3. Блочные шифры. Формальные модели

Пусть $\mathbb{F}_2 = \{0,1\}$ – алфавит двоичных цифр (битов), а

$$\mathbb{F}_2^n = \{(b_0, b_1, \dots, b_{n-1}) \mid b_i \in \mathbb{F}_2,\ i = 1, 2, \dots, n\}$$

– множество n-буквенных слов (n-битовых блоков) в этом алфавите. *Блочным шифром* называется любое отображение $\mathcal{E}: \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$, которое каждому блоку $P \in \mathbb{F}_2^n$ открытого текста сопоставляет некоторый блок $C = \mathcal{E}(P) \in \mathbb{F}_2^n$ шифртекста. При этом отображение $\mathcal{E}$ должно быть взаимно однозначным, поскольку обычно требуется, чтобы для произвольного блока C шифртекста можно было бы всегда восстановить исходный блок P открытого текста.
Всего имеется

$$\nu_n = (2^n)! \approx \sqrt{\pi 2^{n+1}} \left(\frac{2^n}{e}\right)^{2^n} e^{\frac{\theta}{12 \cdot 2^n}},\ 0 < \theta < 1,\ (\text{при } n \longrightarrow \infty)$$

взаимно однозначных отображений множества $\mathbb{F}_2^n$ на себя. Перенумеруем указанные отображения, используя тот или иной способ нумерации. Номера соответствующих отображений будем называть *ключами* шифрования. Тогда, выбирая тот или иной ключ, мы можем задать вполне определенное отображение (функцию шифрования). Для задания всех функций шифрования необходимо, чтобы длина ключа (в битах) была равна, по меньшей мере,

$$m = log_2\, \nu_n \sim n2^n.$$

Такие рассуждения являются доводом в пользу того, что ключи шифрования должны быть достаточно длинными. На практике используются, конечно, более короткие ключи, но m-битовые ключи $K \in \mathbb{F}_2^{\boldsymbol{m}}$ позволяют охватить только 2^m (не более) шифрующих отображений. Отображение с ключом K будем обозначать $\mathcal{E}_k$, а обратное отображение – $\mathcal{D}_k$ или $\mathcal{E}_k^{-1}$.

Формально понятие блочного шифра можно определить следующим образом. Рассмотрим пятерку (P, C, K, $\mathcal{E}$, $\mathcal{D}$), где P и C – множества соответственно преобразуемых (открытых) и преобразованных (зашифрованных) блоков данных фиксированных длин, K – множество (пространство) ключей, а $\mathcal{E}$ и $\mathcal{D}$ – соответственно прямое и обратное отображения:

$\mathcal{E} : P \times K \to C$,

$\mathcal{D} : C \times K \to P$,

причем для любых $c \in C$ и $k \in K$ уравнение $\mathcal{E}_k(p) = c$ разрешимо относительно $p \in P$ и $\mathcal{D}_k\,(\mathcal{E}_k(p)) = p$ (здесь $\mathcal{E}_k\,(p) = \mathcal{E}(p,\,k)$ и $\mathcal{D}_k(c) = D(c,\,k)$ – сужения отображений $\mathcal{E}$ и D на множестве $P \times \{k\}$ и $C \times \{k\}$). Пятёрка (P, C, K, $\mathcal{E}$, $\mathcal{D}$) называется *блочным алгоритмом преобразования данных* или просто *блочным шифром.* С практической точки зрения наибольший интерес представляет случай $P = C = \mathbb{F}_2^n$, $K = \mathbb{F}_2^m$. Поскольку $\mathcal{E}_k$ и $\mathcal{D}_k$ – взаимно обратные отображения на множестве $\mathbb{F}_{\boldsymbol{2}}^{\boldsymbol{n}}$, то в этом случае блочный шифр однозначно определяется тройкой $(\mathbb{F}_2^n, \mathbb{F}_2^m, \mathcal{E})$.

В отображении $\mathcal{E}_k: \mathbb{F}_2^n \times \mathbb{F}_2^m \longrightarrow \mathbb{F}_2^n$ по сути заключено целое семейство (а именно 2^m) отображений $\mathcal{E}_k: \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$, получаемых из $\mathcal{E}$ путем выбора конкретного ключа $k \in \mathbb{F}_{\boldsymbol{2}}^{\boldsymbol{m}}$. Значение m (длина ключа) должно быть достаточно большим, чтобы исключить попытку полного (тотального) перебора ключей k и соответственно отображений $\mathcal{E}_k$. Рекомендуемые значения $m = 128$, 196 или 256 исключают возможность указанного перебора. Отображения $\mathcal{E}_k$ и $\mathcal{D}_k$ будем называть соответственно *функциями зашифрования и расшифрования.*

Конкретную функцию зашифрования $\mathcal{E}_k: \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$ можно представить в виде 2^m-строчной таблицы (так называемой *электронной кодовой книги*):

$$\begin{array}{cc} & \cdots \\ x_0 x_1\, \dots x_{n-1} & y_0 y_1\, \cdots y_{n-1} \\ & \cdots \end{array}$$

,

в которой каждому блоку $X = x_0 x_1\, \dots x_{n-1} \in \mathbb{F}_2^n$ открытого текста сопоставлен соответствующий блок $Y = \mathcal{E}_k(X) = y_0 y_1\, \dots y_{n-1} \in \mathbb{F}_2^n$ шифртекста. Другой способ заключается в использовании формул алгебры логики: значение y_i (i-го бита шифртекста, $i = 0, 1, \dots, n$-1) может быть выражено в виде булевой функции $f_i\,(x_0, x_1, \dots, x_{n-1})$ от битовых переменных x_i, образующих блок $X = x_0 x_1\, \dots x_{n-1}$ открытого текста (функция f_i при этом может быть представлена либо в виде СДНФ – совершенной дизъюнктивной нормальной формы, либо в виде полинома Жегалкина). Указанные способы задания функции шифрования являются наиболее общими, но их практическое применение возможно лишь при относительно небольших n: если n велико, то таблица, задающая $\mathcal{E}_k$, становится чрезмерно большой (содержащей $n2^n$ бит, причем следует иметь в виду, что каждому ключу будет соответствовать отдельная шифровальная книга), а каждая из булевых функций f_i будет иметь, как правило, большую сложность (порядка 2^n по числу членов, входящих в формулу, реализующую f_i). На практике, как правило, используется алгоритмический способ задания функций шифрования. Шифры, задаваемые в виде алгоритма, составляют лишь небольшую долю возможных отображений, но зато они описываются компактно, и именно такие шифры могут быть использованы на практике.

Криптографическое преобразование $\mathcal{E}_k$ обычно конструируется как произведение (композиция или суперпозиция) некоторого числа достаточно простых для задания и реализации преобразований $\mathcal{E}_k^{(i)}$, $i = 1, 2, \ldots, r$, т.е.

$$\mathcal{E}_k = \mathcal{E}_k^{(r)} \circ \mathcal{E}_k^{(r-1)} \circ \ldots \circ \mathcal{E}_k^{(1)},$$

где произведение $f \circ g$ отображений f и g определяется условием: $f \circ g(u) = f(g(u))$, $\forall u \in \mathbb{F}_2^n$. Построенные таким способом шифры обычно называют *композиционными.* Лежащая в их основе идея состоит в том, что сложное криптографическое преобразование может быть построено путем многократного применения относительно простых криптографических преобразований. К. Шеннон, теоретически обосновавший такой принцип конструирования блочных шифров, предложил использовать в качестве простых преобразований операции подстановки ($substitution$) и перестановки ($permutation$). Схемы, реализующие эти преобразования, получили название *подстановочно-перестановочных сетей* ($SP - networks$).

Многократное использование этих преобразований позволяет обеспечить следующие свойства, которые должны быть присущи стойким шифрам: *рассеивание* ($diffusion$) и *перемешивание* ($confusion$).

Рассеивание – это свойство шифра, заключающееся в распространении влияния одного бита (знака) открытого текста или ключа на значительное количество битов шифртекста. Наличие такого свойства:

- позволяет скрыть статистическую зависимость между битами (знаками) открытого текста (т.е. маскирует статистические свойства исходного текста);
- не позволяет криптоаналитику противника восстанавливать неизвестный ему ключ по частям.

К числу операций, обладающих свойством рассеивания, относится, например, обычная перестановка битов открытого текста.

Перемешивание − это свойство шифра маскировать взаимосвязи статистических и аналитических свойств открытого и шифрованного текстов. Простой метод создания перемешивания – подстановка (например, замена подслов открытого текста другими).

Один из способов достижения хорошего рассеивания и перемешивания − построение композиционного шифра, в котором последовательно применяются подстановки (S), перестановки (P) и линейные преобразования (L). Результирующее преобразование в этом случае выглядит следующим образом:

$$\mathcal{E} = P_r \circ L_r \circ S_r \circ \ldots \circ P_2 \circ L_2 \circ S_2 \circ P_1 \circ L_1 \circ S_1.$$

Ключ k, определяющий разнообразие получающихся при этом отображений, может использоваться при определении какого-либо одного типа преобразований (P, L или S), или сразу во всех типах.

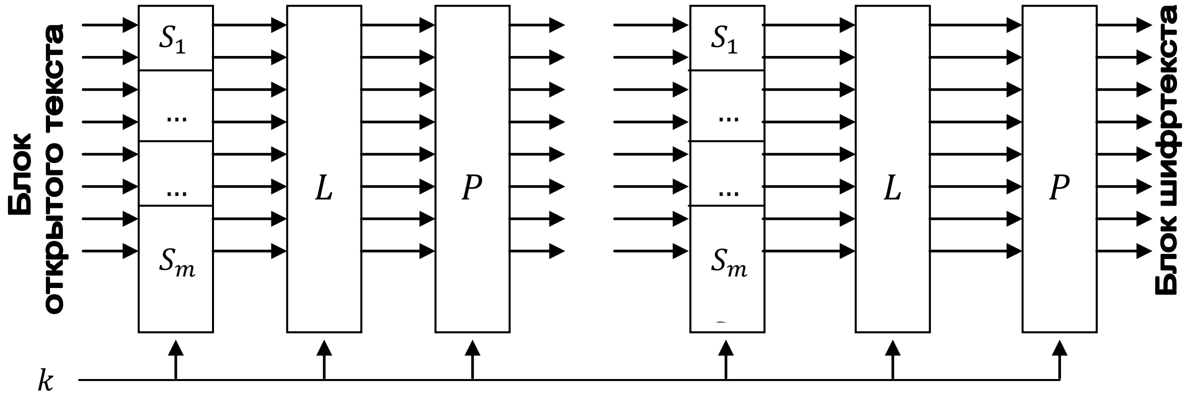

Рис. 2.1. *Структура композиционного шифра на основе подстановок, перестановок и линейных преобразований*

Процедура зашифрования для композиционного шифра $\mathcal{E}_k$ осуществляется по схеме:

Вход: Блок $B \in \mathbb{F}_2^n$ открытого текста.

$\boldsymbol{for}\ i := 1\ \boldsymbol{to}\ r\ \boldsymbol{do}\ B := \mathcal{E}^{(i)}(B, k_i)$

Выход: Блок $B \in \mathbb{F}_2^n$ шифртекста.

Дадим необходимые пояснения к приведенной схеме. Преобразование $B := \mathcal{E}^{(i)}(B, k_i)$ называется *раундом* (или *циклом*) *зашифрования,* а $\mathcal{E}_k^{(i)}(B,\ k_i)$ – *раундовой* (или *цикловой*) *функцией зашифрования.* Отметим, что базовый ключ k в каждом раунде, вообще говоря, целиком не применяется, а используются *раундовые ключи* $k_1, k_2, \ldots, k_r$, которые формируются на основе базового ключа k на этапе предвычислений. Объединение раундовых ключей $Q = k_1 \| k_2 \| \ldots \| k_r$ обычно называют *расширенным ключом.* Базовый ключ k является секретным элементом криптосистемы и называется *секретным* (или *основным*) ключом. Обратное преобразование (расшифрование) $\mathcal{D}_k = \mathcal{E}_k^{-1}$ для композиционного шифра выполняется по схеме:

$\boldsymbol{for}\ i := r\ \boldsymbol{downto}\ 1\ \boldsymbol{do}\ B := \mathcal{D}^{(i)}(B, k_i),$

где $\mathcal{D}^{(i)} = \left(\mathcal{E}_k^{(i)}\right)^{-1}$.

Если во всех раундах шифрования используется одна и та же раундовая функция, т.е.

$$\mathcal{E}_k^{(1)} = \mathcal{E}_k^{(2)} = \cdots = \mathcal{E}_k^{(r)} = \mathcal{E}_k,$$

то такой композиционный шифр называется *итеративным.*

Схема (*сеть*) *Фейстеля* ($H. Feistel$) является разновидностью итеративного шифра. При шифровании блок открытого текста разбивается на две равные части – левую (L) и правую (R). В каждом раунде одна из частей преобразуется при помощи функции F и раундового ключа k_i. Результат операции суммируется покомпонентно по модулю 2 (операция xor) с другой частью. Затем левая и правая части меняются местами (исключая последний раунд). Схема Фейстеля представлена на рис. 2.2. Эта схема примечательна тем, что для зашифрования и расшифрования может быть использован один и тот же алгоритм с той лишь разницей, что при расшифровании раундовые ключи используются в обратном порядке. Отметим, что преобразование, осуществляемое в схеме Фейстеля, является обратимым и позволяет восстановить входные данные функции F на каждом раунде. Сама же функция F при этом не обязательно должна быть обратимой. Это важное свойство, поскольку оно освобождает разработчика от необходимости выбора только таких преобразований, для которых обратные преобразования имеют невысокую сложность.

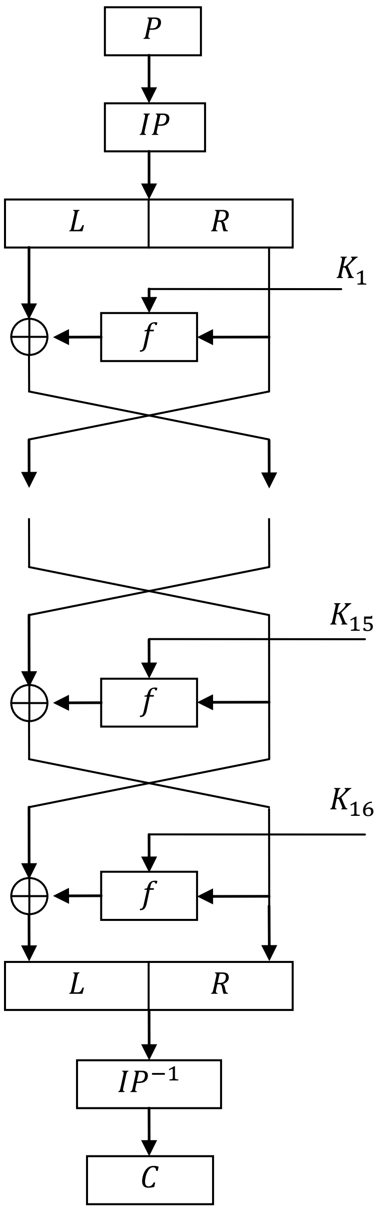

Рис. 2.2. *Схема Фейстеля на примере алгоритма DES*