Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

В. В. Захаров

СРЕДСТВА И МЕТОДЫ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

В двух частях

Часть 1

Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве пособия для специальности 1-40 03 01 «Искусственный интеллект»

Минск БГУИР 2019

УДК 004.056(076.5)
ББК 32.972.5я73
З-38

Р е ц е н з е н т ы:
кафедра информационно-вычислительных систем учреждения образования
«Военная академия Республики Беларусь»
(протокол №17 от 08.05.2018);

ведущий инженер-программист
многопрофильного научно-производственного частного унитарного
предприятия «Тетраэдр»
кандидат технических наук, доцент Ю. В. Виланский

Захаров, В. В.

З-38 Средства и методы обеспечения информационной безопасности. Лабораторный практикум. В 2 ч. Ч. 1 : пособие / В. В. Захаров. – Минск : БГУИР, 2019. – 64 с. : ил.

ISBN 978-985-543-479-6 (ч. 1).

Даны рекомендации по выполнению лабораторных работ, рассмотрены примеры решения задач, поставленных в рамках лабораторных работ.

УДК 004.056(076.5)
ББК 32.972.5я73

ISBN 978-985-543-479-6 (ч. 1)
ISBN 978-985-543-478-9

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Данное пособие является первой частью лабораторного практикума, направленного на формирование устойчивых знаний о современных и перспективных методах и средствах обеспечения информационной безопасности, способах применения средств обеспечения информационной безопасности в информационных системах у студентов.

Целью лабораторных занятий, предлагаемых в данном пособии, является закрепление теоретического курса, приобретение навыков оценки и применения методов и средств аутентификации, криптографии, а также межсетевого экранирования в системах защиты информации.

1. ГЕНЕРАЦИЯ ПАРОЛЕЙ

1.1. Теоретические сведения

Базовыми процедурами, лежащими в основе любых систем защиты информационных систем, являются идентификация и аутентификация. Большинство механизмов защиты работают на фоне субъектно-объектной модели. В этой модели информационная система представляется как совокупность субъектов и объектов доступа.

Субъект доступа – активная сущность информационной системы, которая может изменять состояние системы путем порождения процессов над объектами, в том числе порождать новые объекты и инициализировать порождение новых субъектов. Чаще всего в качестве субъектов доступа выступают пользователи и процессы, действующие от имени пользователя.

Объект доступа – пассивная сущность информационной системы, над которой выполняют действия субъекты. Объектами являются информационные ресурсы системы.

Предполагается, что существует безошибочный критерий различения субъектов и объектов по свойству активности.

Все субъекты и объекты информационной системы должны быть идентифицированы.

Идентификация в информационных системах – процедура, в результате выполнения которой для субъекта идентификации выявляется его идентификатор, однозначно идентифицирующий этого субъекта в информационной системе. Для выполнения процедуры идентификации в информационной системе субъекту предварительно должен быть назначен соответствующий идентификатор (то есть проведена регистрация субъекта в информационной системе).

Идентификация обеспечивает выполнение следующих функций:

– установление подлинности и определение полномочий субъекта при его входе в систему;

– контроль установленных полномочий в процессе сеанса работы;

– регистрация и учет действий.

Идентификация связана с аутентификацией (установлением подлинности).

Аутентификация – это процедура проверки принадлежности субъекту доступа предъявленного им идентификатора и подтверждения его подлинности.

В общем виде алгоритм процедуры идентификации и аутентификации пользователя может быть следующим:

1) Ввод идентификатора пользователя.

2) Проверка правильности идентификатора. Если идентификатор неверен, то переход к шагу 6. Если идентификатор верен, то переход к шагу 3.

3) Вызов процедуры аутентификации.

4) Пользователь идентифицирован? Если нет, то переход к шагу 6. Если да, то переход к шагу 5.

5) Уведомление пользователя о входе в систему. Завершение работы процедуры.

6) Проверка превышения допустимого числа проверок. Если не превышено, то переход к шагу 7. Если превышено, то переход к шагу 8.

7) Уведомление пользователя об ошибке. Переход к шагу 1.

8) Уведомление о несанкционированном доступе. Временная блокировка ввода. Переход к шагу 1.

Методы аутентификации, как правило, классифицируют по используемым отличительным чертам субъекта, называемым *фактором*. Обычно выделяют три фактора:

– то, что пользователь знает: пароль – секретное слово, известное только пользователю, комбинация знаков для замка, личный идентификационный номер (PIN);

– то, что пользователь имеет: личная печать, ключ от замка, файл данных, содержащих характеристику, смарт-карта и т. д.;

– то, что является непосредственным физическим признаком пользователя (биометрические признаки).

В последнее время экспериментируют с таким фактором, как информация, ассоциируемая с пользователем, например координаты его места нахождения.

Аутентификация с использованием паролей. Наиболее распространенным и привычным являются методы аутентификации с использованием паролей. При вводе субъектом пароля подсистема аутентификации сравнивает его с паролем или его образом, хранящимся в базе данных эталонов. При совпадении пароля с его образом подсистема аутентификации разрешает доступ к объектам системы.

По степени сменяемости паролей различают две группы методов:

– методы, использующие многоразовые пароли;

– методы, использующие одноразовые пароли.

В большинстве информационных систем используют многоразовые пароли.

Введенный пользователем пароль может передаваться по сети двумя способами:

– незащищенно в открытом виде на основе протокола парольной аутентификации (PAP – Password Authentication Protocol);

– в защищенном виде с использованием шифрования (SSL или TLS) или функции хеширования.

Проблемы защищенности многоразовых паролей:

1) Незашифрованный пароль может быть перехвачен атакующим и затем использован для несанкционированного получения прав доступа пользователя. Поэтому с точки зрения обеспечения безопасности целесообразно перед передачей пароль зашифровать и передавать зашифрованный текст.

2) В случае если в базе эталонов хранятся пароли в открытом виде, атакующий может получить доступ к хранящимся паролям и затем использовать их для неправомочного получения доступа.

3) Если в базе хранятся зашифрованные пароли, то где хранится ключ расшифрования? Если ключ хранится на том же сервере, он может быть найден атакующим и повторяется проблема 2.

Наиболее целесообразным решением может быть передача пароля в зашифрованном виде и хранение на сервере в качестве эталонного образа значения, вычисленного из пароля с помощью криптографически стойкой хеш-функции.

В этом случае последовательность действий при аутентификации, будет следующей:

– пароль зашифровывается на стороне пользователя;

– зашифрованное значение передается по сети серверу;

– сервер расшифровывает сообщение и восстанавливает пароль;

– сервер вычисляет хеш-значение, используя в качестве аргумента расшифрованный пароль;

– сервер сравнивает вычисленное хеш-значение со значением, хранящимся в базе эталонов. Если сравнение верно, то аутентификация успешно завершена. В противном случае выдается сообщение об ошибке аутентификации.

4) Субъект должен помнить свой многоразовый пароль. Атакующий может получить пароль непосредственно у пользователя с помощью методов социальной инженерии (*социальная инженерия* – совокупность приемов, методов и технологий создания такого пространства, условий и обстоятельств, которые максимально эффективно приводят к конкретному необходимому результату с использованием социологии и психологии).

5) Если пользователь сам выбирает пароль, то для простоты запоминания он может выбрать слова естественного языка или хорошо известные ему сочетания символов, например дату своего рождения. В этом случае атакующий

может успешно применить *словарную атаку* – перебор словарного справочника, составленного из наиболее часто используемых на практике ключей имен, фамилий, инициалов, номеров телефонов, дат рождения в различных способах их написания. При использовании уже хорошо известных словарей сложность подбора слова из словаря может быть эквивалентна сложности подбора случайной двоичной строки длиной 16 бит. Решением этой проблемы является использование случайных, достаточно длинных, паролей, ограничение срока действия паролей, ограничение числа попыток неправильного ввода пароля.

Более безопасными являются методы аутентификации с использованием одноразовых или динамически изменяемых паролей. В этом случае пароль действителен только для одного входа в систему. При каждом последующем запросе доступа требуется новый пароль.

Известны следующие методы применения одноразовых паролей:

– с использованием генераторов псевдослучайных чисел, одинаковых для субъекта аутентификации и проверяющей стороны;

– с использованием временных меток в системе единого времени;

– с использованием базы случайных паролей, одинаковой для субъекта аутентификации и проверяющей стороны.

В случае использования первого метода субъект аутентификации генерирует пароль, вычисляет его хеш-значение и отправляет проверяющей стороне. Проверяющая сторона генерирует свой пароль, вычисляет его хеш-значение и сравнивает с полученным.

В качестве генератора паролей может быть использовано функциональное преобразование $f(x)$, такое, что для известного пароля x невозможно (или вычислительно трудно) вычислить новый пароль $y = f(x)$ и, кроме того, по известным x и y невозможно (или вычислительно трудно) определить вид функции $f(x)$.

Во втором случае метки времени могут использоваться в качестве динамического параметра T функции $f_T(x)$. При этом субъект аутентификации и проверяющая система должны заранее договориться о стартовом значении x_0 и периодичности модификации функции $f_T(x)$, тогда новый пароль вычисляется как $y_i = f_T(x_{i-1})$. Такой прием позволяет защититься от атаки повторного использования пароля человеком посредине.

Метод использования общей базы случайных паролей широко используется в банковской системе. Банк выдает клиенту пронумерованный список одноразовых паролей и такой же список хранит у себя. При попытке входа клиента в банковскую систему она запрашивает пароль, одновременно сообщая клиенту его номер. Клиент вводит пароль с соответствующим номером.

Каждый пароль может быть использован только один раз. После использования всех паролей из списка список должен быть обновлен.

Аутентификация с использованием уникальных предметов. Метод предполагает предъявление субъектом аутентификации таких предметов, как личная печать, ключ от замка, файл данных, содержащих характеристику, смарт-карта и т. д. с последующей проверкой их подлинности системой.

В информационных системах наибольшее распространение получили карточки пассивные и активные. Пассивные карточки содержат только устройства памяти, а активные – устройство памяти и микропроцессор.

Пассивные карточки часто используют совместно с парольной аутентификацией. Для получения доступа в систему пользователь вводит свой идентификационный номер (PIN), который сравнивается системой с образом номера, хранящимся в запоминающем устройстве карточки. При их совпадении система предоставляет доступ. Такая аутентификация называется двухфакторной. Она позволяет защитить систему от несанкционированного вторжения путем использования карточки злоумышленником при ее потере или краже.

По сравнению с парольной аутентификацией применение пассивных карточек имеет следующие недостатки:

– затраты на производство карточек, устройств записи и чтения;

– необходимость поддержания организационных мер по учету и распределению карточек;

– необходимость действий клиента по защите от попадания карточек к злоумышленнику.

Активные карточки, имеющие собственный микропроцессор, более универсальны, могут быть использованы для более широкого спектра задач, в том числе таких, как выполнение финансовых операций, организация защищенных информационных каналов и т. д.

Недостатком является более высокая стоимость по сравнению с пассивными.

Аутентификация с использованием биометрических данных. Свойствами субъекта, используемыми для аутентификации, являются:

– отпечатки пальцев;

– геометрия руки;

– рисунок радужной оболочки глаза;

– форма ушной раковины;

– лицо;

– голос;

– моторика рук (совокупность скоординированных действий, направленных на выполнение мелких и точных движений, например набор текста на клавиатуре);

– рукописная подпись.

Отпечатки пальцев. Вероятность обнаружения одинаковых отпечатков у разных людей (биологическая повторяемость) с учетом погрешностей современных сканеров составляет порядка 10^{-7}. Специализированные устройства считывания – сканеры отпечатка – компактны и относительно недороги. Метод является наиболее широко применяемым в биометрических паспортах, устройствах распознавания владельцев персональных компьютеров, смартфонов и т. д. Вероятность ошибки может резко возрастать при загрязнении или повреждении кожи пальцев. Требует непосредственного контакта со сканером.

Геометрия руки. Вероятность биологического повторения значительно выше – порядка 10^{-2}. Применяется в случае затруднений считывания отпечатка пальцев. Требует непосредственного контакта со сканером.

Радужная оболочка глаза. Вероятность биологического повторения очень низкая – порядка 10^{78}. Сканеры более дорогие. Требует расстояния от сканера до глаза порядка нескольких сантиметров.

Форма ушной раковины. Перспективное направление, пока не нашедшее широкого распространения, возможно использование совместно с распознаванием лица.

Лицо. Главное достоинство – возможность распознавания человека с расстояния до нескольких десятков метров. В определенных условиях вероятность распознавания порядка 3 %. В качестве сканера могут быть использованы универсальные устройства, например веб-камеры.

Голос. Используется в телекоммуникационных приложениях передачи речи. Не требует специализированного оборудования. Используются универсальные микрофоны и звуковые платы. Вероятность распознавания порядка 2 %.

Моторика рук. Используются параметры скорости и интервалов между нажатиями клавиш. Вероятность распознавания невысока – порядка 5–10 %.

Распознавание подписи. Статическое изображение рукописной подписи подделать довольно легко. Однако трудно подделать скорость написания и давление пишущего предмета. Предполагается использовать наряду с обычными паролями в банковских приложениях. Имеются проблемы нестабильности движения подписанта в различном психологическом состоянии и при болезни. В

настоящее время является недостаточно эффективным способом аутентификации.

Биометрическая аутентификация, несмотря на свою привлекательность, подвержена ряду существенных угроз.

Первая группа угроз связана с вероятностным характером измерений, которые могут привести к ошибкам первого и второго рода:

– ошибки первого рода – успешная аутентификация лица, не являющегося полномочным пользователем;

– ошибки второго рода – неправомерный отказ в доступе правомочному пользователю. Защита от этих угроз заключается в совершенствовании методов распознавания образов и чувствительных элементов сканеров.

Вторая группа угроз связана со статичностью базы образов субъекта аутентификации. Это дает возможность мимикрии атакующего под правомочного пользователя путем изготовления искусственных объектов (артефактов), позволяющих имитировать признаки субъекта аутентификации. Защита от такой угрозы заключена в контроле среды, затрудняющем использование артефактов (например, видеонаблюдение).

Возможно зашумление канала передачи данных от субъекта к серверу аутентификации, что приводит к увеличению вероятности ошибок второго рода.

Требуются повышенные меры защиты базы данных, содержащих эталоны биометрических образцов, с целью недопущения их разглашения и модификации. Разглашение биометрических данных может привести к дальнейшей невозможности их использования для аутентификации в связи с невозможностью их смены (в отличие от паролей).

С целью достижения снижения риска ошибок все более широкое распространение находит многофакторная аутентификация, однако этот путь требует дополнительных затрат.

1.2. Задание для самостоятельного выполнения

1) Разработать программу на языке C++, реализующую следующие функции:

– генерация строки с заданной пользователем длиной, состоящей из символов алфавита в соответствии с вариантом задания (использовать функции rand( ), srand( ) и инициализацию от таймера);

– поверка равномерности распределения символов путем визуализации частотного распределения;

– вычисление среднего времени подбора пароля, выбираемого из сгенерированной строки.

2) Построить график зависимости среднего времени подбора пароля от его длины.

3) Дать практические рекомендации по выбору пароля исходя из предположений об алфавите пароля; ценности информации, доступ к которой защищается с помощью этого пароля; производительности вычислительного средства атакующего и времени атаки.

Варианты алфавита для генерации пароля:

1) Латиница строчные.

2) Латиница строчные и прописные.

3) Буквы русского языка строчные.

4) Буквы русского языка строчные и прописные.

5) Арабские цифры.

6) Латиница строчные и арабские цифры.

7) Латиница строчные, прописные и арабские цифры.

8) Буквы русского языка строчные и арабские цифры.

9) Буквы русского языка строчные, прописные и арабские цифры.

10) Все символы таблицы ASCII.

Вариант выбирается в соответствии с номером студента в рамках группы. Если студентов в группе больше, чем вариантов в списке, то варианты снова повторяются начиная с единицы.

2. ПРОСТЕЙШИЕ КРИПТОГРАФИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ

2.1. Теоретические сведения

С древнейших времен криптография использовалась для защиты военной и дипломатической связи. Необходимость защиты правительственной связи вполне очевидна, и до недавнего времени широкое применение криптографии было почти исключительно правом государства. В настоящее время большинство правительств контролирует или сами исследования в этой области, или по крайней мере производство криптографического оборудования и программного обеспечения. В Республике Беларусь деятельность по технической и (или) криптографической защите информации является лицензируемой. С началом информационного века возникла срочная необходимость использования криптографии в частном секторе. Сегодня огромное количество конфиденциальной информации (такой, например, как персональные данные, истории болезней, юридические документы, данные о финансовых операциях) передается между ЭВМ по обычным линиям связи. Поэтому возникает необходимость обеспечения секретности и подлинности подобной информации.

Криптология (происходит от греч. kryptos – тайный и logos – слово) – наука о шифровании и дешифровании.

Шифрование – метод, используемый для преобразования исходных данных в зашифрованный текст (криптограмму) для того, чтобы они могли быть прочитаны только пользователем, обладающим соответствующим ключом шифрования для расшифрования содержимого.

Криптология делится на две части: криптографию (шифрование) и криптоанализ. Киптография занимается разработкой методов обеспечения секретности и (или) аутентичности (подлинности) сообщений. Криптоанализ предназначен для решения обратной задачи – раскрытия (взлома) шифра с целью получения возможности несанкционированного чтения зашифрованного сообщения или осмысленной подделки такого сообщения. Кроме того, криптоанализ применяют при исследовании шифров с целью улучшения их свойств, например криптографической стойкости.

Исходное сообщение, подлежащее зашифрованию, называется открытым текстом сообщения или просто открытым текстом, а результат зашифрования – шифрованным текстом сообщения – зашифрованным текстом, шифртекстом, или криптограммой.

Текстом называют упорядоченный набор элементов алфавита.

Алфавит – конечное множество X_n, используемых для кодирования информации знаков, где n – мощность алфавита.

В современных информационных системах используют большое разнообразие алфавитов, например:

– алфавит X_{26} – 26 букв латинского алфавита;

– алфавит X_{32} – 32 буквы алфавита русского языка;

– алфавит X_{44} – 43 буквы алфавита русского языка, знаки препинания и пробел;

– алфавит X_{256} – символы стандартных кодов ASCII и КОИ-8;

– алфавит $X_2 = \{0,1\}$ – двоичный алфавит;

– шестнадцатеричный алфавит $X_{16} = \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$ и т. д.

Для обеспечения процессов зашифрования и расшифрования в симметричных системах шифрования используют некоторую секретную информацию, называемую ключом.

Процедуру, с помощью которой из открытого текста получают криптограмму и наоборот, называют криптографическим преобразованием.

Криптографическая система представляет собой семейство T преобразований текста. Члены этого семейства индексируются каким-нибудь символом, например k. Параметр k является ключом. Пространство ключей K – это набор возможных значений ключа.

Как правило (но не всегда), создатель зашифрованного сообщения передает этот секретный ключ по защищенному каналу человеку (или машине) – получателю.

Под защищенным каналом понимают канал связи, недоступный для наблюдения злоумышленнику.

В современных информационных системах ключ – это двоичная строка, которая может быть интерпретирована набором символов другого алфавита.

Цель криптографической системы чаще всего заключается в том, чтобы зашифровать осмысленный исходный текст, получив в результате зашифрованный текст, бессмысленный с точки зрения постороннего наблюдателя. Получатель, которому он предназначен, должен быть способен расшифровать этот криптотекст, восстановив таким образом соответствующий ему открытый текст. При этом противник (криптоаналитик) должен быть неспособен раскрыть исходный текст.

Способность криптосистемы противостоять получению исходного текста из криптограммы без знания ключа называют стойкостью (криптостойкостью).

Чаще всего в качестве показателей криптостойкости используют:

– мощность ключевого поля (количество всех возможных ключей);

– среднее время, необходимое для дешифрования криптотекста.

В системах общего использования, а такими являются большинство реальных информационных систем, криптоаналитик почти всегда может тем или иным способом получить алгоритм шифрования. Поэтому при проектировании современных шифров исходят из предположения, что заведомо криптоаналитику известен алгоритм шифрования за исключением значения секретного ключа.

Общепринятым допущением в криптографии является наличие у криптоаналитика противника полного текста криптограммы. В этом случае говорят, что криптоаналитик может осуществить атаку на основании известного криптотекста.

Если криптоаналитик имеет некоторое количество открытого текста и соответствующего ему зашифрованного текста, образованного с использованием секретного ключа, то он может предпринять атаку на основе открытого текста.

Если криптоаналитик способен ввести свой открытый текст и получить криптограмму, образованную с помощью секретного ключа, то он может произвести анализ на основе выбранного открытого текста.

Если предположить, что криптоаналитик противника может подставить фиктивные криптограммы и получить текст, в который они превращаются при шифровании, то он может произвести анализ на основе выбранного шифртекста.

Если принять оба последних допущения, то возможен анализ на основе выбранного текста.

Разработчики большинства современных шифров обеспечивают их стойкость к анализу на основе выбранного открытого текста даже в том случае, когда предполагается, что криптоаналитик противника сможет прибегнуть к анализу только на основе шифртекста.

Многие приемы, используемые в качестве элементов современных шифров, были разработаны сотни лет назад. Такими приемами являются простейшие криптографические преобразования подстановки (или замены) и перестановки.

Далее рассмотрим эти приемы на примерах наиболее известных исторических шифров.

Первым шагом в криптографии, были моноалфавитные шифры, использующие простые подстановки. Ярким историческим примером шифров этого семейства является шифр Цезаря.

Шифр Цезаря – один из наиболее известных и простых методов шифрования, использованный Гаем Юлием Цезарем для секретной переписки.

Шифр Цезаря – это один из шифров подстановки, в котором каждый символ открытого текста заменяется символом, находящимся на некотором, определяемом ключом, постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом вправо на 3 А заменяют на Г, Б станет Д и т. д. В табл. 2.1 приведен алфавит подстановки.

Таблица 2.1

Алфавит подстановки

Исходный алфавит	абвгдеёжзийклмнопрстуфхцчшщъыьэюя
Зашифрованный алфавит	гдеёжзийклмнопрстуфхцчшщъыьэюяабв

При использовании простейших шифров подобного типа, как правило, перед зашифрованием из текста удаляли пробелы, знаки препинания и использовали только строчные или прописные буквы. Пример шифрования приведен в табл. 2.2.

Таблица 2.2

Пример шифрования

Исходный текст	съешьжеещёэтихмягкихфранцузскихбулокдавыпейчаю
Зашифрованный текст	фэзыяйззьиахлшпвёнлшчугрщцкфнлшдцоснжгеютзмъгб

Математическая модель шифра Цезаря

Каждой букве алфавита сопоставляют его порядковый номер, начиная с 0. Тогда процедуры зашифрования и расшифрования описываются следующими выражениями:

$$y = (x + k) \bmod n;$$

$$x = (y - k + n) \bmod n;$$

где x – символ открытого текста;

y – символ зашифрованного текста;

n – мощность алфавита;

k – ключ;

(*) mod n – остаток от деления * на n.

Важно отметить, что суперпозиция двух шифрований на ключах k1 и k2 является шифрованием на ключе k3 = k1 + k2. Множество шифрующих преобразований шифра Цезаря образует группу Z_n.

Шифр Цезаря может быть легко взломан даже в случае, когда взломщик знает только зашифрованный текст. Рассмотрим две ситуации:

1) Взломщик знает (или предполагает), что использовался простой шифр подстановки, но не знает, что это схема Цезаря.

2) Взломщик знает, что использовался шифр Цезаря, но не знает значение сдвига.

В первом случае шифр может быть взломан теми же методами, что и для простого шифра подстановки, такими как частотный анализ и т. д. Используя эти методы, взломщик, вероятно, быстро заметит регулярность в решении и поймет, что используемый шифр – это шифр Цезаря.

Во втором случае взлом шифра является более простым. Существует ровно n вариантов значений сдвига (для английского языка – 26, для русского – 33). Они легко могут быть проверены методом «грубой силы». В табл. 2.3 приведен пример для зашифрованного текста «EXXEGOEXSRGI»; открытый текст немедленно опознается визуально в четвертой строке.

Таблица 2.3

Пример для зашифрованного текста «EXXEGOEXSRGI»

Сдвиг	Открытый текст
0	exxegoexsrgi
1	dwwdfndwrqfh
2	cvvcemcvqpeg
3	buubdlbupodf
4	attackatonce
5	zsszbjzsnmbd
…	…
25	fyyfhpfytshj

Другой подход к применению метода «грубой силы» для взлома – частотный анализ. Изобразив диаграммой частоты встречи букв в зашифрованном тексте и зная ожидаемое распределение букв для обычного текста на рассматриваемом языке, можно легко определить сдвиг, взглянув на смещение некоторых характерных черт на диаграмме. Например, в тексте на английском языке частот букв E, T (наиболее частых) и Q, Z (более редких) особенно различается.

Полиалфавитные шифры

Поскольку шифр моноалфавитной замены оказался легко взламываемым, особенно в связи с открытием частотного анализа, то возникла потребность в его совершенствовании. Следующим шагом в развитии стали шифры полиалфавитной замены. В середине XV века итальянский ученый Леон

Баттиста Альберти предложил использовать вместо одного алфавита несколько, меняя алфавит при зашифровании каждого последующего символа исходного текста по некоторому правилу. Однако сделать свой шифр достаточно совершенным он не смог. Дальнейшее развитие полиалфавитные шифры получили в работе французского криптографа Блеза де Виженера «Трактат о шифрах». Основное отличие от предшественников заключалось в правиле выбора алфавита подстановки для очередной буквы. Было предложено использовать в качестве ключа другой открытый текст.

Шифр Виженера. Метод является простой формой многоалфавитной (полиалфавитной) замены. Многоалфавитная замена определяется ключом, содержащим не менее двух вариантов замен.

В шифре Виженера в качестве ключа используется секретное слово.

В полиалфавитном шифре несколько одноалфавитных шифров применяются циклически.

При зашифровании вручную может использоваться таблица алфавитов, называемая квадрат Виженера. Для латинского алфавита квадрат Виженера состоит из 26 строк по 26 символов (рис. 2.1). Каждая следующая строка образуется сдвигом предыдущей на одну позицию. Таким образом, в таблице получается 26 алфавитов. Для зашифрования каждой последующей буквы циклически используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
BCDEFGHIJKLMNOPQRSTUVWXYZA
CDEFGHIJKLMNOPQRSTUVWXYZAB
DEFGHIJKLMNOPQRSTUVWXYZABC
EFGHIJKLMNOPQRSTUVWXYZABCD
FGHIJKLMNOPQRSTUVWXYZABCDE
GHIJKLMNOPQRSTUVWXYZABCDEF
HIJKLMNOPQRSTUVWXYZABCDEFG
…
YZABCDEFGHIJKLMNOPQRSTUVWX
ZABCDEFGHIJKLMNOPQRSTUVWXY

Рис. 2.1. Квадрат Виженера

Пусть имеется исходный текст: SECRETLETTERFORYOU.

Шифровальщик записывает ключевое слово (например, PRIVATE) циклически до тех пор, пока его длина не сравняется с длиной исходного текста: PRIVATEPRIVATEPRIVA.

Первая буква исходного текста S зашифровывается последовательностью P, задаваемой первой буквой ключа. Первая буква H зашифрованного текста находится на пересечении строки P и столбца S в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста V получается на пересечении строки R и столбца E. Остальная часть исходного текста зашифровывается аналогично.

Исходный текст: SECRETLETTERFORYOU

Ключ: PRIVATEPRIVATEPRIVA

Зашифрованный текст: HVKMEMPTKBZRYSGPWP

Расшифрование. Находим в таблице Виженера строку, соответствующую первой букве ключевого слова. В этой строке находим первую букву зашифрованного текста. Столбец, в котором находится эта буква, соответствует первой букве исходного текста. Следующие буквы зашифрованного текста расшифровываются аналогично.

Пусть n – мощность алфавита исходного текста, x_i – буквы исходного текста, k_i – буквы ключа, тогда шифрование Виженера может быть описано следующим образом:

$$y_i = x_i + k_i \pmod{n}.$$

Расшифрование:

$$x_i = y_i - k_i \pmod{n}.$$

Шифр Виженера позволяет скрыть частотные характеристики букв исходного текста, однако не полностью. Главная проблема шифра Виженера заключается в циклическом повторении ключа.

В 1863 году Фридрих Касиски нашел способ вскрытия шифра Виженера с коротким кодовым словом. А именно короткие кодовые слова, как правило, и использовали. В случае когда шифрование производилось с помощью соразмерного с открытым текстом ключа, кодовая фраза может быть подобрана при условии, что она состоит из осмысленных слов.

Криптоанализ шифра включает два этапа:

1) Определение длины ключа, например, методом прореживания. Для этого берут текст, содержащий каждую вторую букву криптограммы, затем каждую третью и т. д. Для каждого из прореженных текстов определяют частотное распределение букв. Эту процедуру выполняют до тех пор, пока не получат частотное распределение, существенно отличающееся от равномерного. В этом случае можно с высокой степенью вероятности говорить о найденной длине ключа.

2) Криптоанализ совокупности из j шифров Цезаря, где j – найденная длина ключа.

В ходе Первой мировой войны майор армии США Джозеф Моборн предложил использовать одноразовый шифрблокнот. На каждую страницу блокнота наносили последовательность из случайных букв, которая использовалась в качестве одноразового ключа для шифрования сообщения. Такой же блокнот создавался для получателя зашифрованного сообщения, чтобы он мог его расшифровать. После использования соответствующая страница обоих блокнотов уничтожалась. Этот шифр достаточно успешно использовался еще более пятидесяти лет. Однако с ростом объемов передаваемой информации он быстро утрачивал широту применения в связи с трудоемкостью, а следовательно, и низкой скоростью процесса шифрования.

Возникла потребность в механизации процесса шифрования. В конце XIX века Томас Джефферсон предложил одну из первых роторных машин, облегчающую процесс полиалфавитного шифрования. Машина получила название цилиндр Джефферсона. Роторные машины получили широкое распространение в течение XX века в связи со Второй мировой войной. Наиболее известными из роторных шифровальных машин стали немецкая Enigma, американская Sigaba и английская Typex.

«Энигма» – семейство переносных электромеханических роторных машин, разработанных в Германии и применявшихся с 20-х годов XX века в коммерческих и военных целях. Наибольшую известность получила военная модель.

Все роторные машины построены по одному принципу, который мы рассмотрим на примере «Энигмы». «Энигма» состоит из комбинации двух подсистем: механической и электрической. Механическая подсистема включает в себя клавиатуру, набор вращающихся дисков (роторов), расположенных на общем валу и прилегающих к нему, и ступенчатый механизм, смещающий один или несколько роторов при нажатии клавиши. Электрическая часть состоит из электрической схемы, соединяющей между собой клавиатуру, коммутационную панель, лампочки и роторы. Роторы соединяются скользящими контактами.

Принцип функционирования заключается в следующем: при каждом нажатии клавиши самый правый ротор сдвигается на одну позицию, а при выполнении некоторых условий сдвигаются и другие роторы. Движение роторов приводит к различным криптографическим преобразованиям при каждом следующем нажатии клавиши на клавиатуре.

Механические части двигались, замыкая контакты и образуя меняющийся электрический контур, который собственно и реализовывал процесс

шифрования. При нажатии клавиши на клавиатуре происходило замыкание электрического контура, ток проходил через различные цепи и в результате включал одну лампочку из набора, отображавшую букву криптограммы. Например, при зашифровании сообщения, начинающегося с ANX…, оператор вначале нажимал клавишу A – загоралась лампочка Z, то есть Z и становилась первой буквой криптограммы. Далее оператор нажимал N и продолжал шифрование таким же образом далее. Сформированный таким образом шифртекст затем передавали по каналу связи (например, по радио или телеграфу).

Изменение электрической цепи (при зашифровании каждой очередной буквы исходного текста), через которую шел ток вследствие вращения роторов, позволяло реализовать многоалфавитный шифр подстановки, достаточно надежный для своего времени.

Каждому экземпляру шифровальной машины на определенный интервал времени задавались определенные настройки. Чтобы сообщение было идентично зашифровано и расшифровано, машины отправителя и получателя должны были быть одинаково настроены. Эти настройки оговаривались заранее и записывались в специальных шифровальных книгах.

Ключ шифрования «Энигмы» включает следующие параметры:

– расположение роторов: выбор роторов и их расположение;

– первоначальные позиции роторов: выбранные оператором, различные для каждого сообщения;

– настройка колец: позиция алфавитного кольца, совпадающая с роторной схемой;

– настройки штепселей: соединения штепселей на коммутационной панели.

Система обеспечивала криптостойкость даже в том случае, если к криптоаналитику попадала в руки сама машина. При этом ему становились известны роторы, но все остальные параметры настройки – неизвестны. Количество вариантов остальных настроек достаточно велико. Разработчики были уверены в неуязвимости системы.

Однако уже в 1929 году первые образцы «Энигмы» попали к польским криптоаналитикам и атака началась.

Математики сконцентрировали основные усилия на анализе уязвимости протокола обмена сообщениями, а именно – на повторении ключа сообщения. Из ежедневных сообщений выбирались первые шесть букв и на их основе составлялась таблица соответствия. Особенность полного варианта таблицы заключалась в том, что пока дневной ключ остается без изменений, содержимое

таблицы также не меняется. Можно было бы составить каталог таблиц, однако их количество равно 26!, что в тех условиях (без применения вычислительной техники) было невозможным в обозримое время. Далее криптоаналитики сумели выделить из таблиц некоторые шаблоны или найти некоторые структурные закономерности. Конкретные буквы зависели от дневной настройки «Энигмы» полностью, количество цепочек и букв в них задавалось только настройками роторов. Так как количество роторов первой модели машины было равно трем (порядок размещения любой), а начальная настройка состояла из трех букв латинского алфавита, то число вариантов было равно 3!*26**3=105456, что значительно меньше чем 26!. Это позволило составить каталог, содержащий все возможные цепочки. Результат годовой работы криптоаналитиков дал на короткое время возможность читать некоторые сообщения.

В свою очередь немецкие криптологи постоянно совершенствовали протоколы смены ключа, повышая частоту смены настроек, увеличивали количество роторов вплоть до восьми, и на длительное время криптоаналитики снова стали перед неразрешимой задачей.

С 1939 года после оккупации Польши Германией работа по взлому «Энигмы» была передана в центр британской разведки Station X. Одним из основных теоретиков группы криптоаналитиков был Алан Тьюринг. В августе 1940 года была построена криптоаналитическая машина Bombe. Со временем их работало более 200, что позволило взламывать до трех тысяч зашифрованных сообщений в день.

Шифр Скитала – один из первых шифров перестановки, использованный спартанцами и афинянами в III веке до нашей эры. Исторически идея шифра заключена в следующем: на некоторый цилиндр наматывается виток к витку узкая лента пергамента, на который поперек витков наносится текст шагом в одну букву. Затем лента отдельно от цилиндра пересылалась с гонцом. Для расшифрования сообщения требовался цилиндр такого же диаметра.

Метод вскрытия этого шифра приписывается Аристотелю. Он предложил заточить конусом длинный брус, обернуть вокруг него ленту и начать сдвигать ее по конусу от малого диаметра до самого большого. В том месте, где диаметр конуса совпадал с диаметром Скиталы, буквы складывались в слоги и слова. После этого требовалось изготовить жезл нужного диаметра и читать сообщение.

Дальнейшим развитием шифра Скитала стал метод простых шифрующих таблиц.

Алгоритм шифрования. Открытый текст построчно, начиная с верхней строки, вписывают в таблицу, состоящую из m строк и n столбцов. Если открытый текст содержит больше букв, чем произведение m на n, то его

предварительно разбивают на блоки соответствующего размера. Если текст меньше размера таблицы, то незаполненные клетки заполняют произвольными символами. Криптограмму получают путем последовательного считывания букв из таблицы по столбцам. Ключом шифрования является размер таблицы. Перед зашифрованием также полезно удалить из исходного текста пробелы и знаки препинания.

Пример шифрования. Пусть имеется таблица из 5 столбцов и 4 строк. Исходный текст «ЭТО ШИФР ДЕВНЕЙ СПАРТЫ» вписываем в таблицу по строкам, начиная с верхней, предварительно исключив из текста пробелы.

ЭТОШИ
ФРДРЕ
ВНЕЙС
ПАРТЫ

После прочтения по столбцам, начиная с левого, получаем криптотекст «ЭФВПТРНАОДЕРШРЙТИЕСЫ».

Алгоритм расшифрования. Зашифрованный текст вписывают в таблицу, состоящую из m строк и n столбцов по столбцам, начиная с левого. Если открытый текст содержит больше букв, чем произведение m на n, то его предварительно разбивают на блоки соответствующего размера. Считывают текст по строкам, начиная с верхней. Если в конце последнего расшифрованного блока имеется текст, не имеющий смысла, то его отбрасывают. Четкий критерий наличия смысла отсутствует. Поэтому решение отдается на выбор расшифровывающего человека.

В отличие от оригинального шифра Скитала, в котором атакующему неизвестен только один параметр ключа – диаметр цилиндра, такой метод кажется более надежным, так как неизвестны два параметра – количество строк и столбцов таблицы. Очевидно, что такой шифр может быть легко взломан простым перебором размеров таблицы, если атакующему известен язык исходного текста.

2.2. Задание для самостоятельного выполнения

1) Реализовать в виде программы шифр (зашифрование и расшифрование) в соответствии с вариантом. Язык исходного текста русский или английский по выбору исполнителя.

2) Реализовать в виде программы атаку полным перебором ключа, используя для оценки правильности выбора ключа визуальный метод или исходный текст для автоматического сравнения результата дешифрования.

3) Оценить криптографическую стойкость реализованного шифра.

4) Предложить варианты усложнения шифра. Предложенные варианты оформить в виде алгоритма.

Варианты для реализации.

1) Шифр Цезаря.

2) Шифр Виженера.

3) Шифр Скитала.

4) Шифр перестановки, использующий простые (прямоугольные) таблицы.

Вариант выбирается в соответствии с номером студента в рамках группы. Если студентов в группе больше, чем вариантов в списке, то варианты снова повторяются, начиная с единицы.

3. РЕЖИМЫ ПРИМЕНЕНИЯ БЛОЧНЫХ ШИФРОВ

3.1. Теоретические сведения

В настоящее время для обеспечения защиты данных от несанкционированного чтения при передаче, хранении и обработке в информационных системах наиболее широкое применение нашли блочные шифры, обеспечивающие достаточно высокую практическую стойкость при использовании относительно не длинных (по сравнению с совершенно секретными шифрами) ключей.

Блочными шифрами называют разновидность симметричных шифров, в которых исходный текст перед зашифрованием разбивают на блоки фиксированной длины (по 64–256 бит) и затем зашифровывают их с помощью выбранного алгоритма шифрования и секретного ключа независимо друг от друга.

Такой шифр является более практичным по сравнению с совершенно секретными шифрами в связи с необходимостью обмена ключами, как правило, значительно более короткими, чем исходное сообщение. В блочных шифрах используют практически одинаковые (за исключением порядка выбора ключа и порядка действий) алгоритмы для зашифрования и расшифрования, что упрощает их программную и аппаратную реализацию.

Функции зашифрования E и расшифрования D могут быть представлены в следующем виде:

$$\mathrm{E_K(M)} \coloneqq \mathrm{E(K,M)} : \{0,1\}^{\mathrm{k}} \times \{0,1\}^{\mathrm{n}} \to \{0,1\}^{\mathrm{n}};$$

$$\mathrm{D_K(C)} \coloneqq \mathrm{D(K,C)} : \{0,1\}^{\mathrm{k}} \times \{0,1\}^{\mathrm{n}} \to (0,1)^{\mathrm{n}};$$

где M – блок исходных данных размером n бит;

K – ключ размером k бит;

C – блок зашифрованного текста (криптограммы) размером n бит.

Функция расшифрования является обратной функцией зашифрования:

$$\mathrm{D} = \mathrm{E}^{-1},$$

$$\forall \mathrm{K} : \mathrm{D_K}\big(\mathrm{E_K(M)}\big) = \mathrm{M} \text{ и}$$

$$\mathrm{E_K}\big(\mathrm{D_K(C)}\big) = \mathrm{C}.$$

Поскольку шифр является симметричным, то для зашифрования и расшифрования используется один и тот же ключ.

Для обеспечения простоты реализации и высокой скорости шифрования блочные шифры строятся как итеративные, то есть блоки исходного текста фиксированной длины преобразуются в блоки шифртекста такой же длины с помощью циклически повторяемых обратимых преобразований, называемых

раундовыми преобразованиями. На каждом раунде используется ключ раунда, полученный из ключа шифрования с использованием функции расширения ключа:

$$C_i = R_{K_i}(C_{i-1}),$$

где C_i – выход i-го раунда;

C_0 – вход первого раунда, являющийся блоком исходного текста;

K_i – ключ i-го раунда, полученный из исходного ключа шифрования.

В случае, если размер исходного текста не является кратным размеру блока, то последний блок дополняется до полного путем добавления произвольной информации, которая после расшифрования просто отбрасывается.

Каждый раунд шифрования представляет собой комбинацию преобразований подстановки и перестановки, что позволяет обеспечить выполнение следующих требований:

– рассеивание информации – распространение влияния одного бита блока исходного текста и одного бита ключа на все биты блока криптотекста с целью исключения возможности восстановления исходного текста или ключа по частям;

– перемешивание информации – усложнение обнаружения зависимости между ключом и блоком криптотекста.

Модули, реализующие преобразование подстановки принято называть S-бокс (сокращение от английского слова Substitution), модули перестановки называют P-бокс (P – Permutation).

Сочетание этих преобразований образует SP-сеть (Substitution-Permutation network, подстановочно-перестановочная сеть). Пример SP-сети для трех раундов приведен на рис. 3.1.

SP-сеть представляет собой «сэндвич» из двух типов (P и S) чередующихся и многократно повторяющихся слоев.

P-слой состоит из P-блока большой разрядности.

S-слой состоит из нескольких S-блоков меньшей разрядности.

Частным случаем SP-сети является сеть Фейстеля.

DES – Data Encryption Standard. Первым официальным стандартом симметричного блочного шифра, принятым правительством США в 1977 году, стал алгоритм DES.

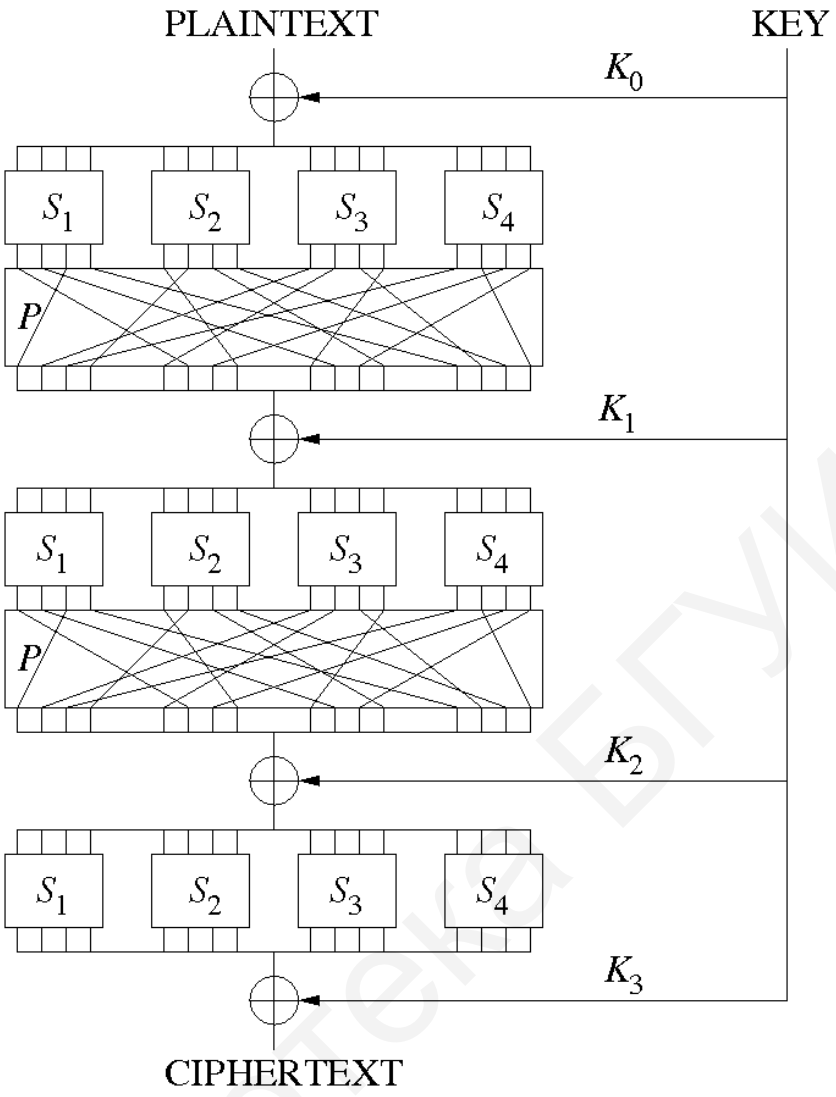

Рис. 3.1. Пример SP-сети для трех раундов

Размер блока равен 64 битам, размер ключа 64 бита, из них собственно ключом, используемым для шифрования, являются только 56 бит, а оставшиеся восемь бит используются для проверки целостности ключа методом контроля четности. Архитектура шифра – классическая сеть Фейстеля.

Алгоритм зашифрования DES включает в себя начальную перестановку, 16 раундов шифрования (циклов преобразования Фейстеля) и конечную перестановку.

Схема алгоритма шифрования DES представлена на рис. 3.2.

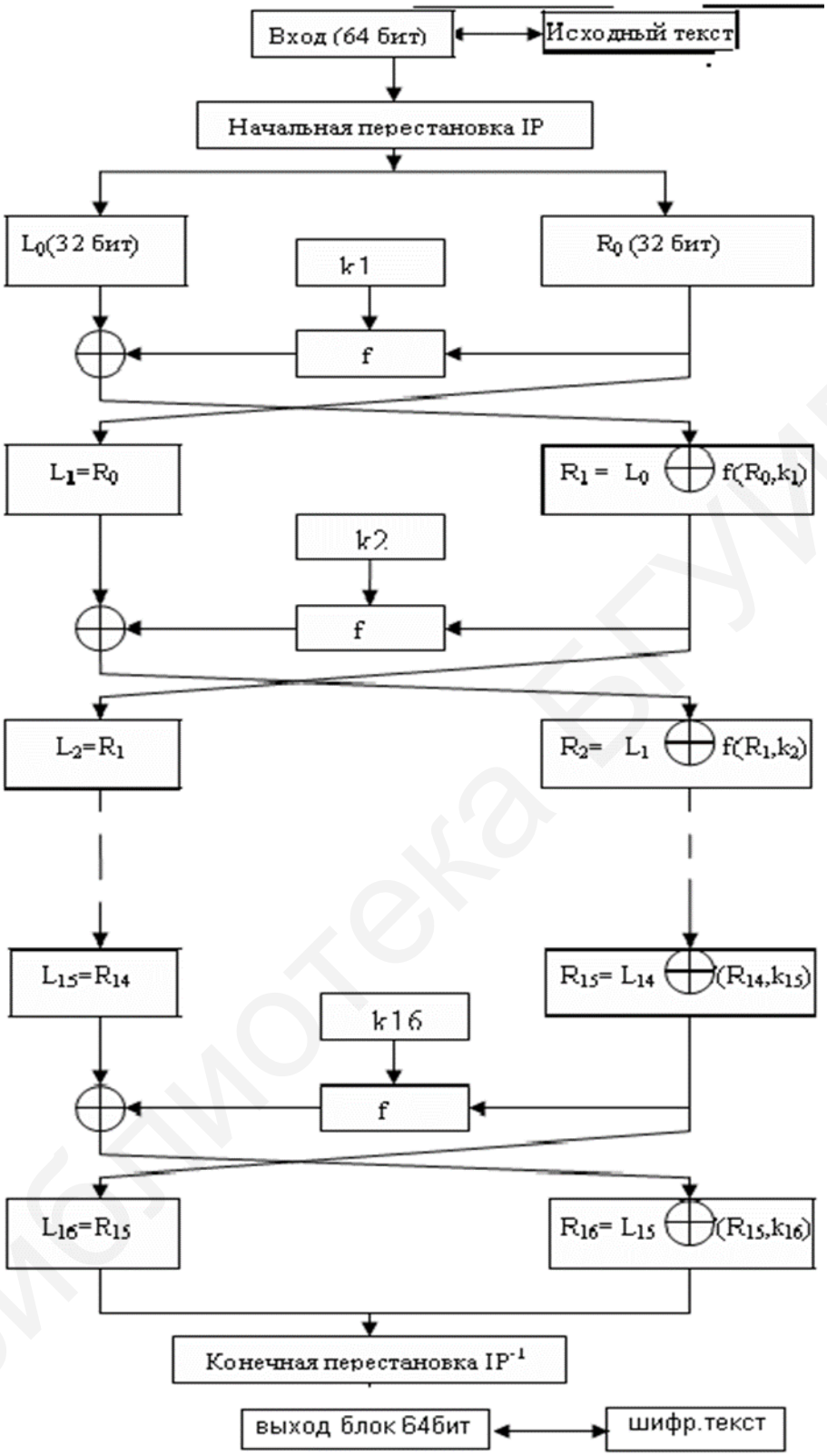

Рис. 3.2. Схема алгоритма шифрования DES

Рассмотрим далее один цикл преобразования Фейстеля.

Блок исходного текста IP(T) на две части по 32 бита L_0, R_0 – старшие и младшие биты соответственно:

$$T_0 = IP(T) = L_0R_0.$$

Пусть $T_{i-1} = L_{i-1}R_{i-1}$ результат (i-1) итерации, тогда результат i-й операции $T_i = L_iR_i$ вычисляют следующим образом:

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i).$$

Левой половине присваивают значение правой половины вектора входного значения предыдущего раунда $L_{i-1}R_{i-1}$.

Правой половине R_i присваивают значение результата побитового сложения по модулю 2 (операция XOR) L_{i-1} и $f(R_{i-1}, k_i)$.

Функция f называется основной функцией шифрования сети Фейстеля и включает в себя следующие операции:

– расширение правой часть блока R_i до 48 бит с помощью таблицы, определяющей перестановку бит и расширение на 16 бит;

– результат предыдущей операции суммируется по модулю 2 с 48-битным подключом k_i раунда;

– 48-битный результат предыдущей операции делится на 8 частей по 6 бит, над каждой частью выполняется подстановка с помощью S-боксов, причем S-бокс имеет 6-битный вход и 4-битный выход;

– результат подстановки – 8 блоков по 4 бита объединяют в 32-битное значение и выполняют битовую перестановку P, не зависящую от ключа.

Такой выбор функции шифрования обеспечивает высокую зависимость всех битов результата от битов исходного значения и ключа и максимальное переупорядочивание битов.

Алгоритм расшифрования DES заключается в выполнении всех действий зашифрования, но в обратном порядке. Используется обратное преобразование Фейстеля:

$$R_{i-1} = L_i;$$
$$L_{i-1} = R_i \oplus f(L_i, k_i).$$

Алгоритм DES более 20 лет успешно противостоял атакам криптоаналитиков.

Однако уже в 1997 году шифр DES был взломан программистами-добровольцами с помощью сети персональных компьютеров, взаимодействующих через Интернет. Управлял сетью компьютер с процессором Intel Pentium с тактовой частотой 90 МГц и оперативной памятью 16 Мбайт. DES перестал быть надежной защитой конфиденциальных данных.

На смену ему, в результате победы на конкурсе, проведенном для выбора алгоритма нового стандарта шифрования, пришел алгоритм Rijndael. А новый стандарт, принятый 26 мая 2002 года, получил название AES.

AES – Advanced Encryption Standard. AES – стандарт шифрования, основанный на алгоритме Rijndael, разработанном двумя бельгийскими криптографами Винсентом Рейменом и Джоаном Дейменом.

Алгоритм Rijndael представляет собой итеративный блочный шифр с возможностью выбора длины блока и длины ключа 128, 192 или 256 бит. Для стандарта AES длина блока только 128 бит, а длина ключа 128, 192 или 256 бит.

В стандарте используются следующие обозначения:

– Block – последовательность битов, из которых состоит input, output, State и Round Key;

– CipherKey – секретный ключ, который используется процедурой Key Expansion для генерации набора ключей раундов шифрования (RoundKeys) и может быть представлен как прямоугольный массив байтов, имеющий четыре строки и Nk колонок;

– Ciphertext – выходные данные алгоритма шифрования;

– KeyExpansion – процедура генерации RoundKeys из CipherKey;

– RoundKey – RoundKeys получают из CipherKey использованием процедуры KeyExpansion. Они применяются к State при зашифровании и расшифровании;

– State – промежуточный результат шифрования, который может быть представлен как прямоугольный массив байтов имеющий четыре строки и Nb столбцов;

– S-box – фиксированная (не зависящая от ключа) нелинейная таблица замен, использующаяся в нескольких преобразованиях замены байтов и в процедуре KeyExpansion;

– Nb – число столбцов (32-битных слов), составляющих State Nb = 4 (для Rijndael Nb = 4, 6 или 8);

– Nk – число 32-битных слов, составляющих ключ CipherKey, Nk = 4, 6, или 8);

– Nr – число раундов, которое является функцией Nk и Nb. Nr = 10, 12, 14;

– Rcon[ ] – массив из битов 32-разрядного слова, постоянный для данного раунда.

Архитектура алгоритма называется square – квадрат. Название алгоритма связано с тем, что блок данных (Block) представляется в виде прямоугольного массива из четырех строк и Nb столбцов. Операции производятся над отдельными байтами и над независимыми строками и столбцами.

В начале зашифрования входной блок input копируют в массив State следующим образом:

$$\text{state}[r, c] = \text{input}[r + 4c], \text{ для } 0 \le r < 4,\ 0 \le c < Nb.$$

Затем к State применяют процедуру AddRoundKey(). Далее State проходит, в зависимости от длины ключа, 10, 12 или 14 раундов шифрования.

После завершения последнего раунда State копируют в output (блок шифртекста) следующим образом:

$$\text{output}[r + 4c] = \text{state}[r, c], \text{ для } 0 \le r < 4, 0 \le c < Nb.$$

Схема раунда шифрования AES приведена на рис. 3.3.

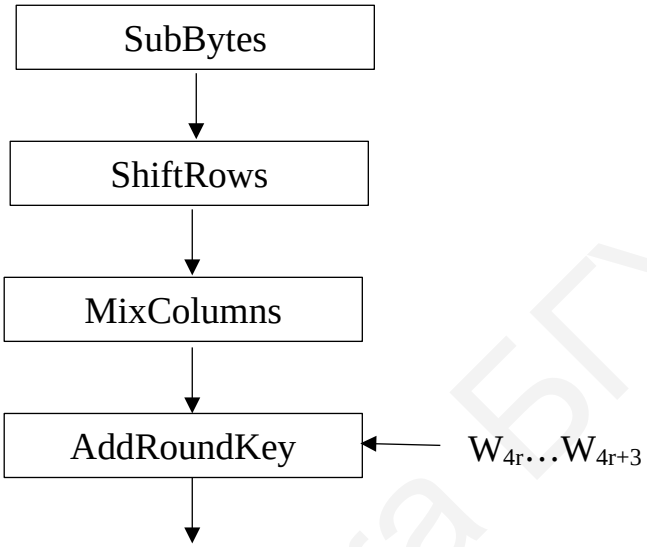

Рис. 3.3. Схема раунда шифрования AES

Операция SubBytes выполняет табличную замену байтов массива. Каждый байт в State a_{ij} заменяют соответствующим элементом b_{ij} в фиксированной 8-битной таблице поиска, S; $b_{ij} = S(a_{ij})$.

Операция ShiftRows выполняет циклический сдвиг влево строк массива. Причем каждая строка сдвигается на свое количество байт. Нулевая строка сдвигается на 0 байт, первая на 1 байт, вторая на 2 и третья на 3 байта.

Операция MixColumns выполняет умножение по модулю $x^4 + 1$ каждого столбца на физический полином $a(x)$:

$$a(x) = 3x^3 + x^2 + x + 2.$$

Вместе с ShiftRows MixColumns вносит диффузию в шифр.

Операция AddRoundKey выполняет добавление к массиву данных материала ключа. Для этого выполняют побитовую операцию XOR (сложение по модулю 2) каждого байта State с каждым битом RoundKey.

В последнем раунде, в отличие от предыдущих, не выполняют операцию MixColumns.

Ключи шифрования раундов RoundKeys получают из CipherKey использованием процедуры KeyExpansion. Алгоритму шифрования требуется Nr+1 наборов по Nb слов ключа. Один набор требуется перед первым раундом шифрования и по одному набору на каждом из Nr раундов шифрования. Полученный массив обозначается как $w[i], 0 \leq i < Nb * (Nr + 1)$.

В 2003 году Агентство национальной безопасности США признало шифр AES достаточно надежным для защиты сведений, составляющих государственную тайну (classified information). Для защиты информации до уровня SECRET включительно разрешено использовать ключи длиной 128 бит, а для уровня TOP SECRET – ключи длиной 192 и 256 бит.

Все блочные шифры обладают одним неприятным для защищающих свою информацию свойством, а именно – при зашифровании одинаковых блоков исходного текста получают одинаковые блоки криптотекста, поэтому существует потенциальная возможность утечки информации о повторяющихся блоках данных, шифруемых на одинаковом ключе. Этот факт в некоторых случаях может привести к получению существенной информации о содержании исходного текста при наблюдении криптограммы без ее расшифрования.

Для устранения этой проблемы используют различные режимы шифрования.

Режимы шифрования. Режимом шифрования называют способ применения алгоритма блочного шифра для преобразования последовательности блоков исходного текста в последовательность блоков шифртекста.

Наиболее широкое распространение получили следующие режимы шифрования:

– **ECB** (Electronic Code Book) – режим электронной кодовой книги, или режим простой замены;

– **CBC** (Cipher Block Changing) – режим сцепления блоков шифра;

– **CFB** (Cipher Feedback) – режим обратной связи по шифртексту, или режим гаммирования с обратной связью;

– **OFB** (Output Feedback) – режим обратной связи по выходу;

– **CTR** (Counter Mode) – режим счетчика.

– **ECB** (Electronic Code Book) – режим электронной кодовой книги, или режим простой замены. Схема шифрования в режиме ECB приведена на рис. 3.4. Зашифрование/расшифрование i-го блока исходного текста/шифртекста выполняется независимо от остальных блоков:

$$c_i = E_k(m_i),$$
$$m_i = D_k(c_i),$$

где c_i, m_i – блок исходного текста;

c_i – блок шифртекста;

i – номер блока;

E_k – блочная функция зашифрования;

D_k – блочная функция расшифрования;

k – ключ.

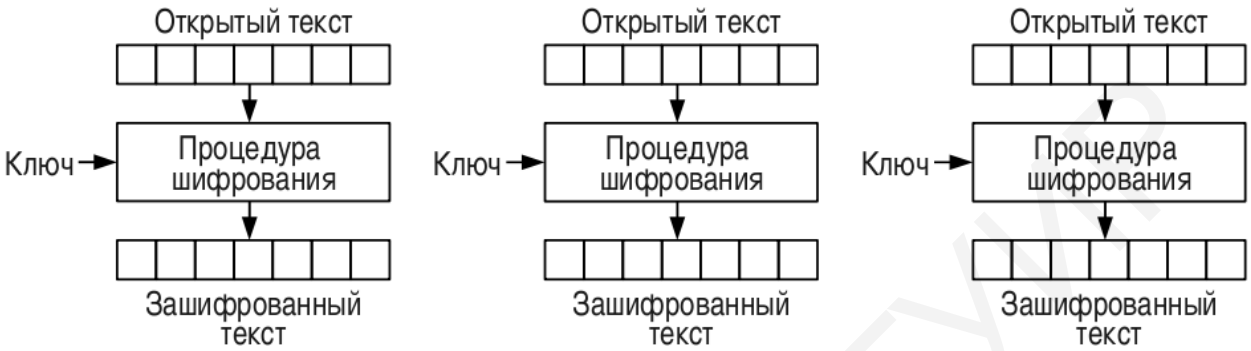

Рис. 3.4. Схема шифрования в режиме ECB

Недостатком режима ECB является то, что одинаковые блоки исходного текста преобразуются в одинаковые блоки шифртекста при использовании одинакового ключа.

Достоинства режима ECB:

– простота реализации;

– возможность распараллеливания шифрования.

CBC (Cipher Block Changing) – режим сцепления блоков шифра. Схема шифрования в режиме CBC приведена на рис. 3.5. Каждый блок открытого текста перед зашифрованием суммируется по модулю 2 с предыдущим блоком шифртекста:

$$c_i = E_k(m_i \oplus c_{i-1}),$$
$$m_i = D_k(c_i) \oplus c_{i-1},$$
$$c_0 = IV,$$

где IV – вектор инициализации (случайное число).

Недостатки режима CBC:

– распространение ошибки с i-го на i+1-й блок;

– невозможность параллельной обработки блоков при зашифровании и расшифровании.

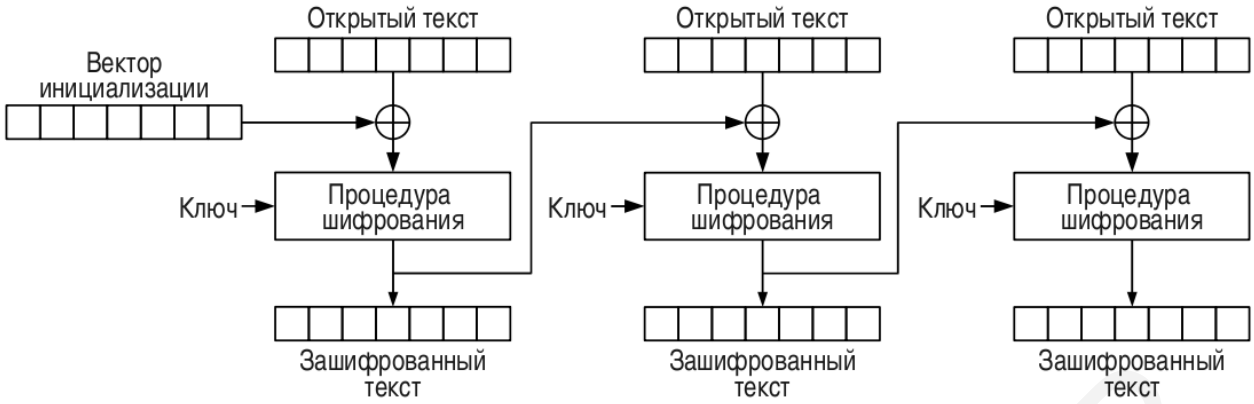

Рис. 3.5. – Схема шифрования в режиме CBC

Достоинства режима CBC:

– одинаковые блоки исходного текста после зашифрования дают разные блоки криптотекста;

– невозможна независимая манипуляция с каждым блоком криптотекста отдельно.

CFB (Cipher Feedback) – режим обратной связи по шифртексту или режим гаммирования с обратной связью. Схема шифрования в режиме CFB приведена на рис. 3.6. Модифицирует блочный шифр в синхронный поточный. Режим CFB зашифровывает текущий блок исходного текста путем его сложения по модулю два с предыдущим блоком шифртекста:

$$c_0 = IV,$$
$$c_i = E_k(c_{i-1}, k) \oplus m_i,$$
$$m_i = E_k(c_i, k) \oplus c_i.$$

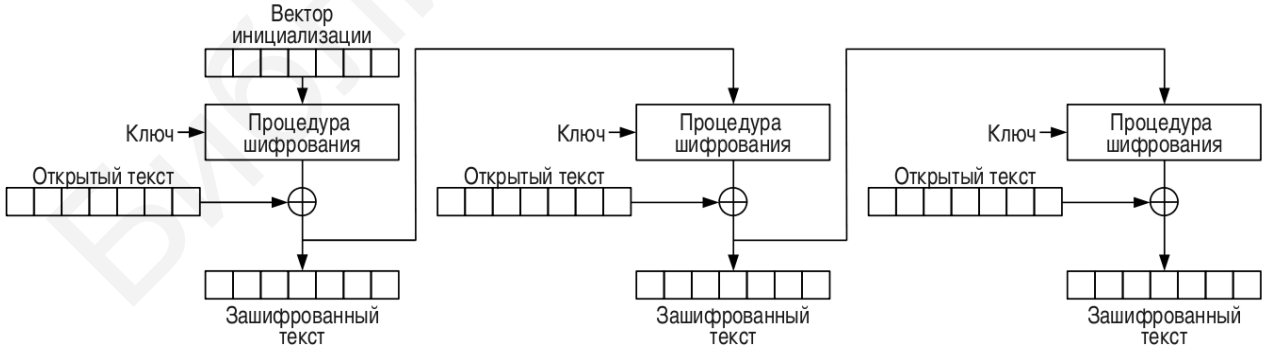

Рис. 3.6. Схема шифрования в режиме CFB

Режим CFB по своим возможностям совпадает с режимом CBC, но не требует дополнения последнего блока исходного текста в случае, если его длина не кратна размеру блока, что позволяет формировать шифртекст того же размера,

что и исходное сообщение. Ошибка в одном бите зашифрованного текста приведет к ошибочному расшифрованию двух блоков. Ошибочно будет расшифрован один бит одноименного блока и весь следующий блок. Отсутствует возможность распараллеливания зашифрования. В режиме CFB при ошибках в векторе инициализации искаженным окажется по крайней мере первый сегмент шифртекста. Будут ли испорчены остальные сегменты зависит от положения самого правого бита в IV (в худшем случае пострадают b/s сегментов шифртекста, где b – длина блока, s – длина сегмента).

OFB (Output Feedback) – режим обратной связи по выходу. Схема шифрования в режиме OFB приведена на рис. 3.7. Режим OFB модифицирует блочный шифр в синхронный поточный. Он генерирует поток ключевых блоков, которые затем суммируются по модулю два с блоками исходного текста. В следствие симметрии операции сложения по модулю два операции зашифрования и расшифрования полностью совпадают:

$$c_i = m_i \oplus O_i,$$
$$m_i = c_i \oplus O_i,$$
$$O_i = E_k(O_{i-1}),$$
$$O_0 = IV.$$

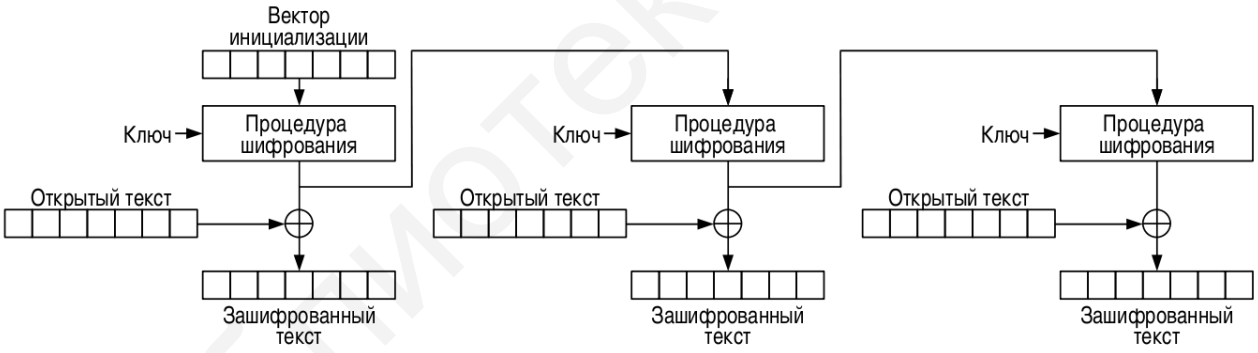

Рис. 3.7. Схема шифрования в режиме OFB

Каждая операция зашифрования/расшифрования блока зависит от всех предыдущих, поэтому не может быть распараллелена. Однако потоковый ключ шифрования блоков может быть сгенерирован заранее и затем операцию сложения по модулю два окончательного шифрования можно выполнять параллельно с открытым текстом. Ошибки, возникающие при передаче по зашумленным каналам связи, локализуются в пределах одного бита и не распространяются на весь блок и соседние блоки. Однако даже один бит ошибки в блоке вектора инициализации IV искажает каждый блок шифртекста в соответствующем сообщении.

CTR (Counter mode) – режим счетчика. Режим CTR делает из блочного шифра поточный. Он генерирует ключевую последовательность путем зашифрования значения счетчика, которую затем суммирует по модулю два с исходным текстом:

$$c_i = m_i \oplus E_k(Ctr_i), i = 1,2, \dots, n;$$
$$m_i = c_i \oplus E_k(Ctr_i), i = 1,2, \dots, n,$$

где Ctr_i – значение счетчика для i-го блока.

Алгоритмы зашифрования и расшифрования в режиме CTR могут выполняться параллельно. Вычисления, связанные с зашифрованием значений счетчика, могут быть выполнены до появления исходного или зашифрованного текстов, что дает преимущество режиму CTR по быстродействию относительно CFB и OFB.

Для режима CTR один бит ошибки в значении счетчика приводит к тому, что любой бит в расшифрованном сообщении может быть искаженным с вероятностью, близкой к 50 %.

Режимы ECB, CBC, CFB и OFB обрабатывают входной текст, длина которого должна быть кратна длине одного блока. Если это свойство не выполняется, то к сообщению необходимо добавить недостающее количество произвольных битов, называемых дополнением (padding). В стандарте ISO/IEC 9797-1 предложено добавлять в конец сообщения один единичный бит, а оставшиеся заполнять нулями.

При использовании этого метода для правильного расшифрования получатель должен точно знать, что в сообщении содержится дополнение. Это можно обеспечить, прикрепляя дополнение к каждому сообщению, даже если оно не требуется (в этом случае его посылают отдельным блоком) или посылать с каждым сообщением служебную информацию о его длине.

Кроме возникновения бита ошибки (инверсии бита) в блоке шифртекста может произойти удаление или вставка бита. Такое искажение приводит к нарушению границ всех последующих блоков шифртекста, и результаты их расшифрования будут полностью искаженными, пока не восстановится синхронизация границ блоков. При использовании режима 1-битного CFB синхронизация восстанавливается автоматически спустя b+1 позиций после вставленного или удаленного бита. В остальных режимах автоматического восстановления синхронизации не происходит.

Выбор режима шифрования зависит от поставленной цели и требований системы, в которую встраивают шифрование.

3.2. Инструментальные средства для выполнения задания

Для выполнения задания использовать программу EModes.exe. Программа EModes.exe позволяет выполнять зашифрование изображения различных типов изображений и отображать исходное изображение и результаты шифрования, интерпретируя шифртекст как изображение.

Типы изображений можно выбрать из заранее подготовленных рисунков следующего типа:

– компьютерный рисунок;

– диаграмма;

– текстура;

– фотография с небольшим количеством деталей;

– фотография с большим количеством деталей.

Возможен выбор алгоритмов шифрования:

– AES (размер блока – 128 бит);

– DES (размер блока – 64 бит).

Программа позволяет выбирать применение алгоритмов шифрования в следующих режимах:

ECB (Electronic Code Book) – режим электронной кодовой книги, или режим простой замены;

CBC (Cipher Block Changing) – режим сцепления блоков шифра;

CTR (Counter mode) – режим счетчика.

3.3. Задание для самостоятельного выполнения

1) Зашифровать предложенные изображения всеми возможными алгоритмами во всех возможных режимах. Результаты шифрования отразить в отчете в виде скриншотов.

2) Оценить полученные результаты и объяснить их причины.

3) Дать рекомендации по применению алгоритмов шифрования и их режимов в зависимости от типов изображения, шифрования и особенностей применения.

4) Дать ответ на вопрос: как влияет размер блока шифра на результат шифрования и почему?

4. ОТКРЫТОЕ РАСПРОСТРАНЕНИЕ КЛЮЧЕЙ

4.1. Теоретические сведения

Современные симметричные шифры обладают высокой практической криптографической стойкостью, обеспечивающей требуемую конфиденциальность каналов связи. Однако симметричная криптография сталкивается с существенными трудностями при решении двух задач.

1) Распределение секретного ключа. До начала обмена зашифрованными сообщениями обе стороны должны иметь одинаковый общий секретный ключ. Исторически эта задача решалась путем личной встречи или передачи ключа с надежным курьером. Однако в современных информационных системах такое решение практически не реализуемо.

2) Обеспечение защиты сообщения от подделки и подтверждение авторства сообщения (цифровая подпись). Не решается симметричными системами без наличия третьего доверенного лица.

Для решения этих задач в 1976 году Уитфилд Диффи (Whitfield Diffie), Мартин Хеллман (Martin Hellman) и Ральф Меркл (Ralph Merkle) предложили идею использования односторонних функций и односторонних функций с потайным ходом (с секретом). Эта идея положила начало асимметричной криптографии, в рамках которой проблемы симметричной криптографии оказываются элегантно разрешимыми.

Понятие односторонней функции является базовым в криптографии.

Односторонняя функция – это некоторая функция f, такая, что для любого x из ее области определения f(x) легко вычислима; однако практически для всех y из ее области значений нахождение x, для которого $y = f(x)$, вычислительно неосуществимо.

Однако такое определение оставляет открытым вопрос о том, что означает «вычислительно неосуществимо».

Поэтому более информативным является другое определение.

Пусть $\{0,1\}^n$ – множество всех двоичных строк длиной n.

Функция $f: \{0,1\}^* \to \{0,1\}^*$ является односторонней функцией, если она эффективно вычисляется за полиномиальное время на детерминированной машине Тьюринга, но не существует полиномиальной вероятностной машины Тьюринга, которая обращает эту функцию с более чем экспоненциально малой вероятностью. То есть для любой вероятностной полиномиальной машины M, для любого полинома $p(n)$ и достаточно большого $n \in N$ выполняется:

$$\Pr\left[M\left(f(m)\right) \in f^{-1}(m)\right] < 1/p(n),$$

где m – случайная равновероятная строка из множества $\{0,1\}^n$.

Время работы машины M ограничено полиномом от длины искомого прообраза. Время работы машины M определяется полиномом от длины искомого образа.

В настоящее время существование односторонних функций не доказано. Проблема заключается в следующем. Если f является односторонней функцией, то нахождение обратной функции является трудновычислимой, но легко проверяемой задачей. Тогда из существования односторонней функции следует, что $P \neq NP$. Однако неизвестно, следует ли из $P \neq NP$ существование односторонних функций.

В информационных системах нашли широкое применение односторонние функции, **сохраняющие длину**, – односторонние функции, битовая длина значения которых равна битовой длине аргумента. Такие функции используют для построения генераторов псевдослучайных последовательностей чисел. Для построения криптографических хеш-функций используют односторонние функции, длина значения которых постоянна при любой длине аргумента.

Построение систем шифрования на основе односторонних функций является достаточно трудной задачей, что хорошо видно из следующего примера. Пусть имеется односторонняя функция f. Необходимо построить криптосистему с открытым ключом. Криптосистема определяется семейством функций зашифрования E и семейством функций расшифрования D:

$$(\forall m)\ D\big(E(m)\big) = m,$$

где m – исходное сообщение.

Пусть для вычисления криптограммы c использовалась функция f, т. е. $c = f(m)$. Тогда противник, перехвативший зашифрованное сообщение c, может вычислить исходное сообщение m с пренебрежимо малой вероятностью. Однако и полномочный получатель столкнется с той же проблемой. Кроме того, из того что f является односторонней функцией, следует, что противник не сможет вычислить сообщение целиком, но не очевидно, что не сможет вычислить часть сообщения, что является существенной уязвимостью такой криптосистемы.

Кандидаты в односторонние функции. В настоящее время существует несколько десятков кандидатов в односторонние функции, наиболее широкое применение в криптографии нашли следующие:

– умножение и факторизация;

– возведение в квадрат и вычисление квадратного корня по модулю;

– возведение в степень по модулю и дискретное логарифмирование;

– криптографические хеш-функции.

Умножение и факторизация

Аргументами функции f являются пары взаимно простых чисел p и q, значение функции

$$N = f(p, q) = pq.$$

Значение функции может быть вычислено за время порядка $O(n^2)$, где n – сумма длин значений аргументов в битах.

Для нахождения значений обратной функции требуется разложить N на пару взаимно простых целых множителей.

Существует несколько методов разложения на простые множители, например:

– метод факторизации Ферма;

– метод эллиптической кривой;

– квадратичное решето;

– квадратичное решето в числовом поле.

Некоторые из них эффективны только для чисел специального вида. Верхняя оценка сложности метода Ферма и полного перебора порядка – $O(\sqrt{p})$.

Возможна факторизация с полиномиальной сложностью на квантовом компьютере методом Шнорра.

Возведение в квадрат и вычисление квадратного корня по модулю

Аргументами функции f являются пара простых чисел x и N, где $N = pq$, p и q – простые числа:

$$f(x) = x^2 \bmod N.$$

Для нахождения обратной функции требуется вычисление квадратного корня по модулю N, то есть нахождение x, если известно y при $x^2 \bmod N = y$. Эта задача имеет такую же сложность, как и разложение N на множители.

На основе функции возведения в квадрат и вычисления квадратного корня по модулю построена, а также задачи факторизации построена криптографическая схема Рабина.

Возведение в степень по модулю и дискретное логарифмирование

Параметрами функции f являются простое число p и целое число a, $0 < a < p$. Аргументом функции f является целое число $0 < x < p$:

$$f(x) = a^x \bmod p.$$

Функция f может быть вычислена за время $O(n^3)$, где $n = \log_2 p$.

Пусть $(G,*)$ – конечная абелева группа. Задача вычисления дискретных логарифмов заключается в нахождении целого числа x, удовлетворяющего соотношению $a^x \bmod p = B$, при известных a, B.

Сложность нахождения дискретного логарифма существенно зависит от вида группы $(G,*)$.

Дискретное логарифмирование аналогично обычному логарифмированию в поле действительных чисел. Однако, в отличие от последней задачи, в которой решение является приближенным, задача о вычислении дискретного логарифма имеет точное решение.

Криптографические хеш-функции

Примером односторонних криптографических хеш-функций является SHA-2 – семейство криптографических алгоритмов однонаправленных хеш-функций, включающее в себя алгоритмы SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256 и SHA-512/224. Хеш-функции предназначены для создания образов фиксированной длины для сообщений произвольной длины. Применяются в различных приложениях или компонентах, связанных с защитой информации, таких, как аутентификация, контроль целостности, цифровая подпись.

Протокол Диффи – Хеллмана

Протокол Диффи – Хеллмана – это криптографический протокол, позволяющий двум и более сторонам сгенерировать общий секрет, используя незащищенный канал связи. Под незащищенным каналом понимают канал, который может быть наблюдаем (прослушан) третьей стороной. Общий секрет используют для генерации ключа, который затем применяют в симметричных алгоритмах шифрования (для зашифрования и расшифрования) и защищенного обмена сообщениями.

Пусть имеются два несекретных числа g и P, известные всем заинтересованным сторонам. P – большое простое число, g – является первообразным корнем по модулю P. Первообразным корнем по модулю P называют такое число g, что все его степени по модулю P принимают значения всех чисел, взаимно простых с P.

Пусть также имеются два абонента Алиса и Боб, которые поочередно выполняют следующие шаги:

1) Алиса генерирует целое число a и держит его в секрете, затем вычисляет $A = g^a \bmod P$ и пересылает его Бобу.

2) Боб генерирует целое число b и держит его в секрете, затем вычисляет $B = g^b \bmod P$ и пересылает его Алисе.

3) Алиса вычисляет значение $B^a \bmod P = g^{ab} \bmod P$.

4) Боб вычисляет значение $A^b \bmod P = g^{ab} \bmod P$.

Нетрудно видеть, что Алиса и Боб вычислили одно и то же число:

$$K = g^{ab} \bmod P = g^{ba} \bmod P.$$

Теперь Алиса и Боб могут сгенерировать ключ шифрования, используя общий секрет в качестве стартового значения генератора (одинакового у Алисы и Боба). В качестве такого генератора часто используют криптографические функции хеширования.

Использование протокола Диффи – Хеллмана не ограничивается двумя участниками. Он может быть применен на неограниченное количество пользователей. Например, в случае трех участников A, B и C они вычисляют общий секрет:

$$K = g^{abc} \bmod P = g^{cba} \bmod P = g^{bac} \bmod P.$$

При этом злоумышленник может наблюдать в открытом канале промежуточные значения $g^a, g^b, g^c, g^{ab}, g^{ac}, g^{bc}$, но не может вычислить общий секрет g^{abc}.

Возвращаясь снова к протоколу с двумя участниками, предполагается, что злоумышленник может получить значения A, B, g и P, но не модифицировать. Если числа A, B и P выбраны достаточно большими, то при попытке вычислить общий секрет атакующий встретится с задачей, неразрешимой за разумное время. Величина числа g на стойкость протокола не влияет, поэтому его обычно выбирают пределах первого десятка (из соображений простоты вычислений).

Величину P на практике выбирают не менее 1024 бит.

В 2016 году была представлена работа, показавшая возможности по подготовке специальных конечных полей для алгоритма Диффи – Хеллмана. Выбранное исследователями простое число p специального вида (размером 1024 бита) выглядит обычным для пользователей, но упрощает на несколько порядков сложность вычислений по методу SNFS (Специальный метод решета числового поля – Special Number Feld Sieve) для решения задачи дискретного логарифмирования. Для борьбы с атакой предлагается увеличить размер модуля до 2048 бит.

При правильном выборе параметров протокол Диффи – Хеллмана устойчив к пассивным атакам. Однако он не способен противостоять атаке «человек посредине». Пусть имеются два абонента Алиса – А и Боб – Б, а также атакующий Джо – Д. Джо может перехватывать и модифицировать сообщения. Последовательность действий участников при атаке «человек посредине» приведена в табл. 4.1.

Таким образом, Джо получает общий секрет для защищенного обмена сообщениями с Алисой и секрет для общения с Бобом, выдавая себя за Боба для Алисы и за Алису для Боба. Защитой от такой атаки являются протоколы аутентификации сторон.

Поскольку при выполнении шагов протокола показатели степенной функции достаточно велики, а также велико значение числа *P*, то для ускорения вычислений используют **алгоритм быстрого возведения в степень методом повторяющихся возведений в квадрат и умножения**.

Таблица 4.1

Атака «человек посредине»

№	Алиса	Джо	Боб
1	$g^a \bmod P \rightarrow$	$g^a \bmod P$	–
2	$g^j \bmod P \leftarrow$	$g^j \bmod P$	–
3	$g^{ja} \bmod P$	$g^{aj} \bmod P$	–
4	–	$g^j \bmod P \rightarrow$	$g^j \bmod P$
5	–	$g^b \bmod P \leftarrow$	$g^b \bmod P$
6	–	$g^{bj} \bmod P$	$g^{jb} \bmod P$

Метод заключается в следующем.

Пусть требуется вычислить $x^a \bmod n$.

Представим показатель степени в виде

$$a = a_{j-1}2^{j-1} + a_{j-2}2^{j-2} + \cdots + a_2 2^2 + a_1 2^1 + a_0,$$

где $a_j = (0,1)$.

Далее представим $x^a \bmod n$ в виде

$$x^a \bmod n = x^{a_{j-1}2^{j-1}+a_{j-2}2^{j-2}+\cdots+a_2 2^2 + a_1 2^1 + a_0} \bmod n =$$

$$= (x^2)^{a_{j-1}2^{j-2}+a_{j-2}2^{j-3}+a_1 2^0} x^{a_0} \bmod n =$$

$$= ((x^2)^2)^{a_{j-1}2^{j-3}+a_{j-2}2^{j-4}+\cdots+a_2 2^0} (x^2)^{a_1} x^{a_0} \bmod n =$$

$$= (\ldots((x^2)^2 \ldots)^2)^{a_{j-1}} \ldots (x^8)^{a_3} (x^4)^{a_2} (x^2)^{a_1} x^{a_0} \bmod n.$$

Затем вычисляют $x^2 \bmod n$ и выполняют замену в преобразованном выражении. Вычисление производят до тех пор, пока не будет получен результат.

4.2. Задание для самостоятельного выполнения

Для заданного простого P (в соответствии с вариантом) найти g – примитивный элемент конечного поля GF(P) и выполнить генерацию общего секрета. Для нахождения g воспользуйтесь методом перебора по возрастанию, возведения в степень по модулю P и проверки того факта, что все степени принимают значения от 0 до P – 1.

Варианты:

1) 5717	11) 3877	21) 4877
2) 9721	12) 1877	22) 2957
3) 2111	13) 1973	23) 2971
4) 3917	14) 4937	24) 3137
5) 4231	15) 7237	25) 1123
6) 9001	16) 9011	26) 9679
7) 8699	17) 8233	27) 8329
8) 8447	18) 8581	28) 7351
9) 7489	19) 7573	29) 7673
10) 7759	20) 7883	30) 6823

Вариант выбирается в соответствии с порядковым номером студента в рамках группы. Если студентов в группе больше, чем вариантов в списке, то варианты снова повторяются, начиная с единицы.

Отчет должен содержать:

1) Листинги программ:

а) для проверки g (первообразный корень по модулю P);

б) для вычисления $B^a \bmod P = g^{ab} \bmod P$ и $A^b \bmod P = g^{ab} \bmod P$.

2) Описание шагов, выполняемых участниками протокола – Алисой и Бобом для вычисления общего секрета.

3) Выводы, содержащие:

а) модель атакующего и оценки длины ключа;

б) возможные угрозы протоколу и предложения по защите от них.

Важно!

Возведение в степень выполнять методом последовательного возведения в квадрат и умножения.

При возведении в степень по модулю операцию взятия по модулю выполнять на каждом шаге возведения в квадрат и умножения для исключения переполнения.

5. АСИММЕТРИЧНОЕ ШИФРОВАНИЕ И ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ

5.1. Теоретические сведения

Электронная цифровая подпись ЭЦП – реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).

Существуют две основные схемы построения цифровой подписи:

1) Схема арбитражной подписи (рис. 5.1). В этой схеме необходимо наличие в системе арбитра – третьего лица, пользующегося доверием сторон, генерирующих и проверяющих подпись. Схема строится с использованием симметричных алгоритмов шифрования.

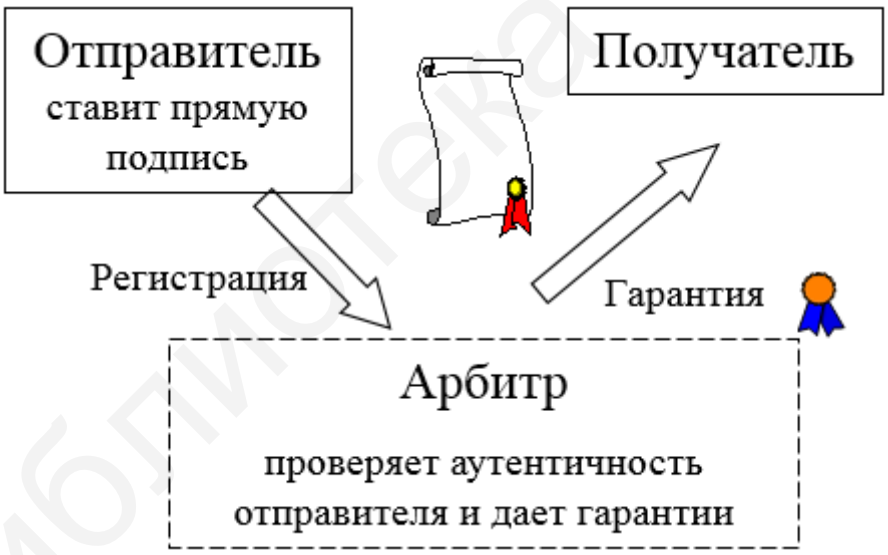

Рис. 5.1. Схема арбитражной подписи

2) Схема прямой подписи (рис. 5.2). Схема строится на основе алгоритмов асимметричного шифрования. Для формирования и проверки подписи третья сторона не нужна. Однако наличие арбитра требуется в случае возникновения конфликтной ситуации между сторонами.

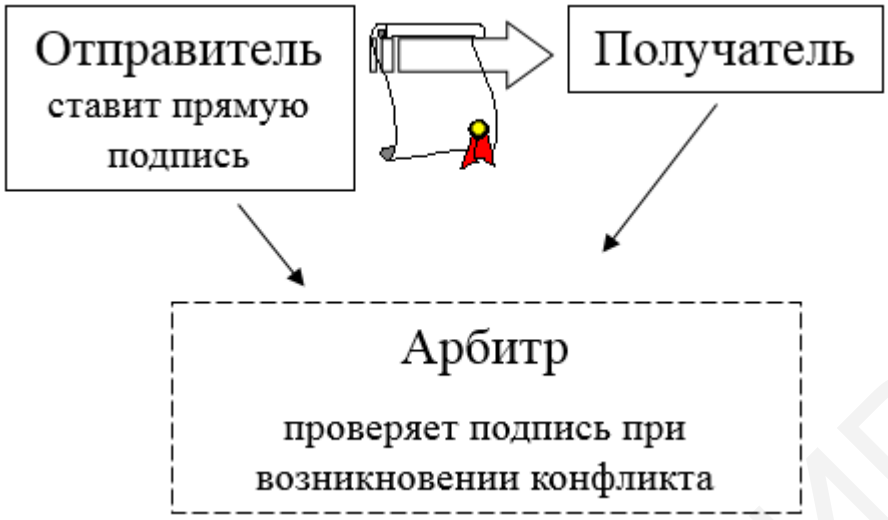

Рис. 5.2. Схема прямой подписи

Кроме этого, существуют другие виды цифровых подписей (групповая подпись, неоспоримая подпись, доверенная подпись), которые являются вариациями двух основных схем. Их появление обусловлено разнообразием задач, решаемых с помощью ЭЦП.

Основными функциями цифровой подписи являются:

– контроль целостности электронного документа;

– защита от изменений (подделки) документа;

– обеспечение невозможности отказа от авторства документа;

– формирование доказательств подтверждения авторства документа.

В настоящее время наибольшее распространение получили системы, использующие схему прямой подписи. Формирование и проверка прямой цифровой подписи основана на использовании идеи ***односторонних функций с потайным ходом***.

В статье, опубликованной Диффи и Хеллманом в 1976 году, односторонняя функция была определена как семейство обратимых функций f_z с параметром z, таких, что для данного z можно найти алгоритмы E_z и D_z, позволяющие легко вычислить значение $f_z(x)$ для всех x из области определения, а также $f_z^{-1}(y)$ для всех y из области значений, однако практически для всех значений параметра z и практически для всех значений y из области значений f_z нахождение $f_z^{-1}(y)$ вычислительно неосуществимо даже при известном E_z.

Независимо от выбранного вида односторонней функции с потайным ходом схемы прямой подписи относятся к асимметричным криптографическим системам с открытым ключом. В рамках такой системы каждый пользователь имеет пару ключей – секретный (private key) и открытый (public key).

Использовать их можно как для цифровой подписи, так и для шифрования сообщений (чаще всего для обмена секретными ключами по открытым каналам связи).

Для формирования цифровой подписи подписант использует свой секретный ключ, а для проверки проверяющий использует открытый ключ подписанта.

Для зашифрования сообщения отправитель использует открытый ключ получателя, а для расшифрования получатель использует свой секретный ключ.

Схема формирования цифровой подписи включает в себя три процесса:

1) Генерация ключевой пары. Секретный ключ равновероятным образом выбирают из множества возможных секретных ключей, затем вычисляют соответствующий ему открытый ключ.

2) Формирование ЭЦП. Для блока данных с помощью секретного ключа вычисляют значение ЭЦП.

3) Проверка (верификация) ЭЦП. Для блока данных и значения ЭЦП с помощью открытого ключа проверяют действительность ЭЦП.

Для эффективного использования цифровой подписи необходимо выполнение следующих условий:

– верификация ЭЦП должна выполняться с помощью открытого ключа, соответствующего секретному ключу, использовавшемуся для формирования ЭЦП;

– без обладания секретным ключом должно быть вычислительно сложно сформировать легитимную цифровую подпись.

Для обеспечения выполнения этих условий, как правило, используют алгоритмы, основанные на следующих задачах:

– задача дискретного логарифмирования;

– задача факторизации, то есть разложение числа на простые множители.

В августе 1977 года трое ученых из Массачусетского технологического института Рональд Ривест, Ади Шамир и Леонард Адлеман предложили **криптосистему RSA**, основанную на задаче факторизации.

Алгоритм создания ключа:

1) Выбирают два случайных простых числа p и q заданного размера, $p \neq q$.

2) Вычисляют произведение $n = p \cdot q$.

3) Вычисляют значение функции Эйлера от числа n:

$$\varphi(n) = (p-1)(q-1).$$

4) Выбирают целое число e, $1 < e < \varphi(n)$, взаимно простое со значением функции $\varphi(n)$. Обычно в качестве e берут простые числа, содержащие небольшое количество единичных битов в двоичной записи, например простые

числа Ферма 17, 257 или 65537 (что обеспечивает высокую скорость шифрования, так как время шифрования пропорционально количеству единичных битов в числе e). Число e называют открытой экспонентой.

5) Вычисляют число d, мультипликативно обратное к числу e по модулю n, удовлетворяющее сравнению:

$$d \cdot e \equiv 1 \bmod \varphi(n).$$

Число d называют секретной экспонентой. Для вычисления d используют расширенный алгоритм Евклида.

Пара чисел $\{e, n\}$ публикуется в качестве открытого ключа RSA.

Пара $\{d, n\}$ играет роль закрытого ключа RSA и держится в секрете.

Шифрование. В качестве исходного сообщения m выступают целые числа в интервале от 0 до $n - 1$.

Для зашифрования используют открытый ключ $\{e, n\}$ получателя сообщения и вычисляют криптограмму с следующим образом:

$$c = E(m) = m^e \bmod n.$$

Для расшифрования используют секретный ключ $\{d, n\}$ получателя сообщения и вычисляют исходное сообщение следующим образом:

$$m = D(c) = c^d \bmod n.$$

На практике RSA не применяют для шифрования сообщений, а используют смешанный алгоритм, состоящий из RSA и блочного симметричного шифра, например AES.

С помощью RSA зашифровывается секретный ключ блочного алгоритма шифрования, а затем с помощью этого ключа шифруют сообщения блочным симметричным шифром. Как правило, такой ключ используют в течение одного сеанса связи, а затем уничтожают. Поэтому ключ называют сеансовым.

Если сеансовый ключ больше n, то перед зашифрованием его разделяют на блоки и зашифровывают поблочно.

Цифровая подпись. Алгоритм цифровой подписи отличается от алгоритма шифрования использованием ключей.

При создании цифровой подписи входом является исходный текст m и секретный ключ подписанта $\{d, n\}$.

Для создания цифровой подписи s с помощью секретного ключа $\{d, n\}$ вычисляют

$$s = m^d \bmod n.$$

Затем формируют пару $\{m, s\}$ и отправляют получателю.

Для проверки цифровой подписи входом является пара $\{m, s\}$ и открытый ключ подписанта $\{e, n\}$.

Проверяющий вычисляет прообраз сообщения из подписи

$$m^* = s^e \bmod n$$

и сравнивает m и m^*. Если $m = m^*$, значит подпись верна, в противном случае – ложна.

Важным является то, что создать подпись может только автор – владелец секретного ключа $\{e, n\}$, а проверить любой, имеющий доступ к открытому ключу подписанта.

Цифровая подпись не обеспечивает конфиденциальность подписанного сообщения, так как не зашифровывает его.

Для обеспечения конфиденциальности автор должен подписать исходное сообщение, затем зашифровать пару исходное сообщение – подпись $\{m, s\}$ с помощью открытого ключа получателя. Получатель расшифровывает полученное сообщение с помощью своего секретного ключа, восстанавливая пару $\{m, s\}$, затем проверяет цифровую подпись.

Стойкость RSA. Стойкость алгоритма RSA базируется на предположении о сложности вычисления функции, обратной по отношению к функции шифрования

$$c = E(m) = m^e \bmod n.$$

Поскольку атакующему известна тройка $\{c, e, n\}$, то для вычисления m необходимо найти d, удовлетворяющее условию

$$e \cdot d \equiv 1 \bmod \varphi(n).$$

Для решения этой задачи атакующему необходимо знать значение функции Эйлера $\varphi(n)$. Поскольку

$$\varphi(n) = (p - 1)(q - 1),$$
$$n = p \cdot q,$$

то для нахождения $\varphi(n)$ надо разложить число n на простые множители p и q, т. е. решить задачу факторизации.

В настоящее время самым быстрым из известных методов факторизации является общий метод решета числового поля. Его скорость для целого числа длиной k бит оценивается как

$$\exp\left(\left(c + o(1)\right)k^{\frac{1}{3}}\log^{\frac{2}{3}}k\right) \text{ для некоторого } c < 2.$$

В 2010 году были успешно взломаны данные, зашифрованные RSA с длиной ключа 768 бит, и был сделан вывод о том, что надежной может считаться система RSA с длиной ключа не менее 1024 бит. Браузер Mozilla с 2013 года не поддерживает сертификаты удостоверяющего центра с длиной ключа менее 2048 бит, с 2014 года протокол TLS для сертификата сервера также рекомендует использовать RSA с ключом 2056 бит и более.

5.2. Задание для самостоятельного выполнения

Разработать программное обеспечение, реализующее функции генерации секретного и открытого ключей, шифрования и цифровой подписи для алгоритма RSA. Обмен входными и выходными данными должен осуществляться через файлы:

– открытого ключа;

– секретного ключа;

– исходного сообщения;

– зашифрованного сообщения.

Для повышения скорости шифрования использовать метод последовательного возведения в квадрат и умножения.

Выполнить тестирование разработанного программного обеспечения на 10 наборах тестовых данных.

Длина чисел p и q должна быть не менее 1024 бит.

6. МЕЖСЕТЕВОЕ ЭКРАНИРОВАНИЕ

6.1. Теоретические сведения

Межсетевой экран является одним из основных средств, используемых при построении многоуровневой защиты корпоративной сети от внешних угроз.

Межсетевой экран – это комплекс программно-аппаратных средств межсетевой защиты, позволяющий реализовать частную политику безопасности (набор правил), определяющую условия прохождения пакетов данных через границу из одного участка информационной сети в другой. Как правило, межсетевой экран устанавливается между локальной сетью предприятия и глобальной сетью Интернет. Межсетевой экран является также программным решением, являющимся элементом операционных систем семейства Windows, начиная с Windows NT и более поздних. В качестве другого, часто используемого названия для межсетевого экрана, используют термины firewall (английский) и брандмауэр (русская транскрипция с немецкого).

Для защиты от несанкционированного доступа (предотвращения или обнаружения) межсетевой экран должен располагаться между защищаемым участком локальной сети и потенциально опасной внешней сетью, например, как показано на рис. 6.1.

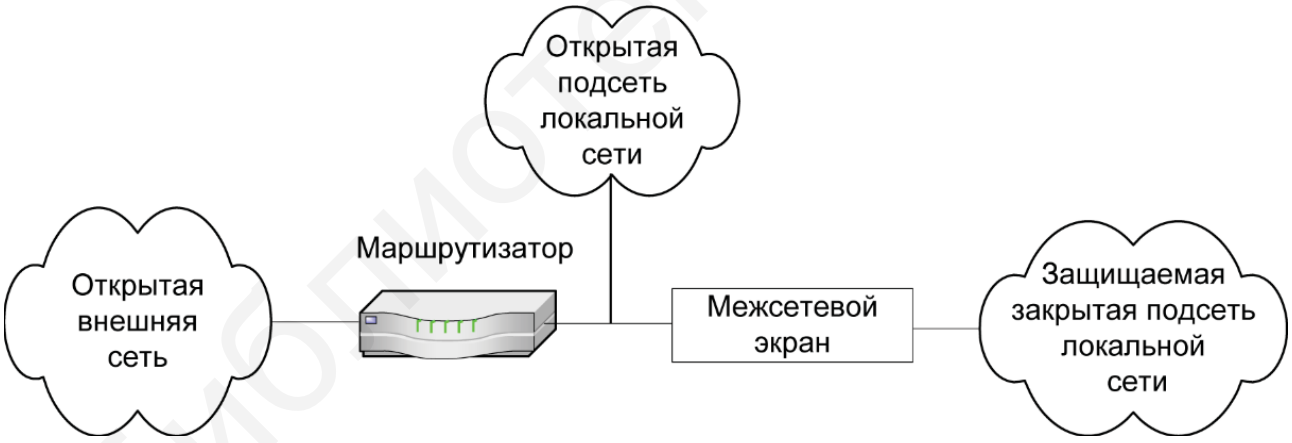

Рис. 6.1. Вариант схемы подключения межсетевого экрана

Основными задачами, решаемыми межсетевыми экранами, являются:

1) Ограничение доступа внешних пользователей к ресурсам защищаемой внутренней сети.

2) Разграничение доступа пользователей защищаемой сети к внешним ресурсам. Такое разграничение позволяет, например, ограничить доступ своих сотрудников к внешним ресурсам, не требуемым для выполнения своих функциональных обязанностей.

Классифицировать межсетевые экраны можно по нескольким признакам.

По уровню функционирования относительно модели OSI:

– пакетный фильтр;

– шлюз сеансового уровня;

– прикладной шлюз;

– шлюз экспертного уровня.

По используемой технологии:

– контроль состояния протокола (SPI – Stateful Packet Inspection);

– модули посредники (прокси).

По исполнению:

– аппаратно-программный;

– программный.

По схеме подключения:

– схема единой защиты сети;

– схема с защищаемым закрытым и незащищаемым открытым сегментами сети (см. рис. 6.1);

– схема с раздельной защитой закрытого и открытого сегментов сети.

Пакетный фильтр представляет собой маршрутизатор или работающую на сервере программу, сконфигурированные таким образом, чтобы фильтровать входящие и исходящие пакеты. Фильтрация осуществляется на основе анализа следующих данных:

– IP-адреса источника и приемника;

– тип протокола (TCP, UDP, ICMP и т. д.);

– номера портов отправителей и получателей TCP и UDP пакетов;

– другие данные заголовков пакетов (например, флаги TCP заголовка).

Анализ выполняется путем сравнения данных заголовка с сконфигурированной таблицей правил. При создании таблиц правил, кроме данных внешних пакетов, приведенных ранее, можно использовать данные, внешние по отношению к пакетам, такие как дата, время прохождения пакета и др.

Пакетные фильтры просты в использовании, дешевы, оказывают минимальное влияние на производительность вычислительной системы. Пакетные фильтры могут быть реализованы аппаратно или программно (например, в ОС семейства Windows и Unix). Пакетный фильтр может быть установлен как на устройстве, расположенном на границе между двумя сетями, так и на рабочей станции пользователя, повышая тем самым ее защищенность. Пакетные фильтры являются составной частью практически всех межсетевых экранов, использующих контроль состояния.

Основные недостатки пакетных фильтров:

1) Отсутствие возможности анализа трафика на прикладном уровне, на котором осуществляются такие атаки, как проникновение вирусов, интернет-червей, отказ в обслуживании и т. д. Пакетный фильтр анализирует только заголовок и не анализирует поле данных, которое может содержать информацию, противоречащую политике безопасности защищаемой системы.

2) Сложность настройки и администрирования, которые требуют создания как минимум двух правил для каждого типа разрешенного взаимодействия (для входящего и исходящего трафика). При создании большого количества правил могут быть созданы противоречивые правила. Большое количество правил может снижать производительность межсетевого экрана.

3) Слабая аутентификация трафика, выполняемая только на основе адреса отправителя, что позволяет подменить его любым из адресов, принадлежащих адресному пространству IP-протокола.

Шлюз сеансового уровня контролирует допустимость сеанса связи. Он исключает прямое взаимодействие двух узлов на сеансовом уровне, выступая в качестве посредника (proxy), перехватывающего все запросы одного узла на доступ к другому и устанавливающего соединение после проверки допустимости таких запросов. Затем шлюз сеансового уровня копирует пакеты, передаваемые в рамках одной сессии между двумя узлами без дополнительной фильтрации. Сразу после установления авторизованного соединения шлюз помещает в таблицу соединений адреса отправителя и получателя, сведения о состоянии соединения, информацию о номере последовательности и т. д. После завершения сеанса связи запись о нем удаляется из таблицы. Все последующие пакеты, которые могут быть сформированы злоумышленником, отбрасываются.

Достоинство технологии заключается в исключении прямого контакта между внутренним и внешним хостами. Для связи между защищаемой внутренней и внешней сетью используется только адрес шлюза сеансового уровня. Поскольку соединение между узлами устанавливается только после проверки его допустимости, то шлюз предотвращает подмену адреса, что возможно в пакетных фильтрах.

Основным недостатком технологии является невозможность проверки содержания поля данных. Злоумышленник может передать в защищаемую сеть деструктивные программные средства. Возможность перехвата TCP-сессии позволяет злоумышленнику атаковать защищаемую сеть в рамках разрешенной сессии.

Шлюз прикладного уровня проверяет содержимое каждого проходящего через шлюз пакета и может фильтровать отдельные виды команд или информации в протоколах прикладного уровня, которые им поручено

обслуживать. Это более совершенный и надежный тип межсетевого экрана, использующий программы-посредники (proxies) прикладного уровня, или агенты. Агенты составляются для конкретных служб сети Интернет (HTTP, FTP, telnet и т. д.) и служат для проверки сетевых пакетов на наличие достоверных данных.

Шлюз прикладного уровня снижает уровень производительности системы из-за повторной обработки в программе-посреднике. Это незаметно при работе в Интернете, при работе по низкоскоростным каналам, но существенно при работе во внутренней сети.

Межсетевые экраны экспертного уровня сочетают в себе элементы всех трех описанных выше категорий.

Как и межсетевые экраны с фильтрацией пакетов, они работают на сетевом уровне модели OSI, фильтруя входящие и исходящие пакеты на основе проверки IP-адресов и номеров портов.

Межсетевые экраны экспертного уровня выполняют функции шлюза сеансового уровня, определяя, относятся ли пакеты к соответствующему сеансу.

Межсетевые экраны выполняют функции шлюза прикладного уровня, оценивая содержимое каждого пакета в соответствии с политикой безопасности, выработанной в конкретной организации.

Вместо применения связанных с приложениями программ-посредников брандмауэры экспертного уровня используют специальные алгоритмы распознавания и обработки данных на уровне приложений. С помощью этих алгоритмов пакеты сравниваются с известными шаблонами данных, что обеспечивает более эффективную фильтрацию пакетов.

Кроме основной функции фильтрации трафика межсетевые экраны могут выполнять следующие функции:

1) Трансляция сетевых адресов, позволяющая скрывать конфигурацию защищаемой сети от внешнего наблюдателя. Трансляция может быть динамической или статической. При динамической трансляции адрес выделяется узлу в момент обращения к межсетевому экрану. После завершения соединения адрес освобождается и может быть использован любым другим узлом защищаемой сети. При статической трансляции адрес узла всегда привязывается к одному адресу межсетевого экрана.

2) Аутентификация пользователей с помощью пары идентификатор – пароль или с помощью других, более надежных, методов, например с помощью цифровых сертификатов.

3) Регистрация событий безопасности. К таким событиям относятся пропуск или блокирование сетевых пакетов, установление или отказ сеанса

связи, изменение правил разграничения доступа администратором безопасности и другие действия. Регистрация позволяет анализировать данные журнала регистрации с целью обнаружения и предотвращения атак, совершенствования политики безопасности, сбора доказательств для судебных разбирательств или внутреннего расследования в случае нарушения безопасности.

Межсетевые экраны могут быть реализованы программно или аппаратно.

Программная реализация дешевле, но обладает меньшей производительностью и требует значительных ресурсов системы.

Аппаратные межсетевые экраны представляют собой программно-аппаратные комплексы, использующие специализированные или обычные операционные системы, модифицированные для выполнения защитных функций. Чаще всего используют операционные системы на базе FreeBSD или Linux. Операционная система должна быть максимально ограниченной по функционалу и отвечать следующим требованиям:

– иметь средства разграничения доступа к ресурсам системы;

– блокировать доступ к ресурсам в обход сервиса разграничения доступа;

– запрещать привилегированный доступ к своим ресурсам из защищаемой сети;

– иметь средства аудита событий безопасности, включая все действия администратора.

К достоинствам аппаратных решений можно отнести следующее:

1) Простота внедрения межсетевого экрана в информационную систему обеспечивается поставкой с предустановленной и настроенной операционной системой и защитными механизмами.

2) Простота управления. Аппаратные межсетевые экраны могут управляться с любой рабочей станции, работающей под управлением операционных систем семейства Windows или Unix. Взаимодействие консоли управления с аппаратным межсетевым экраном осуществляется по стандартным или (предпочтительно) защищенным протоколам, например: Telnet, SNMP, SSL, TLS.

3) Производительность системы не снижается (или снижается незначительно) благодаря аппаратной поддержке некоторых функций, что снижает нагрузку на центральный процессор сервера или рабочей станции, а также использованию специализированной операционной системы, из которой исключены все ненужные сервисы и подсистемы.

4) Имеется возможность использования механизмов обеспечения программной и аппаратной отказоустойчивости.

5) Аппаратные межсетевые экраны относительно легко объединяются в кластеры.

К недостаткам межсетевых экранов можно отнести:

1) Необходимость реконфигурации архитектуры сетей, использующих некоторые технологии и сервисы, например широковещательную рассылку сообщений.

2) Отсутствие эффективных решений от разрушающих программных средств, использующих мобильный код JavaScript, VBScript, Java applets, ActiveX controls, Flash animations и т. д.

3) Сложности использования в межсетевых экранах интегрированных антивирусных средств защиты, способных контролировать весь передаваемый трафик.

4) Снижение производительности сети в тех случаях, когда межсетевой экран выполняет полный анализ содержания каждого пакета.

Межсетевой экран позволяет осуществлять фильтрацию только того трафика, формат представления данных в котором ему доступен. Защищенные протоколы, такие как TLS, SSH, IPsec и SRTP, используют криптографию для того, чтобы скрыть содержимое, из-за чего их трафик не может быть проинтерпретирован. Также некоторые протоколы, такие как OpenPGP и S/MIME, шифруют данные прикладного уровня, из-за чего фильтровать трафик на основании информации, содержащейся на данном сетевом уровне, становится невозможно. Еще одним примером ограниченности анализа межсетевых экранов является туннелированный трафик, так как его фильтрация является невозможной, если межсетевой экран «не понимает» используемый механизм туннелирования. Во всех этих случаях правила, сконфигурированные на межсетевом экране, должны явно определять, что делать с трафиком, который они не могут интерпретировать.

При подключении локальной или корпоративной сети к глобальной сети необходимо решить следующие задачи безопасности:

– защита корпоративной или локальной сети от несанкционированного доступа со стороны глобальной сети;

– сокрытие информации о структуре сети и ее компонентов от наблюдения и анализа со стороны глобальной сети;

– разграничение доступа в защищаемую сеть из глобальной сети и из защищаемой сети в глобальную сеть.

Для решения этих задач межсетевой экран должен быть правильно подключен и настроен. Эта проблема решается в два этапа:

– формирование политики межсетевого взаимодействия;

– реализация политики межсетевого взаимодействия путем выбора схемы подключения и настройки параметров межсетевого экрана.

Политика межсетевого взаимодействия является компонентом общей политики безопасности организации, определяет правила безопасного информационного обмена организации с внешним миром и включает в себя:

– политику доступа к сетевым сервисам;

– политику функционирования межсетевого экрана.

Политика доступа к сетевым сервисам определяет правила предоставления и использования всех возможных сервисов защищаемой компьютерной сети:

– определены все сервисы, предоставляемые через межсетевой экран;

– указаны допустимые адреса клиентов для каждого сервиса;

– указаны правила, описывающие, когда и какие пользователи каким сервисом и на каком компьютере могут воспользоваться.

Политика работы межсетевого экрана задает один из двух базовых подходов функционирования межсетевого экрана:

1) Запрещено все, что не разрешено.

2) Разрешено все, что не запрещено.

Первый подход позволяет успешно реализовать принцип минимальных привилегий и является наиболее безопасным. Администратор безопасности создает для каждого разрешенного типа взаимодействия не менее одного правила доступа, что не позволяет случайно оставить разрешенными какие-либо полномочия, запрещенные по умолчанию. Лишние доступные сервисы могут быть использованы для нарушения безопасности. Однако этот подход может доставить неудобства пользователям и снизить их лояльность к правилам безопасности.

Второй подход предполагает блокирование только явно запрещенных межсетевых взаимодействий. В этом случае обеспечивается удобство пользователей в ущерб безопасности. Администратор может не предусмотреть все действия, запрещенные пользователем. Поэтому предпочтительно не использовать этот подход.

Выбор схемы подключения межсетевых экранов зависит от условий функционирования и конфигурации защищаемой сети, а также количества и типа сетевых интерфейсов и др.

Наибольшее распространение получили следующие схемы подключения:

– схемы с использованием экранирующего маршрутизатора;

– схемы единой защиты локальной сети;

– схемы с защищаемой закрытой и незащищаемой открытой подсетями;

– схемы с раздельной защитой закрытой и открытой подсетей.

Схема защиты с использованием экранирующего маршрутизатора состоит из экранирующего маршрутизатора, расположенного между защищаемой сетью и открытой внешней сетью (рис. 6.2). Экранирующий маршрутизатор (пакетный фильтр) фильтрует входящие и исходящие пакеты на основе анализа их адресов и портов.

Компьютеры, находящиеся в защищаемой сети, имеют прямой доступ в сеть Интернет, а доступ к ним из Интернета избирательно блокируется. Основным достоинством является относительно низкая стоимость за чет реализации функций маршрутизации и фильтрации в одном модуле.

Основными недостатками схемы с использованием экранирующего маршрутизатора являются:

– сложность правил фильтрации;

– невозможность полноценного тестирования правил фильтрации, это приводит к незащищенности сети от непроверенных атак;

– сложность регистрации событий, приводящая к трудности анализа состояния безопасности маршрутизатора.

Схема единой защиты локальной сети является наиболее простым решением (рис. 6.3), при котором между маршрутизатором и межсетевым экраном имеется только один путь, по которому идет весь трафик. Межсетевой экран целиком экранирует локальную сеть от потенциально враждебной внешней сети. Схема позволяет наиболее просто реализовать политику безопасности, основанную на принципе «запрещено все, что явно не разрешено», при этом пользователю недоступны все службы, кроме тех, для которых определены полномочия. Межсетевой экран является единственным видимым снаружи сетевым устройством и скрывает конфигурацию внешней сети от внешнего наблюдения.

Рис. 6.2. Схема с использованием экранирующего маршрутизатора

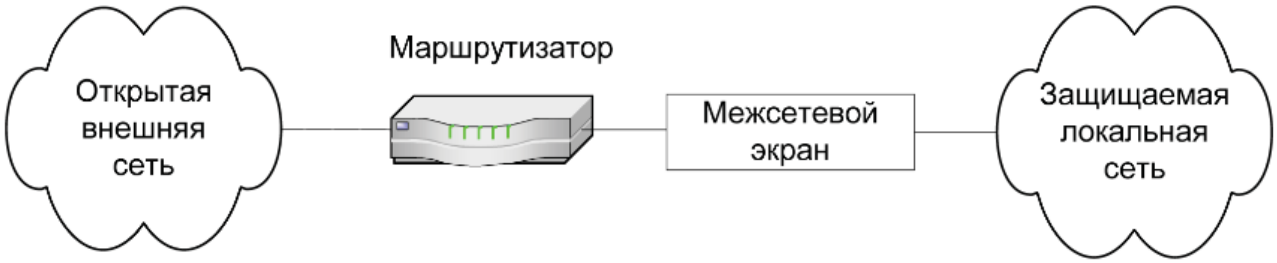

Рис. 6.3. Схема единой защиты локальной сети

Данную схему подключения предпочтительно использовать при отсутствии в локальной сети открытых серверов или когда имеющиеся открытые серверы делаются доступными из внешней сети только для ограниченного числа доверенных пользователей.

Поскольку межсетевой экран использует отдельный хост, то на нем могут быть установлены программы для усиленной аутентификации пользователей и протоколирования доступа.

Схема с защищаемой закрытой и незащищаемой открытой подсетями. Если в составе локальной сети имеются общедоступные открытые серверы, тогда их целесообразно вынести как открытую подсеть до межсетевого экрана (рис. 6.4). Данный способ обладает более высокой защищенностью закрытой части локальной сети, но обеспечивает пониженную безопасность открытых серверов, расположенных до межсетевого экрана. Схему подключения межсетевого экрана с защищаемой закрытой подсетью и незащищаемой открытой подсетью целесообразно использовать в случае, если не предъявляются высокие требования по безопасности к открытой подсети.

Рис. 6.4. Схема с защищаемой закрытой и незащищаемой открытой подсетями

Если к безопасности открытых серверов предъявляются повышенные требования, тогда необходимо использовать **схему с раздельной защитой**

закрытой и открытой подсетей (рис. 6.5). Она может быть построена с использованием одного межсетевого экрана с тремя сетевыми интерфейсами (см. рис. 6.5) или двух межсетевых экранов с двумя сетевыми интерфейсами (рис. 6.6).

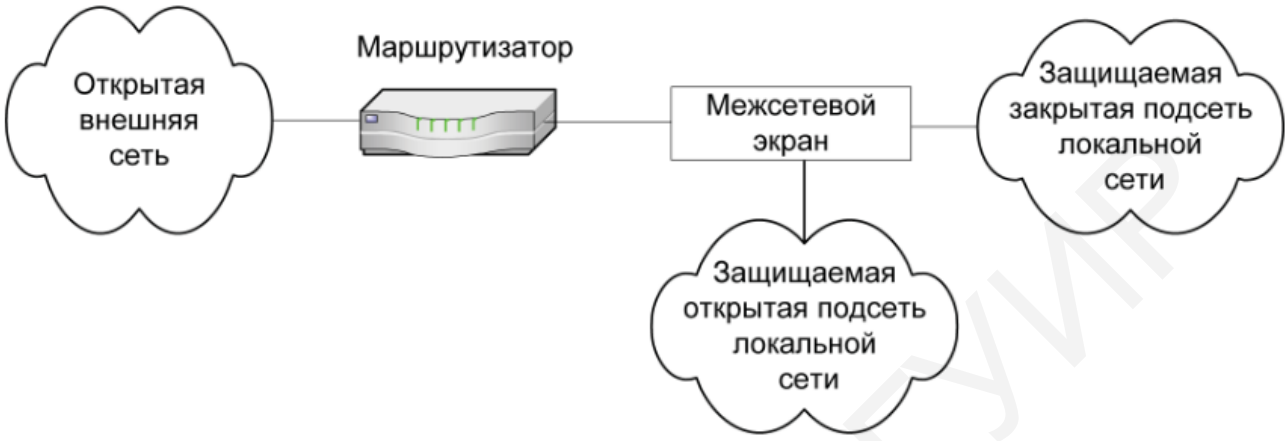

Рис. 6.5. Схема с раздельной защитой закрытой и открытой подсетей с использованием одного межсетевого экрана с тремя сетевыми интерфейсами

В обоих случаях доступ ко внутренним локальным подсетям возможен только через межсетевой экран. Получение доступа к открытой подсети не означает получение доступа к закрытой подсети. Наибольший уровень безопасности обеспечивает схема с использованием двух межсетевых экранов, так как каждый из межсетевых экранов обеспечивает свой эшелон защиты для закрытой подсети, что усложняет задачу атаки злоумышленнику. Средства мониторинга позволяют с высокой эффективностью обнаружить попытку злоумышленника преодолеть два «эшелона» обороны.

Персональные сетевые экраны. Наиболее уязвимым местом корпоративной сети являются рабочие станции конечных пользователей, находящиеся за пределами защищаемого периметра, и особенно мобильные устройства, которые имеют, как правило, низкий уровень защиты. Традиционные межсетевые экраны построены так, что защищаемые пользователи и ресурсы должны находиться под их защитой с внутренней стороны корпоративной или локальной сети, что является невозможным для удаленных пользователей. Поэтому почти повсеместное распространение получила технология персонального сетевого экранирования. Такой персональный экран (Personal Firewall) обеспечивает фильтрование всего исходящего и входящего трафика независимо от других защитных средств.

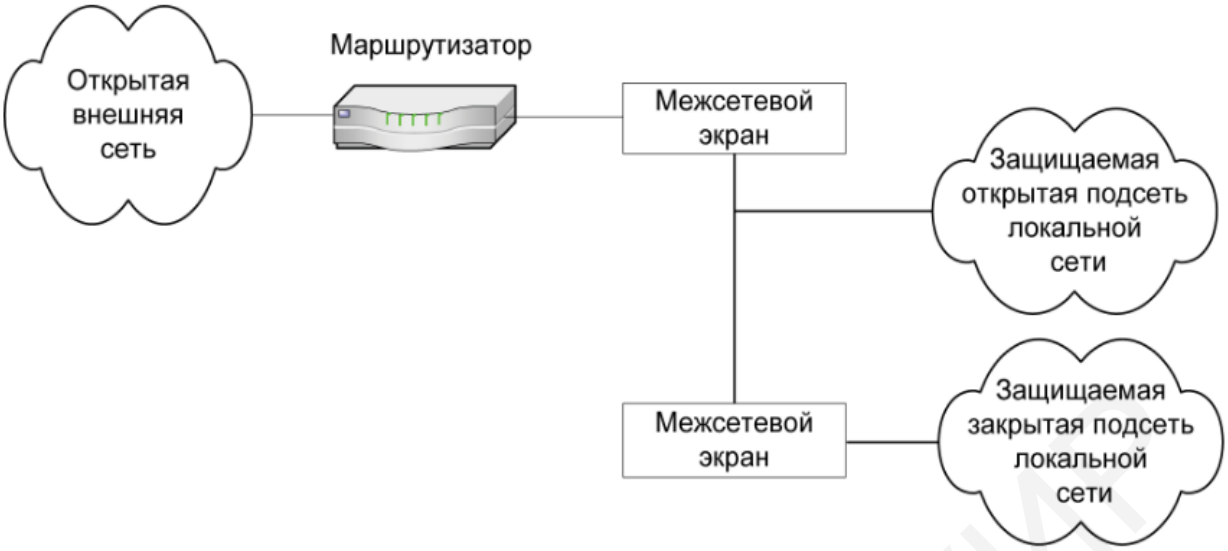

Рис. 6.6. Схема с раздельной защитой закрытой и открытой подсетей с использованием двух межсетевых экранов с двумя сетевыми интерфейсами

Ярким представителем персональных экранов может служить межсетевой экран – брандмауэр Windows, являющийся компонентом операционной системы семейства Windows, начиная с WindowsNT и последующих версий продуктов фирмы Microsoft.

Брандмауэр Windows не позволяет большинству вредоносных программ проникать в систему, обеспечивая защиту от хакеров, вирусов и компьютерных червей, которые пытаются получить доступ к компьютеру через Интернет.

Персональный сетевой экран встроен практически во все виды современного антивирусного программного обеспечения, например, компонент «Сетевой экран», входящий в состав продукта Kaspersky Endpoint Security или Avast Antivirus.

Персональные сетевые экраны являются обязательным компонентом системы распределенного сетевого экранирования.

Распределенный межсетевой экран представляет собой централизованно управляемую совокупность межсетевых экранов, защищающих отдельные компьютеры сети. Функциональные компоненты распределенных межсетевых экранов распределяются по узлам сети и, как правило, обладают различной функциональностью. При обнаружении признаков возможной атаки управляющие модули распределенного межсетевого экрана могут адаптивно изменять конфигурацию, состав и расположение компонентов.

В настоящее время более 60 % сетевых атак выполняются изнутри локальной сети, поэтому традиционный подход к защите периметра становится недостаточно эффективным. В связи с этим наиболее целесообразным будет

сочетание защиты точек входа в Интернет в периметре сети со средствами защиты отдельных компьютеров, серверов и участков локальной сети. Что обеспечивается совместным применением технологий распределенных и персональных межсетевых экранов.

6.2. Задание для самостоятельного выполнения

1) Создать папку с общим доступом на одной из виртуальных машин.

2) Настроить брандмауэр, применив различные политики:

а) доступ к разделяемому ресурсу разрешен только компьютеру с данным IP-адресом;

б) доступ к виртуальной машине разрешен только по заданным портам (например, www или ftp);

в) доступ к виртуальной машине разрешен только по заданным портам (например, www или ftp) и только компьютерам с данным IP-адресом (адресами);

г) доступ к внешним ресурсам разрешен только конкретным программам;

д) конкретной программе разрешен доступ к ресурсам удаленного компьютера с данным IP-адресом по заданному порту;

е) запретить запрос входящего эха (ICMP).

3) Оформить отчет, подтверждающий применение указанных политик.

Для выполнения задания использовать два ПК с ОС Windows и WMware Workstation.

В случае, если студент для выполнения задания использует ПК под управлением Windows 10, то дополнительно можно воспользоваться специальной программой от компании Microsoft – Firewall Control, которая позволяет выполнить более детальные настройки брандмауэра Windows.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Encyclopedia of Cryptography and Security / ed. : Henk C. A. van Tilborg, Sushil Jajodia. – 2nd ed. – Springer, 2011. – 1457 p.

2. Аутентификация. Теория и практика обеспечения доступа к информационным ресурсам : учеб. пособие для вузов / А. А. Афанасьев [и др.] ; под ред. А. А. Шелупанова, С. Л. Груздева, Ю. С. Нахаева. – 2-е изд., стер. – М. : Горячая линия – Телеком, 2012. – 550 с.

3. Фергюсон, Н. Практическая криптография / Н. Фергюсон, Б. Шнайер ; пер. с англ. – М. : Изд. дом «Вильямс», 2004. – 432 с.

4. Фаронов, А. Е. Основы информационной безопасности при работе на компьютере / А. Е. Фаронов. – 2-е изд., испр. – М. : Национальный открытый университет ИНТУИТ, 2016. – 154 с.

5. Лебедь, С. В. Межсетевое экранирование. Теория и практика защиты внешнего периметра / С. В. Лебедь. – М. : МГТУ им. Н. Э. Баумана, 2002. – 304 с.

6. Лапонина, О. Р. Межсетевое экранирование : учеб. пособие / О. Р. Лапонина. – М. : Интернет-университет информационных технологий ; Бином. Лаборатория знаний, 2007. – 343 с.

Св. план 2018, поз. 19

Учебное издание

Захаров Владимир Владимирович

СРЕДСТВА И МЕТОДЫ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

В двух частях

Часть 1

ПОСОБИЕ

Редактор *Е. И. Костина*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *М. В. Касабуцкий*

Подписано в печать 29.08.2019. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 3,84. Уч.-изд. л. 4,0. Тираж 60 экз. Заказ 281.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/338 от 24.03.2014,
№2/133 от 07.04.2014, №3/615 от 07.04.2014.
Ул. П. Бровки, 6, 220013, г. Минск