# 3. Блочные шифры. Формальные модели

Пусть – алфавит двоичных цифр (битов), а

**,**

**− множество *n*-буквенных слов (*n*-битовых блоков) в этом алфавите. *Блочным шифром* называется любое отображение : , которое каждому блоку открытого текста сопоставляет некоторый блок шифртекста. При этом отображение должно быть взаимно однозначным, поскольку обычно требуется, чтобы для произвольного блока шифртекста можно было бы всегда восстановить исходный блок открытого текста.**

**Всего имеется**

**, 0, (при )**

**взаимно однозначных отображений множества 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,**

**.**

**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 позволяют охватить только (не более) шифрующих отображений. Отображение с ключом будем обозначать , а обратное отображение − или .**

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

**: *P* ´ *K* ® *C*,**

**: *C* ´ *K* ® *P*,**

**причем для любых *C* и *K* уравнение разрешимо относительно  *P* и**

**() (здесь ) =, и , – сужения отображений и на множестве *P* и *C*×)*.* Пятёрка (*P*, *C*, *K*, , ) называется *блочным алгоритмом преобразования данных* или просто *блочным шифром.* С практической точки зрения наибольший интерес представляет случай *P* = *C* = , *K* = . Поскольку и – взаимно обратные отображения на множестве , то в этом случае блочный шифр однозначно определяется тройкой ().**

**В отображении по сути заключено целое семейство (а именно ) отображений , получаемых из путем выбора конкретного ключа . Значение (длина ключа) должно быть достаточно большим, чтобы исключить попытку полного (тотального) перебора ключей и соответственно отображений . Рекомендуемые значения , или исключают возможность указанного перебора. Отображения и будем называть соответственно *функциями зашифрования и расшифрования.***

**Конкретную функцию зашифрования можно представить в виде -строчной таблицы (так называемой *электронной кодовой книги*):**

|  |
| --- |
| **…**    **…**  **,** |

**в которой каждому блоку открытого текста сопоставлен соответствующий блок  шифртекста. Другой способ заключается в использовании формул алгебры логики: значение (-го бита шифртекста, ) может быть выражено в виде булевой функции от битовых переменных , образующих блок открытого текста (функция 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 велико, то таблица, задающая , становится чрезмерно большой (содержащей бит, причем следует иметь в виду, что каждому ключу будет соответствовать отдельная шифровальная книга), а каждая из булевых функций будет иметь, как правило, большую сложность (порядка по числу членов, входящих в формулу, реализующую ). 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 обычно конструируется как произведение (композиция или суперпозиция) некоторого числа достаточно простых для задания и реализации преобразований , , ,…, , т.е.**

**,**

**где произведение отображений и определяется условием: , . 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 () и перестановки (). Схемы, реализующие эти преобразования, получили название *подстановочно-перестановочных сетей .***

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

**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 (), перестановки () и линейные преобразования (). Результирующее преобразование в этом случае выглядит следующим образом:**

**.**

**Ключ , определяющий разнообразие получающихся при этом отображений, может использоваться при определении какого-либо одного типа преобразований , или сразу во всех типах.**

S1

S1

S1

**Рис. 2.1. *Структура композиционного шифра***

***на основе подстановок, перестановок и линейных преобразований***

**Процедура зашифрования для композиционного шифра осуществляется по схеме:**

***Вход*: Блок открытого текста.**

***Выход*:Блок шифртекста.**

**Дадим необходимые пояснения к приведенной схеме. Преобразование называется *раундом* (или *циклом*) *зашифрования,* а (,) − *раундовой* (или *цикловой*) *функцией зашифрования.* Отметим, что базовый ключ в каждом раунде, вообще говоря, целиком не применяется, а используются *раундовые ключи* ,,…, , которые формируются на основе базового ключа на этапе предвычислений. Объединение раундовых ключей обычно называют *расширенным ключом.* Базовый ключ является секретным элементом криптосистемы и называется *секретным* (или *основным*)ключом. Обратное преобразование (расшифрование) для композиционного шифра выполняется по схеме:**

**,**

**где .**

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

**,**

**то такой композиционный шифр называется *итеративным.***

***Схема* (*сеть*) *Фейстеля* () является разновидностью итеративного шифра. При шифровании блок открытого текста разбивается на две равные части – левую () и правую (). В каждом раунде одна из частей преобразуется при помощи функции и раундового ключа . Результат операции суммируется покомпонентно по модулю (операция ) 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 на каждом раунде. Сама же функция 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*