Лекция №2

Часть 1. Блочный шифр.

Елена Киршанова

Курс “Основы криптографии”

Блок-шифры: мотивация

Симметричные шифр-схемы могут быть основаны на

1. Потоковом шифре
2. Блочном шифре

Блочные шифры лежат в основе шифрования с аутентификацией.

Блок-шифр – это детерминированный шифр $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ с $\mathcal{K}, \mathcal{X} := \mathcal{M} = \mathcal{C}$, и эффективной функцией

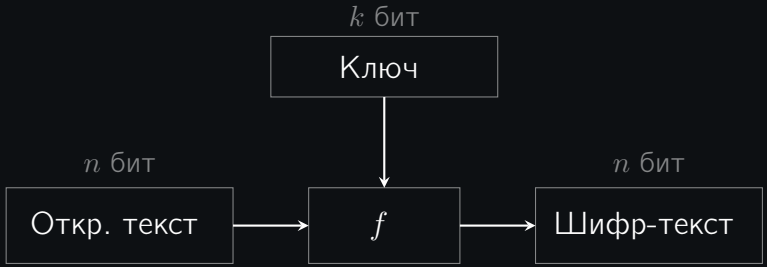
$$\mathsf{Enc} = f(k, \cdot) : \mathcal{X} \to \mathcal{X}.$$

При этом выполняется

- корректность $\implies$ $f(k, \cdot)$ – биекция для всех $k \in \mathcal{K}$
- $|\mathcal{X}| < \infty$.

То есть, $f(k, \cdot)$ – перестановка на $\mathcal{X}$.

Шифрование блока



Примеры:

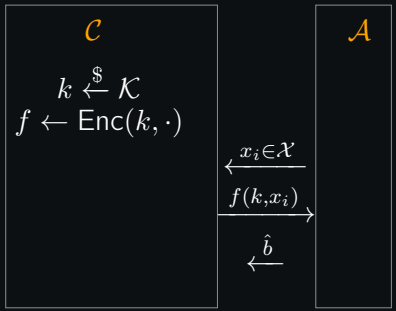
- AES: $n = 128$, $k = 128, 192, 256$
- ГОСТ 34.12-2018: $n = 128$, $k = 256$ (Кузнечик)

Безопасность блочного шифра $\Pi = (\mathsf{KeyGen}, \mathsf{Enc} = f, \mathsf{Dec} = f^{-1})$

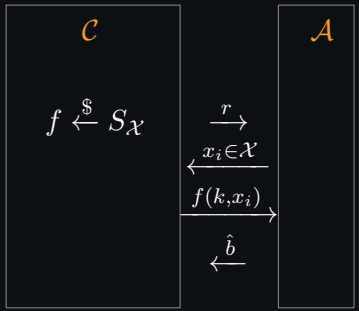
$f(k,\cdot) : \mathcal{X} \to \mathcal{X}$ должна быть вычислительно неотличима от случайной перестановки на $\mathcal{X}$.

$S_{\mathcal{X}}$ – множество всех перестановок на $\mathcal{X}$.

Эксперимент 0



Эксперимент 1



Выигрыш $\mathcal{A}$: $\mathsf{BlockAdv}\,[\mathcal{A}, \Pi] = |\Pr[b == \hat{b}] - 1/2|$.

Блок-шифр Π безопасный, если $\mathsf{BlockAdv} = \mathrm{negl}(\cdot)$ для всех ppt $\mathcal{A}$.

Немного истории

- 70'e: IBM опубликовывает шифр Lucifer. $k = 128, n = 128$
- '76: DES – стандарт $k = 56, n = 64$
- '98: 3DES – стандарт $k = 168, n = 64$
- '00: конкурс AES побеждает Rejndael $k = \{128, 192, 256\}, n = 128$

Российские стандарты:

- '89: ГОСТ 28147-89 $k = 256, n = 64$
- '15 : ГОСТ Р 34.12-2015/2018, RFC 7801 $k = 256, n = 128$

Две основные парадигмы в дизайне блочных шифров

- Сеть Фейстеля (создатель: Horst Feistel)

  Примеры: DES, ГОСТ 28147-89

- Substitution-Permutation Network (SPN)
  Подстановочно-перестановочная сеть

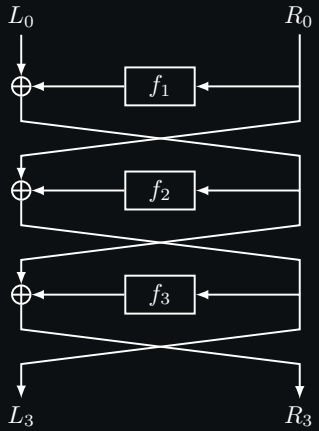
  Примеры: AES, ГОСТ 34.12-2018

Шифр Фейстеля

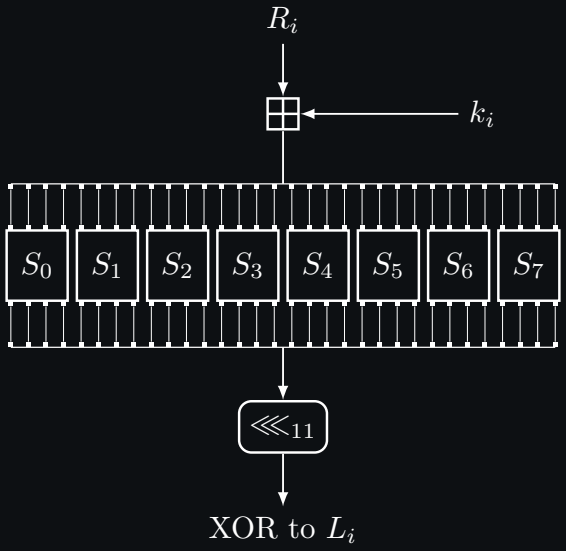
Сесть Фейстеля предлагает общий метод построения перестановки из *любой* функции

Дано $f(k, \cdot) : \{0,1\}^n \to \{0,1\}^n$

построить обратимую $F(k, \cdot) : \{0,1\}^{2n} \to \{0,1\}^{2n}$



Пример: Раундовая функция f в ГОСТе'89



.tex код avanzi-tikz-defs.tex

Что такое S-бокс?

$$S := \{0,1\}^n \to \{0,1\}^m - \text{ таблица подстановки}$$

- Реализована таблицей поиска (Lookup table)
- В одном блочном шифры может быть использовано несколько S-боксов
- S-бокс не должен содержать фиксированных точек: $S(x) \neq x$, $S(x) \neq \bar{x}\ \forall x$
- S-бокс не должен быть линейной или аффинной булевой функцией

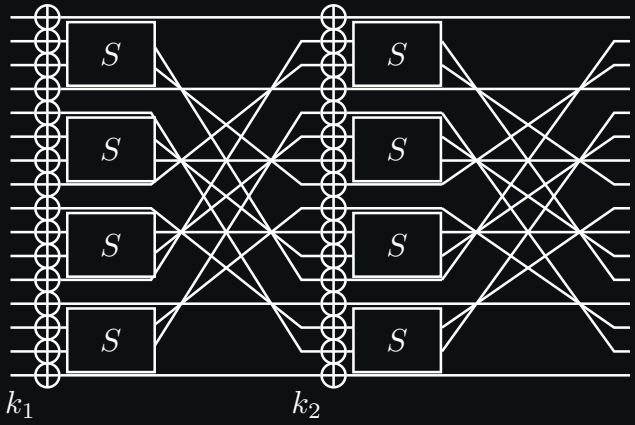
Пример: S-боксы в ГОСТе'89

$$S := \{0,1\}^4 \to \{0,1\}^4$$

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

Wikipedia, https://ru.wikipedia.org/wiki/%D0%93%D0%9E%D0%A1%D0%A2_28147-89

Подстановочно-перестановочная сеть(SPN)



AES: SPN шифр

- стандартизирован в 2001 году (FIPS PUB 197: Advanced Encryption Standard (AES), ISO/IEC 18033-3: Block ciphers)
- Длина блока $n = 128$ бит, длины ключей $k = \{128, 192, 256\}$
- Количество раундов: 10 ($k = 128$), 12 ($k = 192$), 14 ($k = 256$)
- 128 бит организованы в матрицу 4×4 байт

$$\begin{pmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{pmatrix}$$

Перестановка f_{AES}

Перестановка f_{AES} состоит из трёх обратимых операций:

1. `SubBytes` (единственная нелинейная операция)

$$S : \{0,1\}^8 \to \{0,1\}^n - \text{S-бокс}$$

2. `ShiftRows`

$$\begin{pmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{pmatrix} \to \begin{pmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_5 & b_9 & b_{13} & b_1 \\ b_{10} & b_{14} & b_2 & b_6 \\ b_{15} & b_3 & b_7 & b_{11} \end{pmatrix}$$

3. `MixColumns` – столбцы перемешиваются по определённому правилу (см. `en.wikipedia.org/wiki/Advanced_Encryption_Standard`)

Принцип: минимизировать успешность известных атак.

Расширение ключа в AES

Задача: расширить ключ $k \in \{0,1\}^{128}$ до 10 раундовых ключей $k_i \in \{0,1\}^{128}$

- $k_0 = k = (w_{0,0}, w_{0,1}, w_{0,2}, w_{0,3})$, $w_{0,1} \in \{0,1\}^{32}$
- $k_i = (w_{i,0}, w_{i,1}, w_{i,2}, w_{i,3})$, где

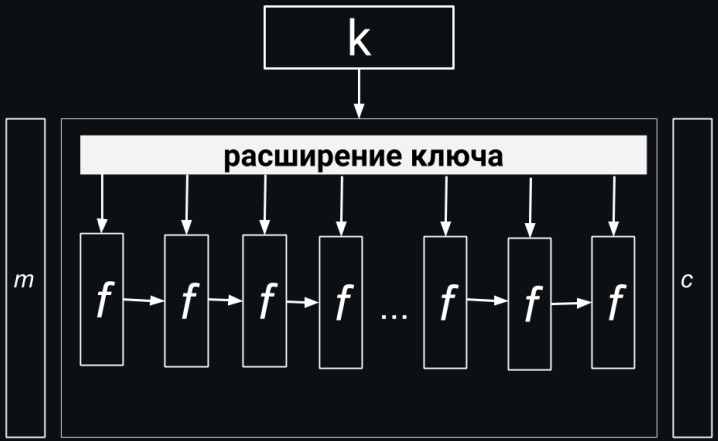
$$
\begin{aligned}
w_{i,0} &= w_{i-1,0} \oplus g_i(w_{i-1,3}) \\
w_{i,1} &= w_{i-1,1} \oplus w_{i,0} \\
w_{i,2} &= w_{i-1,2} \oplus w_{i,1} \\
w_{i,3} &= w_{i-1,3} \oplus w_{i,2}
\end{aligned}
$$

$g_i : \{0,1\}^{32} \to \{0,1\}^{32}$ – функция, состоящая из сдвигов, подстановки `SubBytes` и `XOR` с раундовыми константами c_i.

Часть II

Атаки на блочные шифры

Блок-шифр



Алгоритм перебора

Идея: перебор ключа $k \in \{0,1\}^\kappa$

Для DES/AES/GOST: достаточно двух пар (открытый текст, шифр-текст) $(m_1, c_1 = \mathsf{Enc}(k, m_1)), (m_2, c_2 = \mathsf{Enc}(k, m_2))$, чтобы определить k с большой вероятностью.

Сложность: $\mathcal{O}(2^\kappa)$

Пример: DES $k \in \{0,1\}^{56}$:

- **'99-е** 22 часа на DeepCrack: дорогое железо+ распределенная сеть
- **'07-е** 13 дней COPACOBANA: FPGA, дешевле

Атаки на дизайн

- Линейный криптанализ:
  аппроксимация S-box линейной функцией
- дифференциальный криптоанализ

Атаки на реализацию

- Атаки по сторонним каналам (side-channel attacks):
  замер **времени** или **мощности**, используемых в процессе $\mathrm{Enc}, \mathrm{Dec}$

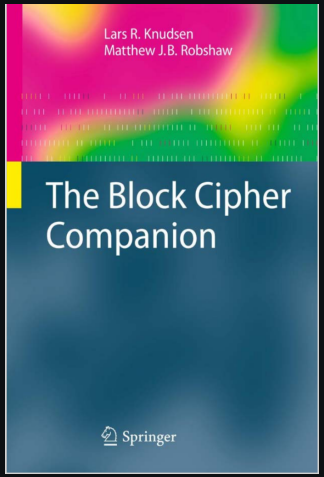
  Эти величины не должны зависеть от секретного ключа.

- Внесение неисправностей (Fault-injection attacks)

  внешние воздействия на устройство, порождения аппаратных ошибок (нагрев, ЭМ волны)

1. **Не** изобретайте **свой собственный** блок-шифр
2. **Используйте** реализации блок-шифров из проверенных временем библиотек

Что почитать



Часть III

Режим шифрования (modes of operation)

Как правильно использовать блочный шифр для шифрования сообщений?

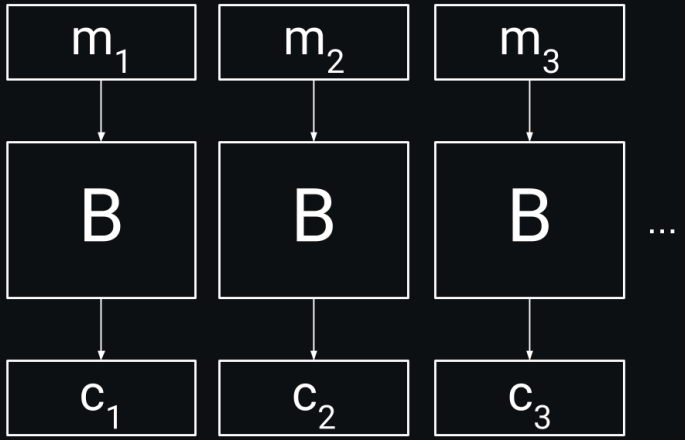
Режимы шифрования

1. Режим электронной кодовой книги или режим простой замены (Electronic Block Code, EBC)
2. Режим сцепления блоков шифротекста (Cipher Block Chain, CBC)
3. Режим счётчика (Counter mode, CTR)

Электронная кодовая книга, Electronic Block Code (EBC)

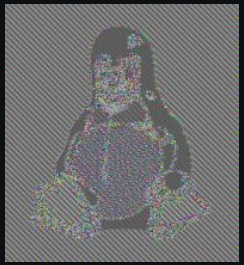
Пусть $m = (m_1, m_2, m_3, ...), m_i \in \{0,1\}^n$ – открытый текст.

Наивный способ использования блочного шифра B



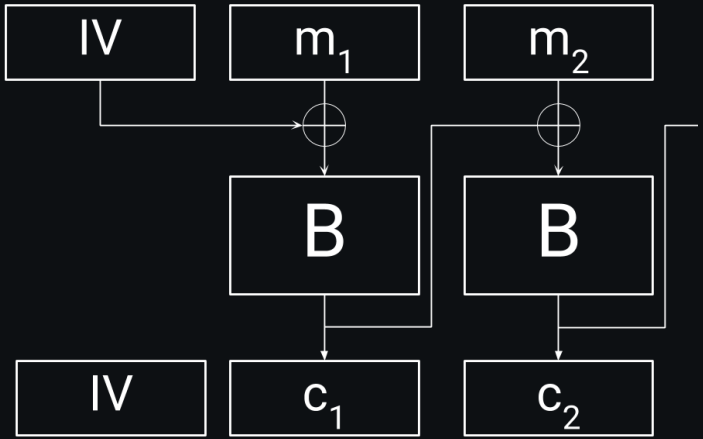
Это небезопасный метод: если $m_1 = m_3$, то $c_1 = c_3$

Если $m_1 = m_2$, то $c_1 = c_2$



Режим сцепления блоков, Cipher Block Chain (CBC)

IV – инициализирующий (начальный) вектор – случайная строка n-бит



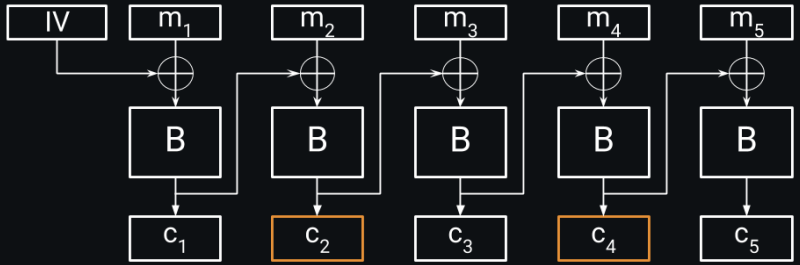
IV передается с шифр-текстом (публично известно).

Безопасность режима сцепления блоков

- Начальное значение IV должно быть случайным (если атакующий может предсказать IV, шифрование CBC небезопасно).
  См. атаку на TLS 1.1.
- IV необходимо обновлять

Безопасность режима сцепления блоков

Положим, мы используем одно и тоже IV для длинного сообщения $m = (m_1, \ldots m_t)$ при $t > 2^{n/2}$.



Парадокс Дней Рождений: имея $2^{n/2}$ блоков шифр-текста c_i, с большой вероятностью мы увидим два одинаковых c_i.

$$c_1 \oplus m_2 == c_3 \oplus m_4$$

Далее применяются статистические атаки на m.

Парадокс Дней Рождений

Определить вероятность того, что в комнате из 30 человек двое родились в один день.

Парадокс Дней Рождений

Определить вероятность того, что в комнате из 30 человек двое родились в один день.

$$\left(1-\frac{1}{365}\right)\left(1-\frac{2}{365}\right)\left(1-\frac{3}{365}\right)\cdot\ldots\cdot\left(1-\frac{29}{365}\right)\approx 0.294$$

Значит, с вероятностью $1-0.294>0.7$ найдутся двое таких людей.

Обобщение Парадокса Дней Рождений

Для m человек и N возможных дней рождений, вероятность того, что все m человек имеют разные дни рождения:

$$P := \prod_{i=1}^{m-1} \left(1 - \frac{i}{N}\right) \approx e^{-m^2/2N}$$

Для $m = \sqrt{2N \ln 2}$, $P \approx 1/2$. Вероятность P быстро увеличивается при росте m.

Обобщение Парадокса Дней Рождений

Для m человек и N возможных дней рождений, вероятность того, что все m человек имеют разные дни рождения:

$$P := \prod_{i=1}^{m-1} \left(1 - \frac{i}{N}\right) \approx e^{-m^2/2N}$$

Для $m = \sqrt{2N \ln 2}$, $P \approx 1/2$. Вероятность P быстро увеличивается при росте m.

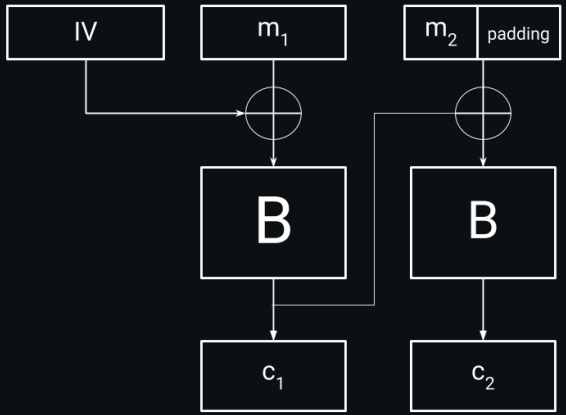
Для блок-шифра с длиной блока n, имеем 2^n всевозможных блоков шифр-текстов.
При $m = \mathcal{O}(2^{n/2})$ шифр-блоков c_i's, некоторые два из них равны с константной вероятностью.

Для режима CBC: $c_i == c_j$ при $m = (m_1, \ldots, m_t)$, $t \approx 2^{n/2}$:

$$c_{i-1} \oplus m_i == c_{j-1} \oplus m_j$$

Набивка (Padding) для режима CBC

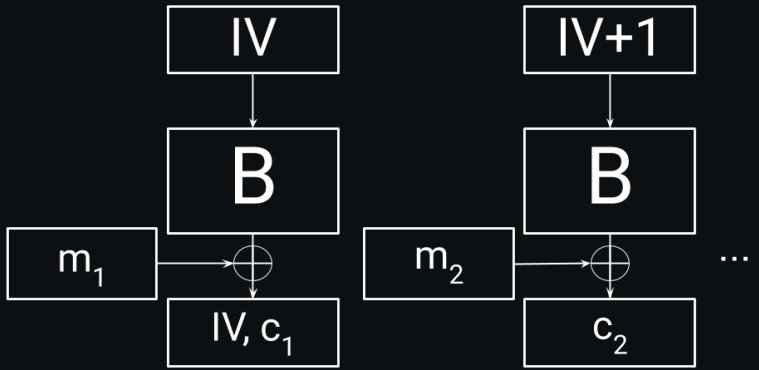
CBC подразумевает, что все блоки m_i фиксированной длины. Для этого используется “набивка”.



Обычно ℓ-байтная набивка состоит из ℓ копий of ℓ.
Набивка из 5 байт: $5|5|5|5|5$.
Если m занимает меньше n-бит, добавляется фиктивный (dummy) блок.

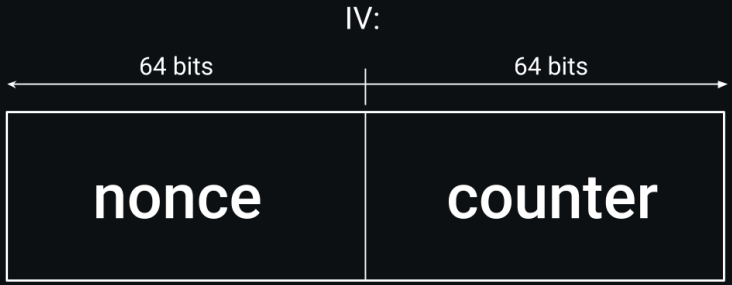
Режим счетчика, Counter Mode (CTR)

Один из самых популярных режимов шифрования
Здесь IV - начальное значение счетчика.
Счетчик увеличивается для каждого нового блока.



CTR создает потоковое шифрование из блок-шифра

Как выглядит IV



- Nonce (нонс) должен быть псевдослучайным (64-битный выход PRG) и не должен повторяться для одного и того же ключа k
- Счетчик увеличивается для каждого нового блока
- Значение счетчика не передается в протоколах, обеспечивающих последовательную доставку пакетов (https)
- Нонс обновляется после 2^{64} зашифрованных блоков.

Преимущества режима Counter Mode