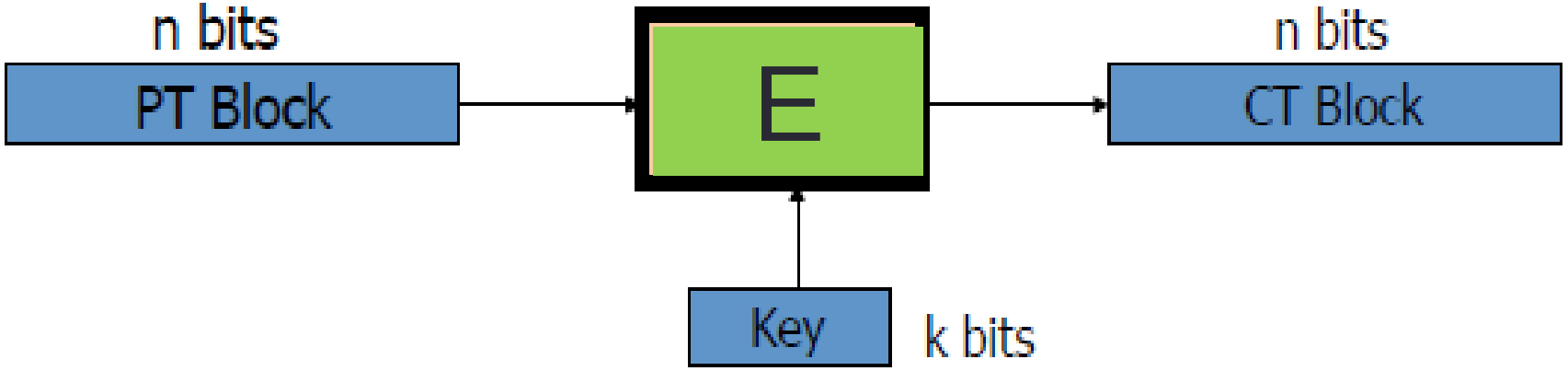
СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

Блоковые шифры

Блоковые шифры



Примеры:

DES n = 64 бита, k = 56 бит

AES n = 128 бит, k = 128, 192 или 256 бит

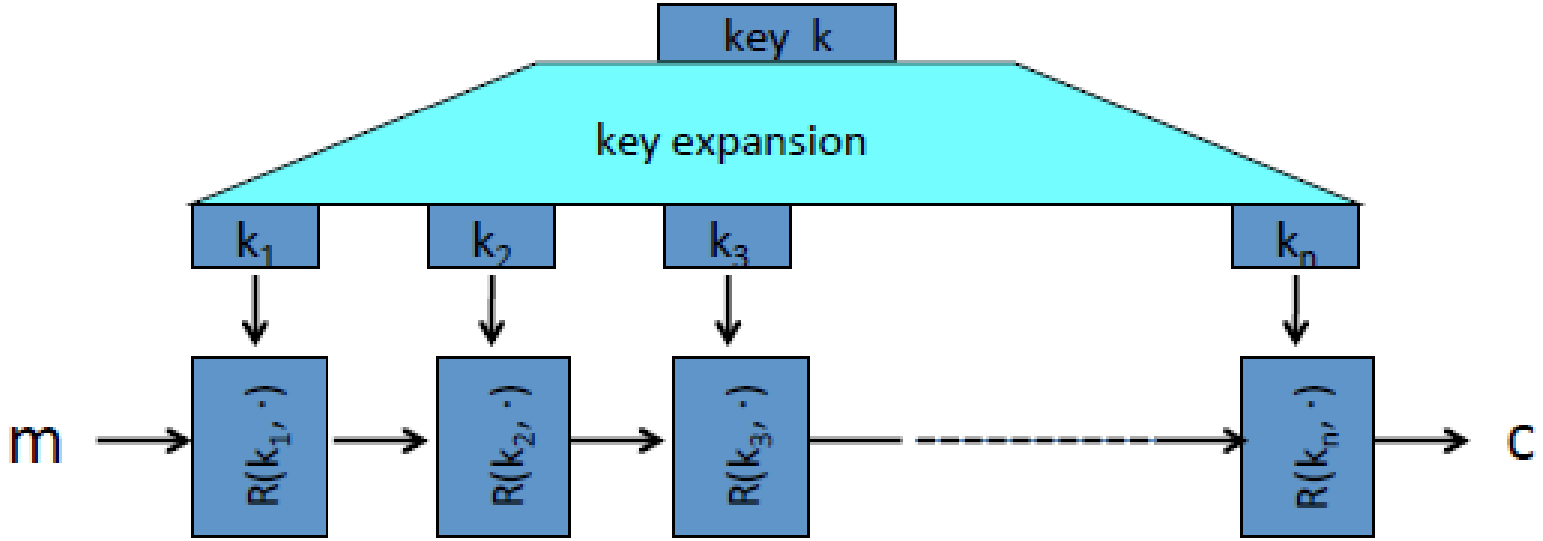
Желаемое

- Я хочу зашифровать блок длины n бит
- Решение: указать в качестве ключа номер перестановки всех возможных блоков
- Каков будет размер ключа?
- Всего n-битных блоков 2^n, тогда количество перестановок на этих блоках -- $2^n!$
- Длина ключа $\log_2(2^n!) \approx n2^n - \frac{1}{\ln 2}2^n + \frac{n}{2}$ бит

Действительное

- Случайную замену всего блока указать сложно
- Шифр собирается из подстановок и перестановок меньших размеров
- Это композиция более простых перестановок и замен
- Например: блок 64 бита можно разбить на куски по 8 бит
- Если все подстановки и перестановки делать над 8 битами – легко взломать
- Объединить в раунды.

Блоковые шифры работают итеративно



- R(k, m) – один раунд шифрования
- Количество раундов определяет стойкость

Блоковые шифры значительно медленнее потоковых

AMD Opteron, 2.2 GHz (Linux)

	Cipher	Block/key size	Speed (MB/sec)
stream	RC4		126
	Salsa20/12		643
	Sosemanuk		727
block	3DES	64/168	13
	AES-128	128/128	109

Понятия PRF и PRP

- Псевдослучайная функция (PRF):
  - определена над (K, X, Y)

$$F: K \times X \to Y$$

  - Существует «эффективный» алгоритм вычисления *F(k,x)*

- Псевдослучайная перестановка (PRP):
  - определена над (K, X)

$$E: K \times X \to X$$

  - существует «эффективный» детерминированный алгоритм вычисления *E(k,x)*
  - функция *E(k,*)* является отображением «один-к-одному»
  - существует «эффективный» алгоритм вычисления обратной функции *D(k,y)*

Примеры

- Функции шифрования DES, AES являются PRP:

$$AES: K \times X \to X \text{ где } K = X = \{0,1\}^{128}$$

$$DES: K \times X \to X \text{ где } K = \{0,1\}^{56}, X = \{0,1\}^{64}$$

- Любая PRP является PRF:
  - PRP – это PRF, у которой X=Y и определена обратная функция

Стойкость PRF

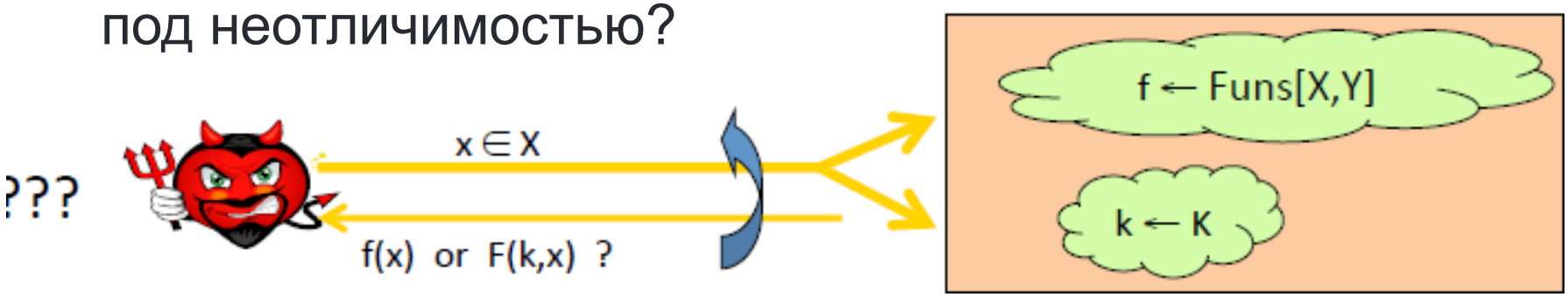
- Рассмотрим псевдослучайную функцию $F: K \times X \to Y$

- Funs[X,Y]: Множество **всех** отображений X в Y

- $S_F = \{ F(k,\cdot) \text{ s.t. } k \in K \} \subseteq$ Funs[X,Y]

Псевдослучайная функция F является стойкой, если случайная функция из множества **Funs[X,Y]** неотличима от случайной функции из S_F

Что мы будем понимать
под неотличимостью?



- Пусть $F: K \times X \to \{0,1\}^{128}$ стойкая псевдослучайная функция, тогда рассмотрим функцию G

$$G(k,x) = \begin{cases} 0^{128}, \text{если } x = 0 \\ F(k,x), \text{если } x \neq 0 \end{cases}$$

- Будет ли функция G(k,x) стойкой?
  - Нет, так как легко отличить *G(k,x)* от случайной функции
  - Да, так как успешная атака на *G(k,x)* будет приводить к вскрытию *F(k,x)*
  - Результат зависит от типа функции *F(k,x)*

Возможные приложения: PRF ⇒PRG

- Пусть $F: K \times \{0,1\}^n \rightarrow \{0,1\}^n$ стойкая псевдослучайная функция, тогда на ее основы можно построить стойкий псевдослучайный генератор $G: K \rightarrow \{0,1\}^{nt}$

$$G(k) = F(k,0) \,||\, F(k,1) \,||\, \cdots \,||\, F(k,t-1)$$

- Полезные свойства: распараллеливание
- Стойкость PRG следует непосредственно из стойкости PRF

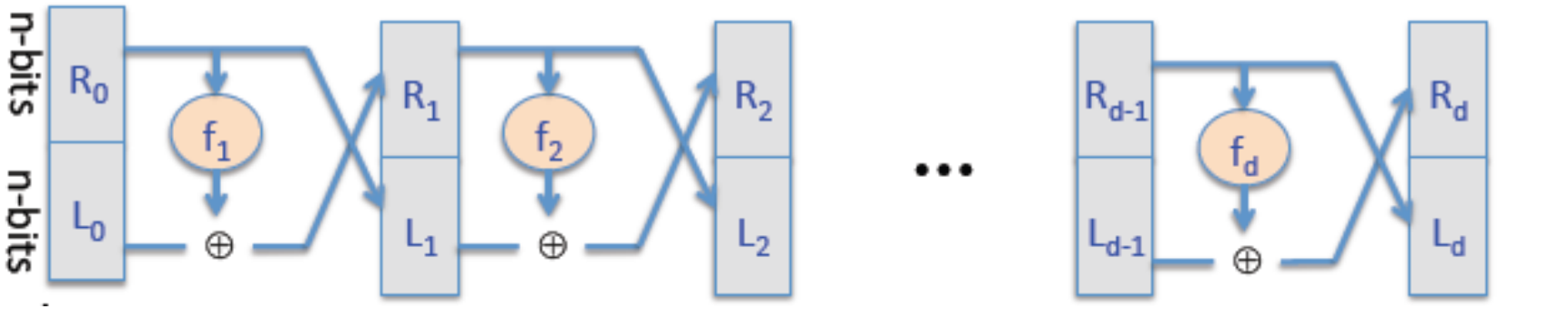
История DES

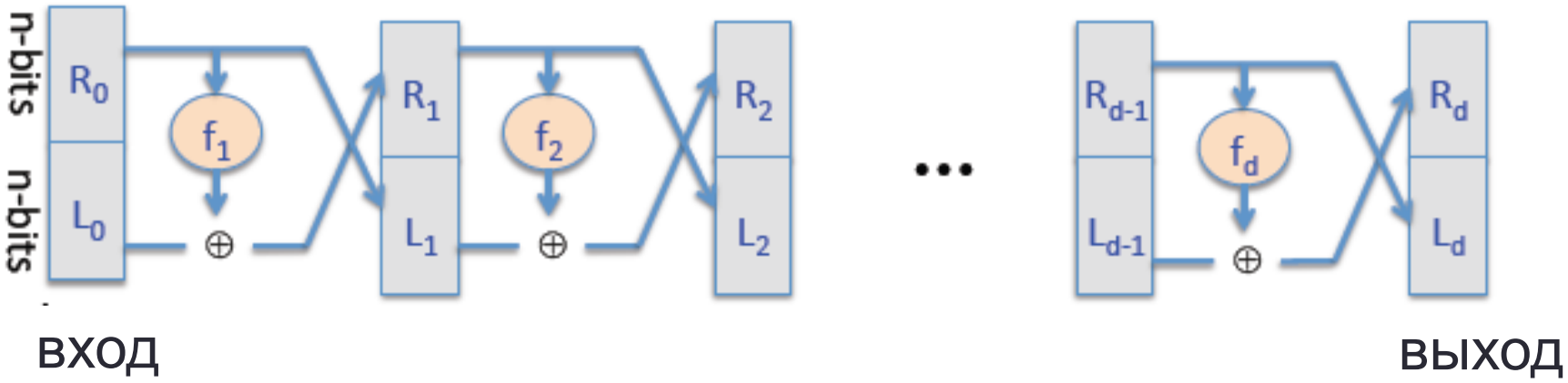
- Был разработан в начале 1970-х, принят как стандарт в 1976 (длина блока 64 бита, длина ключа 56 бит).
- Разрабатывало IBM, но NSA (National Security Agency) принимало активное участие, меняло протоколы.
- Думали, что особенности DES позволяли NSA его дешифровывать.
- На самом деле функции выбирались так, чтобы их было труднее взламывать разностным криптоанализом (dierential cryptoanalysis). Это метод для взлома блочных шифров; Eli Biham, Adi Shamir, конец 1980-х; но, видимо, IBM и NSA знали об этих атаках еще в 1970-х (Don Coppersmith, 1994).
- В 1997 году взломан полным перебором

DES: ключевая идея

- Сеть Фейстеля
- Даны функции $f_1, f_2, \ldots, f_d: \{0,1\}^n \to \{0,1\}^n$
- Цель: построить обратимую функцию

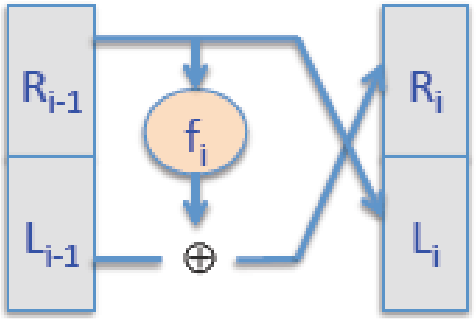
  $F: \{0,1\}^{2n} \to \{0,1\}^{2n}$



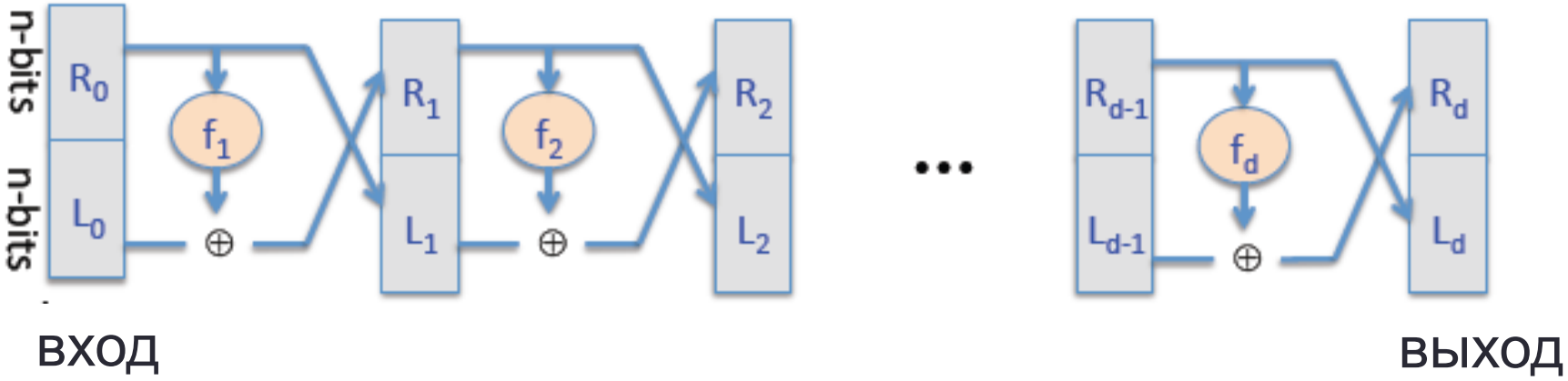


- Докажем, что для любых $f_1, f_2, \ldots, f_d: \{0,1\}^n \to \{0,1\}^n$

  сеть Фейстеля $\mathrm{F}: \{0,1\}^{2n} \to \{0,1\}^{2n}$ обратима
- Док-во: Построим обратную функцию



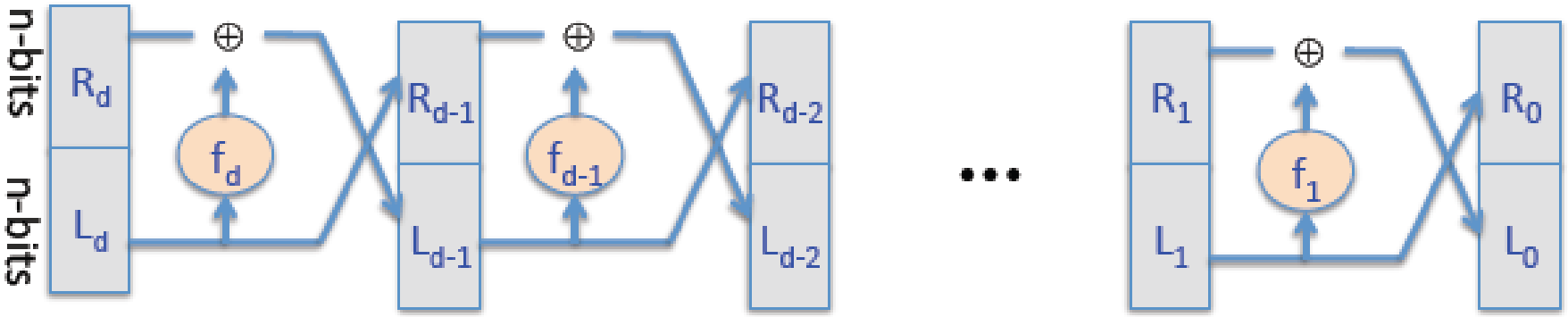
$$R_{i-1} = L_i$$
$$L_{i-1} = ?$$



- Докажем, что для любых $f_1, f_2, \dots, f_d: \{0,1\}^n \to \{0,1\}^n$

  сеть Фейстеля $\mathrm{F}: \{0,1\}^{2n} \to \{0,1\}^{2n}$ обратима
- Док-во: Построим обратную функцию

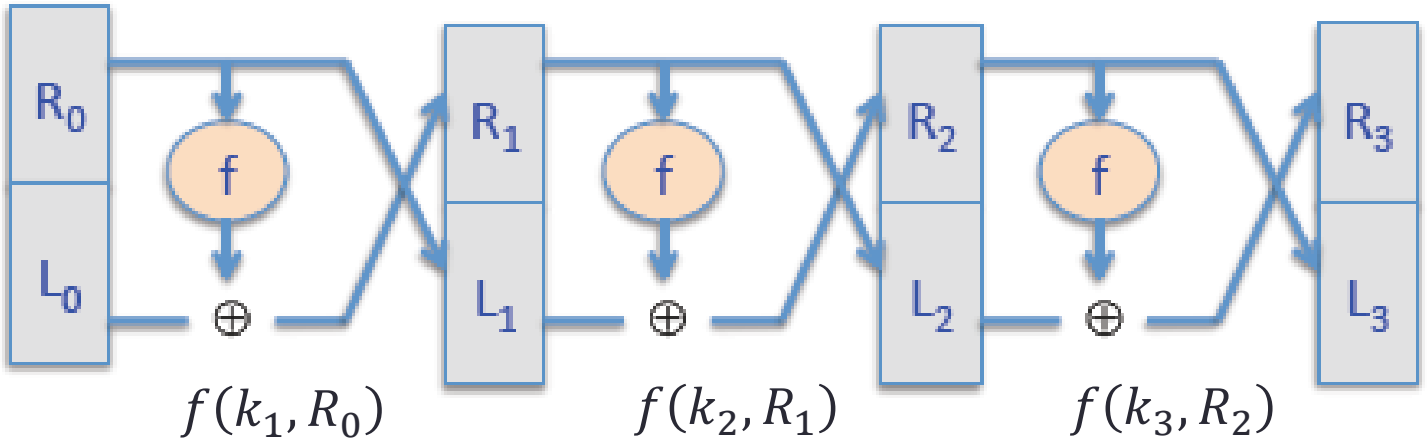
Дешифрующая сеть



- Обратное преобразование состоит в применении тех же функций $f_1, f_2, \ldots, f_d: \{0,1\}^n \to \{0,1\}^n$, только в обратном порядке
- Основной способ построения обратимой функции из набора произвольных
- Широко распространен в конструкции блоковых шифров

- Теорема (Luby-Rackoff'85):

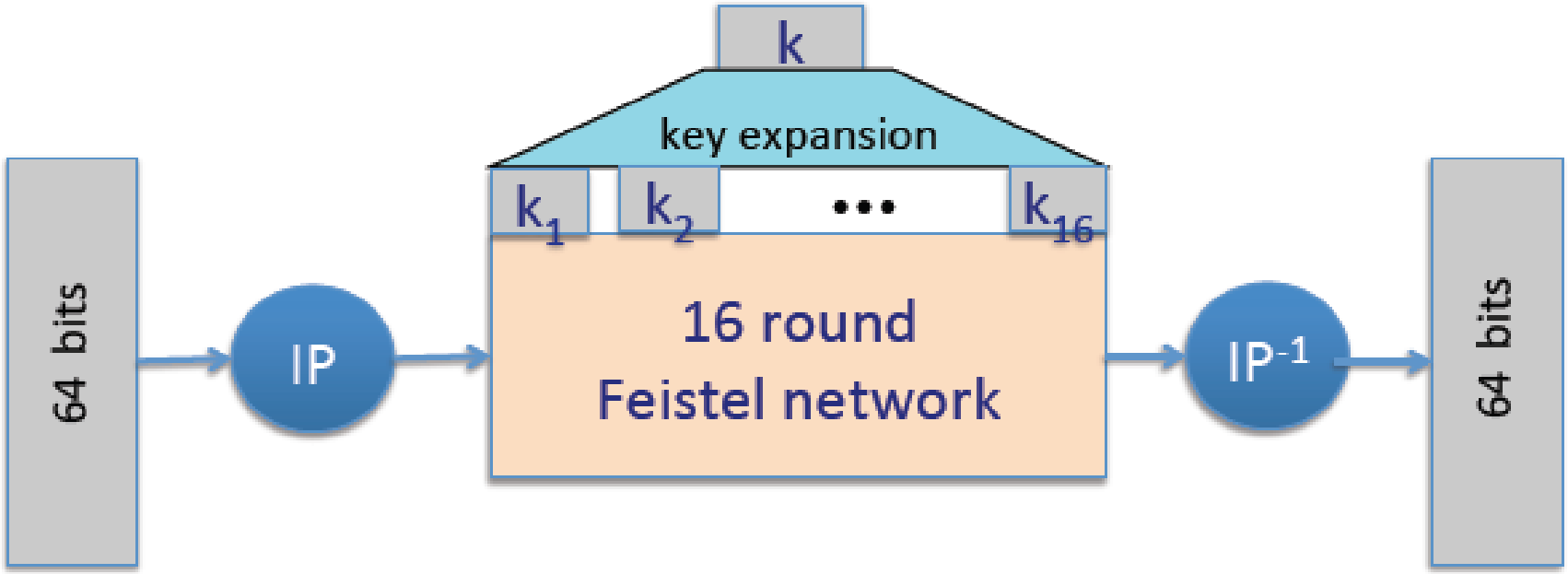
- Если $f: K \times \{0,1\}^n \to \{0,1\}^n$ стойкая PRF, то
- 3-х раундовая сеть Фейстеля $\mathrm{F}: K^3 \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ стойкая PRP



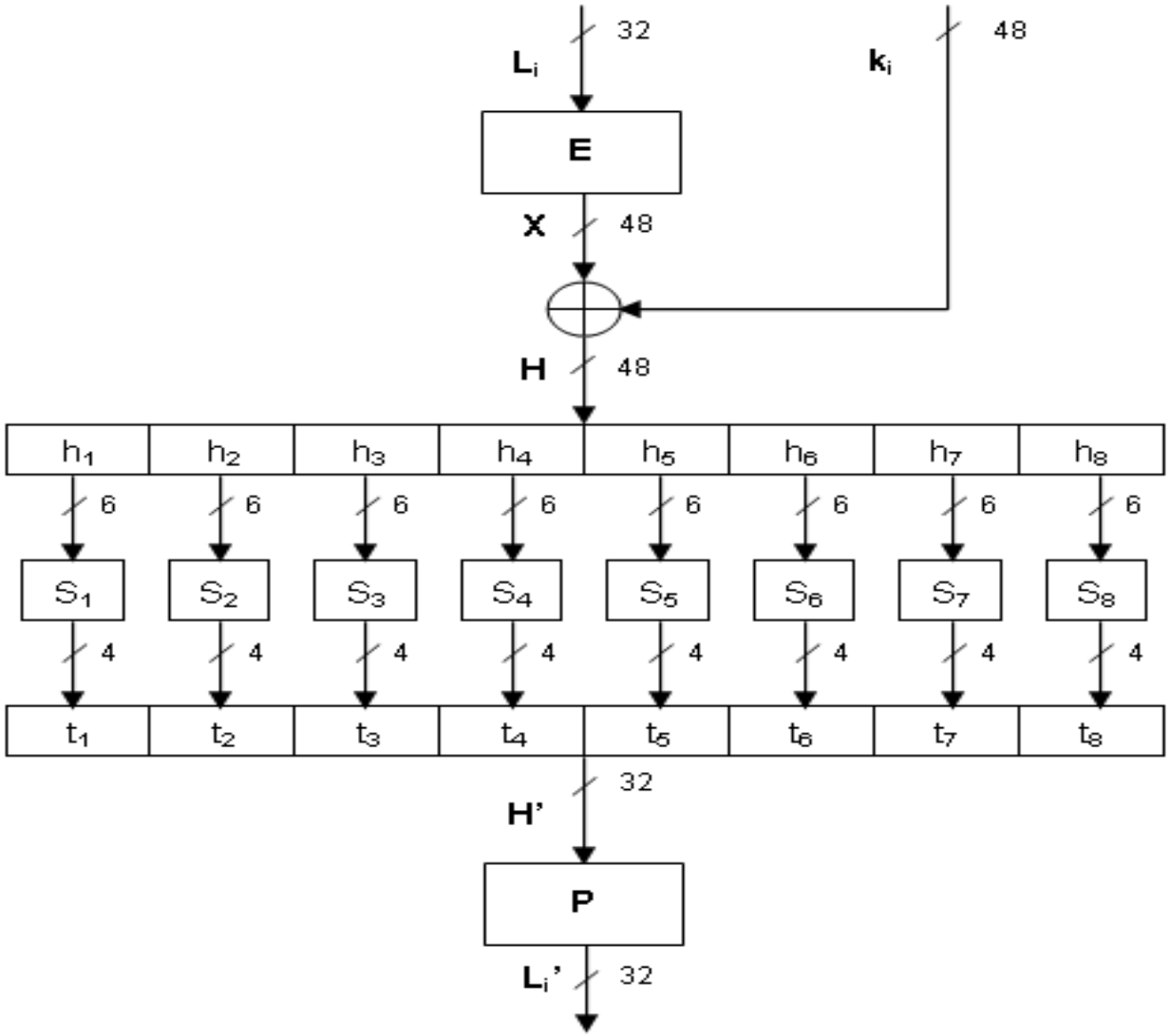
Нужны 3 независимых ключа!

DES 16-раундовая сеть Фейстеля

- функции $f_1, f_2, \ldots, f_{16}: \{0,1\}^{32} \to \{0,1\}^{32}, \quad f_i(x) = F(k_i, x)$
- для дешифрования ключи используются в обратном порядке



Конструкция функции $F(k_i,x)$



S-box

Каждый блок S представляет собой табличную сжимающую подстановку

$$S_i : \{0,1\}^6 \rightarrow \{0,1\}^4$$

S_5		Middle 4 bits of input															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Пример плохого выбора S-box

- Предположим

$$S_i(x_1, x_2, \ldots, x_6) = (x_2 \oplus x_3,\ \ x_1 \oplus x_4 \oplus x_5,\ \ x_1 \oplus x_6,\ \ x_2 \oplus x_3 \oplus x_6)$$

- Тогда можно записать $S_i(\mathbf{x}) = A_i \cdot \mathbf{x} \pmod 2$

$$\begin{bmatrix} 0&1&1&0&0&0 \\ 1&0&0&1&1&0 \\ 1&0&0&0&0&1 \\ 0&1&1&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x_2 \oplus x_3 \\ x_1 \oplus x_4 \oplus x_5 \\ x_1 \oplus x_6 \\ x_2 \oplus x_3 \oplus x_6 \end{bmatrix}$$

В таком случае S_i линейная функция

- В таком случае вся функция шифрования будет линейной

$$DES(k,m) = \underset{64}{\overset{832}{B}} \cdot \begin{matrix} m \\ k_1 \\ k_2 \\ \vdots \\ k_{16} \end{matrix} = c \pmod 2$$

Тогда

$$DES(k,m_1) \oplus DES(k,m_2) \oplus DES(k,m_3) = DES(k, m_1 \oplus m_2 \oplus m_3)$$

$$B\begin{matrix} m_1 \\ k \end{matrix} \oplus B\begin{matrix} m_2 \\ k \end{matrix} \oplus B\begin{matrix} m_3 \\ k \end{matrix} = B\begin{matrix} m_1 \oplus m_2 \oplus m_3 \\ k \oplus k \oplus k \end{matrix}$$

Принципы построения S-box

- Основным требованием к S-box является нелинейность, но этого недостаточно
- В 1989 годы было доказано, что нельзя выбирать таблицы случайным образом – это сокращает перебор до 2^{24}
- Принципы построения S -боксов:
  - каждая строка S -бокса – некоторая перестановка чисел {0;1;2;...;15}
  - S -боксы не могут быть линейными или аффинными преобразованиями входных данных;
  - изменение одного бита входных данных должно на выходе из S -бокса изменить хотя бы 2 бита;
  - если на вход S -бокса поступил вектор x , а потом вектор $y = x \oplus (0,0,1,1,0,0)$, то векторы S(x) и S(y) на выходе должны отличаться хотя бы 2 битами.