Н. А. Молдовян, А. А. Молдовян, М. А. Еремеев

Криптография

От примитивов к синтезу алгоритмов

НАУЧНОЕ ИЗДАНИЕ

bhv®

УДК 681.3
ББК 32.81
М75

***Рецензенты*:**
доктор технических наук, профессор В.И. Коржик
доктор технических наук, профессор А.М. Чуднов

Молдовян Н. А., Молдовян А. А., Еремеев М. А.
М75 Криптография: от примитивов к синтезу алгоритмов. — СПб.: БХВ-Петербург, 2004. — 448 с.: ил.

ISBN 5-94157-524-6

В книге приводятся элементы математических основ криптографии. Раскрывается содержание симметричных и асимметричных шифров, систем цифровой электронной подписи и хэш-функций и основные требования к ним. Излагаются новые результаты в направлении проектирования скоростных шифров на основе управляемых преобразований.

Представлена классификация управляемых примитивов, на основе которых синтезируются новые классы операций, зависящих от преобразуемых данных. Анализируются основные свойства управляемых примитивов. Дается описание ряда новых криптографических примитивов и алгоритмов с оценкой их стойкости к дифференциальному, линейному и другим методам криптоанализа.

Для специалистов в области безопасности информации, криптографии, прикладной математики, информатики и электроники, а также для преподавателей, студентов и аспирантов инженерно-технических вузов.

УДК 681.3
ББК 32.81

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Игорь Шишигин*
Зав. редакцией	*Григорий Добин*
Компьютерная верстка	*Сергея Матвеева*
Корректор	*Евгений Камский*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.05.04.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 36,12.
Тираж 2000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02
от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-524-6

Содержание

ГЛАВА 1

Вопросы одноключевой криптографии

1.1. Варианты реализации шифров

В настоящее время алгоритмы шифрования, представляющие практический интерес, являются для ручного шифрования достаточно сложными. Также в историю вошли алгоритмы, лежавшие в основе механических и электромеханических шифраторов. Современные алгоритмы шифрования обычно реализуются в виде некоторого электронного устройства, основной частью которого является крипточип – интегральная схема, реализующая алгоритм шифрования, и в виде программ для ЭВМ. Алгоритмы, разрабатываемые как стандарты для широкого использования, должны обеспечивать высокую производительность как при программной реализации, так и при недорогой аппаратной реализации. Однако для многих технологических применений предполагается использование либо устройств шифрования, либо программ шифрования. Очевидно, что в таких случаях разумно ориентироваться только на один из указанных вариантов реализации. Это позволит повысить производительность шифрующих программ, а при аппаратной реализации — существенно уменьшить стоимость криптографических устройств. Способы построения аппаратно-ориентированных шифров рассмотрены в книге [14].

К программным шифрам относятся криптосистемы, которые обеспечивают высокую производительность шифрования данных при их реализации в виде компьютерных программ. Система команд микропроцессоров является достаточно ограниченной, однако она позволяет реализовать достаточно производительные алгоритмы шифрования. В значительной степени это связано с тем, что в системе команд всегда присутствует команда пересылки (чтения) содержимого некоторой ячейки оперативной памяти в один из регистров процессора. Данная операция, реализующая выборку из некоторого массива в памяти, представляет собой не что иное, как нахождение значения некоторой функции, заданной табличным способом, по некоторому значению аргумента, заданному как адрес ячейки памяти. Табличным способом могут быть заданы произвольные функции, в том числе и операции подстановок, которые являются базовым криптографическим примитивом во многих современных криптосистемах.

В конечном счете, любой блочный шифр представляет собой чрезвычайно большое множество подстановок большого размера (число возможных входных значений 2^{64} для 64-битового шифра или 2^{128} для 128-битового), выбираемых в зависимости от секретного ключа. Однако такое непосредственное задание шифра практически нереализуемо, поскольку требует неимоверно большого объема памяти. Но такие под-

становки можно генерировать. Соответствующий генератор и представляет собой алгоритм шифрования. Если бы мы могли выбирать непосредственным образом некоторую секретную подстановку, то в принципе мы могли бы выбрать ее случайным образом. При использовании генератора формируемые подстановки не являются случайными, даже если мы выберем случайный секретный ключ, который задает выбор конкретной подстановки. Проблема разработки алгоритмов шифрования – это проблема задания такого множества подстановок, каждая из которых являлась бы псевдослучайной (практически неотличимой от случайной). Иными словами, разработка стойкого шифра связана с построением генератора псевдослучайных подстановок, управляемого секретным ключом.

Что касается подстановок малого размера (4×4, 8×8 и даже 16×16), то они легко реализуются программным и аппаратным способом и используются как базовые операции при проектировании шифров. При этом учитывается, что при увеличении размера подстановки резко возрастает ресурс, необходимый для их реализации. Наиболее эффективными для аппаратной реализации представляются подстановки размера 4×4, а для программной – 4×4 и 8×8. Подстановки размера 8×8 также имеют приемлемую сложность схемотехнической реализации. Именно эти варианты подстановок и представляют собой криптографический примитив, который позволяет сочетать эффективность программной и аппаратной реализации разрабатываемого шифра. Другой операцией, органически дополняющей операцию подстановки, является перестановка битов преобразуемых данных. Произвольная перестановка аппаратным путем реализуется практически без затрат ресурсов, а при программной реализации с использованием современных процессоров широкого назначения она вносит существенную задержку в процесс шифрования. Этот недостаток может быть полностью устранен путем расширения системы команд универсального процессора командой управляемой битовой перестановки, что и было предложено в работах [3, 20].

Выбор размера и конкретных таблиц подстановок при построении шифров является одной из главных задач, которые должны быть решены, но сама возможность эффективного осуществления операций подстановок при программной реализации обеспечивается стандартной системой элементарных команд процессора. Вместо перестановок произвольного типа в программных шифрах используются частные виды перестановок, например, операция циклического сдвига. Наряду с базовой операцией подстановки могут быть использованы другие элементарные команды процессора:

- операции циклического сдвига на фиксированное число двоичных разрядов;
- операции циклического сдвига на фиксированное число двоичных разрядов, зависящее от ключа;
- операции циклического сдвига на фиксированное число двоичных разрядов, зависящее от преобразуемых данных;
- поразрядное суммирование по модулю два;
- суммирование и/или вычитание по модулю 2^{32};
- операции поразрядного логического умножения и/или сложения двух n-битовых двоичных векторов и др.

В целом разработка программных шифров связана с учетом специфики обработки данных в компьютерных системах, что позволяет получить высокие скорости шифрования при использовании микропроцессоров широкого применения. Практическая потребность решения проблемы защиты электронной информации в массовом масштабе обусловливает актуальность разработки программных шифров и перспективы их широкого применения.

При выборе операций подстановок можно использовать следующие возможности, связанные с программной реализацией:

- можно использовать таблицы подстановок, зависящие от секретного ключа (в этом случае таблицы подстановок формируются по ключу на этапе предвычислений, выполняемых при инициализации криптосистемы);
- можно использовать таблицы подстановок достаточно большого размера;
- можно использовать табличные подстановки, зависящие от преобразуемых данных (данный тип подстановок реализуется с помощью некоторого множества пронумерованных таблиц подстановок).

Операции подстановок являются только частным случаем табличного задания функций отображения. Подстановки являются биективными отображениями, т. е. позволяют однозначно выполнить обратное преобразование. Это свойство необходимо для многих схем построения алгоритмов шифрования. Но оно не является необходимым условием для обеспечения возможности осуществления расшифрования закрытого сообщения. Например, в криптосхеме Фейстеля при построении раундовой функции могут быть использованы произвольные операции преобразования. Таким образом, представляют интерес не только подстановки, но и операции отображения более общего типа. Последние также могут быть эффективно реализованы программными средствами. Вместо операций подстановок или дополнительно к ним в программных шифрах можно использовать операции табличного отображения (фиксированные, зависящие от ключа и/или от преобразуемых данных). Одним из важных вариантов реализации операций табличного отображения является механизм выборки подключей в зависимости от преобразуемых данных, который успешно использован при разработке ряда скоростных программных шифров [93-96].

С программной реализацией криптосистем связана возможность применения достаточно сложных процедур предвычислений, реализуемых как этап инициализации криптосистемы, осуществляемый после ввода секретного ключа. В частности, инициализация может включать процедуру настройки алгоритма шифрования по секретному ключу [40]. При аппаратной реализации использование сложных предвычислений приводит к значительному повышению стоимости устройств шифрования и снижению их производительности при частой смене ключей. Шифры, в которых алгоритм шифрования формируется в зависимости от секретного ключа, называются гибкими или недетерминированными. В гибких шифрах каждому ключу соответствует уникальная модификация алгоритма шифрования. Поскольку множество ключей ограничено, то это означает, что гибкий шифр представляет собой множест-

во алгоритмов шифрования, описываемых с помощью некоторого алгоритма, который задает правило формирования алгоритма шифрования в зависимости от секретного ключа. Формирование секретных таблиц подстановок, таблиц операции отображения и настройка алгоритма шифрования предполагают использование этапа настройки шифра, выполняемой однократно при введении секретного ключа. После настройки криптосистема многократно выполняет процедуры шифрования и расшифрования данных. При сравнительно редкой смене ключей (например, ключ сменяется каждые 10 секунд) наличие этапа настройки практически не изменяет среднюю скорость шифрования.

В случае ограничения длины k секретного ключа (например, $k < 40$ бит) возможность значительного усложнения этапа предвычислений, используемого в программных шифрах, может быть использована для повышения стоимости (сложности) раскрытия ключа. Это позволит уменьшить вероятность раскрытия ключа со стороны большого числа потенциальных нарушителей. Еще более надежную страховку может обеспечить использование алгоритмов, настраиваемых по некоторому дополнительному параметру, который должен быть известен узкому кругу пользователей и иметь достаточный размер r (например, $r = 80$ бит). В этом случае для внешнего нарушителя переборная сложность задачи раскрытия криптосистемы может быть оценена как 2^{k+r} шифрований, что существенно превышает сложность задачи криптоанализа (2^k шифрований) для внутренних пользователей.

1.2. Повышение стойкости шифрования при ограничении длины секретного ключа

В случае коротких ключей наиболее результативным способом нападения на шифр может стать силовая атака, которая заключается в переборе всех возможных вариантов секретного ключа (или пароля, если последний используется в качестве ключа). Наличие этапа предвычислений служит определенным барьером против силовой атаки. Наличие предвычислений, которые необходимо выполнить для каждого испытываемого варианта ключа, существенно затрудняет такую атаку. Время выполнения процедур настройки t может быть задано достаточно большим путем усложнения алгоритма настройки или путем многократного его использования. Трудоемкость силовой атаки можно оценить по формуле:

$$W \approx 2^k W_t/2,$$

где W_t – трудоемкость алгоритма предвычислений, k – длина секретного ключа в битах (предполагается, что ключ является случайным и равновероятным по множеству всех ключей длины k). Мы полагаем, что атакующий имеет некоторый известный исходный текст и соответствующий ему шифртекст, причем процедура его зашифрования имеет сложность намного меньше значения W_t (т. е. время шифрования известного текста пренебрежимо по сравнению с t и составляет, например, одну миллисекунду). В случае пароля, состоящего из букв естественного языка или выбираемого из словаря, эта формула также может быть применена, если в качестве значения k использовать некоторую эффективную длину $k_e < k$.

Требуемое время для выполнения силовой атаки можно оценить по формуле:

$$T \approx 2^{k-1} t,$$

где t – время, затрачиваемое на выполнение алгоритма предвычислений. Для многих приложений значение t равное от 0.5 до 1 секунды, является вполне приемлемым, поскольку для законного пользователя эта задержка относится только к однократно выполняемой инициализации криптосистемы. Однако, для нарушителя при длине ключа, составляющей $k = 32$ бит, в среднем время силового взлома составит более 50 лет работы однопроцессорной ЭВМ широкого применения (для которой $t = 0.5$ с). Очевидно, что это делает стоимость раскрытия ключа для многих потенциальных нарушителей (например, хакеров) неприемлемой и заставит их отказаться от осуществления атаки. Для проведения атаки за разумное время потребуется использование очень большого числа ЭВМ. Например, для того чтобы раскрыть один ключ за один месяц, потребуется непрерывная работа более 500 компьютеров широкого применения либо применение специализированных многопроцессорных ЭВМ, которые есть в наличии только у весьма ограниченного числа организаций и являются весьма дорогостоящими, равно как и их эксплуатация.

Аналогичные оценки для секретных ключей, имеющих длину от 8 до 10 байт (т.е. от 64 до 80 бит), показывают, что силовой взлом шифров с предвычислениями легко сделать практически неосуществимым даже для нарушителя, обладающего очень мощными вычислительными ресурсами. Несомненно, противодействие силовой атаке путем увеличения длины ключа является наиболее эффективным общим приемом для всех криптосистем, допускающих произвольный выбор длины секретного ключа, однако при ограничении его длины этот прием не может быть использован в полной мере.

С пользовательской точки зрения, проблема выбора пароля (секретного ключа) и повышения стойкости к атакам на основе подбора пароля заслуживает внимания, поскольку для всех широко используемых систем защиты требуется выбирать хорошие (случайные) пароли, которые трудно запомнить. В средствах защиты можно использовать активный контроллер паролей, т.е. систему, блокирующую выбор плохих паролей. Другое возможное решение проблемы совмещения удобства пользователей с высоким уровнем секретности состоит в использовании паролей-фраз, т.е. не отдельных слов, а целых фраз, которые удобны для запоминания, но трудны для подбора из-за большой длины. Целесообразно построение таких механизмов парольного доступа к ресурсам ЭВМ, которые содержат встроенный механизм противодействия атакам на основе перебора возможных паролей (или парольных фраз). Такие механизмы могут состоять, например, в использовании достаточно сложных процедур вычисления однонаправленной функции от пароля, требующих времени около 1с для процессоров широкого применения. В постоянной памяти ЭВМ будет храниться таблица значений этой однонаправленной функции от паролей всех законных пользователей (таблица образов паролей), а проверка подлинности текущего пользователя будет осуществляться как вычисление значения этой функции от значения текущего пароля и сравнение полученного значения с соответствующим значением из таблицы образов паролей.

Применение долговременных ключевых элементов при ограничении длины секретного ключа

Наиболее результативным способом повышения стойкости криптосистем в условиях применения коротких секретных ключей (например, имеющих длину $k = 32$ бит) является использование долговременных ключей. Примером криптосистемы с долговременным ключом является российский стандарт шифрования ГОСТ 28147–89 [8], в котором таблицы подстановок являются секретными. В соответствии с общепризнанным принципом Керхкоффа при оценке стойкости алгоритм шифрования предполагается известным атакующему. В более общей трактовке этот принцип можно выразить так: *все долговременные элементы механизмов защиты информации необходимо считать известными потенциальному нарушителю*. Это связано с тем, что обычно долговременные элементы известны многим участникам разработки шифра.

Идея использовать долговременные ключи большого размера или долговременные секретные элементы алгоритма шифрования для повышения стойкости является отступлением от принципа Керхкоффа, но в случае ограничения длины секретного ключа этот прием является обоснованным тем, что этот элемент усиления не имеет цель обеспечить гарантированную стойкость. Преследуемая цель состоит только в существенном повышении сложности раскрытия короткого ключа, например, со стороны внешних нарушителей. Ограниченная длина ключа предполагает, что предусматривается защита от потенциального нарушителя с весьма ограниченными вычислительными ресурсами. При наличии у нарушителя больших вычислительных ресурсов подбор ключа реализуем за приемлемое время. Использование долговременного ключа для обоих типов нарушителей потребует определения долговременного ключа, а это может быть сделано либо путем более сложного криптоанализа, либо получением этого ключа «некриптографическими» методами. Долговременный ключ может представлять собой:

- секретные константы;
- секретные таблицы подстановок (отображений);
- секретный алгоритм предвычислений (настройки);
- секретный алгоритм шифрования.

При правильном построении криптосистемы короткий ключ вычислить практически невозможно (при использовании сколь угодно больших реально существующих вычислительных ресурсов) без знания перечисленных секретных элементов (если таковые используются). Это сделает невозможной задачу раскрытия короткого секретного ключа при атаке на криптосистему, осуществляемую субъектами, не владеющими долговременным ключом.

С точки зрения общей оценки криптографической стойкости, использование долговременных ключей не приводит к повышению секретности, поскольку долговременный ключ предполагается известным фиксированному кругу пользователей,

внутри которого реально действующим является только короткий ключ. Примером систем с долговременным ключом является стандарт шифрования ГОСТ 28147–89. Используемое в нем «заполнение таблиц блока подстановки», которое «является долговременным ключевым элементом, общим для сети ЭВМ», должно рассматриваться известным атакующей стороне в некоторых возможных ситуациях. Согласно описанию этого стандарта «заполнение таблиц блока подстановки является секретным элементом и поставляется в установленном порядке». Однако роль такого секретного параметра должна быть оценена с учетом упомянутых выше замечаний.

1.3. Криптосистемы с гибким алгоритмом и требования к алгоритму предвычислений

Использование долговременных (хотя даже и сменяемых) секретных частей шифрующих систем в общем случае не приводит к существенному повышению стойкости. Однако сама идея использования не только секретных параметров, но и секретных элементов алгоритма шифрования заслуживает внимания. Если секретные элементы алгоритма сделать легко сменяемыми, то в этом случае можно говорить о гибких криптосистемах или шифрах с гибким алгоритмом. Такие конструкции могут быть легко реализованы в программных шифрах. С примерами можно ознакомиться в работах [14, 42, 94]. В гибких шифрах долговременным элементом являются процедуры настройки алгоритма шифрования, а конкретная его модификация и ключ шифрования являются сменными элементами криптосистемы, которые автоматически заменяются одновременно со сменой паролей (ключей) и являются уникальными для каждого пользователя (или каждой пары абонентов защищенной сети связи).

Хранение описания секретной модификации алгоритма шифрования приводит к определенным неудобствам для пользователей. Кроме того, при очень большом числе таких модификаций возникают проблемы их хранения. Устранение этих проблем связано с использованием генератора модификаций алгоритма шифрования. Формируется алгоритм, генерирующий конкретную модификацию по конкретному вводимому секретному параметру. Этим параметром может быть дополнительный или основной секретный ключ. В последнем случае только длина секретного ключа будет определять переборную стойкость криптосистемы. Тем не менее, секретность конкретной модификации алгоритма приводит к существенному повышению стойкости к другим типам атак, которые являются наиболее опасными. Действительно, переборную атаку легко устранить простым удлинением ключа, тогда как противодействие некоторым другим типам атак требует тщательной проработки всех элементов криптосистемы.

Процедура формирования алгоритма шифрования по секретному ключу является существенно более сложной по сравнению с непосредственным шифрованием данных. Очевидно, что настройку алгоритма шифрования разумно выполнить как часть предвычислений. Алгоритм предвычислений является частью криптосистемы, поэтому он также вносит свой вклад в задание общей секретности (стойкости) шифра. Эта часть не является столь критичной как сам алгоритм непосредственного шифрования. Кроме того, для реализации предвычислений могут быть использованы значи-

тельные вычислительные ресурсы, что упрощает задачу построения необходимых процедур предвычислений. В общем случае секретная модификация алгоритма непосредственного шифрования и некоторые сгенерированные в зависимости от секретного ключа параметры или массивы данных могут быть рассмотрены как расширенный ключ. Рассмотрим общие требования к алгоритму предвычислений. Одним из требований к алгоритму формирования ключа шифрования является следующее: *количество возможных выходных последовательностей не должно быть существенно меньше числа возможных секретных ключей*. Желательно, чтобы число возможных расширенных ключей было равно числу различных значений секретного ключа. Это требование связано с тем, что число различных расширенных ключей может быть только равным или меньшим. Действительно, длина расширенного выходного ключа превышает длину секретного ключа, но для определенных процедур предвычислений может оказаться, что различным секретным ключам будут соответствовать одинаковые расширенные ключи.

В случае применения односторонних преобразований на этапе формирования ключей шифрования значительное сужение пространства ключей шифрования является маловероятным, однако желательно получение гарантии того, что мощность множества различных ключей шифрования равна мощности множества секретных ключей длиной $l < L$, где L – длина расширенного ключа в байтах. Последнее условие достигается, если используемые процедуры предвычислений на каждом шаге преобразований задают подстановку L-байтового блока данных M, полученного как первые L байт периодического ряда, представляющего собой многократное повторение секретного ключа. В качестве составной части алгоритма предвычислений можно использовать некоторый известный блочный шифр, используемый для преобразования сообщения M в режиме сцепления блоков шифра. При этом в качестве ключа можно использовать некоторое специфицированное значение Q. Криптограмма $C = \mathrm{E}_Q(M)$ может служить в качестве расширенного ключа. При этом выполняется также требование *псевдослучайности расширенного ключа*.

Рассмотренные выше два требования к процедурам формирования расширенного ключа представляются достаточными. При желании без труда можно предложить алгоритм настройки, удовлетворяющий некоторым другим (дополнительным) требованиям. Можно принять требование вычислительной сложности определения секретного ключа по расширенному ключу и известным процедурам предвычислений. Приемлемые алгоритмы построения расширенного ключа описаны в работах [14, 41]. Смысл использования сложных процедур настройки состоит в том, чтобы заставить нападающего отказаться от их рассмотрения и принять предположение о случайности ключа шифрования.

1.4. Управляемые операции как криптографический примитив

Управляемые операции давно привлекают внимание разработчиков алгоритмов шифрования. Одной из наиболее ранних работ, посвященных проектированию криптосистем на основе управляемых операций, является статья [78], в которой рассмот-

рено использование управляемой подстановочно-перестановочной сети. Другие попытки [105, 123] были связаны с применением управляемых перестановочных сетей в качестве криптографического примитива. Однако в предложенных схемах выбор конкретной модификации реализуемой операции осуществлялся в зависимости от секретного ключа. В таком применении управляемых операций фиксируется конкретная их модификация, которая не изменяется при шифровании большого числа блоков данных. В случае управляемых битовых перестановок мы имеем некоторую фиксированную перестановку, которая является линейной операцией, хотя и неизвестной атакующему. Детальные исследования стойкости различных вариантов криптосхем на основе управляемых операций, зависящих от секретного ключа, показало, что они не могут конкурировать с другими шифрами по производительности.

Другим типом управляемых операций являются операции, зависящие от преобразуемых данных. Их особенностью является изменчивость реализуемых модификаций, что позволяет использовать термин «переменные операции». Наиболее известными алгоритмами, использующими переменные операции в качестве базового криптографического примитива, являются итеративные блочные шифры DES [89, 113], RC5 [109], RC6 [103, 110].

Первый алгоритм использует управляемые табличные подстановки размера 4×4, реализованные как блоки подстановок размера 6×4, обеспечивающие выбор одной из четырех возможных подстановок 4×4. Однако размерность векторов, которые преобразуются этими блоками подстановок, невелика, а при увеличении размера табличных подстановок существенно возрастет как сложность выбора оптимальных табличных подстановок, так и сложность их реализации. В алгоритмах RC5 и RC6 в качестве управляемых операций применяются операции циклического сдвига на число битов, выполняемые в зависимости от преобразуемых данных. Несмотря на то, что упомянутые типы переменных операций обладают сравнительно малым числом различных реализуемых модификаций, они являются эффективным криптографическим примитивом. Таким образом, переменные операции с малым числом реализуемых модификаций оказываются более эффективными по сравнению с операциями, зависящими от ключа, хотя последние и обладают очень большим числом модификаций.

Это сопоставление приводит к идее применения операций с большим числом модификаций в качестве переменных операций [92, 97]. Наиболее детально эта идея проработана по отношению к управляемым битовым перестановкам, выполняемым над подблоками данных размером 32 и 64 бит [62, 72, 75, 82, 98]. Переход к произвольным перестановкам дал возможность существенно увеличить число различных модификаций, реализуемых переменной операцией (до $n!$, где n – длина преобразуемого вектора). Для обеспечения возможности выполнения расшифровывающего преобразования шифруемый блок данных разбивают на два подблока – управляемый и преобразуемый, которые обычно имеют одинаковый размер. В этом случае число изменяющихся модификаций ограничивается размером управляющего подблока данных и равно 2^n, где n – размер последнего. Очевидно, что перестановка, зависящая от преобразуемых данных, описывается как операция подстановки частного вида, выполняемая над всем преобразуемым блоком данных и оставляющая управляющий подблок без изменения. Эффективность переменной перестановки можно объ-

яснить тем, что это подстановка, выполняемая над всем преобразуемым блоком данных. Некоторые ее слабости (линейность суммы выходов, сохранение веса Хемминга) связаны именно с тем, что эта подстановка относится к специальному типу.

В монографии [14] приведены описание и анализ ряда скоростных блочных шифров, основанных на битовых перестановках, зависящих от преобразуемых данных. Несмотря на то, что переменные перестановки являются линейным криптографическим примитивом, их комбинирование с операциями, имеющими «небольшую» нелинейность, эффективно нейтрализуют линейный криптоанализ [62, 73, 82]. Это объясняется тем, что единственной линейной комбинацией выходов переменной перестановки является сумма всех выходных битов, а при наличии некоторой дополнительной нелинейной операции трудоемкость линейной атаки становится достаточно высокой.

Другим интересным типом управляемых операций, обладающих большим числом модификаций, являются управляемые сумматоры [9, 12], представляющие собой частный случай управляемых двухместных операций [27, 74]. Данные операции также могут быть применены в качестве переменных операций. Они представляют как самостоятельный интерес, так и для комбинирования с переменными перестановками в единой криптосистеме. В *главе 4* будет дано обобщение управляемых перестановок и построен класс управляемых операционных подстановок, реализуемых на основе подстановочно-перестановочных сетей с использованием управляемых элементов минимального размера. В *главах 5* и *6* будут рассмотрены управляемые операционные подстановки, реализуемые с использованием других типов управляемых элементов. Следует отметить, что в настоящее время имеется достаточно большое число различных типов управляемых операций, обладающих большим числом реализуемых модификаций и перспективных для использования в качестве переменных операций.

Эффективность аппаратной реализации шифров на основе переменных битовых перестановок показана в работах [38, 84]. При использовании сравнительно малых аппаратных ресурсов обеспечивается высокая производительность при реализации как в заказных, так и в программируемых СБИС [119]. Вопросы построения операционных блоков управляемых перестановок (БУП), обладающих заданными свойствами, рассмотрены в работах [10, 11]. На самом деле БУП представляют собой перестановочные сети, которые ранее были широко исследованы в области параллельных вычислений и телефонии [53, 54, 102, 124], однако криптографическое их применение требует рассмотрения некоторых других свойств таких сетей, например, линейных характеристик [2].

1.5. Аппаратная реализация шифров на основе битовых перестановок, зависящих от преобразуемых данных

Представляют интерес два следующих основных варианта аппаратной реализации алгоритмов шифрования:

- в заказных сверхбольших интегральных схемах (СБИС);

- в программируемых логических матрицах – программируемых логических интегральных схемах (ПЛИС).

Первый вариант используется для серийного производства шифраторов, поскольку в этом случае он обеспечивает более низкую стоимость единицы изделия. Другим преимуществом является более высокое быстродействие изготавливаемых специализированных СБИС. Второй вариант является предпочтительным при штучном производстве устройств шифрования. Кроме того, при разработке технологии производства заказных СБИС он применяется как предварительный для изготовления и тестирования экспериментальных устройств. Достоинствами использования ПЛИС являются малый срок разработки и возможность многократного перепрограммирования под другие алгоритмы или при модифицировании алгоритмов. Другими важными направлениями использования ПЛИС являются выполнение вычислительных экспериментов большого объема и решение криптоаналитических задач.

Аппаратную реализацию итеративных шифров осуществляют в соответствии с двумя основными типами архитектур:

- итеративной;
- конвейерной.

В первом случае схемно реализуется только один раунд шифрования. Раундовая функция шифрования используется многократно для выполнения всех раундов шифрования. Дополнительные схемные компоненты осуществляют смену раундовых ключей и другие функции, необходимые для правильного осуществления вычислений, предписываемых алгоритмом. Важнейшим достоинством итеративной архитектуры является обеспечение практически одинаковой производительности в режиме электронной кодовой книги (независимое шифрование блоков данных) и в режиме сцепления блоков шифра.

Конвейерная архитектура предполагает схемную реализацию всех раундов шифрования и возможность одновременного преобразования многих блоков данных. Обычно на выходе каждого раунда устанавливается регистр для хранения промежуточного значения преобразуемого блока данных и количество одновременно преобразуемых блоков данных равно числу раундов шифрования R. Раундовые ключи хранятся в регистре ключей постоянно при выполнении шифрования сообщения. Если схема одного раунда имеет достаточно малое время задержки, то за один такт работы устройства осуществляется преобразование сразу R блоков данных, причем один из них проходит последний раунд шифрования. Таким образом, в конвейерной архитектуре за время одного такта осуществляется преобразование одного блока данных и производительность устройства оказывается примерно в R раз более высокой по сравнению с итеративной архитектурой. Недостатками конвейерной архитектуры являются существенно более высокая стоимость реализации и невозможность сохранения высокой производительности в режиме сцепления блоков шифра.

В таблице 1.1 приводятся параметры реализации следующих шифров, основанных на переменных перестановках: CIKS-1 [92] и SPECTR-H64 [72], DDP-64 (*см. гл. 4*), COBRA-H64 [99] и COBRA-H128 (*см. гл. 8*). В таблице 1.2 дается сопоставление характеристик реализации шифров SPECTR-H64 и COBRA-H128 с широко из-

вестными криптосистемами DES [79], AES и IDEA для случая использования ПЛИС. Шифр SPECTR-H64 обеспечивает более высокую скорость при меньших затратах аппаратных ресурсов по сравнению с криптосистемами AES и IDEA. Стоимость реализации SPECTR-H64 несколько превышает стоимость реализации алгоритма DES, однако, первый обеспечивает многократное превышение по производительности. Сравнение 128-битовых шифров COBRA-H128 и AES показывает, что при примерно одинаковых аппаратных затратах первый из них обеспечивает существенно более высокую производительность.

Таблица 1.1

Характеристика аппаратной реализации шифров, основанных на переменных битовых перестановках, с использованием программируемых и заказных СБИС

Шифр	**Арх-ра**	**ПЛИС (Xilinx Vitrex)**			**Заказные СБИС (0.33 мкм)**		
		К-во блоков CLB*	Частота МГц	Скорость шифров. Гбит/с	Площадь, sqmil**	Частота, МГц	Скорость шифров. Гбит/с
CIKS-1	Итер.	907	81	0.648	3456	93	0.744
	Конв.	6346	81	5.184	21036	95	5.824
SPECTR-H64	Итер.	713	83	0.443	3194	91	0.485
	Конв.	7021	83	5.312	32123	94	6.016
DDP-64	Итер.	615	85	0.544	2620	92	0.589
	Конв.	3440	95	6.1	14050	101	6.5
COBRA-H64	Итер.	615	82	0.525	2694	100	0.640
	Конв.	3020	85	5.5	14640	110	7.1
COBRA-H128	Итер.	2364	86	0.917	6364	90	1.00
	Конв.	12080	90	11.5	48252	95	12.1

* CLB (Configurable Logic Blocks) – конфигурируемые логические блоки, являющиеся типовыми логическими элементами ПЛИС.

** Площадь используемой поверхности кремниевого кристалла указана в единицах sqmil; 1 sqmil = 7.45 10^{-4} мм^2

Таблица 1.2

Сравнение результатов аппаратной реализации различных шифров с использованием программируемых СБИС

Шифр	**Размер входа, бит**	**Арх-ра**	**ПЛИС (Xilinx Vitrex)**		
			К-во блоков CLB*	Частота, Мбит/с	Скорость шифрования Гбит/с
COBRA-H128	128	Итер.	2364	86	0.917
		Конв.	12080	90	11.5

Шифр	Размер входа, бит	Арх-ра	ПЛИС (Xilinx Vitrex)		
			К-во блоков CLB*	Частота, Мбит/с	Скорость шифрования Гбит/с
SPECTR-H64 [119]	64	Итер.	713	83	0.443
		Конв.	7021	83	5.312
AES [117]	128	Итер.	2358	22	0.259
		Конв.	17314	28.5	3.650
IDEA [66]	128	Итер.	2878	150	0.600
DES [79]	64	Итер.	722	11	0.181

Приведенные характеристики различных вариантов реализаций шифров, основанных на переменных перестановках, показывают, что использование операций, зависящих от преобразуемых данных, обеспечивает возможность создания высокоскоростных аппаратных шифраторов, которые могут быть изготовлены с минимальным потреблением аппаратных ресурсов.

1.6. Особенности проектирования блочных шифров на основе управляемых операций

1.6.1. Управляемые операции и отображения

Любую операцию, используемую при построении блочных шифров, можно представить как отображение векторного пространства h-мерных двоичных векторов $W = (w_1, w_2, \ldots, w_h)$ в векторное пространство n-мерных двоичных векторов $Y = (y_1, y_2, \ldots, y_n)$, где для всех $j \in \{1, \ldots, h\}$ и $i \in \{1, \ldots, n\}$ имеем $w_j, y_i \in \mathrm{GF}(2)$. Такое отображение можно записать в виде $\mathrm{GF}(2)^h \to \mathrm{GF}(2)^n$. В вероятностных шифрах используются отображения, относящиеся к случаю $h < n$, в котором выходное значение зависит от некоторого случайного значения. В таких случаях по выходному значению однозначно определяется входной двоичный вектор. При $h > n$ в общем случае нельзя определить однозначно входной вектор по выходному. Операции такого типа могут применяться при построении шифров на основе криптосхемы Фейстеля, которая для произвольной раундовой функции задает корректное построение блочного шифра, т. е. обеспечивает возможность правильного расшифрования. Операции, соответствующие случаю $h = n$ и устанавливающие взаимно-однозначное соответствие между входными и выходными векторами, задают преобразование исходного векторного пространства. Такие операции задают некоторую подстановку. Подстановками иногда называют также операции, соответствующие случаю $h > n$ (см., например, подстановки типа 6×4 в алгоритме DES).

При $h > n$ отображения можно интерпретировать как управляемые операции с размером управляющего входа равным $m = h - n$. Однако часто более удобным, наглядным и полезным для анализа оказывается рассмотрение операции как управляе-

мой, хотя всегда надо иметь в виду, что наиболее общим является описание в виде указанного отображения. Мы можем конструировать управляемые операции, исходя из тех или иных соображений и механизмов, оставаясь в каком-то частном классе отображений. При малых значениях m и n отображение можно задать табличным способом, который является наиболее общим. Но при больших значениях m и n (например, $m = 64$ и $n = 32$) табличный способ оказывается неприменимым. В этом случае можно строить некоторый генератор, который будет формировать отображения такого типа и называться операцией преобразования. В некоторых частных вариантах такой генератор можно назвать управляемой операцией.

Обычно в управляемой операции можно выделить информационный вход (или просто вход) и управляющий вход. Отображаемый вектор W длины h представляется в виде конкатенации (X, V) преобразуемого вектора X длины n и управляющего вектора длины $m = h - n$. Такие управляемые операции можно назвать одноместными, поскольку на информационный вход поступает только один операнд. Операция при фиксированном V называется модификацией управляемой операции. Если при каждом фиксированном V реализуется биективное (взаимно-однозначное) отображение пространства входных n-битовых векторов в выходное пространство n–битовых векторов, то можно говорить об управляемой операции преобразования или об управляемом преобразовании. При $2n < h$ можно говорить о двухместных управляемых операциях с размером управляющего входа $m = h - 2n$, в которых на информационный $2n$-разрядный вход поступают два n-битовых вектора. Целесообразным является синтез таких управляемых операций, множество модификаций которых можно было бы отнести естественным способом к некоторому классу. Характерным примером являются управляемые битовые перестановки.

Подход к разработке блочных шифров, опирающийся на использование переменных операций, связан с использованием управляемых операций с очень большим числом возможных модификаций, когда значение m в два и более раза превышает значение n. Однако в алгоритмах шифрования предполагается разбиение преобразуемого блока данных на равные подблоки. Обычно разбиение осуществляют на два подблока. Очевидно, что один из подблоков подлежит преобразованию (ведь мы хотим его зашифровать), а другой может быть использован для формирования управляющего вектора. Простейшим вариантом такого формирования является случай использования каждого бита управляющего подблока данных для задания нескольких битов управляющего вектора. При аппаратной реализации это реализуется простым разветвлением проводников и практически не требует затрат схемотехнических ресурсов. Блок такого разветвления будем называть блоком расширения E.

1.6.2. Расписание использования ключа

Ниже при рассмотрении нескольких типовых итеративных схем синтеза блочных шифров на основе управляемых операций мы остановимся на построениях, которые обеспечивают возможность зашифрования и расшифрования с помощью одной и той же электронной схемы. Смена режима шифрования в таких криптосистемах осуществляется изменением расписания использования ключа или простым обращением очередности использования раундовых ключей. В блочных шифрах использование