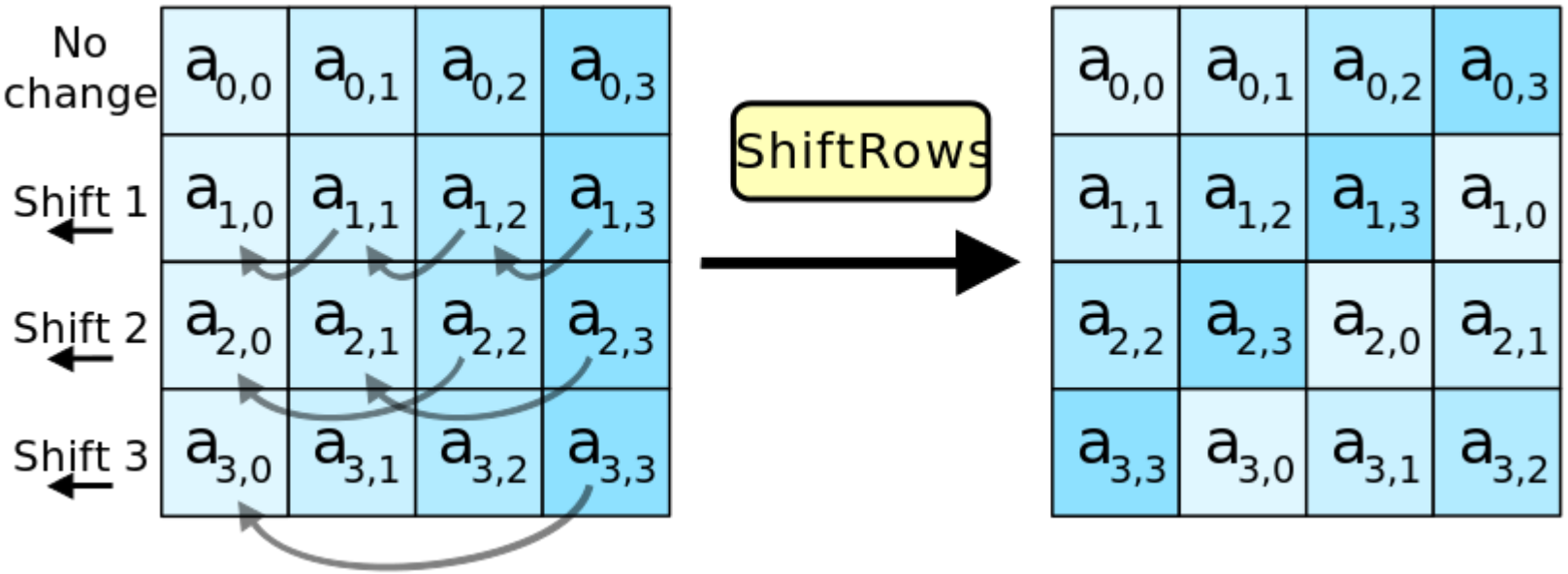


  - Вычислимая

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

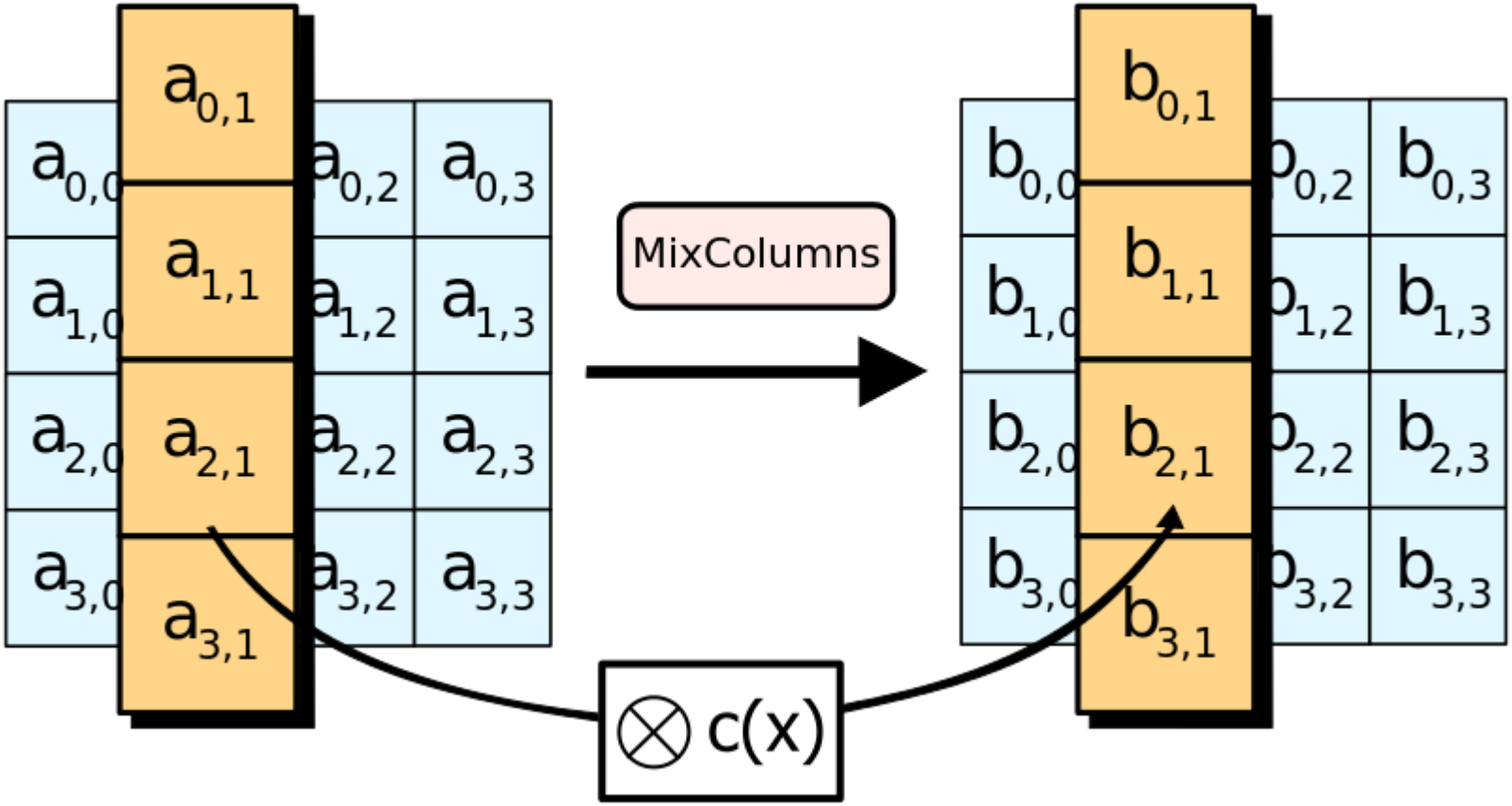
ShiftRow

- Байты в каждой строке циклически сдвигаются влево



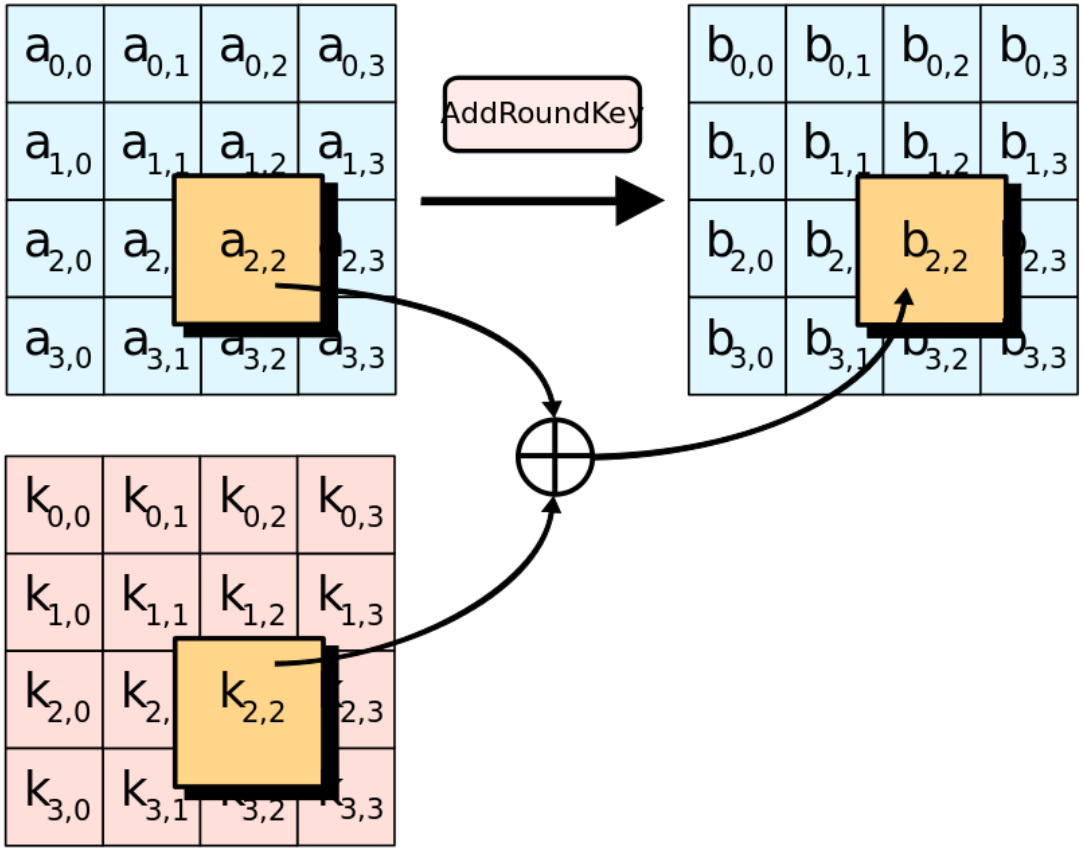
MixColumn

- Каждая колонка состояния перемножается с фиксированным многочленом $c(x) = 3x^3 + x^2 + x + 2$



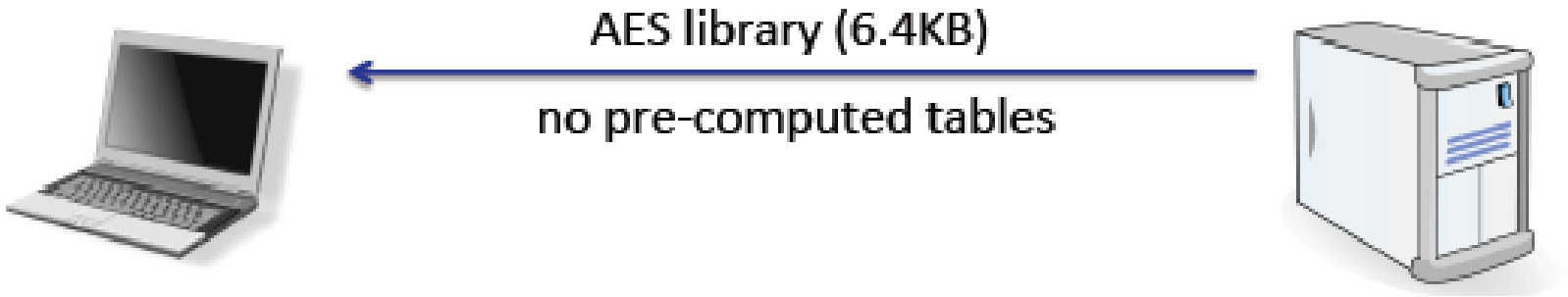
AddRoundKey

- Каждый байт состояния объединяется с ключом раунда, используя операцию XOR(⊕).



Пример реализации: Javascript AES

- Использование AES в браузере:



  - Перед началом шифровая вычисляются таблицы
  - Шифрование по таблицам

- Аппаратная реалиция:
  - Intel Westmere (ускорение в 14 раз OpenSSL)
  - AMD Bulldozer

Исчерпывающий поиск

- **Цель атаки:** по небольшому количеству пар $(m_i, c_i = E(k, m_i)), i = 1, \dots, l$ найти ключ k.

- Лемма: Пусть DES идеальный шифр (т.е. представляет собой 2^{56} случайных обратимых функций $\pi_i: \{0,1\}^{64} \to \{0,1\}^{64}$), тогда для $\forall\, m, c\ \exists$ не более $1\ k: c = DES(k, m)$ с вероятностью $Pr \geq 1 - \frac{1}{256} \approx 99{,}5\%$
- Док-во:
- $Pr\{\exists k' \neq k: c = DES(k, m) = DES(k', m)\} \leq$ $\sum_{k' \in \{0,1\}^{56}} Pr\{DES(k, m) = DES(k', m)\} \leq 2^{56} \cdot \frac{1}{2^{64}} = \frac{1}{2^8}$

Достаточное количество пар

- Для двух пар сообщений DES $(m_1, c_1 = DES(k, m_1)), (m_2, c_2 = DES(k, m_2))$ вероятность найти уникальный ключ $\approx 1 - \frac{1}{2^{71}} \geq 99{,}9\%$

- Для двух пар сообщений AES-128 $(m_1, c_1 = AES(k, m_1)), (m_2, c_2 = AES(k, m_2))$ вероятность найти уникальный ключ $\approx 1 - \frac{1}{2^{128}} \geq 99{,}9\%$

- Значит для исчерпывающего поиска достаточно 2-х пар открытый-закрытый текст

Реализация атаки

- Нам осталось только найти ключи… как?

- Перебор всех возможных ключей один за другим, пока не получено совпадение
  - иначе называется: метод грубой силы
  - работает для всех блоковых шифров

- Сложность атаки определяется длиной ключа N:
  - В худшем случае 2^N
  - В среднем половина 2^{N-1}

Усиление DES

- **Triple-DES**

- Пусть E: $K \times M \rightarrow M$ некоторый блоковый шифр

- Определим **3E**: $K^3 \times M \rightarrow M$ как

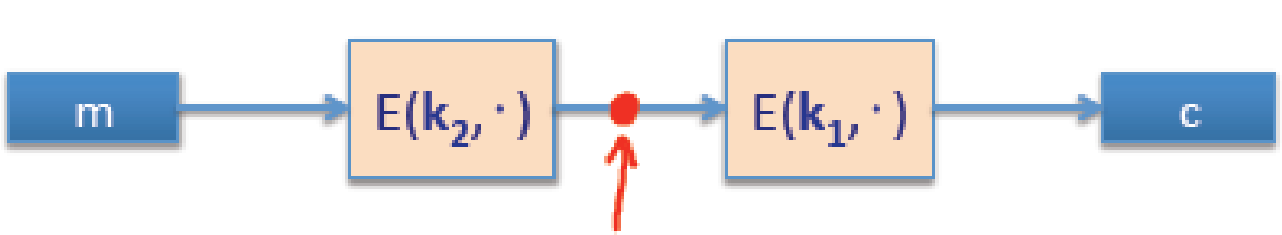
$$\mathbf{3E}((k_1, k_2, k_3), m) = E(k_1, D(k_2, E(k_3, m))),$$

если $k_1 = k_2 = k_3$ => стандартное шифрование

- Для 3DES:
  - длина ключа 3*56 = 168 бит (сложность атаки $\approx 2^{118}$)
  - сложность шифрования в 3 раза выше.

Почему не 2DES?

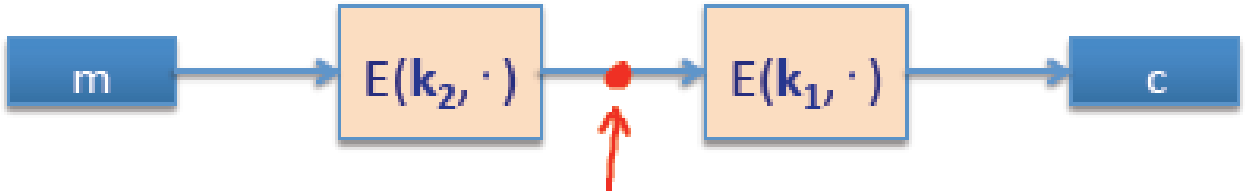
- Определим функцию $2E((k_1, k_2), m) = E(k_1, E(k_2, m))$
  - Для DES длина ключа 56*2 = 112 бит



- Определим атаку как
  - поиск пары (k_1, k_2): $E(k_1, E(k_2, m))=c$

- Аналогично:
  - поиск пары (k_1, k_2): $E(k_2, m)=D(k_1, c)$

Атака: Встреча посередине



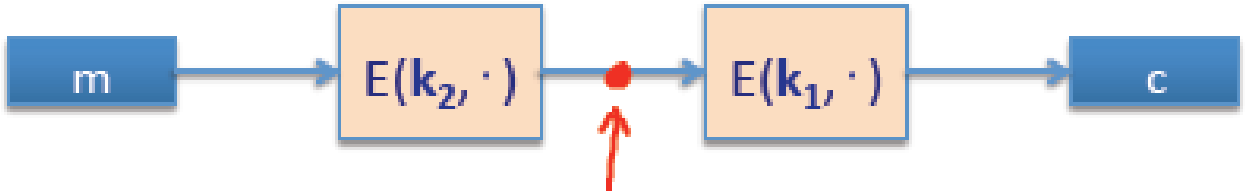
- Атака: M, C
  - Шаг 1: Построить таблицу:
  - отсортировать по 2-му столбцу

$k^0 = 00\ldots00$	$E(k^0, M)$
$k^1 = 00\ldots01$	$E(k^1, M)$
$k^2 = 00\ldots10$	$E(k^2, M)$
⋮	⋮
$k^N = 11\ldots11$	$E(k^N, M)$

  - Шаг 2: Для всех возможных
  - $k \in \{0,1\}^{56}$ проверить, что D(k,C) есть в таблице,
  - если это так, то $E(k^i, M) = D(k, C) \Rightarrow (k^i, k) = (k_1, k_2)$

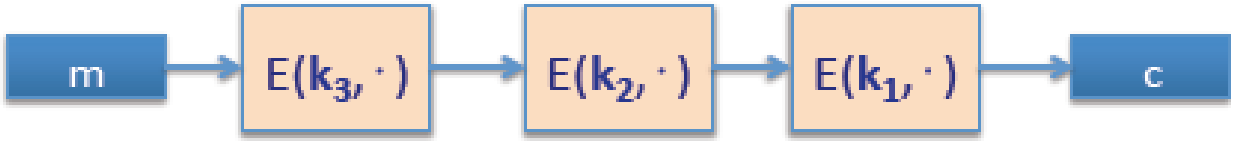
- Сложность?

Встреча посередине: сложность



- $Time = 2^{56}\log(2^{56}) + 2^{56}\log(2^{56}) < 2^{63} \ll 2^{112}$
- Память $\approx 2^{56}$

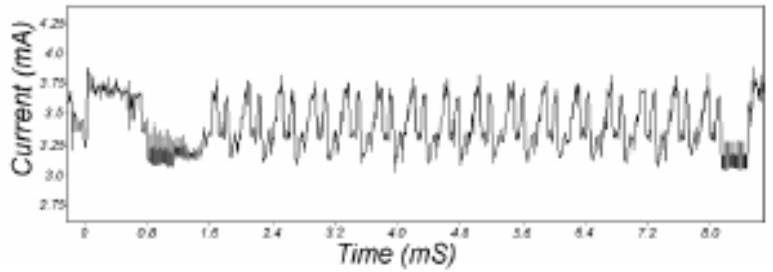
- Применение атаки к 3DES:
  - Time $= 2^{118}$
  - Память $\approx 2^{56}$



Атаки на реализацию

- Атака по сторонним (или побочным) каналам:
  - Измерение времени работы шифр/дешифр
  - Измерение потребляемой энергии при шифр/дешифр

smartcard



[Kocher, Jaffe, Jun, 1998]

- Физические атаки (Fault attack)
  - Внешними физическими воздействиями добиваются сбоя системы (результат: разглашение ключей, принятие неверного пароля…)
  - При нарушении шифрования на последнем раунде – публикация ключа.

Линейный криптоанализ

- Шон Мерфи – 1990, Мицуро Мацуи - 1992
- комбинированный методом, сочетающий:
  - поиск линейных статаналогов для уравнений шифрования,
  - статистический анализ имеющихся открытых и шифрованных текстов,
  - методы согласования и перебора
- Позволил получить наиболее сильные результаты по раскрытию ряда итерационных систем блочного шифрования, в т.ч. DES.

$$\Pr\left[m[i_1] \oplus \cdots \oplus m[i_r] \oplus c[j_1] \oplus \cdots \oplus c[j_v] = k[l_1] \oplus \cdots \oplus k[l_u] \right] = \tfrac{1}{2} + \varepsilon$$

биты сообщения биты шифротекста биты ключа

Для алгоритма DES была получена оценка $\varepsilon = 1/2^{21} \approx 0.0000000477$

Линейный криптоанализ

$$\Pr\left[m[i_1]\oplus\cdots\oplus m[i_r] \oplus c[j_j]\oplus\cdots\oplus c[j_v] = k[l_1]\oplus\cdots\oplus k[l_u] \right] = \tfrac{1}{2} + \varepsilon$$

- Теорема: Если нам даны $\frac{1}{\varepsilon^2}$ пар сообщений $(m, c = DES(k, m))$, то значения бит ключа равно

$$k[l_1, \dots, l_u] = \mathrm{MAJ}\left[m[i_1, \dots, i_r] \oplus c[j_1, \dots, j_v]\right]$$

- с вероятностью $\geq 97{,}7\%$

$\Longrightarrow$ при наличии $\frac{1}{\varepsilon^2}$ пар PT/CT рзначения битов ключа $k[l_1, \dots, l_u]$ может быть найдено за $\frac{1}{\varepsilon^2}$ шагов

Линейный криптоанализ DES

- Так как для DES $\varepsilon = {}^{1}/_{2^{21}} \Rightarrow$

  при получении 2^{42} PT/CT пар можно найти биты ключа $k[l_1, \dots, l_u]$ за 2^{42} операций

- Точнее 14 бит ключа можно найти за 2^{42} шагов

- Остальные биты получить перебором за $2^{56-14} = 2^{42}$

- Общее время атаки: 2^{42} при получении 2^{42} пар текстов

Дифференциальный криптоанализ

- 1990, Эди Бихам, Ади Шамир
- Использует неравновероятность в распределении значений разности 2-х шифртекстов, полученных из пары имеющих некоторую фиксированную разность открытых текстов
- Привел к появлению требования на равномерность распределения разности шифртекстов
- Применим и к хеш-функциям

Дифференциальный криптоанализ

- Основные идеи
  - Chosen-plaintext метод
  - Выбираем пары входных текстов с фиксированной разностью $\Delta X = X1 \oplus X2$, смотрим, как отличаются шифры от них $\Delta Y = Y1 \oplus Y2$
  - Анализируя много таких пар, находим наиболее вероятный ключ K

Дифференциальный анализ DES

- Пусть в алгоритме есть S-box с n-битовым входом и m-битовым выходом
- Дифференциал – это пара: разность входных данных и разность выходных данных нашего преобразования

$$\Delta X \xrightarrow{p} \Delta Y$$

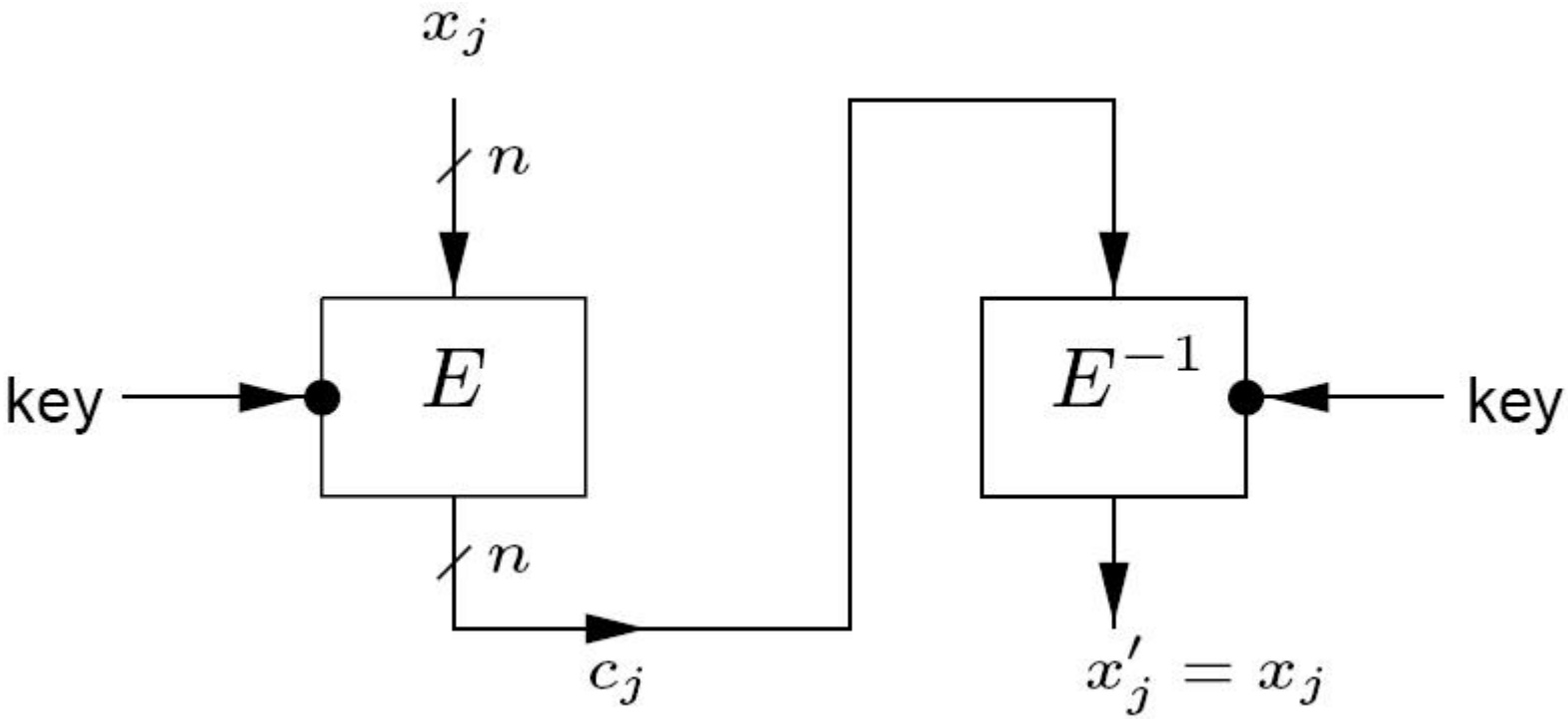
- Если Q - это количество различных пар входов, дающих этот дифференциал, то $p=Q/2^n$ – вероятность этого дифференциала
- Дифференциальная характеристика – это последовательность разностей после каждого раунда $(\Delta_0; \Delta_1; \dots; \Delta_R)$
- Построив дифференциальную характеристику на раундах с 1го по предпоследний, проводим атаку перебором ключа на последнем раунде.
- Для взлома DES необходимо 2^{47} выбираемых нами входных текстов

Режимы шифрования

- Постановка задачи:
  - Предположим, что у нас есть хороший метод кодировать блоки длиной 128 бит (например AES).
  - Но сообщение длиннее. Что делать?
  - Мы сейчас будем рассматривать разные методы построения шифров для длинных сообщений из маленьких блоков.

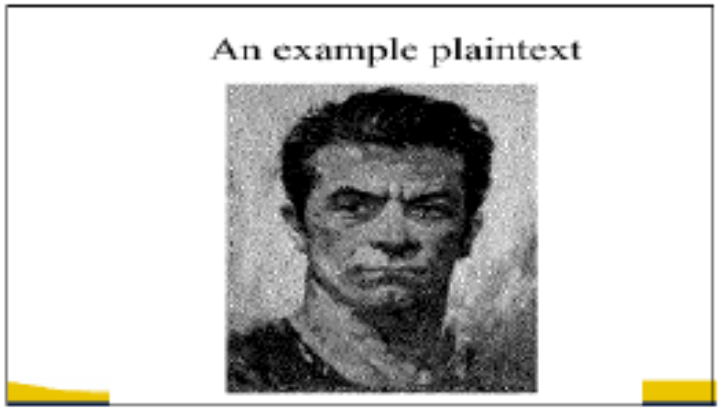
ECB

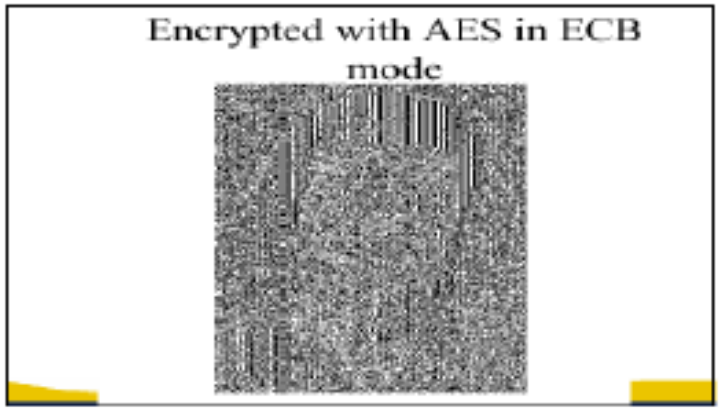
- Самое простое ECB (electronic codebook); блоки кодируются независимо друг от друга. Что плохо?



ECB: недостатки

- Одинаковые блоки переходят в одинаковый код.
- Блоки кодируются независимо; следовательно, их можно переставлять.
- Ошибка в одном блоке дальше этого блока не идет.

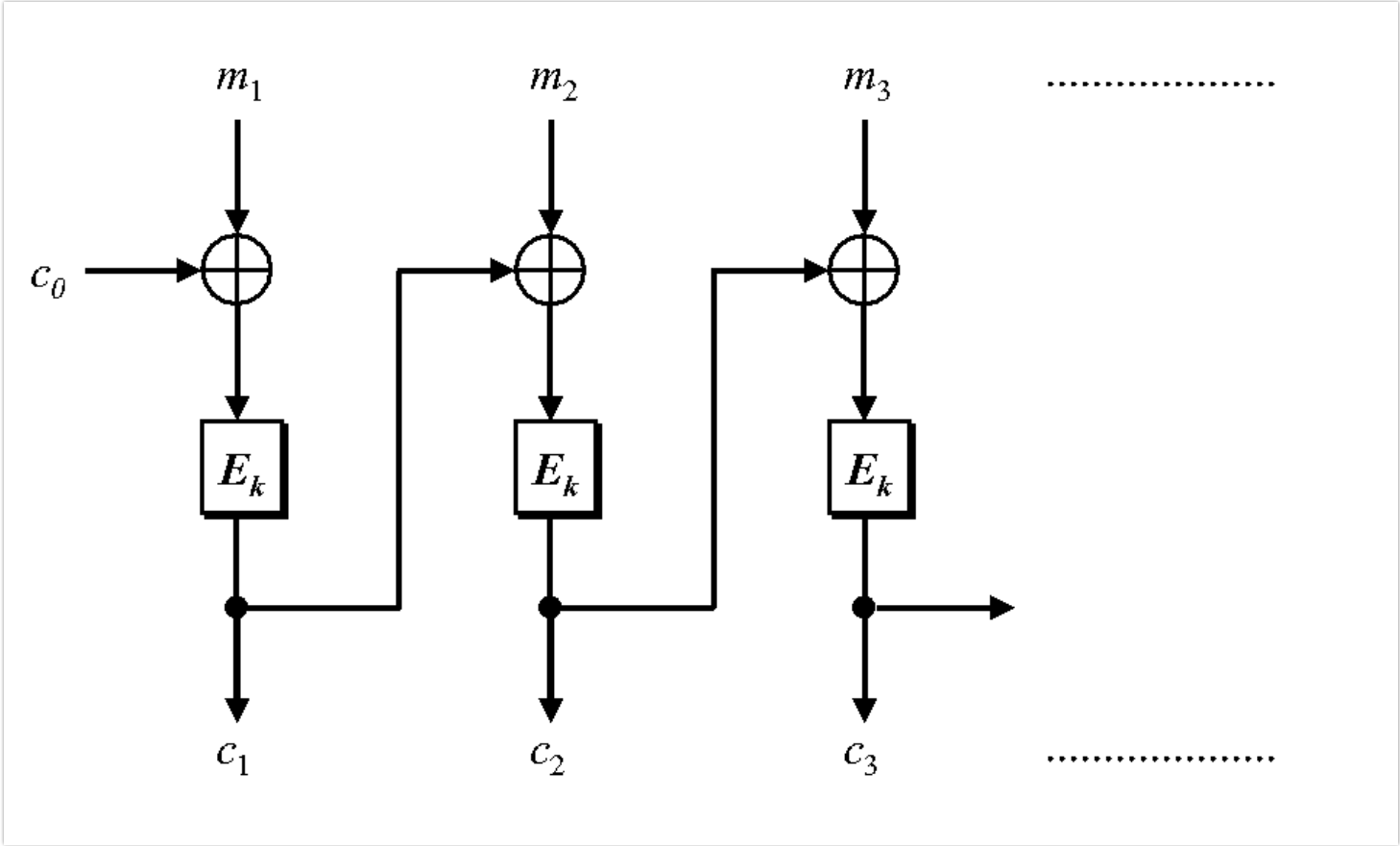




Не рекомендуется использовать!

CBC mode

- CBC (cipher block chaining): $c_0 = IV, c_j = E\left(k, m_j \oplus c_{j-1}\right)$



CBC

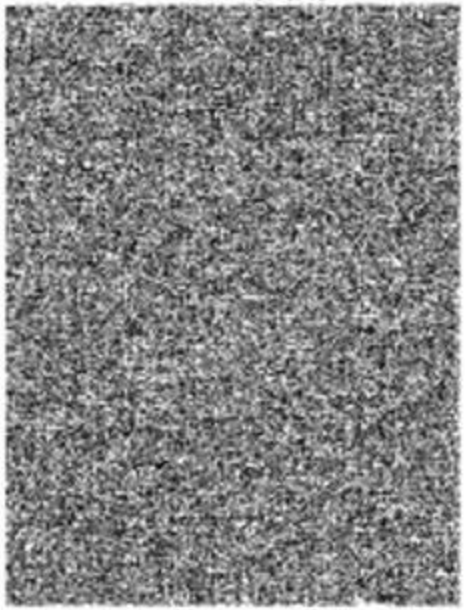
- Идентичные сообщения переходят в идентичные коды, только если IV тоже совпадает.
- Блок зависит от предыдущих, переставлять нельзя.
- Ошибка в блоке c_j вызывает ошибку в m_{j+1} на том же месте, дальше все OK; но m_j теряется полностью.
- Т.е. враг может подкорректировать m_{j+1} , но только ценой того, что m_j превратится в мусор.

CBC mode

- На примере изображения

An example plaintext

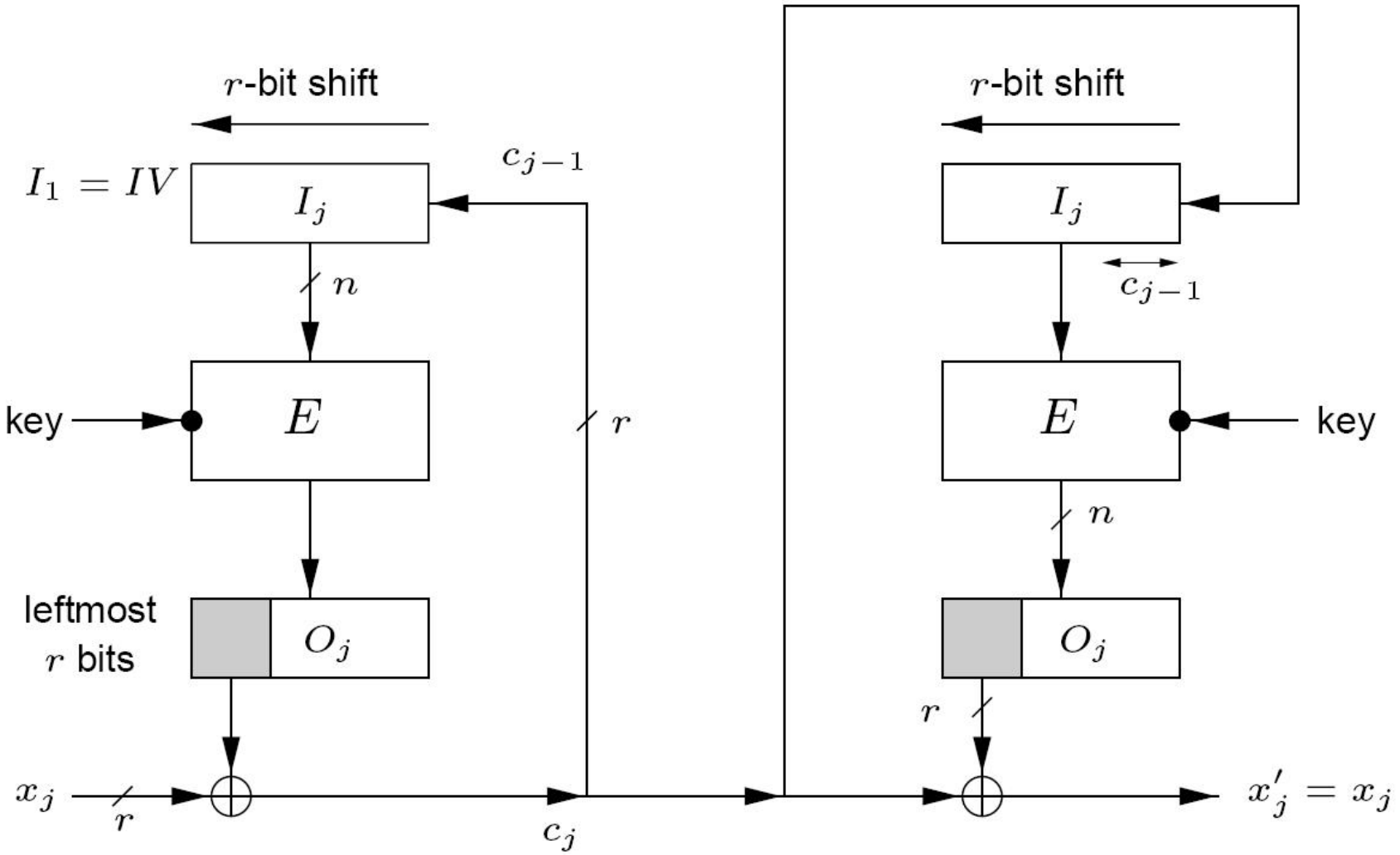
Encrypted with AES in CBC mode



CFB

- CFB (cipher feedback): когда нужно иногда отправлять сообщения размером меньше длины блока, скажем r бит. Тогда:
  - берем $l_0 = IV$,
  - считаем код $O_j = E(k, l_j)$,
  - берем t_j как r крайних слева битов O_j,
  - подсчитываем $c_j = t_j \oplus m_j$,
  - сдвигаем $l_{j+1} = 2^r \cdot l_j + c_j \ mod \ 2^n$

CBF


r-bit shift
$I_1 = IV$
I_j
c_{j-1}
n
key
E
r
leftmost
r bits
O_j
x_j
r
c_j
r-bit shift
I_j
c_{j-1}
E
key
n
O_j
r
$x'_j = x_j$

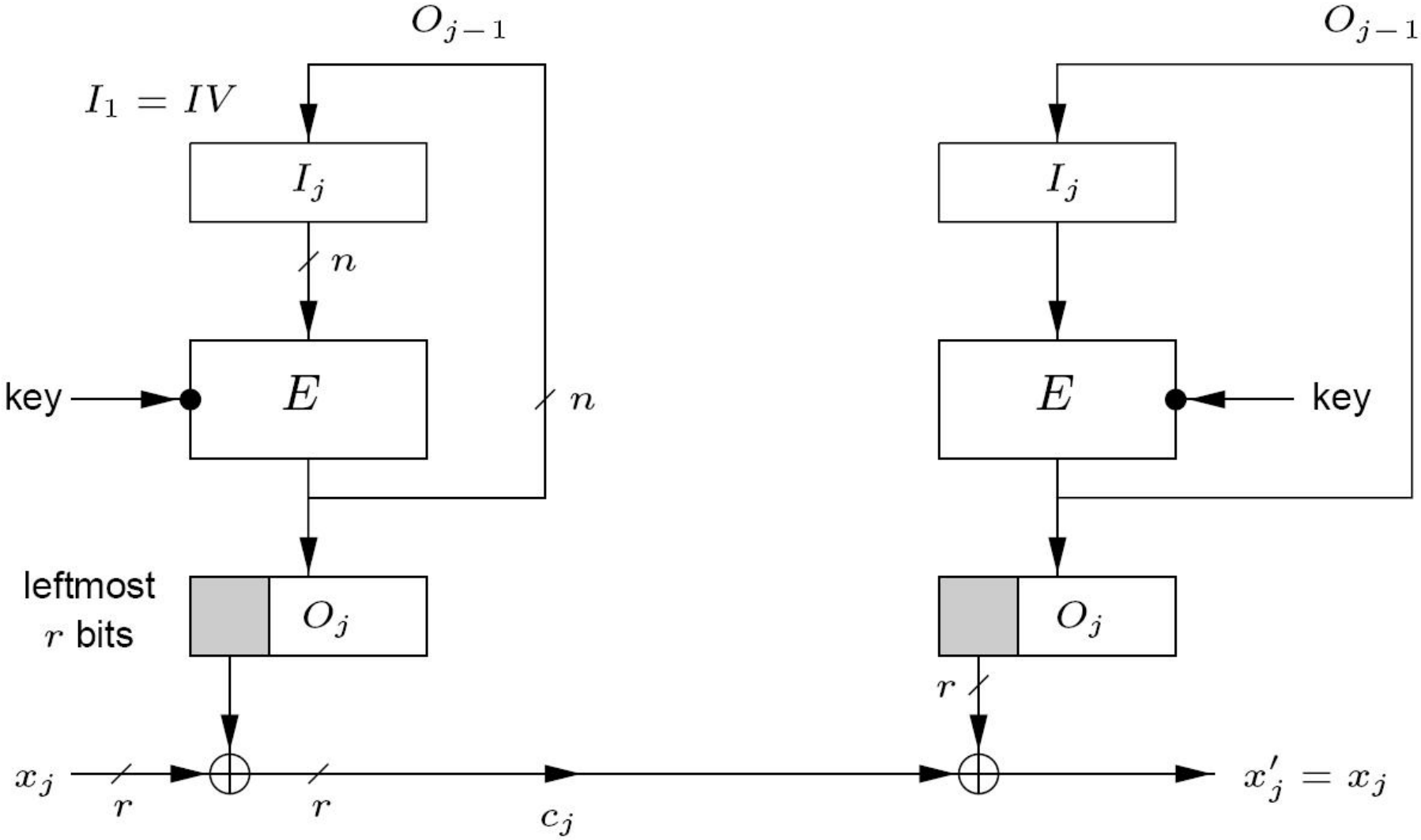
CBF:свойства режима

- Идентичные сообщения переходят в идентичные коды, только если IV тоже совпадает.
- Блок зависит от предыдущих, переставлять нельзя. Чтобы блок декодировался правильно, надо, чтобы предыдущие [n/r] были правильным.
- Ошибка в блоке c_j вызывает ошибку в [n/r] последующих блоках.
- Каждый запуск E выдает только r битов, а не n, как мог бы.

OFB

- OFB (output feedback): когда нехорошо, чтобы ошибка так далеко распространялась. Тогда:
  - берем $l_0 = IV$,
  - считаем код $O_j = E(k, l_j)$,
  - берем t_j как r крайних слева битов O_j,
  - подсчитываем $c_j = t_j \oplus m_j$,
  - берем следующий $l_{j+1} = O_j$.

OFB: схема


O_{j-1}
$I_1 = IV$
I_j
n
key
E
n
leftmost
r bits
O_j
x_j
r
r
c_j
O_{j-1}
I_j
E
key
O_j
r
$x'_j = x_j$

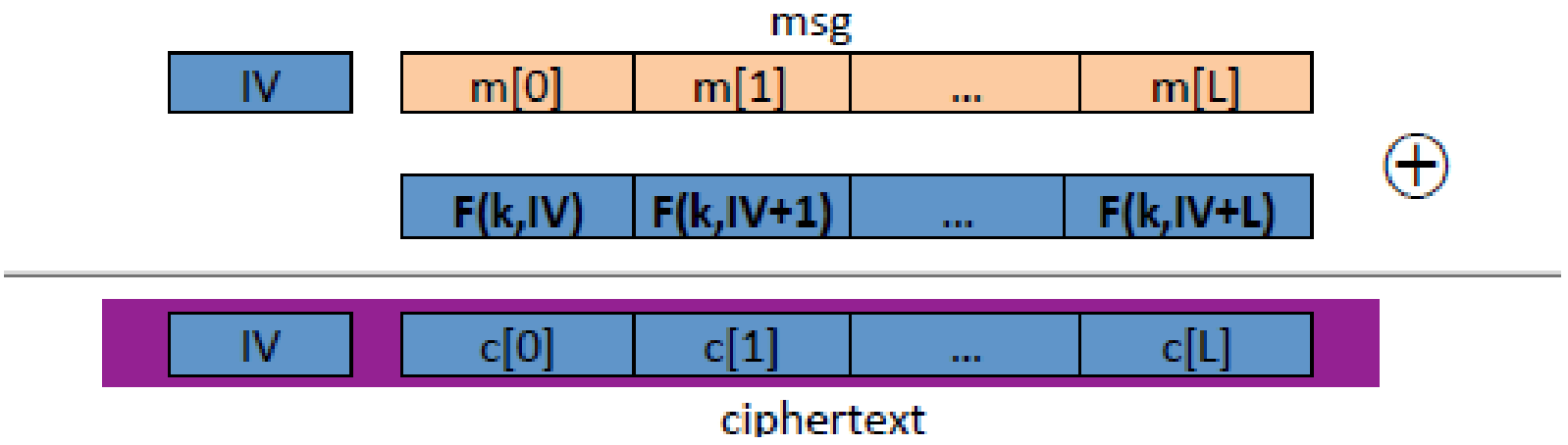
OFB: свойства

- Идентичные сообщения переходят в идентичные коды, только если IV тоже совпадает.
- Блок не зависит от предыдущих. Поэтому, в частности, нельзя использовать тот же ключ с тем же IV два раза.
- Ошибка в блоке c_j к другим ошибкам не приводит, но зато потерялась самосинхронизация.
- Каждый запуск E выдает только r битов, а не n, как мог
- бы; но зато теперь кодовый поток можно вычислять
- заранее.

CTR mode

- CTR (counter): когда необходимо, чтобы вычисления можно было производить параллельно. Тогда:
  - берем $l_0 = IV$,
  - считаем код $t_j = E(k, l_j)$,
  - подсчитываем $c_j = t_j \oplus m_j$,
  - берем следующий $l_{j+1} = l_j + 1$.

CTR: схема



CTR: свойства

- Идентичные сообщения переходят в идентичные коды, только если IV тоже совпадает.
- Блок не зависит от предыдущих. Поэтому, в частности, нельзя использовать тот же ключ с тем же IV два раза.
- Ошибка в блоке c_j к другим ошибкам не приводит, но зато потерялась самосинхронизация.
- Кодовый поток можно вычислять заранее или кодировать блоки параллельно.