Рассмотрим анализ объекта защиты информации на следующем примере: **Офис консалтинговой компании.**

Основные сведения: Офис консалтинговой компании расположен на 2-м этаже кирпичного 5-этажного здания с пропускным режимом. Сотрудников — 10 человек. Посетителей до 5 человек в день.

Офисные помещения находятся в аренде, арендодатель — Институт истории СО РАН. Офис представляет собой помещение из трех комнат: кабинет руководства и отдела организации обучения

(в кабинете работают пять сотрудников); кабинет отдела продаж (в кабинете работают пять сотрудников) и кухни. Вход в кабинеты сотрудников из общего коридора помещения. Вход в кухню через кабинет отдела продаж. Вход в кабинеты осуществляется путем получения на вахте ключа от комнаты после предъявления пропуска. При получении и сдаче ключа сотрудник фиксирует дату, время и ФИО в журнале регистрации получения и сдачи ключей от помещений.

1. Описать направления деятельности объекта защиты информации.

- бизнес-консалтинг (организационный, управленческий, кадровый, маркетинговый и рекламный);
- бизнес-тренинги (корпоративные и открытые программы);
- личностный менеджмент (тренинги, направленные на развитие личности).

2. Описать организационную структуру объекта защиты информации

Организационная структура консалтинговой компании:

- генеральный директор;
- коммерческий директор;
- отдел организации обучения;
- отдел продаж.

Бухгалтерские услуги, а также услуги по системному администрированию и техническому обслуживанию находятся на аутсорсинге.

3. Описать основные виды информации в бумажном и электронном виде, в том числе внутреннюю нормативную документацию.

Основной вид информации в бумажном и электронном виде:

1) данные о клиентах компании:

- общая информация;
- персональные данные сотрудников компаний-клиентов;
- результаты тестирования и обучения сотрудников компаний-клиентов;

2) коммерческие предложения;

3) договоры с клиентами и акты исполнения работ;

4) персональные данные сотрудников компании.

Внутренняя нормативная документация:

- должностная инструкция тренера-консультанта;
- должностная инструкция руководителя отдела продаж;
- должностная инструкция менеджера по продажам;
- политика безопасности компании;
- форма договора об оказании услуг;
- форма договора трудового найма;
- форма заявления о согласии сотрудника на сбор и обработку персональных данных;
- форма заявления о согласии сотрудника компании-клиента на сбор и обработку персональных данных.

4. Представить перечень конфиденциальной информации (включая персональные данные)

1) *сведения о компаниях-клиентах*:

- протоколы переговоров;
- технические задания;
- результаты тестирования сотрудников;
- состав обучающих программ;
- результаты обучения сотрудников;
- персональные данные сотрудников компаний-клиентов (фамилия, имя, отчество сотрудника, год, месяц, дата рождения, личный электронный адрес, внутренний телефон в компании, личный мобильный телефон, сведения об пройденных обучающих программах, сведения о предпочтениях в области обучения, сведения о предыдущем месте работы, результаты тестирования в процессе обучения);

2) *сведения об управлении компанией*:

- планы развития компании;
- результаты аудита деятельности компании;
- управленческие решения;

3) *конфиденциальные договоры с клиентами*;

4) *данные об уровне заработной платы сотрудников*;

5) *персональные данные сотрудников*:

- фамилия, имя, отчество сотрудника;
- год, месяц, дата рождения;
- место рождения;
- место регистрации;
- место проживания;
- паспортные данные;
- семейное положение;
- образование;
- уровень дохода;
- сведения о предыдущем месте работы;
- сведения о состоянии здоровья;

- результаты тестирования при приеме на работу;

6) *стоимость обучающих программ*;

7) *бухгалтерская отчетность*;

8) *налоговая отчетность.*

5. Представить спецификацию оборудования и программных средств, используемых на объекте защиты информации (при этом объект защиты информации рассматривается как комплексная вычислительная система, выполняющая определенный набор функций).

Спецификация оборудования и программных средств ЛВС: компьютеры с процессором семейства Intel; объем оперативной памяти 2 Гбайта и выше; манипуляторы типа мышь и клавиатура; операционные системы семейства *MS Windows*, а также *MS Windows Server 2003*; браузеры *MS Internet Explorer* и *Opera*; количество сотрудников, имеющих ПК — 10. Топология ЛВС — звезда. Кабельная система — беспроводная сеть и витая пара. Вид сети (сетевой операционной системы) — бессерверная сеть. Все компьютеры сети подключены к коммутатору витой парой. Оба кабинета оборудованы многофункциональными аппаратами (принтер-сканер—копир—факс).

На рис. 1 представлена схема расположения ПК.

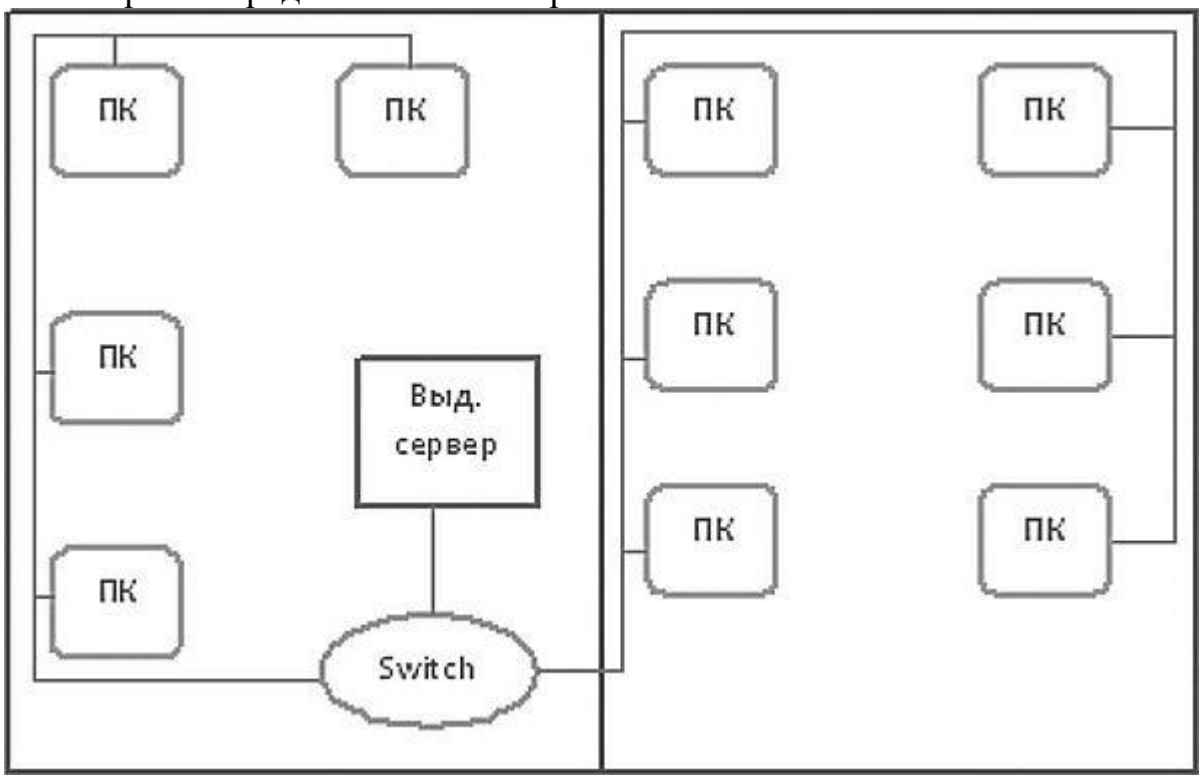


Рис. 1 Схема расположения ПК

Для обработки информации используется 10 компьютеров и 1 ноутбук. К каждому рабочему месту привязан внутренний телефонный номер (общих телефонных номеров — два). Хранилище бумажных документов находится в кабинете отдела продаж.

6. Описать средства защиты информации:

На объекте используются следующие меры по защите информации.

Все сотрудники компании наняты по договору трудового найма, включающего в себя пункт по сохранению персональных данных и коммерческой тайны.

Вход в здание осуществляется по пропускам через пункт охраны. Получение ключей от помещений осуществляется через регистрацию получения ключа в журнале регистрации получения и сдачи ключей от помещений на вахте после предъявления пропуска в здание.

Все помещения оборудованы системами противопожарной сигнализации.

Осуществляется идентификация и аутентификация пользователей. Существует разграничение прав доступа. Доступ в ИС разрешен только зарегистрированным пользователям. Каждому пользователю настроены соответствующие его должности права доступа на различные объекты ИС. Перечень объектов и субъектов доступа определяется на основе прав, которые определены для каждого пользователя ИС. Защита доступа к ресурсам осуществляется при помощи пароля.

Организационные и технические меры защиты конфиденциальной информации, применяемые в консалтинговой компании.

- Идентификация и аутентификация субъектов и объектов доступа осуществляется встроенными средствами *MS Windows.* Защита обратной связи при вводе информации, используемой для аутентификации, осуществляется при помощи сокрытия ее посредством специальных символов.
- Управление доступом субъектов к объектам доступа. Пользователи и администраторы И С имеют различные обязанности и наделены разными правами.
- Обеспечение целостности информационной системы и информации.

Виды угроз

Угрозы возникают из противоречий экономических интересов различных элементов, взаимодействующих как внутри, так и вне социально-экономической системы -- в том числе и в информационной сфере. Они и определяют содержание и направления деятельности по обеспечению общей и информационной безопасности. Следует отметить, что анализ проблем экономической безопасности необходимо проводить, учитывая взаимосвязи экономических противоречий, угроз и потерь, к которым может приводить реализация угроз.

Угрозы можно классифицировать по нескольким критериям:

– по важнейшим составляющим информационной безопасности (доступность, целостность, конфиденциальность), против которых направлены угрозы в первую очередь;

– по компонентам информационных систем и технологий (данные, программно-аппаратные комплексы, сети, поддерживающая инфраструктура), на которые угрозы непосредственно нацелены;

– по способу осуществления (случайные или преднамеренные действия, события техногенного или природного масштаба);

– по локализации источника угроз (вне или внутри информационной технологии или системы).

Основные нарушения:

– физической целостности (уничтожение, разрушение элементов);

– логической целостности (разрушение логических связей);

– содержания (изменение блоков информации, внешнее навязывание ложной информации);

– конфиденциальности (разрушение защиты, уменьшение степени защищенности информации);

– прав собственности на информацию (несанкционированное копирование, использование).

Три наиболее выраженные угрозы:

– подверженность физическому искажению или уничтожению информации;

– возможность несанкционированной (случайной или злоумышленной) модификации информации;

– опасность несанкционированного (случайного и преднамеренного) получения информации лицами, для которых она не предназначена.

Существует несколько типов атак: атаки, основанные на ошибках реализации программного обеспечения (срыв стека, использование конвейера); атаки, основанные на неправильной конфигурации сервера и атаки, основанные на уязвимости сетевых протоколов (внедрение ложного DNS, перехват трафика).

Даже при условии отсутствия ошибок реализации правильно настроенный почтовый сервер может быть атакован, если не предпринять особых мер предосторожности, о которых осведомлен далеко не каждый администратор.

Ниже представлен перечень некоторых атак, которым подвержены сегодня организации, использующие почтовые e-mail-системы:

– трояны;

– фишинг организаций;

– Backscatter - bounce-сообщения;

– Email Bombing - почтовая бомбардировка;

– Denial of Email Service (DoES) -отказ почтовой службы;

– эксплуатация (эксплойты) открытых релеев

Простой взгляд на Web-сайты большинства компаний может дать атакующему злоумышленнику список e-mail-адресов для реализации, например, фишинг-атаки, рассылки вредоносного вложения или ссылки на хакерский сайт, после посещения которого на компьютеры пользователей

незаметно будет установлено программное обеспечение, ворующее их пароли к корпоративным ресурсам или другую не менее ценную персональную информацию.

Характер происхождения угроз

Угрозы безопасности информации в современных системах ее обработки определяются умышленными (преднамеренные угрозы) и естественными (непреднамеренные угрозы разрушающими и искажающими воздействиями внешней среды, надежностью функционирование средств обработки информации, а также преднамеренным корыстным воздействием несанкционированных пользователей, целью которых является хищение, уничтожение, разрушение, несанкционированная модификация и использование обрабатываемой информации. При этом под умышленными или преднамеренными понимаются такие угрозы, которые обуславливаются злоумышленными действиями людей. Случайными или естественными являются угрозы, не зависящие от воли людей. Классификация угроз:

1. Умышленные факторы:

1) хищение носителей информации;

2) подключение к каналам связи;

3) перехват электромагнитных излучений (ЭМИ);

4) несанкционированный доступ;

5) разглашение информации;

6) копирование данных.

2. Естественные факторы:

1) несчастные случаи (пожары, аварии, взрывы);

2) стихийные бедствия (ураганы, наводнения, землетрясения);

3) ошибки в процессе обработки информации (ошибки пользователя, оператора, сбои аппаратуры).

Пример:

Угрозы, связанные с сервером, содержащим бухгалтерский отчет крупной компании:

Серверы меньше подвержены атакам, поскольку они, как правило, работают в контролируемых условиях, обслуживаются квалифицированными администраторами и обладают одним или несколькими уровнями защиты. Сервер отчетов является сервером без сохранения состояния, который хранит содержимое и данные приложений во внешнем хранилище. Одна из самых больших угроз для установки сервера отчетов — несанкционированный доступ и вмешательство в базу данных и файлы конфигурации сервера отчетов. Менее очевидная, но столь же важная угроза определяется способом создания отчетов и кругом лиц, которым предоставлено разрешение на публикацию содержимого на сервере отчетов. Определения отчета — сборки, которые запускаются на компьютере сервера с полным уровнем доверия. Определения отчетов могут содержать другие пользовательские сборки, которые также выполняются на сервере. Если отчет или пользовательская

сборка содержит вредоносный код, этот код будет выполняться на компьютере сервера отчетов с учетными данными пользователя, запросившего отчет. Причиной других неявных угроз может быть структура отчета, в которой непреднамеренно раскрываются конфиденциальные данные. Например, если сотрудник запускает отчет, в котором в качестве параметра используется идентификатор сотрудника, то он не должен иметь возможности видеть сведения о другом сотруднике, вставляя произвольные идентификаторы в URL-адрес отчета. Наконец, учитывайте методы распространения отчетов в организации. Можно настроить сервер отчетов, чтобы уменьшить возможность доставки отчетов вне организации. Можно также использовать разрешения файловой системы и разрешения сервера отчетов, чтобы только авторизованные пользователи могли открыть отчет.

В следующей таблице описаны угрозы и меры по снижению уязвимости.

Таблица 1

Компонент или операция	Угроза	Меры по снижению уязвимости
Параметры конфигурации, хранящиеся в файле Web.config и файлах конфигурации приложения на компьютере сервера отчетов.	Злоумышленник может получить доступ к компьютеру, обнаружить незашифрованный или незащищенный файл конфигурации и внести изменения в этот файл.	Назначьте разрешения для файла. По умолчанию разрешения предоставляются группам безопасности служб Службы Reporting Services, созданным в процессе установки.
Веб-служба сервера отчетов обрабатывает запросы по требованию, переданные через соединения TCP/IP.	Злоумышленник может запустить атаку типа «запрет в обслуживании», которая принимает следующие формы. На целевой сервер направляются множественные запросы без проверки подлинности. Неполные запросы направляются на целевой сервер, но никогда не завершаются. Запросы чрезмерно велики; злоумышленник начинает запрос, а затем посылает большой объем данных на сервер.	Сервер отчетов удаляет все запросы без проверки подлинности в течение двух минут, что может уменьшить последствия атаки типа «запрет в обслуживании». Двухминутный интервал имеет фиксированную длительность, его нельзя уменьшить. Если атака основана на операциях передачи данных на сервер отчетов, можно уменьшить значение элементаmax Request Lengthв файле Machine.config. По умолчанию устанавливаемая платформой ASP.NET верхняя граница для объема передаваемых на сервер файлов составляет 4 МБ. Обратите внимание, что уменьшение значенияmax Request Length должно быть временной мерой. Необходимо вернуть прежнее значение, если передача больших файлов (например, моделей) выполняется часто. Дополнительные сведения о настройке параметра max Request Lengthв установке служб Reporting Services см. в разделе. Максимальные размеры отчетов и моментальных снимков.
Службы Службы Reporting Services поддерживают расширяемую архитектуру, которая позволяет	Злоумышленник может вставить вредный код в пользовательский модуль.	Развертывайте модули только от доверенных пользователей и организаций.

развертывать сторонние модули обработки данных, модули доставки и модули подготовки отчетов. Также можно развертывать пользовательские конструкторы запросов. Модули должны выполняться с полным уровнем доверия.		

Одним из наиболее распространенных и многообразных способов воздействия на информационную систему, позволяющим нанести ущерб любой из составляющих информационной безопасности является несанкционированный доступ. Несанкционированный доступ возможен из-за ошибок в системе защиты, нерационального выбора средств защиты, их некорректной установки и настройки.

Под несанкционированным доступом понимается получение лицами в обход системы защиты с помощью программных, технических и других средств, а также в силу случайных обстоятельств доступа к обрабатываемой и хранимой на объекте информации.

Рассмотрим относительно полное множество каналов несанкционированного получения информации, сформированного на основе такого показателя, как степень взаимодействия злоумышленника с элементами объекта обработки информации и самой информацией.

К **первому классу** относятся каналы от источника информации при НСД к нему.

1. Хищение носителей информации.
2. Копирование информации с носителей (материально-вещественных, магнитных и т. д.).
3. Подслушивание разговоров (в том числе аудиозапись).
4. Установка закладных устройств в помещение и съем информации с их помощью.
5. Выведывание информации обслуживающего персонала на объекте.
6. Фотографирование или видеосъемка носителей информации внутри помещения.

Ко **второму классу** относятся каналы со средств обработки информации при НСД к ним.

1. Снятие информации с устройств электронной памяти.
2. Установка закладных устройств в СОИ.
3. Ввод программных продуктов, позволяющих злоумышленнику получать информацию.
4. Копирование информации с технических устройств отображения (фотографирование с мониторов и др.).

К **третьему классу** относятся каналы от источника информации без НСД к нему.

2. Получение информации по акустическим каналам (в системах вентиляции, теплоснабжения, а также с помощью направленных микрофонов).
3. Получение информации по виброакустическим каналам (с использованием акустических датчиков, лазерных устройств).
4. Использование технических средств оптической разведки (биноклей, подзорных труб и т. д.).
5. Использование технических средств оптико-электронной разведки (внешних телекамер, приборов ночного видения и т. д.).
6. Осмотр отходов и мусора.
7. Выведывание информации у обслуживающего персонала за пределами объекта.
8. Изучение выходящей за пределы объекта открытой информации (публикаций, рекламных проспектов и т. д.).

К **четвертому классу** относятся каналы со средств обработки информации без НСД к ним.

1. Электромагнитные излучения СОИ (паразитные электромагнитные излучения (ПЭМИ), паразитная генерация усилительных каскадов, паразитная модуляция высокочастотных генераторов низкочастотным сигналом, содержащим конфиденциальную информацию).
2. Электромагнитные излучения линий связи.
3. Подключения к линиям связи.
4. Снятие наводок электрических сигналов с линий связи.
5. Снятие наводок с системы питания.
6. Снятие наводок с системы заземления.
7. Снятие наводок с системы теплоснабжения.
8. Использование высокочастотного навязывания и т. д.
9.

Пример 1

Классы каналов несанкционированного получения информации к данным, хранящимся на сервере организации

Утечка информации, хранящаяся на сервере возможно по следующим каналам:

- Радиоканалы - это каналы передачи информации, при которой в качестве носителя информации используются радиоволны, свободно распространяемые в пространстве.
- Инфракрасный канал - это канал передачи данных, не требующий для своего функционирования проводных соединений.
- Проводные линии - это линии связи, состоящие из направленных средств передачи, предназначенные для организации связи.

В качестве проводных линий при передаче информации к внешним средствам регистрации могут быть использованы:

- сети переменного тока;
- линии телефонной связи;
- радиотрансляционные и технологические линии;
- специально проложенные проводные линии.

Пример 2: сервер, содержащий бухгалтерские отчеты крупной компании

К рассматриваемому объекту из первого класса можно отнести такие угрозы, как:

- выведывание информации обслуживающего персонала на объекте.

Из второго класса:

- ввод программных продуктов, позволяющих злоумышленнику получать информацию.
- Каналы угроз третьего класса, относительно данного объекта:
- выведывания информации у обслуживающего персонала за пределами объекта;
- изучения выходящей за пределы объекта открытой информации (публикаций, рекламных проспектов и т.д.);

Каналы угроз четвертого класса:

- электромагнитные излучения линий связи;
- подключения к линиям связи;
- подключение к базам данных и ПЭВМ по компьютерным сетям.

Целостность информации – точность, достоверность и полнота информации, на основе которой принимаются решения и ее защищенность от возможных непреднамеренных и злоумышленных искажений.

Последствия нарушения целостности информации обычно носят тактический характер, а тяжесть последствий зависит от ситуации.

Угрозы нарушения целостности – это угрозы, связанные с вероятностью модификации той или иной информации, хранящейся в информационных системах (ИС). Нарушение целостности может быть вызвано различными факторами – от умышленных действий персонала до выхода из строя оборудования.

Действия, направленные на нарушение целостности информации, подразделяются на субъективные преднамеренные и объективные преднамеренные.

1. Субъективные

1.1. Преднамеренные

1.1.1. Диверсия (организация пожаров, взрывов, повреждений электропитания и др.)

1.1.2. Непосредственные действия над носителем (хищение, подмена носителей, уничтожение информации)

1.1.3. Информационное воздействие (электромагнитное облучение, ввод в компьютерные системы разрушающих программных средств, воздействие на психику личности психотропным оружием)

1.2. Непреднамеренные

1.2.1. Отказы обслуживающего персонала (гибель, длительный выход из строя)

1.2.2. Сбои людей (временный выход из строя)

1.2.3. Ошибки людей

2. Объективные, непреднамеренные

2.1. Отказы (полный выход из строя) аппаратуры, программ, систем питания и жизнеобеспечения

2.2. Сбои (кратковременный выход из строя) аппаратуры, программ, систем питания и жизнеобеспечения

2.3. Стихийные бедствия (наводнения, землетрясения, ураганы)

2.4. Несчастные случаи (пожары, взрывы, аварии)

2.5. Электромагнитная несовместимость

Пример: Телефонная сеть

Основными причинами утечки информации являются:

1. несоблюдение персоналом норм, требований, правил эксплуатации АС, функционирующих в рамках АТС;

2. ошибки в проектировании систем защиты программного обеспечения компьютерной сети АТС;

3. получение защищаемой информации разведками.

Система защиты информации — совокупность взаимосвязанных средств, методов и мероприятий, направленных на предотвращение уничтожения, искажения, несанкционированного получения конфиденциальных сведений, отображенных полями, электромагнитными, световыми и звуковыми волнами или вещественно-материальными носителями в виде сигналов, образов, символов, технических решений и процессов.

Руководящий документ, подписанный по решению председателя Гостехкомиссии России 30 марта 1992 г. устанавливает классификацию автоматизированных систем, подлежащих защите от несанкционированного доступа к информации, и требования по защите информации в ПК различных классов.

Концепция защиты от несанкционированного доступа к информации

Идейной основой набора "Руководящих документов" является "Концепция защиты СВТ и АС от *НСД* к информации". Концепция "излагает систему взглядов, основных принципов, которые закладываются в основу

проблемы защиты информации от *НСД*, являющейся частью общей проблемы безопасности информации".

В "Концепции" различаются понятия средств вычислительной техники и автоматизированной системы, аналогично тому, как в "Европейских Критериях" проводится деление на продукты и системы. Более точно, "Концепция предусматривает существование двух относительно самостоятельных и, следовательно, имеющих отличие направлений в проблеме защиты информации от *НСД*. Это - направление, связанное с СВТ, и направление, связанное с АС. Отличие двух направлений порождено тем, что СВТ разрабатываются и поставляются на рынок лишь как элементы, из которых в дальнейшем строятся функционально ориентированные АС, и поэтому, не решая прикладных задач, СВТ не содержат пользовательской информации.

Помимо пользовательской информации при создании автоматизированных систем появляются такие отсутствующие при разработке СВТ характеристики АС, как полномочия пользователей, модель нарушителя, технология обработки информации."

Существуют различные способы покушения на информационную безопасность: радиотехнические, акустические, программные и т.п. Среди них *НСД* выделяется как "доступ к информации, нарушающий установленные правила разграничения доступа, с использованием штатных средств, предоставляемых СВТ или АС. Под штатными средствами понимается совокупность программного, микропрограммного и технического обеспечения СВТ или АС."

В "Концепции" формулируются следующие основные принципы защиты от *НСД* к информации:

"...Защита СВТ обеспечивается комплексом программно-технических средств.

Защита АС обеспечивается комплексом программнотехнических средств и поддерживающих их организационных мер.... Защита АС должна обеспечиваться на всех технологических этапах обработки информации и во всех режимах функционирования, в том числе при проведении ремонтных и регламентных работ.

Программно-технические средства защиты не должны существенно ухудшать основные функциональные характеристики АС (надежность, быстродействие, возможность изменения конфигурации АС).

Неотъемлемой частью работ по защите является оценка эффективности средств защиты, осуществляемая по методике, учитывающей всю совокупность технических характеристик оцениваемого объекта, включая технические решения и практическую реализацию средств защиты.

Защита АС должна предусматривать контроль эффективности средств защиты от *НСД*. Этот контроль может быть либо периодическим, либо инициироваться по мере необходимости пользователем АС или контролирующими органами."

"Концепция" ориентируется на физически защищенную среду, проникновение в которую посторонних лиц считается невозможным, поэтому нарушитель определяется как "субъект, имеющий доступ к работе с штатными средствами АС и СВТ как части АС.

Нарушители классифицируются по уровню возможностей, предоставляемых им штатными средствами АС и СВТ. Выделяется четыре уровня этих возможностей. Классификация является иерархической, т.е. каждый следующий уровень включает в себя функциональные возможности предыдущего.

Первый уровень определяет самый низкий уровень возможностей ведения диалога в АС - запуск задач (программ) из фиксированного набора, реализующих заранее предусмотренные функции по обработке информации.

Второй уровень определяется возможностью создания и запуска собственных программ с новыми функциями по обработке информации.

Третий уровень определяется возможностью управления функционированием АС, т.е. воздействием на базовое программное обеспечение системы и на состав и конфигурацию ее оборудования.

Четвертый уровень определяется всем объемом возможностей лиц, осуществляющих проектирование, реализацию и ремонт технических средств АС, вплоть до включения в состав СВТ собственных технических средств с новыми функциями по обработке информации.

В своем уровне нарушитель является специалистом высшей квалификации, знает все о АС и, в частности, о системе и средствах ее защиты."

В качестве главного средства защиты от *НСД* к информации в "Концепции" рассматривается система разграничения доступа (СРД) субъектов к объектам доступа. Основными функциями СРД являются:

- "реализация правил разграничения доступа (ПРД) субъектов и их процессов к данным;
- реализация ПРД субъектов и их процессов к устройствам создания твердых копий;
- изоляция программ процесса, выполняемого в интересах субъекта, от других субъектов;
- управление потоками данных с целью предотвращения записи данных на носители несоответствующего грифа;
- реализация правил обмена данными между субъектами для АС и СВТ, построенных по сетевым принципам."

Кроме того, "Концепция" предусматривает наличие обеспечивающих средств для СРД, которые выполняют следующие функции:

- "идентификацию и опознание (аутентификацию) субъектов и поддержание привязки субъекта к процессу, выполняемому для субъекта;
- регистрацию действий субъекта и его процесса;
- предоставление возможностей исключения и включения новых субъектов и объектов доступа, а также изменение полномочий субъектов;

- реакцию на попытки *НСД*, например, сигнализацию, блокировку, восстановление после *НСД,* тестирование;
- очистку оперативной памяти и рабочих областей на магнитных носителях после завершения работы пользователя с защищаемыми данными;
- учет выходных печатных и графических форм и твердых копий в АС;
- контроль целостности программной и информационной части как СРД, так и обеспечивающих ее средств."

Видно, что функции системы разграничения доступа и обеспечивающих средств, предлагаемые в "Концепции", по своей сути близки к аналогичным положениям "Оранжевой книги". Это вполне естественно, поскольку близки и исходные посылки - защита от несанкционированного доступа к информации в условиях физически безопасного окружения.

Технические средства защиты от *НСД*, согласно "Концепции", должны оцениваться по следующим основным параметрам:

- "степень полноты охвата ПРД реализованной СРД и ее качество;
- состав и качество обеспечивающих средств для СРД;
- гарантии правильности функционирования СРД и обеспечивающих ее средств."

Примечание. Средства СВТ могут рассматриваться как объекты НСД только при записи и хранении в них информации. "Голое" железо интерес для хакеров, как правило, не представляет.

"Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности."

Краткое содержание важнейшего документа "Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности." В этом документе определены семь классов защищенности СВТ от НСД к информации. Самый низкий класс - седьмой, самый высокий первый. Каждый класс наследует требования защищенности предыдущего класса.

Изложенные ниже требования к показателям защищенности предъявляются к общесистемным программным средствам и операционным системам.

Совокупность всех средств СВТ защиты образует комплекс средств защиты (КСЗ).

В зависимости от реализованных моделей защиты и надежности их проверки классы подразделяются на четыре группы. Первая группа включает только один седьмой класс (минимальная защищенность).

Вторая группа характеризуется избирательной защитой и включает шестой и пятый классы. Избирательная защита предусматривает контроль доступа поименованных субъектов к поименованным объектам системы. При этом для каждой пары "субъект-объект" должны быть определены разрешенные типы доступа. Контроль доступа должен быть применим к

каждому объекту и каждому субъекту (индивиду или группе равноправных индивидов).

Третья группа характеризуется полномочной защитой и включает четвертый, третий и второй классы. Полномочная защита предусматривает присвоение каждому субъекту и объекту системы классификационных меток, указывающих место субъекта (объекта) в соответствующей иерархии. Классификационные метки на объекты устанавливаются пользователем системы или специально выделенным субъектом. Обязательным требованием для классов, входящих в эту группу, является реализация диспетчера доступа (в иностранной литературе - reference monitor, монитор ссылок). Контроль доступа должен осуществляться применительно ко всем объектам при явном и скрытом доступе со стороны любого из субъектов. Решение о санкционированности запроса на доступ должно приниматься только при одновременном разрешении его и избирательными и полномочными правилами разграничения доступа (ПРД).

Четвертый класс характеризуется верифицированной защитой и содержит только первый класс.

Для присвоения класса защищенности система должна содержать руководство администратора по системе, руководство пользователя, тестовую и конструкторскую (проектную) документацию.

Перечень показателей по классам защищенности СВТ приведен в таблице 1. Их краткое описание приведено ниже.

Таблица 1. Показатели по классам защищенности СВТ Гостехкомиссии РФ

№№ п/п	Наименование Показателя	Класс защищенности					
		С6 (С1)	С5 (С2)	4 (В1)	3 (В2)	2 (В3)	1 (А1)
Политика безопасности							
1	Избирательная политика безопасности	+	+	+	=	+	=
2	Полномочная политика безопасности	-	-	+	=	=	=
3	Повторное использование объектов	-	+	+	+	=	=
4	Изоляция модулей	-	-	+	=	+	=
5	Маркировка документов	-	-	+	=	=	=
6	Защита ввода и вывода на отчуждаемый физический носитель информации	-	-	+	=	=	=
7	Сопоставление пользователя с устройством	-	-	+	=	=	=
Учет							

8	Идентификация и аутентификация	+	=	+	=	=	=
9	Регистрация	-	+	+	+	=	=
10	Взаимодействие пользователя с КСЗ	-	-	-	+	=	=
Гарантии							
11	Гарантии проектирования	-	+	+	+	+	+
12	Гарантии архитектуры	-	-	-	-	-	+
13	Надежное восстановление	-	-	-	+	=	=
14	Целостность КСЗ	-	+	+	+	=	=
15	Контроль модификации	-	-	-	-	+	=
16	Контроль дистрибуции	-	-	-	-	+	=
17	Тестирование	+	+	+	+	+	=
Документация							
18	Руководство пользователя	+	=	=	=	=	=
19	Руководство по КСЗ	+	+	=	+	+	=
20	Тестовая документация	+	+	+	+	+	=
21	Конструкторская (проектная) документация	+	+	+	+	+	+

Обозначения:

"-" - нет требований к данному классу;

"+" - новые или дополнительные требования,

"=" - требования совпадают с требованиями к СВТ предыдущего класса.

Шестой класс защищенности

КСЗ этого класса должен предоставлять возможности санкционированного изменения ПРД, списка пользователей я и списка защищаемых объектов. Права изменять ПРД должны предоставляться выделенным субъектам (администрации, службе безопасности и т.д.).

Перед работой пользователь должен пройти процедуру проверки подлинности (аутентификацию). По ее результатам ему присваиваются определенные права по доступу к объектам в системе (авторизация пользователя). Неавторизованный пользователь не должен иметь доступа к защищаемым ресурсам.

Пятый класс защищенности

На начальном этапе проектирования СВТ должна быть построена модель защиты. Модель должна включать в себя ПРД к объектам и непротиворечивые правила изменения ПРД.

В КСЗ этого класса должны быть предусмотрены средства управления, ограничивающие распространение прав на доступ.

При первоначальном назначении или при перераспределении внешней памяти КСЗ должен предотвращать доступ субъекту к остаточной информации.

КСЗ должен быть в состоянии осуществлять регистрацию следующих событий:

- использование идентификационного и аутентификационного механизма;
- запрос на доступ к защищаемому объекту ;
- создание и уничтожение объекта;
- действия по изменению ПРД.

Для каждого из регистрируемых событий должны

- регистрироваться:
- дата и время;
- субъект, осуществляющий регистрируемое действие;
- тип события (если регистрируется запрос на доступ, то следует отмечать объект и тип доступа);
- успешно ли осуществилось событие (обслужен запрос на доступ или нет).

КСЗ должен содержать средства выборочного ознакомления с регистрационной информацией.

Периодическому контролю целостности должны быть подвержены программная и информационная части КСЗ.

Четвертый класс защищенности

Проектирование КСЗ должно начинаться с построения модели защиты, включающей в себя непротиворечивые ПРД, непротиворечивые правила изменения ПРД и правила работы с устройствами ввода и вывода информации и каналами связи.

Защита должна осуществляться не только от "явных" (осуществляемых при помощи системных средств, языков высокого уровня), но и от скрытых (осуществляемых другим путем, в т.ч. и с использованием собственных программ работы с устройствами) запросов на получение доступа .

Кроме внешней памяти очистке должны быть подвержена и оперативная память при ее перераспределении.

При наличии в СВТ мультипрограммирования в КСЗ должен существовать программно-технический механизм, изолирующий программные модули одного процесса (одного субъекта), от программных модулей других процессов (других субъектов).

При выводе защищаемой информации на документ в начале и конце КСЗ должен проставить штамп установленного образца.

КСЗ должен различать каждое устройство ввода-вывода и каждый канал связи как произвольно используемые или идентифицированные ("помеченные"). При вводе с "помеченного" устройства (выводе на "помеченное" устройство) КСЗ должен обеспечивать соответствие между меткой вводимого (выводимого) объекта (классификационным уровнем) и меткой устройства. Такое же соответствие должно обеспечиваться при работе с "помеченным" каналом связи. Изменения в назначении и разметке устройств и каналов должны осуществляться только под контролем КСЗ.

Идентифицированный КСЗ должен включать в себя механизм, посредством которого санкционированный пользователь надежно сопоставляется выделенному устройству.

КСЗ должен обладать способностью надежно связывать полученную идентификацию со всеми действиями данного пользователя.

Для СВТ данного класса защищенности требования по регистрации дополнительно включают регистрацию попыток доступа, и всех действий оператора и выделенных пользователей (администраторов защиты и т.п.).

Программы КСЗ должны выполняться в отдельной части оперативной памяти.

Третий класс защищенности

На начальном этапе проектирования КСЗ должна строиться модель защиты, содержащая непротиворечивые правила изменения ПРД, правила работы с устройствами ввода и вывода и формальную модель механизма управления доступом.

Должна предлагаться высокоуровневая спецификация части КСЗ, реализующего механизм управления доступом и его интерфейсов. Эта спецификация должна быть проверена на соответствие заданных принципов разграничения доступа.

Для обеспечения возможности изучения, анализа, проверки и модификации КСЗ должен быть хорошо структурирован, его структура должна быть модульной и четко определенной.

Для третьего класса защищенности КСЗ должен осуществлять очистку оперативной и внешней памяти. Очистка должна производиться путем записи маскирующей информации в память при ее освобождении (перераспределении).

Процедуры восстановления после сбоев и отказов оборудования должны обеспечивать полное восстановление свойств КСЗ.

Необходимо осуществлять периодический контроль за целостностью КСЗ. Программы должны выполняться в отдельной части оперативной памяти.

Второй класс защищенности

ПРД избирательной защиты должны быть эквивалентны ПРД полномочной защиты.

Гарантии изоляции модулей различных субъектов должны быть основаны на архитектуре СВТ.

Дополнительно требуется, чтобы высокоуровневые спецификации КСЗ были отображены последовательно в спецификации одного или нескольких нижних уровней, вплоть до реализации высокоуровневой спецификации КСЗ на языке программирования высокого уровня. При этом методами верификации должно осуществляться доказательство соответствия каждого такого отображения.

При проектировании, построении и сопровождении СВТ должно быть предусмотрено управление конфигурацией СВТ, т.е. контроль изменений в формальной модели, спецификациях разных уровней, документации, исходном тексте, версии в объектном коде. Оригиналы программ должны быть защищены.

Должен осуществляться контроль точности копирования в СВТ при изготовлении копий с образца. Изготовляемая копия должна гарантированно повторять образец.

Первый класс защищенности

Дополнительно требуется верификация соответствия объектного кода тексту КСЗ на языке высокого уровня.

КСЗ должен обладать механизмом, гарантирующим перехват диспетчером доступа всех обращений субъектов к объектам.

Классификация автоматизированных систем по уровню защищенности от НСД

Для автоматизированных систем как объединений средств СВТ ГТК РФ разработала специальную классификацию автоматизированных систем по уровню защищенности от НСД

Классификация автоматизированных систем устроена иначе. Снова обратимся к соответствующему "Руководящему документу".

"...устанавливается девять классов защищенности АС от *НСД* к информации.

Каждый класс характеризуется определенной минимальной совокупностью требований по защите.

Классы подразделяются на три группы, отличающиеся особенностями обработки информации в АС.

В пределах каждой группы соблюдается иерархия требований по защите в зависимости от ценности (конфиденциальности) информации и, следовательно, иерархия классов защищенности АС.

... Третья группа классифицирует АС, в которых работает один пользователь, допущенный ко всей информации АС, размещенной на носителях одного уровня конфиденциальности. Группа содержит два класса - 3Б и 3А.

Вторая группа классифицирует АС, в которых пользователи имеют одинаковые права доступа (полномочия) ко всей информации АС, обрабатываемой и (или) хранимой на носителях различного уровня конфиденциальности. Группа содержит два класса - 2Б и 2А.

Первая группа классифицирует многопользовательские АС, в которых одновременно обрабатывается и (или) хранится информация разных уровней конфиденциальности и не все пользователи имеют право доступа ко всей информации АС. Группа содержит пять классов - 1Д, 1Г, 1В, 1Б и 1А." В таблице 2 собраны требования ко всем девяти классам защищенности АС.

Таблица 2. Требования к защищенности автоматизированных систем.

Классы	Подсистемы и требования								
	3Б	3А	2Б	2А	1Д	1Г	1В	1Б	1
1. Подсистема управления доступом									
1.1. Идентификация, проверка подлинности и контроль доступа субъектов:									
в систему;	+	+	+	+	+	+	+	+	+
к терминалам, ЭВМ, узлам сети ЭВМ, каналам связи, внешним устройствам ЭВМ;	-	-	-	+	-	+	+	+	+
к программам;	-	-	-	+	-	+	+	+	+
к томам, каталогам, файлам, записям, полям записей.	-	-	-	+	-	+	+	+	+
1.2. Управление потоками информации.	-	-	-	+	-	-	+	+	+
2. Подсистема регистрации и учета									
2.1. Регистрация и учет:									
входа/выхода субъектов доступа в/из системы (узла сети);	+	+	+	+	+	+	+	+	+
выдачи печатных (графических) выходных документов;	-	+	-	+	-	+	+	+	+
запуска/завершения программ и процессов (заданий, задач);	-	-	-	+	-	+	+	+	+
доступа программ субъектов доступа к защищаемым файлам, включая из создания и удаления, передачу по линиям и каналам связи;	-	-	-	+	-	+	+	+	+

доступа программ субъектов доступа к терминалам, ЭВМ, узлам сети ЭВМ, каналам связи, внешним устройствам ЭВМ, программам, томам, каталогам, файлам, записям, полям записей;	-	-	-	+	-	+	+	+	+
изменения полномочий субъектов доступа;	-	-	-	-	-	-	+	+	+
создаваемых защищаемых объектов доступа.	-	-	-	+	-	-	+	+	+
2.2. Учет носителей информации.	+	+	+	+	+	+	+	+	+
2.3. Очистка (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ и внешних накопителей.	-	+	-	+	-	+	+	+	+
2.4. Сигнализация попыток нарушения защиты.	-	-	-	-	-	-	+	+	+
3. Криптографическая подсистема									
3.1. Шифрование конфиденциалной информации.	-	-	-	+	-	-	-	+	+
3.2. Шифрование информации, принадлежащей различным субъектам доступа (группам субъектов) на разных ключах.	-	-	-	-	-	-	-	-	+
3.3. Использование аттестованных (сертифицированных) криптографических средств.	-	-	-	+	-	-	-	+	+
4. Подсистема обеспечения целостности									
4.1. Обеспечения целостности программных средств и обрабатываемой информации.	+	+	+	+	+	+	+	+	+
4.2. Физическая охрана средств вычислительной техники и носителей информации.	+	+	+	+	+	+	+	+	+
4.3. Наличие администратора (службы) защиты информации в АС.	-	-	-	+	-	-	+	+	+

4.4. Периодическое тестирование СЗИ *НСД*.	+	+	+	+	+	+	+	+	+
4.5. Наличие средств восстановления СЗИ *НСД*.	+	+	+	+	+	+	+	+	+
4.6. Использование сертифицированных средств защиты.	-	+	-	+	-	-	+	+	+

Обозначения:

"- " - нет требований к данному классу;

"+" - есть требования к данному классу;

"СЗИ *НСД*" - система защиты информации от несанкционированного доступа.

Ниже приведено подробное изложение требований к достаточно представительному классу защищенности - 1В. По существу, перед нами - минимум требований, которым необходимо следовать, чтобы обеспечить конфиденциальность защищаемой информации (реальные АС часто не соответствуют данному классу).

ТРЕБОВАНИЯ К КЛАССУ ЗАЩИЩЕННОСТИ 1В

Подсистема управления доступом:

должна осуществляться идентификация и проверка подлинности субъектов доступа при входе в систему по идентификатору (коду) и паролю условно-постоянного действия длиной не менее шести буквенно-цифровых символов; должна осуществляться идентификация терминалов, ЭВМ, узлов сети ЭВМ, каналов связи, внешних устройств ЭВМ по логическим именам и/или адресам; должна осуществляться идентификация программ, томов, каталогов, файлов, записей, полей записей по именам; должен осуществляться контроль доступа субъектов к защищаемым ресурсам в соответствии с матрицей доступа; должно осуществляться управление потоками информации с помощью меток конфиденциальности. При этом уровень конфиденциальности накопителей должен быть не ниже уровня конфиденциальности записываемой на него информации.

Подсистема регистрации и учета:

должна осуществляться регистрация входа/выхода субъектов доступа в систему/из системы, либо регистрация загрузки и инициализации операционной системы и ее программного останова; должна осуществляться регистрация выдачи печатных (графических) документов на "твердую" копию; должна осуществляться регистрация запуска/завершения программ и процессов (заданий, задач), предназначенных для обработки защищаемых файлов; должна осуществляться регистрация попыток доступа программных средств к следующим дополнительным защищаемым объектам доступа: терминалам, ЭВМ, узлам сети ЭВМ, линиям (каналам) связи, внешним

устройствам ЭВМ, программам, томам, каталогам, файлам, записям, полям записей; должна осуществляться регистрация изменений полномочий субъектов доступа и статуса объектов доступа; должен осуществляться автоматический учет создаваемых защищаемых файлов с помощью их дополнительной маркировки, используемой в подсистеме управления доступом. Маркировка должна отражать уровень конфиденциальности объекта; должен проводиться учет всех защищаемых носителей информации с помощью их любой маркировки; должна осуществляться очистка (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ и внешних накопителей. Очистка осуществляется двухкратной произвольной записью в любую освобождаемую область памяти, использованную для хранения защищаемой информации; должна осуществляться сигнализация попыток нарушения защиты.

Подсистема обеспечения целостности:

должна быть обеспечена целостность программных средств СЗИ *НСД*, а также неизменность программной среды, при этом: целостность СЗИ *НСД* проверяется при загрузке системы по контрольным суммам компонент СЗИ, целостность программной среды обеспечивается использованием трансляторов с языков высокого уровня и отсутствием средств модификации объектного кода программ при обработке и (или) хранении защищаемой информации; должна осуществляться физическая охрана СВТ (устройств и носителей информации), предусматривающая постоянное наличие охраны территории и здания, где размещается АС, с помощью технических средств охраны и специального персонала, использование строгого пропускного режима, специальное оборудование помещений АС; должен быть предусмотрен администратор (служба) защиты информации, ответственный за ведение, нормальное функционирование и контроль работы СЗИ *НСД*. Администратор должен иметь свой терминами и необходимые средства оперативного контроля и воздействия на безопасность АС; должно проводиться периодическое тестирование всех функций СЗИ *НСД* с помощью специальных программных средств не реже одного раза в год; должны быть в наличии средства восстановления СЗИ *НСД*, предусматривающие ведение двух копий программных средств СЗИ НОД и их периодическое обновление и контроль работоспособности; должны использоваться сертифицированные средства защиты."

Пример

Категория информации, обрабатываемой в информационных системах аэропорта практически во всех случаях относится к служебной тайне. Такая информация принадлежит грифу конфиденциально и предназначена для служебного пользования.

Помимо важности обрабатываемой информации, важную роль в определение класса защищенности играют последствия от ее утечки или кражи. Кража такой информации может привести к финансовым потерям,

нарушения юридических прав клиентов данной организации и даже катастрофам.

АСУ аэропорта является многопользовательской, существует раздельный контроль доступа к различным уровням системы и имеет выход в интернет.

Исходя из определяющих признаков можно отнести рассматриваемую информационную систему к первой группе. Первая группа включает многопользовательские ИС, в которых одновременно обрабатывается и (или) хранится информация разных уровней конфиденциальности. Не все пользователи имеют право доступа ко всей информации АСУ.

Информационная система аэропорта имеет класс защищенности 1Г.

Требования к защите информации – это установленные правила или нормы, которые должна быть выполнены при организации и осуществлении защиты информации, или допустимое значение показателя эффективности защиты информации.

Формирование требований к защите информации, содержащейся в информационной системе, осуществляется обладателем информации (заказчиком).

Формирование требований к защите информации, содержащейся в информационной системе, осуществляется с учетом ГОСТ Р 51583 "Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения" (далее - ГОСТ Р 51583) и ГОСТ Р 51624 "Защита информации. Автоматизированные системы в защищенном исполнении. Общие требования" (далее - ГОСТ Р 51624) и в том числе включает:

– принятие решения о необходимости защиты информации, содержащейся в информационной системе;

– классификацию информационной системы по требованиям защиты информации (далее - классификация информационной системы);

– определение угроз безопасности информации, реализация которых может привести к нарушению безопасности информации в информационной системе, и разработку на их основе модели угроз безопасности информации;

– определение требований к системе защиты информации информационной системы.

Требования к системе защиты информации информационной системы определяются в зависимости от класса защищенности

информационной системы и угроз безопасности информации, включенных в модель угроз безопасности информации.

Требования к системе защиты информации информационной системы включаются в техническое задание на создание информационной системы и (или) техническое задание (частное техническое задание) на создание системы защиты информации информационной системы, разрабатываемые с учетом ГОСТ 34.602, ГОСТ Р 51583 и ГОСТ Р 51624, и должны в том числе содержать:

– цель и задачи обеспечения защиты информации в информационной системе;

– класс защищенности информационной системы;

– перечень нормативных правовых актов, методических документов и национальных стандартов, которым должна соответствовать информационная система;

– перечень объектов защиты информационной системы;

– требования к мерам и средствам защиты информации, применяемым в информационной системе;

– стадии (этапы работ) создания системы защиты информационной системы;

– требования к поставляемым техническим средствам, программному обеспечению, средствам защиты информации;

– функции заказчика и оператора по обеспечению защиты информации в информационной системе;

– требования к защите средств и систем, обеспечивающих функционирование информационной системы (обеспечивающей инфраструктуре);

– требования к защите информации при информационном взаимодействии с иными информационными системами и информационно-телекоммуникационными сетями, в том числе с информационными системами уполномоченного лица, а также при применении вычислительных ресурсов (мощностей), предоставляемых уполномоченным лицом для обработки информации.

При определении требований к системе защиты информации информационной системы учитываются положения политик обеспечения информационной безопасности обладателя информации (заказчика), а также политик обеспечения информационной безопасности оператора и уполномоченного лица

в части, не противоречащей политикам обладателя информации (заказчика).

В случае создания информационной системы, функционирование которой предполагается на базе информационно-телекоммуникационной инфраструктуры центра обработки данных, дополнительно определяются требования по защите информации, подлежащие реализации в информационно-телекоммуникационной инфраструктуре центра обработки данных.

Пример

Целью применения мер обеспечения безопасности информации является уменьшение риска либо за счет уменьшения вероятности осуществления угрозы, либо за счет уменьшения эффекта воздействия угрозы.

С экономической точки зрения мера защиты оправдана, если ее эффективность, выраженная через уменьшение ожидаемого экономического ущерба, превышает затраты на ее реализацию.

Рассмотрим концептуальную модель безопасности информации:

Опираясь на виды угроз нашего объекта и на представленную модель можно сформулировать основное требование к защите

информации: обеспечить сохранность, целостность, достоверность информации и исключить любую попытку несанкционированного доступа (с применением правового, организационного и инженерно-физического направлений защиты) с использованием всех способов защиты.

Итак, должны быть обеспечены требования:

1) Необходимо иметь явную и четкую методику обеспечения безопасности;

2) Обеспечение идентификации пользователей, имеющих доступ к информации;

3) Информация должна храниться отдельно и защищаться так, чтобы была возможность отслеживать действия, влияющие на безопасность;

4) Средства защиты информации должны включать механизмы контроля доступа ко всем видам информационных и программных ресурсов, которые в соответствии с принципом обоснованности доступа следует разделять между пользователями;

5) Потоки информации должны разграничиваться в зависимости от уровня ее конфиденциальности;

6) Должна быть предусмотрена очистка ресурсов, содержащих конфиденциальную информацию, до перераспределения этих ресурсов другим пользователям;

7) Предупреждение возможных неблагоприятных последствий нарушения порядка доступа к информации;

8) Проведение мероприятий, направленных на предотвращение неправомерных действий в отношении информации;

9) Своевременное обнаружение фактов неправомерных действий в отношении информации;

10) Недопущение воздействия на технические средства информационной системы общего пользования, в результате которого может быть нарушено их функционирование;

11) Возможность оперативного восстановления информации, модифицированной или уничтоженной вследствие неправомерных действий;

12) Проведение мероприятий по постоянному контролю за обеспечением защищенности;

13) Возможность записи и хранения сетевого трафика;

14) Система защиты информации должна точно выполнять свои функции в соответствии с основными принципами и быть изолированной от пользователей.

Под **несанкционированным доступом к информации** (НСД) согласно руководящим документам Гостехкомиссии будем понимать доступ к информации, нарушающий установленные правила разграничения доступа и осуществляемый с использованием штатных средств, предоставляемых СВТ или АС. НСД может носить случайный или намеренный характер.

Можно выделить несколько обобщенных категорий методов защиты от НСД, в частности:

- организационные;
- технологические;
- правовые.

К первой категории относятся меры и мероприятия, регламентируемые внутренними инструкциями организации, эксплуатирующей информационную систему. Пример такой защиты — присвоение грифов секретности документам и материалам, хранящимся в отдельном помещении, и контроль доступа к ним сотрудников. Вторую категорию составляют механизмы защиты, реализуемые на базе программно-аппаратных средств, например систем идентификации и аутентификации или охранной сигнализации. Последняя категория включает меры контроля за исполнением нормативных актов общегосударственного значения, механизмы разработки и совершенствования нормативной базы, регулирующей вопросы защиты информации. Реализуемые на практике методы, как правило, сочетают в себе элементы нескольких из перечисленных категорий. Так, управление доступом в помещения может представлять собой взаимосвязь организационных (выдача допусков и ключей) и технологических (установку замков и систем сигнализации) способов защиты.

Рассмотрим подробнее такие взаимосвязанные методы защиты от НСД, как идентификация, аутентификация и используемое при их реализации криптографическое преобразование информации.

Идентификация — это присвоение пользователям идентификаторов и проверка предъявляемых идентификаторов по списку присвоенных.

Аутентификация — это проверка принадлежности пользователю предъявленного им идентификатора. Часто аутентификацию также называют подтверждением или проверкой подлинности.

Под безопасностью (стойкостью) системы идентификации и аутентификации будем понимать степень обеспечиваемых ею гарантий того, что злоумышленник не способен пройти аутентификацию от имени другого пользователя, В этом смысле, чем выше стойкость системы аутентификации, тем сложнее злоумышленнику решить указанную задачу. Система идентификации и аутентификации является одним из ключевых элементов инфраструктуры защиты от НСД любой информационной системы.

Различают три группы методов аутентификации, основанных на

наличии у каждого пользователя:

– индивидуального объекта заданного типа;

– знаний некоторой известной только ему и проверяющей стороне информации;

– индивидуальных биометрических характеристик.

К первой группе относятся методы аутентификации, использующие удостоверения, пропуска, магнитные карты и другие носимые устройства, которые широко применяются для контроля доступа в помещения, а также входят в состав программно-аппаратных комплексов защиты от НСД к средствам вычислительной техники.

Во вторую группу входят методы аутентификации, использующие пароли. По экономическим причинам они включаются в качестве базовых средств защиты во многие программно-аппаратные комплексы защиты информации. Все современные операционные системы и многие приложения имеют встроенные механизмы парольной защиты.

Последнюю группу составляют методы аутентификации, основанные на применении оборудования для измерения и сравнения с эталоном заданных индивидуальных характеристик пользователя: тембра голоса, отпечатков пальцев, структуры радужной оболочки глаза и др. Такие средства позволяют с высокой точностью аутентифицировать обладателя конкретного биометрического признака, причем "подделать" биометрические параметры практически невозможно. Однако широкое распространение подобных технологий сдерживается высокой стоимостью необходимого оборудования.

Если в процедуре аутентификации участвуют только две стороны, устанавливающие подлинность друг друга, такая процедура называется непосредственной аутентификацией (direct password authentication). Если же в процессе аутентификации участвуют не только эти стороны, но и другие, вспомогательные, говорят об аутентификации с участием доверенной стороны (trusted third party authentication). При этом третью сторону называют сервером аутентификации (authentication server) или арбитром (arbitrator).

Наиболее распространенные методы аутентификации основаны на применении многоразовых или одноразовых паролей. Из-за своего широкого распространения и простоты реализации парольные схемы часто в первую очередь становятся мишенью атак злоумышленников. Эти методы включают следующие разновидности способов аутентификации:

– по хранимой копии пароля или его свёртке (plaintext-equivalent);

– по некоторому проверочному значению (verifier-based);

– без непосредственной передачи информации о пароле проверяющей стороне (zero- knowledge);

– с использованием пароля для получения криптографического ключа (cryptographic).

В первую разновидность способов входят системы аутентификации, предполагающие наличие у обеих сторон копии пароля или его свертки. Для организации таких систем требуется создать и поддерживать базу данных, содержащую пароли или сверки паролей всех пользователей. Их слабой

стороной является то, что получение злоумышленником этой базы данных позволяет ему проходить аутентификацию от имени любого пользователя.

Способы, составляющие вторую разновидность, обеспечивают более высокую степень безопасности парольной системы, так как проверочные значения, хотя они и зависят от паролей, не могут быть непосредственно использованы злоумышленником для аутентификации.

Наконец, аутентификация без предоставления проверяющей стороне какой бы то ни было информации о пароле обеспечивает наибольшую степень защиты. Этот способ гарантирует безопасность даже в том случае, если нарушена работа проверяющей стороны (например, в программу регистрации в системе внедрен "троянский конь").

Особым подходом в технологии проверки подлинности являются криптографические протоколы аутентификации. Такие протоколы описывают последовательность действий, которую должны совершить стороны для взаимной аутентификации, кроме того, эти действия, как правило, сочетаются с генерацией и распределением криптографических ключей для шифрования последующего информационного обмена. Корректность протоколов аутентификации вытекает из свойств задействованных в них математических и криптографических преобразований и может быть строго доказана.

Обычные парольные системы проще и дешевле для реализации, но менее безопасны, чем системы с криптографическими протоколами. Последние обеспечивают более надежную защиту и дополнительно решают задачу распределения ключей. Однако используемые в них технологии могут быть объектом законодательных ограничений.

Для более детального рассмотрения принципов построения парольных систем сформулируем несколько основных определений.

Идентификатор пользователя — некоторое уникальное количество информации, позволяющее различать индивидуальных пользователей парольной системы (проводить их идентификацию). Часто идентификатор также называют именем пользователя или именем учетной записи пользователя.

Пароль пользователя — некоторое секретное количество информации, известное только пользователю и парольной системе, которое может быть запомнено пользователем и предъявлено для прохождения процедуры аутентификации. Одноразовый пароль дает возможность пользователю однократно пройти аутентификацию. Многоразовый пароль может быть использован для проверки подлинности повторно.

Учетная запись пользователя — совокупность его идентификатора и его пароля.

База данных пользователей парольной системы содержит учетные записи всех пользователей данной парольной системы.

Под **парольной системой** будем понимать программно-аппаратный комплекс, реализующий системы идентификации и аутентификации пользователей АС на основе одноразовых или многоразовых паролей. Как

правило, такой комплекс функционирует совместно с подсистемами разграничения доступа и регистрации событий. В отдельных случаях парольная система может выполнять ряд дополнительных функций, в частности генерацию и распределение кратковременных (сеансовых) криптографических ключей.

Основными компонентами парольной системы являются:

– интерфейс пользователя;
– интерфейс администратора;
– модуль сопряжения с другими подсистемами безопасности;
– база данных учетных записей.

Парольная система представляет собой "передний край обороны" всей системы безопасности. Некоторые ее элементы (в частности, реализующие интерфейс пользователя) могут быть расположены в местах, открытых для доступа потенциальному злоумышленнику. Поэтому парольная система становится одним из первых объектов атаки при вторжении злоумышленника в защищенную систему. Ниже перечислены типы угроз безопасности парольных систем:

1. Разглашение параметров учетной записи через:
– подбор в интерактивном режиме;
– подсматривание;
– преднамеренную передачу пароля его владельцем другому лицу;
– захват базы данных парольной системы (если пароли не хранятся в базе в открытом виде, для их восстановления может потребоваться подбор или дешифрование);
– перехват переданной по сети информации о пароле;
– хранение пароля в доступном месте.
2. Вмешательство в функционирование компонентов парольной системы через:
– внедрение программных закладок;
– обнаружение и использование ошибок, допущенных на стадии разработки;
– выведение из строя парольной системы.

Некоторые из перечисленных типов угроз связаны с наличием так называемого человеческого фактора, проявляющегося в том, что пользователь может:

– выбрать пароль, который легко запомнить и также легко подобрать;
– записать пароль, который сложно запомнить, и положить запись в доступном месте;
– ввести пароль так, что его смогут увидеть посторонние:
– передать пароль другому лицу намеренно или под влиянием заблуждения.

В дополнение к выше сказанному необходимо отметить существование "парадокса человеческого фактора". Заключается он в том, что пользователь нередко стремится выступать скорее противником парольной системы, как, впрочем, и любой системы безопасности, функционирование которой влияет

на его рабочие условия, нежели союзником системы защиты, тем самым ослабляя ее. Защита от указанных угроз основывается на ряде перечисленных ниже организационно-технических мер и мероприятий.

Выбор паролей

В большинстве систем пользователи имеют возможность самостоятельно выбирать пароли или получают их от системных администраторов. При этом для уменьшения деструктивного влияния описанного выше человеческого фактора необходимо реализовать ряд требований к выбору и использованию паролей.

Таблица 1

Требование к выбору пароля	**Получаемый эффект**
Установление минимальной длины пароля	Усложняет задачу злоумышленника при попытке подсмотреть пароль или подобрать пароль методом «тотального опробования»
Использование в пароле различных групп символов	Усложняет задачу злоумышленника при попытке подобрать пароль методом «тотального опробования»
Проверка и отбраковка пароля по словарю	Усложняет задачу злоумышленника при попытке подобрать пароль по словарю
Установление максимального срока действия пароля	Усложняет задачу злоумышленника при попытке подобрать пароль методом «тотального опробования», в том числе без непосредственного обращения к системе защиты (режим off-line)
Установление минимального срока действия пароля	Препятствует попыткам пользователя заменить пароль на старый после его смены по предыдущему требованию
Ведение журнала истории паролей	Обеспечивает дополнительную степень защиты по предыдущему требованию
Применение эвристического алгоритма, бракующего пароли на основании данных журнала истории	Усложняет задачу злоумышленника при попытке подобрать пароль по словарю или с использованием эвристического алгоритма
Ограничение числа попыток ввода пароля	Препятствует интерактивному подбору паролей злоумышленником
Поддержка режима принудительной смены пароля пользователя	Обеспечивает эффективность требования, ограничивающего максимальный срок действия пароля
Использование задержки при вводе неправильного пароля	Препятствует интерактивному подбору паролей злоумышленником

Запрет на выбор пароля самими пользователями и автоматическая генерация паролей	Исключает возможность подобрать пароль по словарю. Если алгоритм генерации паролей не известен злоумышленнику, последний может подбирать пароли только методом «тотального опробования»
Принудительная смена пароля при первой регистрации пользователя в системе	Защищает от неправомерных действия системного администратора, имеющего доступ к паролю в момент создания учетной записи

2. Примеры.

Например 1.

Задание определить время перебора всех паролей, состоящих из 6 цифр.

Алфавит составляют цифры n=10. Длина пароля 6 символов k=6.

Таким образом, получаем количество вариантов: $C=n^k=10^6$

Примем скорость перебора s=10 паролей в секунду. Получаем время перебора всех паролей $t= C/s=10^5$секунд≈1667минут≈28часов≈1,2 дня.

Примем, что после каждого из m=3 неправильно введенных паролей идет пауза в v=5 секунд. Получаем время перебора всех паролей

T=t*5/3=16667секунд≈2778минут≈46часов≈1,9 дня.

$T_{итог}$ = t+T = 1,2 + 1,9 = 3,1 дня

Пример 2.

Определить минимальную длину пароля, алфавит которого состоит из 10 символов, время перебора которого было не меньше 10 лет.

Алфавит составляют символы n=10.

Длина пароля рассчитывается: $k=\log_n C= \lg C$.

Определим количество вариантов C= t * s=10лет*10 паролей в сек. = 10*10*365*24*60*60≈3,15* 10^9 вариантов

Таким образом, получаем длину пароля: $k=\lg (3,15*10^9) = 9,5$ Очевидно, что длина пароля должна быть не менее 10 символов.

Архиваторы – это программы для создания архивов. Архивы предназначены для хранения данных в удобном компактном виде. В качестве данных обычно выступают файлы и папки. Как правило, данные предварительно подвергаются процедуре сжатия или упаковки. Поэтому почти каждый архиватор одновременно является программой для сжатия данных. С другой стороны, любая программа для сжатия данных может рассматриваться как архиватор. Эффективность сжатия является важнейшей характеристикой архиваторов. От нее зависит размер создаваемых архивов. Чем меньше архив, тем меньше места требуется для его хранения. Для передачи нужна меньшая пропускная способность канала передачи или затрачивается меньшее время. Преимущества архивов очевидны, если учесть, что данные уменьшаются в размере и в 2 раза, и в 5 раз.

Сжатие данных используется очень широко. Можно сказать, почти везде. Например, документы PDF , как правило, содержат сжатую информацию. Довольно много исполняемых файлов EXE сжаты специальными упаковщиками. Всевозможные мультимедийные файлы (GIF , JPG , MP 3, MPG) являются своеобразными архивами.

Основным недостатком архивов является невозможность прямого доступа к данным. Их сначала необходимо извлечь из архива или распаковать. Операция распаковки, впрочем, как и упаковки, требует некоторых системных ресурсов. Это не мгновенная операция. Поэтому архивы в основном применяют со сравнительно редко используемыми данными. Например, для хранения резервных копий или установочных файлов.

В данный момент существует много архиваторов. Они имеют разную распространенность и эффективность. Некоторые интересные архиваторы не известны широкому кругу потенциальных пользователей. Особый интерес представляют оценка и сравнение эффективности сжатия популярных архиваторов.

Методы сжатия архиваторов.

Разработано большое количество разнообразных методов, их модификаций и подвидов для сжатия данных. Современные архиваторы, как правило, одновременно используют несколько методов одновременно. Можно выделить некоторые основные.

Кодирование длин серий (RLE - сокращение от run - length encoding - кодирование длин серий).

Очень простой метод. Последовательная серия одинаковых элементов данных заменяется на два символа: элемент и число его повторений. Широко используется как дополнительный, так и промежуточный метод. В качестве самостоятельного метода применяется, например, в графическом формате BMP .

Словарный метод (LZ - сокращение от Lempel Ziv - имена авторов).

Наиболее распространенный метод. Используется словарь, состоящий из последовательностей данных или слов. При сжатии эти слова заменяются на их коды из словаря. В наиболее распространенном варианте реализации в качестве словаря выступает сам исходный блок данных.

Основным параметром словарного метода является размер словаря. Чем больше словарь, тем больше эффективность. Однако для неоднородных данных чрезмерно большой размер может быть вреден, так как при резком изменении типа данных словарь будет заполнен неактуальными словами. Для эффективной работы данного метода при сжатии требуется дополнительная память. Приблизительно на порядок больше, чем нужно для исходных данных словаря. Существенным преимуществом словарного метода является простая и быстрая процедура распаковки. Дополнительная память при этом не требуется. Такая особенность особенно важна, если необходим оперативный доступ к данным.

Энтропийный метод (Huffman - кодирование Хаффмена, Arithmetic

coding - арифметическое кодирование)

В этом методе элементы данных, которые встречаются чаще, кодируются при сжатии более коротким кодом, а более редкие элементы данных кодируются более длинным кодом. За счет того, что коротких кодов значительно больше, общий размер получается меньше исходного.

Широко используется как дополнительный метод. В качестве самостоятельного метода применяется, например, в графическом формате JPG .

Метод контекстного моделирования (CM - сокращение от context modeling - контекстное моделирование)

В этом методе строится модель исходных данных. При сжатии очередного элемента данных эта модель выдает свое предсказание или вероятность. Согласно этой вероятности, элемент данных кодируется энтропийным методом. Чем точнее модель будет соответствовать исходным данным, тем точнее она будет выдавать предсказания, и тем короче будут кодироваться элементы данных.

Для построения эффективной модели требуется много памяти. При распаковке приходится строить точно такую же модель. Поэтому скорость и требования к объему оперативной памяти для упаковки и распаковки почти одинаковы. В данный момент методы контекстного моделирования позволяют получить наилучшую степень сжатия, но отличаются чрезвычайно низкой скоростью.

PPM (PPM - Prediction by Partial Matching - предсказание по частичному совпадению).

Это особый подвид контекстного моделирования. Предсказание выполняется на основании определенного количества предыдущих элементов данных. Основным параметром является порядок модели, который задает это количество элементов. Чем больше порядок модели, тем выше степень сжатия, но требуется больше оперативной памяти для хранения данных модели. Если оперативной памяти недостаточно, то такая модель с большим порядком показывает низкие результаты. Метод PPM особенно эффективен для сжатия текстовых данных.

Предварительные преобразования или фильтрация.

Данные методы служат не для сжатия, а для представления информации в удобном для дальнейшего сжатия виде. Например, для несжатых мультимедиа данных характерны плавные изменения уровня сигнала. Поэтому для них применяют дельта-преобразование, когда вместо абсолютного значения берется относительное. Существуют фильтры для текста, исполняемых файлов, баз данных и другие.

Метод сортировки блока данных (BWT - сокращение от Burrows Wheeler Transform - по имени авторов).

Это особый вид или группа преобразований, в основе которых лежит сортировка. Такому преобразованию можно подвергать почти любые данные. Сортировка производится над блоками, поэтому данные предварительно разбиваются на части. Основным параметром является размер блока, который

подвергается сортировке. Для распаковки данных необходимо проделать почти те же действия, что и при упаковке. Поэтому скорость и требования к оперативной памяти почти одинаковы. Архиваторы, которые используют данный метод, обычно показывают высокую скорость и степень сжатия для текстовых данных.

Непрерывные блоки или непрерывный режим (Solid mode - непрерывный режим).

Во многих методах сжатия начальный участок данных или файла кодируется плохо. Например, в словарном методе словарь пуст. В методе контекстного моделирования модель не построена. Когда количество файлов большое, а их размер маленький, общая степень сжатия значительно ухудшается за счет этих начальных участков. Чтобы этого не происходило при переходе на следующий файл, используется информация, полученная исходя из предыдущих файлов. Аналогичного эффекта можно добиться простым представлением исходных файлов в виде одного непрерывного файла.

Этот метод используется во многих архиваторах и имеет существенный недостаток. Для распаковки произвольного файла необходимо распаковать и файлы, которые оказались в начале архива. Это необходимо для правильного заполнения словаря или построения модели. Существует и промежуточный вариант, когда используются непрерывные блоки фиксированного размера. Потери сжатия получаются минимальными, но для извлечения одного файла, который находится в конце большого архива, необходимо распаковать только один непрерывный блок, а не весь архив.

Сегментирование.

Во всех методах сжатия при изменении типа данных собственно сам переход кодируется очень плохо. Словарь становится не актуальным, модель настроена на другие данные. В этих случаях применяется сегментирование. Это предварительная разбивка на однородные части. Затем эти части кодируются по отдельности или группами.

Особо хочется подчеркнуть, что существует большое количество методов сжатия. Каждый метод обычно ориентирован на один вид или группу реальных данных. Хорошие результаты показывает комплексное использование методов.

Особенности данных

Степень сжатия в основном зависит от исходных данных. Хорошо сжимаются почти все предварительно несжатые данные, например, исполняемые файлы (EXE), тексты (TXT , DOC), базы данных (DBF), простые несжатые изображения (BMP). Ограниченно сжимаются несжатый звук (WAV), сложные несжатые изображения (BMP

). Не сжимаются почти все уже сжатые данные, например, архивы (ZIP , CAB), сжатые документы (PDF), сжатая графика и видео (JPG , GIF , AVI , MPG), сжатый звук (MP 3). Их сжатие находится в пределах пары процентов за счет служебных блоков и небольшой избыточности.

Для сжатия некоторых специфических данных (текст, несжатые

изображения, несжатый звук) существуют специальные методы и архиваторы. Такие архиваторы обеспечивают высокую степень сжатия и высокую скорость. Однако так называемые универсальные архиваторы постепенно дополняются подобными методами. В данный момент только для несжатого звука существуют высокоэффективные специальные архиваторы, такие, как OptimFROG , Monkey Audio . Для текстов и изображений лучшие универсальные архиваторы показывают лучшую степень сжатия. Например, архив изображений получится меньше, если использовать формат BMP и архиватор WinRK вместо специализированных графических форматов, таких как JPEG 2000 (LossLess - сжатие без потерь).

Большое количество типов данных уже являются сжатыми. Использование архиваторов дает мизерное уменьшение размера. Тем не менее даже в таких случаях эффективное сжатие теоретически возможно. Это обусловлено тем, что в большинстве распространенных форматов файлов, использующих сжатие, применены не самые эффективные методы. Например, в основе формата JPG лежит энтропийное сжатие, которое используется после преобразований Фурье. Данные кодируются неоптимальными блоками, что обусловлено желанием сделать формат JPG устойчивым к повреждениям и возможности частичного извлечения информации. Перекодировав файлы JPG при помощи высокоэффективных методов, можно добиться сжатия порядка 75% от исходного файла (архиватор StuffIt). Собственно сам исходный файл JPG сжимается обычными архиваторами только до 96%. Однако подобные манипуляции с файлами JPG стали возможны только недавно и еще не получили распространения. В большинстве случаев сжимать уже сжатые данные бесполезно.

Следует различать собственно программу-архиватор, формат архивов и методы сжатия. Даже один и тот же метод сжатия может иметь варианты реализации. Например, существует более десятка программ-архиваторов, которые могут создавать архивы в формате ZIP. В свою очередь данные в формате ZIP могут быть сжаты различными методами: Deflate, Deflate64, BZip2. Метод Deflate имеет несколько реализаций с разной скоростью и степенью сжатия (разница порядка 5%). С помощью этого метода архиватор 7-zip позволяет создавать архивы в формате ZIP и 7Z.

Обычно архиваторы могут создавать архивы в собственном эксклюзивном формате с использованием своих оригинальных методов. Например, архиватор RAR позволяет создавать архивы RAR. В формате архива и методах сжатия заключаются основные преимущества того или иного архиватора.

В простейшем случае архиватор позволяет только упаковать или распаковать один файл. Кроме собственно сжатия данных, современные архиваторы обеспечивают некоторые дополнительные функции. Можно выделить несколько основных:

– сжатие некоторых файлов и целых директорий;

– создание самораспаковывающихся (SFX) архивов. То есть для распаковки архива программа-архиватор не требуется;

– изменение содержимого архива; шифрование содержимого архива;

– информация для восстановления архива при частичном повреждении и возможность восстановления поврежденных архивов;

– разбивка архива на несколько частей или томов;

– консольная версия программы для работы из командной строки; графическая (GUI) версия программы.

Стоит отметить, что, несмотря на формальное наличие, реализация каждой дополнительной функции может быть выполнена на совершенно разном уровне.

Кроме различий в функциональности, можно разбить архиваторы на две группы: асимметричные и симметричные. Асимметричные архиваторы требуют для операции распаковки значительно меньше времени и оперативной памяти, чем для операции упаковки. Это позволяет быстро получать содержимое архива на маломощных компьютерах. Симметричные архиваторы требуют для операций упаковки и распаковки одинаковое время и объем оперативной памяти. Использование таких архиваторов на широком парке компьютеров или для оперативного доступа к содержимому архива ограничено. Известный архиватор RAR в качестве основного использует асимметричный словарный метод сжатия, а для текстов может использовать симметричный PPM-метод. Таким образом, распаковка архивов RAR, сжатых с максимальной степенью сжатия, может быть невозможна на компьютерах с ограниченным объемом оперативной памяти. Все или почти все передовые архиваторы с высокой степенью сжатия являются симметричными.

Точной статистики по распространенности архиваторов у меня нет. Я выскажу свою субъективную точку зрения на основе личного опыта. Безусловно, самым распространенным архиватором являются ZIP и его модификации. По своей распространенности он значительно превосходит ближайших конкурентов. Следом идут RAR и ACE. В последние годы встречается архиватор 7-zip. Других архиваторов и архивов лично мы не встречали. Исключение составляют некогда популярные ARJ и LHA. В данный момент они не актуальны из-за очень низкой степени сжатия.

Несмотря на очень скромные данные о распространенности архиваторов, их существует большое множество. Основная масса относится к категории экспериментальных и архиваторов с ограниченной функциональностью. Тем не менее, каждый их них позволяет выполнять собственно процедуру сжатия данных. Меньшая распространенность увеличивает вероятность ошибок в программе.

Российская Федерация имеет свой стандарт шифрования. Этот стандарт закреплен ГОСТом №28147-89, принятом, как явствует из его обозначения, еще в 1989 году в СССР. Алгоритм ГОСТ может работать в нескольких режимах:

- режим простой замены;
- режим гаммирования;
- режим гаммирования с обратной связью;
- режим выработки имитовставки.

В данной реализации алгоритм ГОСТ работает в режиме простой замены. Весь алгоритм опирается на два базовых цикла:

- цикл зашифрования (CRYPT);
- цикл расшифрования (DECRYPT);

В свою очередь, каждый из базовых циклов представляет собой многократное повторение одной единственной процедуры, называемой основным шагом криптопреобразования.

Таким образом, чтобы разобраться в ГОСТе, надо понять следующее:

а) что такое основной шаг криптопреобразования;

б) как из основных шагов складываются базовые циклы;

в) как из базовых циклов складываются практические алгоритмы.

Прежде чем перейти к изучению этих вопросов, следует поговорить о ключевой информации, используемой алгоритмами ГОСТа. В соответствии с принципом Кирхгофа, которому удовлетворяют все современные известные широкой общественности шифры, именно ее секретность обеспечивает секретность зашифрованного сообщения. В ГОСТе ключевая информация состоит из двух структур данных. Помимо собственно ключа, необходимого для всех шифров, она содержит еще и таблицу замен. Ниже приведены основные характеристики ключевых структур ГОСТа [3, с.378]:

1. Ключ является массивом из восьми 32-битовых элементов кода, далее в настоящей работе он обозначается символом К: $К= \{K_i\}$, $0<i<7$. Таким образом, размер ключа составляет $32 \cdot 8 = 256$ бит или 32 байта.

2. Таблица замен может быть представлена в виде матрицы размера 8x16, содержащей 4-битовые элементы, которые можно представить в виде целых чисел от 0 до 15. Строки таблицы замен называются узлами замен, они должны содержать различные значения, то есть каждый узел замен должен содержать 16 различных чисел от 0 до 15 в произвольном порядке. Таким образом, общий объем таблицы замен равен: 8 узлов x 16 элементов/узел x 4 бита/элемент = 512 бит или 64 байта.

Таблица 5.1

Пример S-блоков алгоритма ГОСТ, используемых в режиме обучения.

S-блок 1:	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
S-блок 2:	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
S-блок 3:	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
S-блок 4:	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
S-блок 5:	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
S-блок 6:	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
S-блок 7:	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
S-блок 8:	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

Основной шаг криптопреобразования по своей сути является оператором, определяющим преобразование 64-битового блока данных. Дополнительным параметром этого оператора является 32-битовый блок, в качестве которого используется какой-либо элемент ключа. Схема алгоритма основного шага приведена на рис. 5.6.

Блок-схема основного шага алгоритма ГОСТ 28147-89.

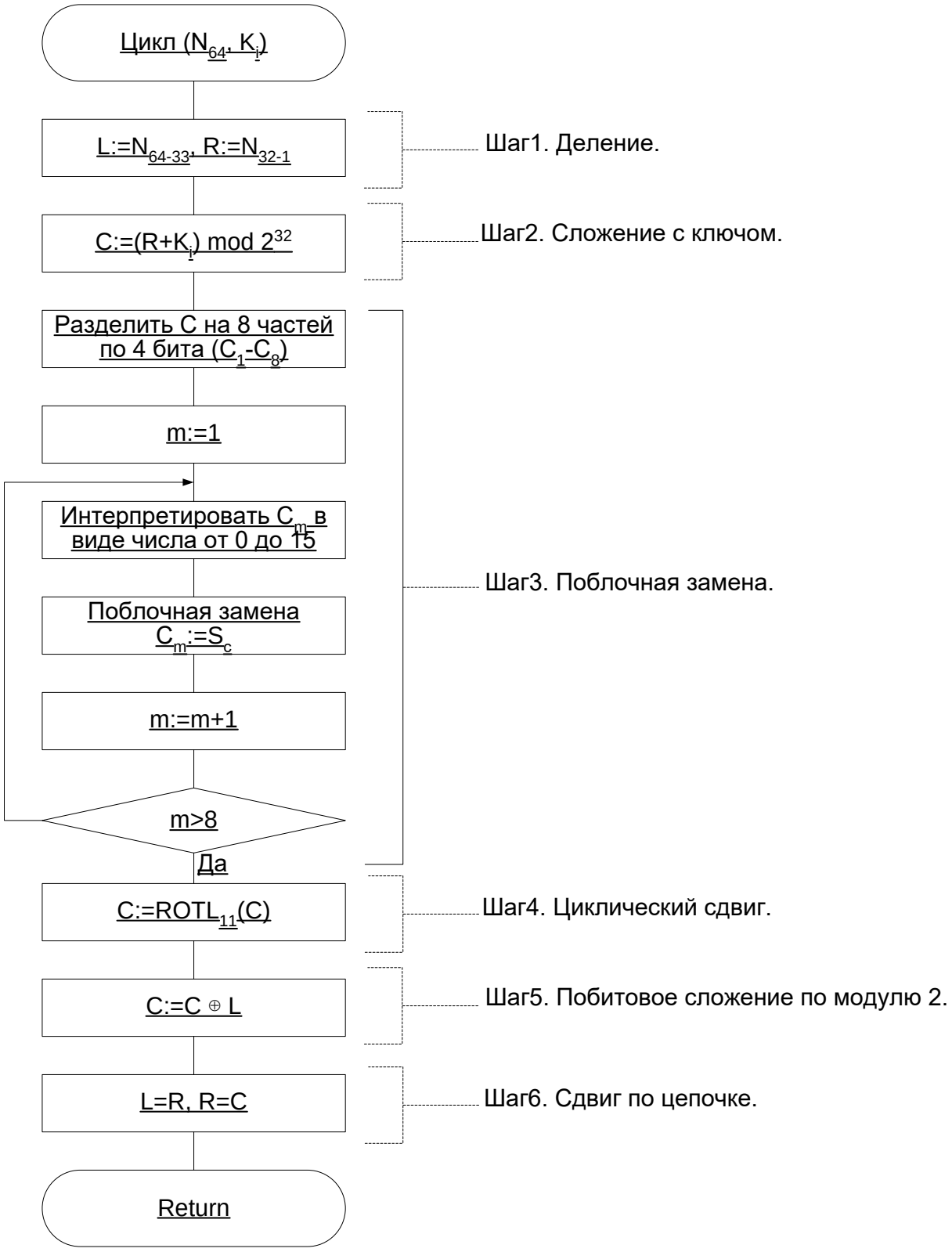


Рис. 5.6

Ниже даны пояснения к рис.5.6 :

- N_{64} – преобразуемый 64-битовый блок данных, в ходе выполнения шага его младшая R и старшая L части обрабатываются как отдельные 32-битовые целые числа без знака.

– X=K_i (где 1<i<32 – номер цикла) – 32-битовый элемент ключа;

Полученные после шестого шага значения старшей (L) и младшей (R) частей преобразуемого блока, каждая из которых размером в 32 бита , сливаются в единое целое и возвращаются как 54-битовый результат выполнения алгоритма основного шага криптопреобразования.

На основном шаге криптопреобразования используются достаточно простые математические преобразования, требующие однако минимум 32 разрядного процессора. Для зашифровки одного блока данных цикл на Рис. 5.4. повторяют 32 раза. Таким образом 256 бит ключа делят на восемь 32 битных подключей ($\boldsymbol{K_0 \ldots K_7}$), которые используются в соответствии с таблицей 5.2. Такой режим работы называется режимом простой замены. Процесс расшифрования аналогичен процессу зашифрования, только ключи используются в обратной последовательности. Поэтому данный алгоритм и является симметричным.

Таблица 5.2

Использование подключей в различных раундах алгоритма ГОСТ.

Раунд	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Подключ	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8

Раунд	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Подключ	1	2	3	4	5	6	7	8	8	7	6	5	4	3	2	1

Основой для лабораторной работы №2 является алгоритм шифрования с открытым ключом Эль Гамаля. В настоящий момент наиболее распространенными системами шифрования с открытым ключом являются системы RSA и Эль Гамаль. Их применение требует больших вычислительных мощностей, а практическая реализация требует решения таких проблем, как

быстрое выполнение вычислений с числами, состоящими из сотен десятичных разрядов.

Алгоритм Эль Гамаля был разработан в 1985 году и основывается на возведении в степень по модулю большого простого числа. Данная система является альтернативой RSA и при равном значении ключа обеспечивает ту же криптостойкость. Это связано с тем что задача разложения на сомножители больших целых чисел и задача вычисления дискретного логарифма сопоставимы между собой по трудоемкости решения.

В отличие от RSA метод Эль-Гамаля основан на проблеме дискретного логарифма, т.е если возводить число в степень в конечном поле достаточно легко, то восстановить аргумент по значению (то есть найти логарифм) довольно трудно.

Основу системы составляют параметры P и W - числа, первое из которых - очень большое простое целое число (несколько сотен десятичных разрядов), а второе - целое. Т.к. программная реализация алгоритмов, которые позволяли бы работать с большими числами, очень трудна, то в лабораторной работе я использую числа до 10^{18} чего вполне достаточно для применения ее в учебных целях.

Число *W* находится из условий:

$$\begin{cases} 0 < W < P-1 \\ W^{P-1} = 1 \bmod P \end{cases} \quad (5.1)$$

Затем генерируется секретный ключ Xa из диапазона $1 \le Xa \le P-1$.

Затем вычисляется открытый ключ *Ya* как степень:

$$Ya = W^{Xa} \bmod P. \quad (5.2)$$

Для того чтобы зашифровать сообщение M выбирается дополнительное число *K* удовлетворяющее условию $1 \le K \le P-1$. Сообщение M – это ничто иное, как 1 символ текста, или, точнее говоря, ASCII-код символа (от 0 до 255).

Затем, выбрав число K, мы вычисляем число R по формуле :

$$R = Ya^K \bmod P. \quad (5.3)$$

Криптограмма, или закодированное сообщение, формируется из двух элементов. Для ее формирования используется операция побитового сложения по модулю 2:

$$\begin{cases} C1 = W^K \bmod P, \\ C2 = M \oplus R. \end{cases} \quad (5.4)$$

Для восстановления по криптограмме C исходного сообщения сначала используя C1 находят *R*, для этого возводят C1 в степень Xa по модулю P:

$$R = C_1^{Xa} \bmod P = W^{KXa} \bmod P = (W^{Xa})^K \bmod P = Ya^K \bmod P. \quad (5.5)$$

Если известно R то дешифрование исходного текста не представляет какой-либо трудности. Дешифрование производится по формуле:

$$M = C2 \oplus R. \quad (5.6)$$

Таким образом в лабораторной работе происходит циклическое считывание из исходного файла по 1 символу, который затем интерпретируется как ASCII-код и помещается в переменную M. Затем происходит сохранение результата сложения по M xor R в выходном файле, что и представляет собой криптограмму C2. В начало же выходного файла (самым первым его байтом) записывается полученное значение элемента криптограммы C1, поэтому размер закодированного файла больше исходного на 1 байт. При декодировании из закодированного файла сначала считывается первый байт C1, который будет необходим для вычисления R, а затем также циклически происходит побайтовое считывание закодированного файла.

Первые алгоритмы с открытым ключом стали известны в то же время, когда проходило DES обсуждение как стандарта шифрования (70-е годы).

В реальном мире алгоритмы с открытыми ключами не заменяют симметричные алгоритмы и используются не для шифрования сообщений, а для шифрования ключей по следующим двум причинам:

Алгоритмы с открытыми ключами работают медленно. Симметричные алгоритмы, по крайней мере, в 1000 раз быстрее, чем алгоритмы с открытыми ключами. Да, компьютеры становятся все быстрее и быстрее, и лет через 15 криптография с открытыми ключами достигнет скоростей, сравнимых с сегодняшней скоростью симметричной криптографии. Но требования к объему передаваемой информации также возрастают, и всегда будет требоваться шифровать данные быстрее, чем это сможет сделать криптография с открытыми ключами.

Криптосистемы с открытыми ключами уязвимы по отношению к вскрытию с выбранным открытым текстом. Если закодированное сообщение $C = E(P)$, где P - открытый текст из n возможных открытых текстов, то криптоаналитику нужно только зашифровать все n возможных открытых текстов и сравнить результаты с C (помните, ключ шифрования общедоступен). Он не сможет раскрыть ключ дешифрирования, но он сможет определить P.

К тому же, ни одна из реализаций систем с открытым ключом, предложенных до сих пор не может конкурировать в скорости с системами с секретным ключом, такими, например, как DES или ГОСТ. Когда необходимо передать большое количество информации, может оказаться, что использование криптоалгоритмов с открытым ключом было бы слишком медленным, тогда как использование симметричных алгоритмов было бы либо невозможным (из-за отсутствия разделенного секретного ключа), либо не отвечающим требованиям секретности.

В такой ситуации может быть полезным использование компромиса. Гибридная (смешанная) криптосистема использует криптосистему с открытым ключом один раз в начале передач сообщений для того, чтобы выделить небольшую часть информации, которая затем используется как ключ

к шифратору или дешифратору для тех текущих сообщений, которые проходят через криптосистему с секретным ключом. Без особого замедления протокола это значительно повышает стойкость гибридной системы по двум причинам: криптосистему с секретным ключом раскрыть легче (при атаке только на основе шифртекста, которая является единственным типом атаки, имеющим смысл в этой ситуации), если доступен большой шифртекст, но даже если криптоаналитику и удастся определить секретный ключ, он сможет расшифровать лишь сообщение, закодированное данным ключом.

В большинстве реализаций криптография с открытыми ключами используется для засекречивания и распространения сеансовых ключей, которые используются симметричными алгоритмами для закрытия потока сообщений. Таким образом, использование криптографии с открытыми ключами для распределения ключей решает очень важную проблему тайного распространения ключей. Это значительно уменьшает риск компрометации сеансового ключа.

В лабораторной работе должны участвовать как минимум 2 компьютера, поэтому перед началом работы необходимо разделиться на группы по 2 человека, которые будут работать на разных компьютерах. Каждый пользователь должен перед началом выполнения работы узнать IP-адрес своей машины (это можно сделать в настройках контрольной панели: локальные соединения, служба TCP/IP). Это необходимо, т.к. связь между компьютерами будет осуществляться по IP-адресу. Затем необходимо уяснить, чье приложение будет серверным (принимать ключ) и чье клиентским (посылать ключ). В принципе это не имеет значения, т.к. в процессе работы участники будут меняться ролями. В начале работы по умолчанию оба приложения играют роль серверных, т.е. они ожидают приема (это отображается в строке состояния), но как только одно из приложений сделает запрос и установит соединение с другим компьютером, то оно сразу же становится клиентским.

Основой для лабораторной работы является алгоритм шифрования с открытым ключом RSA.

RSA (буквенная аббревиатура от фамилий Rivest, Shamir и Adleman) - криптографический алгоритм с открытым ключом. RSA стал первым алгоритмом такого типа, пригодным и для шифрования, и для цифровой подписи. Алгоритм используется в большом числе криптографических приложений.

В криптографической системе с открытым ключом каждый участник располагает как открытым ключом (public key), так и секретным ключом (secret key). Каждый ключ - это часть информации. В криптографической системе RSA каждый ключ состоит из пары целых чисел. Каждый участник создаёт свой открытый и секретный ключ самостоятельно. Секретный ключ каждый из них держит в секрете, а открытые ключи можно сообщать кому угодно или даже публиковать их. Открытый и секретный ключи каждого участника обмена сообщениями образуют «согласованную пару» в том смысле, что они являются взаимно обратными.

Алгоритм RSA состоит из следующих пунктов:

1. Выбрать простые числа p и q заданного размера (например, 512 битов каждое).

В криптографии под случайным простым числом понимается простое число, содержащее в двоичной записи заданное количество битов k, на алгоритм генерации которого накладываются определенные ограничения.

2. Вычислить n = p * q
3. Вычисляется значение функции Эйлера от числа n:

$m = (p - 1) * (q - 1)$

4. Выбрать число d взаимно простое с m

Два целых числа называются взаимно простыми, если они не имеют никаких общих делителей, кроме ±1. Примеры: 14 и 25 взаимно просты, а 15 и 25 не взаимно просты (у них имеется общий делитель 5).

5. Выбрать число e так, чтобы e * d = 1 (mod m)

Числа e и d являются ключами. Шифруемые данные необходимо разбить на блоки - числа от 0 до n - 1.

Шифрование и дешифровка данных производятся следующим образом:

Шифрование: $P(M) = M^e \bmod n$

Дешифровка: $S(C) = C^d \bmod n$

Следует также отметить, что ключи e и d равноправны, т.е. сообщение можно шифровать как ключом e, так и ключом d, при этом расшифровка должна быть произведена с помощью другого ключа.

Схема шифрования и дешифрования RSA представлена на рис.4. Предположим, сторона B хочет послать стороне A сообщение M. Сообщением являются целые числа от 0 до n-1.

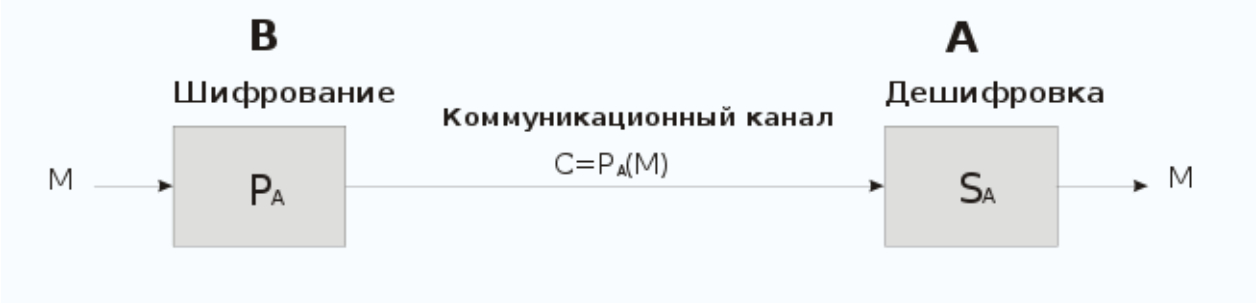


Рис.4 – Схема шифрования/дешифрования RSA

Размер ключа в алгоритме RSA связан с размером модуля n. Два числа p и q, произведением которых является модуль, должны иметь приблизительно одинаковую длину, поскольку в этом случае найти сомножители (факторы) сложнее, чем в случае, когда длина чисел значительно различается. Например, если предполагается использовать 768-

битный модуль, то каждое число должно иметь длину приблизительно 384 бита. Обратите внимание, что если два числа чрезвычайно близки друг к другу или их разность близка к некоторому предопределенному значению, то возникает потенциальная угроза безопасности, однако такая вероятность – близость двух случайно выбранных чисел – незначительна.

Оптимальный размер модуля определяется требованиями безопасности: модуль большего размера обеспечивает большую безопасность, но и замедляет работу алгоритма RSA. Длина модуля выбирается в первую очередь на основе значимости защищаемых данных и необходимой стойкости защищенных данных и во вторую очередь – на основе оценки возможных угроз.

На 2009 год система шифрования на основе RSA считается надёжной, начиная с размера n = p * q в 1024 бита. В лабораторной работе предлагается выбрать размер чисел p и q, и сгенерировать ключ размерами 64 (при выборе размерности простых чисел из выпадающего списка – 32 бита), 128, 256, 512, 1024 бита.

Более подробную информацию о данном алгоритме шифрования и конкретные примеры можно получить из справки, а также из источников, указанных в литературе.

Подсистемы идентификации и аутентификации пользователя играют важную роль в системах защиты информации.

Стойкость подсистемы идентификации и аутентификации пользователя в системе защиты информации (СЗИ) во многом определяет устойчивость к взлому самой СЗИ. Данная стойкость определяется гарантией того, что злоумышленник не сможет пройти аутентификацию, присвоив чужой идентификатор или украв его.

Парольные системы идентификации/аутентификации являются одними из основных и наиболее распространенных в СЗИ методами пользовательской аутентификации. В данном случае информацией, аутентифицирующей пользователя, является некоторый секретный пароль, известный только легальному пользователю.

Парольная аутентификация пользователя, как правило, передний край обороны СЗИ. В связи с этим модуль аутентификации по паролю наиболее часто подвергается атакам со

стороны злоумышленника. Цель последнего в данном случае – подобрать аутентифицирующую информацию (пароль) легального пользователя.

Методы парольной аутентификации пользователя наиболее просты и при несоблюдении определенных требований к выбору пароля являются достаточно уязвимыми.

Основными минимальными требованиями к выбору пароля и к подсистеме парольной аутентификации пользователя являются следующие.

К паролю:

1) минимальная длина пароля должна быть не менее 6 символов;

2) пароль должен состоять из различных групп символов (малые и большие латинские буквы, цифры, специальные символы ‘(’, ‘)’, ‘#’ и т.д.);

3) в качестве пароля не должны использоваться реальные слова, имена, фамилии и т.д.

К подсистеме парольной аутентификации:

1) администратор СЗИ должен устанавливать максимальный срок действия пароля, после чего, пароль следует сменить;

2) в подсистеме парольной аутентификации необходимо установить ограничение числа попыток ввода пароля (как правило, не более трёх);

3) в подсистеме парольной аутентификации требуется установить временную задержку в случае ввода неправильного пароля.

Как правило, для генерирования паролей в СЗИ, удовлетворяющих перечисленным требованиям к паролям, используются программы – автоматические генераторы паролей пользователей.

При выполнении перечисленных требований к паролям и к подсистеме парольной аутентификации единственно возможным методом взлома данной подсистемы злоумышленником является прямой перебор паролей (brute forcing). В данном случае, оценка стойкости парольной защиты осуществляется следующим образом.

Количественная оценка стойкости парольной защиты

Пусть A – мощность алфавита паролей (количество символов, которые могут быть использованы при составлении пароля: если пароль состоит только из малых английских букв, то $A = 26$), L – длина пароля, $S = A^L$ – число всевозможных паролей длины L, которые можно составить из символов алфавита A, V – скорость перебора паролей злоумышленником, T – максимальный срок действия пароля.

Тогда, вероятность P подбора пароля злоумышленником в течение срока его действия V определяется по следующей формуле:

$$P = (V \cdot T) / S = (V \cdot T) / A^L.$$

Эту формулу можно использовать в обратную сторону для решения следующей задачи.

Задача. Определить минимальные мощность алфавита паролей A и длину паролей L, обеспечивающих вероятность подбора пароля злоумышленником не более заданной P, при скорости подбора паролей V, максимальном сроке действия пароля T.

Данная задача имеет неоднозначное решение. При исходных данных V, T, P однозначно можно определить лишь нижнюю границу S^* числа всевозможных паролей. Целочисленное значение нижней границы вычисляется по формуле

$$S^* = [V \cdot P / T], \qquad (1)$$

где [] – целая часть числа, взятая с округлением вверх.

После определения нижней границы S^* необходимо выбрать такие A и L для формирования $S = A^L$, чтобы выполнялось следующее неравенство:

$$S^* \le S = A^L. \quad (2)$$

При выборе S, удовлетворяющего неравенству (2), вероятность подбора пароля злоумышленника (при заданных V и T) будет меньше, чем заданная P.

Следует отметить, что при осуществлении вычислений по формулам (1) и (2), величины должны быть приведены к одним размерностям.

Пример. Исходные данные: $P = 10^{-6}$, T = 7 дней = 1 неделя, V = 10 (паролей / минуту) = $10 \cdot 60 \cdot 24 \cdot 7 = 100800$ паролей в неделю. Тогда, $S^* = [(10800 \cdot 1) / 10^{-6}] = = 108 \cdot 10^8$.

Условию $S^* \le A^L$ удовлетворяют, например, такие комбинации A и L, как A = 26, L = 8 (пароль состоит из восьми малых символов английского алфавита), A = 36, L = 6 (пароль состоит из шести символов, среди которых могут быть малые латинские буквы и произвольные цифры).

Шифры замены.

Шифром замены называется алгоритм шифрования, который производит замену каждой буквы открытого текста на какой-то символ шифрованного текста. Получатель сообщения расшифровывает его путем обратной замены.

В классической криптографии различают 4 разновидности шифров замены:

- *Простая замена,* или *одноалфавитный шифр.* Каждая буква открытого текста заменяется на один и тот же символ шифртекста.
- *Омофонная замена.* Аналогична простой замене с единственным отличием: каждой букве открытого текста ставятся в соответствие несколько символов шифртекста. Например, буква "А" заменяется на цифру 5, 13, 25 или 57 *,* а буква "Б" — на 7, 19, 31 или 43 и так далее.
- *Блочная замена.* Шифрование открытого текста производится блоками. Например, блоку "АБА" может соответствовать "РТК", а блоку "АББ" — "СЛЛ".
- Многоалфавитная замена. Состоит из нескольких шифров простои замены. Например, могут использоваться пять шифров простой замены, а какой из них конкретно применяется для шифрования данной буквы открытого текста, — зависит от ее положения в тексте.

Система шифрования Цезаря

Шифр Цезаря является частным случаем шифра простой замены (одноалфавитной подстановки).

При шифровании исходного текста каждая буква заменялась на другую букву того же алфавита по следующему правилу. Заменяющая буква определялась путем смещения по алфавиту от исходной буквы на К букв. При достижении конца алфавита выполнялся циклический переход к его началу. Цезарь использовал шифр замены при смещении К= 3. Такой шифр замены можно задать таблицей подстановок, содержащей соответствующие пары букв открытого текста и шифртекста. Совокупность возможных подстановок для К= 3 показана в таблице 1

Например, послание Цезаря

VENI VIDI VICI

(в переводе на русский означает "Пришел, Увидел, Победил"), направленное его другу Аминтию после победы над понтийским царем Фарнаком, сыном Митридата, выглядело бы в зашифрованном виде так:

YHQL YLGL YLFL

Таблица 4.1

Одноалфавитные подстановки (К = 3, m = 26)

A → D	J → M	S → V
B → E	K → N	T → W

C → F	L → O	U → X
D → G	M → P	V → Y
E → H	N → Q	W → Z
F → I	O → R	X → A
G → J	P → S	Y → B
H → K	Q → T	Z → C
I → L	R → U	

Шифрующие таблицы Трисемуса

В Таблицу сначала вписывается по строкам ключевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополняется не вошедшими в нее буквами алфавита по порядку. Та как ключевое слово или фразу легко хранить в памяти, то такой подход упрощал процессы шифрования и расшифрования.

Для русского алфавита шифрующая таблица может иметь размер 4х8. Вы берем в качестве ключа слово БАНДЕРОЛЬ. Шифрующая таблица таким ключом показана на рис.4.2.

Шифрующая таблица с ключевым словом БАНДЕРОЛЬ

Б	А	Н	Д	Е	Р	О	Л
Ь	В	Г	Ж	З	И	Й	К
М	П	С	Т	У	Ф	Х	Ц
Ч	Ш	Щ	Ы	Ъ	Э	Ю	Я

Рис.4.2.

При шифровании находят в этой таблице очередную букву открытого текста и записывают в шифртекст букву, расположенную ниже ее в том же столбце. Если буква текста оказывается в нижней строке таблицы, тогда для шифртекста берут самую верхнюю букву из того же столбца.

Например, при шифровании с помощью этой таблицы сообщения

ВЫЛЕТАЕМПЯТОГО

получаем шифртекст

ПДКЗЫВЗЧШЛЫЙСЙ

Такие табличные шифры называются монограммными, так как шифрование выполняется по одной букве. Трисемус первым заметил, что шифрующие таблицы позволяют шифровать сразу по две буквы. Такие шифры называются *биграммными.*

Шифры перестановки.

В качестве ключа в шифрующих таблицах используются:

- размер таблицы;
- слово или фраза, задающие перестановку;
- особенности структуры таблицы.

Простая перестановка

Для простой перестановки ключом служит размер таблицы. Этот метод шифрования сходен с шифром скитала. Например, сообщение

ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ

записывается в таблицу из 5 строк и 7 столбцов поочередно по столбцам.

Заполнение таблицы из 5 строк и 7 столбцов

Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

Рис. 4.3.

После заполнения таблицы текстом сообщения по столбцам для формирования шифртекста считывают содержимое таблицы по строкам. Если шифртекст записывать группами по пять букв, получается такое шифрованное сообщение:

ТНПВЕ ГЛЕАР АДОНР ТИЕЬВ ОМОБТ МПЧИР ЫСООЬ

Отправитель и получатель сообщения должны заранее условиться об общем ключе в виде размера таблицы. Объединение букв шифртекста в 5-буквенные группы не входит в ключ шифра и осуществляется для удобства записи несмыслового текста. При расшифровании действия выполняют в обратном порядке.

Несколько большей стойкостью к раскрытию обладает метод шифрования, называемый одиночной перестановкой по ключу. Этот метод отличается от предыдущего тем, что столбцы таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

Применим в качестве ключа, например, слово

ПЕЛИКАН,

На рисунке 4.4. показаны две таблицы, заполненные текстом сообщения и ключевым словом, при этом левая таблица соответствует заполнению до перестановки, а правая таблица – заполнению после перестановки.

Таблицы, заполненные ключевым словом и тенистом сообщения

П	Е	Л	И	К	А	Н
7	2	5	3	4	1	6
Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

До перестановки

А	Е	И	К	Л	Н	П
1	2	3	4	5	6	7
Г	Н	В	Е	П	Л	Т
О	А	А	Д	Р	Н	Е
В	Т	Е	Ь	И	О	Р
П	О	Т	М	Б	Ч	М
О	Р	С	О	Ы	Ь	И

После перестаноки

Рис. 4.4.

В верхней строке левой таблицы записан ключ, а номера под буквами ключа определены в соответствии с естественным порядком соответствующих букв ключа в алфавите. Если бы в ключе встретились одинаковые буквы, они бы были понумерованы слева направо. В правой таблице столбцы переставлены в соответствии с упорядоченными номерами букв ключа.

При считывании содержимого правой таблицы по строкам и записи шифртекста группами по пять букв получим шифрованное сообщение:

ГНВЕП ЛТООА ДРНЕВ ТЕЬИО РПОТМ БЧМОР СОЫЬИ

Для обеспечения дополнительной скрытности можно повторно зашифровать сообщение, которое уже прошло шифрование. Такой метод шифрования называется ***двойной перестановкой.*** В случае двойной перестановки столбцов и строк таблицы перестановки определяются отдельно для столбцов и отдельно для строк. Сначала в таблицу записывается текст сообщения, а потом поочередно переставляются столбцы, а затем строки. При расшифровании порядок перестановок должен быть обратным.

Пример выполнения шифрования методом двойной перестановки показан на рисунке 4.5. Если считывать шифртекст из правой таблицы построчно блоками по четыре буквы, то получится следующее:

ТЮАЕ ООГМ РЛИП ОЬСВ

Ключом к шифру двойной перестановки служит последовательность номеров столбцов и номеров строк исходной таблицы (в нашем примере последовательности 4132 и 3142 соответственно).

Пример выполнения шифрования методом двойной перестановки

	4	1	3	2
3	П	Р	И	Л

	1	2	3	4
3	Р	Л	И	П

	1	2	3	4
1	Т	Ю	А	Е

1	У	Т	А	Ю
4	В	О	С	Ь
2	М	О	Г	О

Исходная таблица

1	Т	Ю	А	У
4	О	Ь	С	В
2	О	О	Г	М

Перестановка столбцов

2	О	О	Г	М
3	Р	Л	И	П
4	О	Ь	С	В

Перестановка строк

Рис. 4.5

Число вариантов двойной перестановки быстро возрастает.- при увеличении размера таблицы:

- для таблицы 3x3 36 вариантов;
- для таблицы 4x4 576 вариантов;
- для таблицы 5x5 14400 вариантов.

Однако двойная перестановка не отличается высокой стойкостью и сравнительно просто "взламывается" при любом размере . таблицы шифрования.

Применение магических квадратов

Магическими квадратами называют квадратные таблицы с "' вписанными в их клетки последовательными натуральными числа- ми, начиная от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число.

Шифруемый текст вписывали в магические квадраты в соответствии с нумерацией их клеток. Если затем выписать содержи- мое такой таблицы по строкам, то получится шифртекст, сформированный благодаря перестановке букв исходного сообщения.

Пример магического квадрата и его заполнения сообщением

ПРИЛЕТАЮ ВОСЬМОГО

показан на рис. 4.6.

Пример магического квадрата 4x4 и его заполнения сообщением ПРИЛЕТАЮ ВОСЬМОГО

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

О	И	Р	М
Е	О	С	Ю
В	Т	А	Ь
Л	Г	О	П

Рис. 4.6.

Шифртекст, получаемый при считывании содержимого правой таблицы по строкам, имеет вполне загадочный вид:

ОИРМ ЕОСЮ ВТАЬ ЛГОП

Число магических квадратов быстро возрастает с увеличением размера квадрата. Существует только один магический квадрат размером 3x3 (если не учитывать его повороты). Количество магических квадратов 4x4 составляет уже 880, а количество магических квадратов 5x5 — около 250000.

Магические квадраты средних и больших размеров могли служить хорошей базой для обеспечения нужд шифрования того времени, поскольку практически нереально выполнить вручную перебор всех вариантов для такого шифра.

Основой для лабораторной работы является блочный симметричный алгоритм шифрования IDEA.

IDEA (англ. International Data Encryption Algorithm, международный алгоритм шифрования данных) — симметричный блочный алгоритм шифрования данных, запатентованный швейцарской фирмой Ascom. Известен тем, что применялся в пакете программ шифрования PGP. В ноябре 2000 года IDEA был представлен в качестве кандидата в проекте NESSIE в рамках программы Европейской комиссии IST (англ. Information Societes Technology, информационные общественные технологии).

Так как IDEA использует 128-битный ключ и 64-битный размер блока, открытый текст разбивается на блоки по 64 бит. Если такое разбиение невозможно, используются различные режимы шифрования. Каждый исходный незашифрованный 64-битный блок делится на четыре подблока по 16 бит каждый, так как все алгебраические операции, использующиеся в процессе шифрования, совершаются над 16-битными числами. Для шифрования и расшифрования IDEA использует один и тот же алгоритм.

Используемые обозначения операций показаны на рис.4:

Рис.4 – Обозначение операций

Фундаментальным нововведением в алгоритме является использование операций из разных алгебраических групп, а именно:

- Сложение по модулю 2^{16}
- Умножение по модулю $2^{16} + 1$
- Побитовое исключающее ИЛИ (XOR).

Эти три операции несовместимы в том смысле, что:

1. Никакие две из них не удовлетворяют дистрибутивному закону, то есть

$$a * (b + c) \; <> \; (a * b) + (a * c) \tag{1.1}$$

2. Никакие две из них не удовлетворяют ассоциативному закону, то есть

$$a + (b \oplus c) \; <> \; (a + b) \oplus c \tag{1.2}$$

Применение этих трех операций затрудняет криптоанализ IDEA по сравнению с DES, который основан исключительно на операции исключающее ИЛИ, а также позволяет отказаться от использования S-блоков и таблиц замены. IDEA является модификацией сети Фейстеля.

Из 128-битного ключа для каждого из восьми раундов шифрования генерируется по шесть 16-битных подключей, а для выходного преобразования генерируется четыре 16-битных подключа. Всего потребуется 52 = 8 x 6 + 4 различных подключей по 16 бит каждый. Процесс генерации пятидесяти двух 16-битных ключей заключается в следующем:

Первым делом, 128-битный ключ разбивается на восемь 16-битных блоков. Это будут первые восемь подключей по 16 бит каждый —

Затем этот 128-битный ключ циклически сдвигается влево на 25 позиций, после чего новый 128-битный блок снова разбивается на восемь 16-битных блоков. Это уже следующие восемь подключей по 16 бит каждый —

Процедура циклического сдвига и разбивки на блоки продолжается до тех пор, пока не будут сгенерированы все 52 16-битных подключа.

Таблица подключей для каждого раунда показана на рис.5:

Номер раунда	Подключи
1	$K_1^{(1)} K_2^{(1)} K_3^{(1)} K_4^{(1)} K_5^{(1)} K_6^{(1)}$
2	$K_1^{(2)} K_2^{(2)} K_3^{(2)} K_4^{(2)} K_5^{(2)} K_6^{(2)}$
3	$K_1^{(3)} K_2^{(3)} K_3^{(3)} K_4^{(3)} K_5^{(3)} K_6^{(3)}$
4	$K_1^{(4)} K_2^{(4)} K_3^{(4)} K_4^{(4)} K_5^{(4)} K_6^{(4)}$
5	$K_1^{(5)} K_2^{(5)} K_3^{(5)} K_4^{(5)} K_5^{(5)} K_6^{(5)}$
6	$K_1^{(6)} K_2^{(6)} K_3^{(6)} K_4^{(6)} K_5^{(6)} K_6^{(6)}$
7	$K_1^{(7)} K_2^{(7)} K_3^{(7)} K_4^{(7)} K_5^{(7)} K_6^{(7)}$
8	$K_1^{(8)} K_2^{(8)} K_3^{(8)} K_4^{(8)} K_5^{(8)} K_6^{(8)}$

выходное преобразование	$K_1^{(9)} K_2^{(9)} K_3^{(9)} K_4^{(9)}$

Рис.5 – Таблица подключей

Структура алгоритма IDEA показана на рис.6:

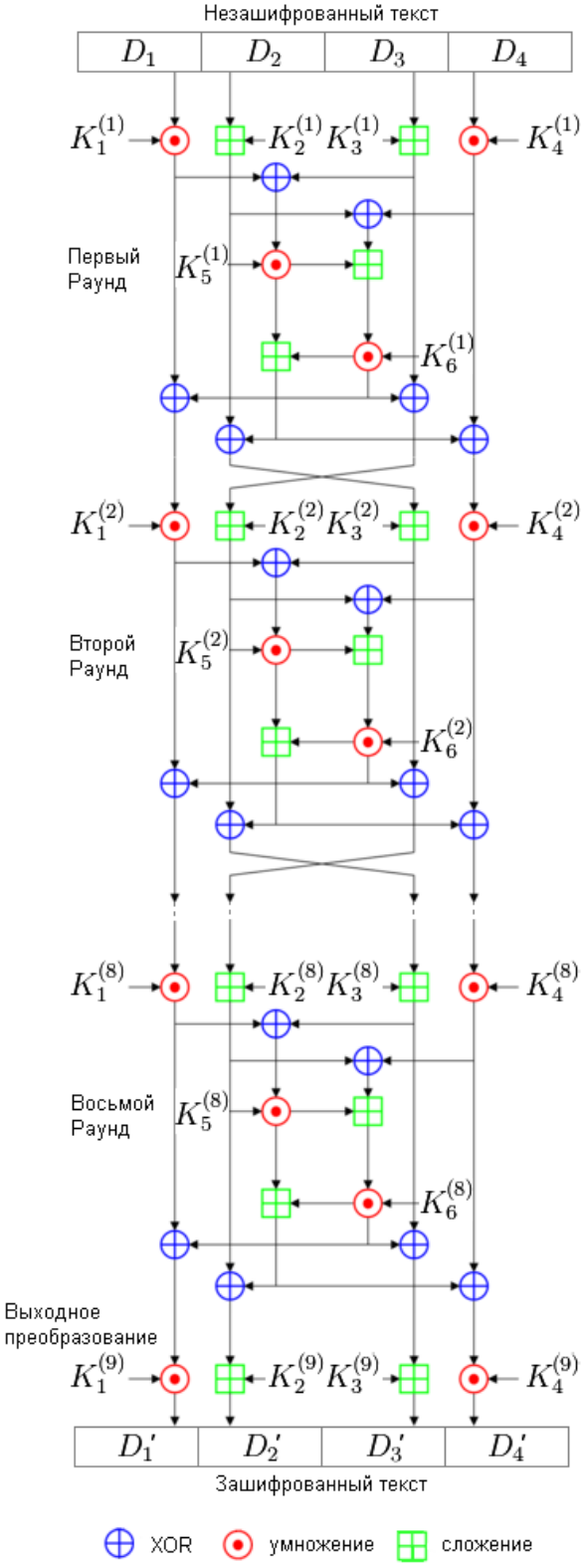


Рис.6 – Структура алгоритма IDEA

Процесс шифрования состоит из восьми одинаковых раундов шифрования и одного выходного преобразования. Исходный незашифрованный текст делится на блоки по 64 бита. Каждый такой блок делится на четыре подблока

по 16 бит каждый. На рис.6 эти подблоки обозначены D1, D2, D3, D4. В каждом раунде используются свои подключи согласно таблице подключей (рис.5). Над 16-битными подключами и подблоками незашифрованного текста производятся следующие операции:

- Умножение по модулю $2^{16} + 1 = 65537$, причем вместо нуля используется 216
- Сложение по модулю 2^{16}
- Побитовое исключающее ИЛИ

В конце каждого раунда шифрования имеется четыре 16-битных подблока, которые затем используются, как входные подблоки для следующего раунда шифрования. Выходное преобразование представляет собой укороченный раунд, а именно, четыре 16-битных подблока на выходе восьмого раунда и четыре соответствующих подключа подвергаются операциям:

- Умножение по модулю $2^{16} + 1$
- Сложение по модулю 2^{16}

После выполнения выходного преобразования конкатенация подблоков D1', D2', D3' и D4' представляет собой зашифрованный текст. Затем берется следующий 64-битный блок незашифрованного текста и алгоритм шифрования повторяется. Так продолжается до тех пор, пока не зашифруются все 64-битные блоки исходного текста.

Блок открытого текста размером 64 бит делится на четыре равных подблока размером по 16 бит

$$(D_1^{(0)}, D_2^{(0)}, D_3^{(0)}, D_4^{(0)}) \tag{1.3}$$

Для каждого раунда (i = 1...8) вычисляются:

$$A^{(i)} = D_1^{(i-1)} * K_1^{(i)} \tag{1.4}$$

$$B^{(i)} = D_2^{(i-1)} + K_2^{(i)} \tag{1.5}$$

$$C^{(i)} = D_3^{(i-1)} + K_3^{(i)} \tag{1.6}$$

$$D^{(i)} = D_4^{(i-1)} * K_4^{(i)} \tag{1.7}$$

$$E^{(i)} = A^{(i)} \oplus C^{(i)} \tag{1.8}$$

$$F^{(i)} = B^{(i)} \oplus D^{(i)} \tag{1.9}$$

$$D_1^{(i)} = A^{(i)} \oplus ((F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \quad (1.10)$$

$$D_2^{(i)} = C^{(i)} \oplus ((F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \quad (1.11)$$

$$D_3^{(i)} = B^{(i)} \oplus (E^{(i)} * K_5^{(i)} + (F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \quad (1.12)$$

$$D_4^{(i)} = D^{(i)} \oplus (E^{(i)} * K_5^{(i)} + (F^{(i)} + E^{(i)} * K_5^{(i)}) * K_6^{(i)}) \quad (1.13)$$

Результатом выполнения восьми раундов будут следующие четыре подблока:

$$(D_1^{(8)}, D_2^{(8)}, D_3^{(8)}, D_4^{(8)}) \quad (1.14)$$

Выполняется выходное преобразование (i = 9):

$$D_1^{(9)} = D_1^{(8)} * K_1^{(9)} \quad (1.15)$$

$$D_2^{(9)} = D_3^{(8)} + K_2^{(9)} \quad (1.16)$$

$$D_3^{(9)} = D_2^{(8)} + K_3^{(9)} \quad (1.17)$$

$$D_4^{(9)} = D_4^{(8)} * K_4^{(9)} \quad (1.18)$$

Результатом выполнения выходного преобразования является зашифрованный текст:

$$(D_1^{(9)}, D_2^{(9)}, D_3^{(9)}, D_4^{(9)}) \quad (1.19)$$

Метод вычисления, использующийся для расшифровки текста по существу такой же, как и при его шифровании. Единственное отличие состоит в том, что для расшифровки используются другие подключи. В процессе расшифровки подключи должны использоваться в обратном порядке, показаны на рис.7:

Номер раунда	Подключи
1	$1/K_1^{(9)} \; -K_2^{(9)} \; -K_3^{(9)} \; 1/K_4^{(9)} \; K_5^{(8)} \; K_6^{(8)}$
2	$1/K_1^{(8)} \; -K_3^{(8)} \; -K_2^{(8)} \; 1/K_4^{(8)} \; K_5^{(7)} \; K_6^{(7)}$

3	$1/K_1^{(7)} \; -K_3^{(7)} \; -K_2^{(7)} \; 1/K_4^{(7)} \; K_5^{(6)} \; K_6^{(6)}$
4	$1/K_1^{(6)} \; -K_3^{(6)} \; -K_2^{(6)} \; 1/K_4^{(6)} \; K_5^{(5)} \; K_6^{(5)}$
5	$1/K_1^{(5)} \; -K_3^{(5)} \; -K_2^{(5)} \; 1/K_4^{(5)} \; K_5^{(4)} \; K_6^{(4)}$
6	$1/K_1^{(4)} \; -K_3^{(4)} \; -K_2^{(4)} \; 1/K_4^{(4)} \; K_5^{(3)} \; K_6^{(3)}$
7	$1/K_1^{(3)} \; -K_3^{(3)} \; -K_2^{(3)} \; 1/K_4^{(3)} \; K_5^{(2)} \; K_6^{(2)}$
8	$1/K_1^{(2)} \; -K_3^{(2)} \; -K_2^{(2)} \; 1/K_4^{(2)} \; K_5^{(1)} \; K_6^{(1)}$
Выходное преобразование	$1/K_1^{(1)} \; -K_2^{(1)} \; -K_3^{(1)} \; 1/K_4^{(1)}$

Рис.7 – Таблица подключей для расшифровки

Первый и четвёртый подключи i-го раунда расшифровки получаются из первого и четвёртого подключа (10-i)-го раунда шифрования мультипликативной инверсией. Для 1-го и 9-го раундов второй и третий подключи расшифровки получаются из второго и третьего подключей 9-го и 1-го раундов шифрования аддитивной инверсией. Для раундов со 2-го по 8-й второй и третий подключи расшифровки получаются из третьего и второго подключей с 8-го по 2-й раундов шифрования аддитивной инверсией. Последние два подключа i-го раунда расшифровки равны последним двум подключам (9-i)-го раунда шифрования.

Мультипликативная инверсия подключа K обозначается 1/K и

$$(1/K) * K = 1 \; mod \; (2^{16} + 1) \qquad (1.20)$$

Так как 2^{16} + 1 — простое число, каждое целое не равное нулю K имеет уникальную мультипликативную инверсию по модулю 216 + 1. Аддитивная инверсия подключа K обозначается -K и

$$-K + K = 0 \; mod \; (2^{16}) \qquad (1.21)$$

Алгоритм DES представляет собой блочный шифр, предназначенный для шифрования данных 64-битовыми блоками. С одного конца поступает 64-битовый блок открытого текста, а с другого конца выходит 64-битовый блок шифротекста. DES относится к симметричным

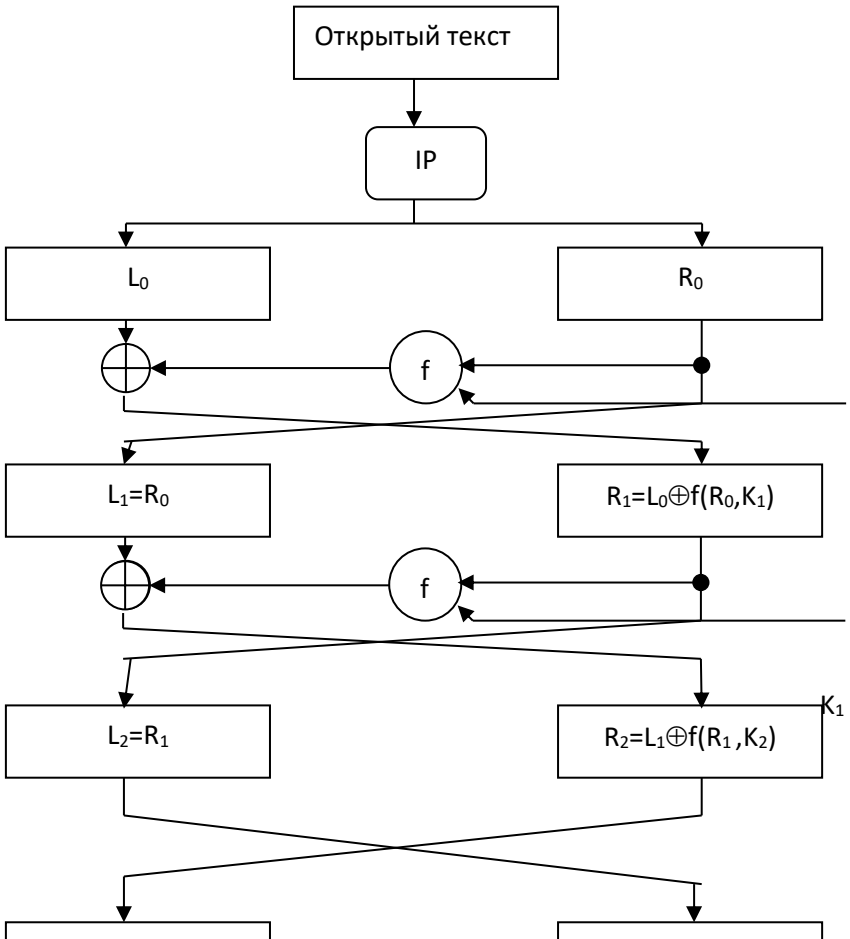
алгоритмам, т.е. для расшифрования и зашифрования ипользуються одинаковые алгоритм и ключ (за исключением небольших различий в процедуре развертки ключа).

Длина ключа равна 56 бит. (ключ обычно представляется 64-битовым числом, но каждый

восьмой бит используется для конторля четности и игнорируется. Биты четности –наименнее значащие биты в байтах ключа). Ключ , которым может служить любое 56-битовое число, можно изменить в любой момент времени. Ряд чисел полагаются слабыми ключами, но число чрезвычайно мало по сравнению с возможным числом всех ключей. Секретность всецело 70 определяется ключом.

На простейшем уровне алгоритм представляет собой комбинацию двух основных методов шифрования: рассеивания и перемешивания.Фундаментальным строительным блоком Des является применение к тексту единичной комбинации этих методов (подстановка, а за ней перестановка), зависящей от ключа. Такой блок называется раундом. DES включает 16 раундов, одна и та же комбинация методов применяется к открытому тексту 16 раз (см. рис.3.1.).

В алгоритме используется только стандартная арифметика и логические операции над 64-битовыми числами,поэтому он без труда реализоывается в аппаратуре второй половины 70 годов. Изобилие повторений в алгоритме делает его идеальным для реализации специализированными микросхемами.



K_2

K_{16}

Рис.3.1. Алгоритм DES

3.1.1. Схема алгоритма

DES оперирует 64-битными блоками открытого текста . После первоначальной перестановки блок разбивает на правую и левую половины длиной по 32 бита. Затем выполняет 16 раундов одинаковых действий, называемых функцией f, в которых данные объединяются с ключом. После 16 раунда правая и левая половины объединяются, и алгоритм завершается заключительной перестановкой(обратной первоначальной).

На каждом раунде (см. рис 3.2.) биты ключа сдвигаются, а затем из 56 битов ключа выбираются 48 битов. Правая половина данных увеличивается до 48 битов путем перестановки с расширением, ксорится с 48 битами смещенного и переставленного ключа, проходит через 8 S-блоков, образуя 32 новых бита, и переставляется снова. Эти 4 операции выполняются функцией f. Затем результат выполнения ксорится с левой половиной. В итоге этих действий появляется новая правая половина, а старая правая становится левой. Эти действия повторяются 16 раз, образуя 16 раундов алгоритма DES.

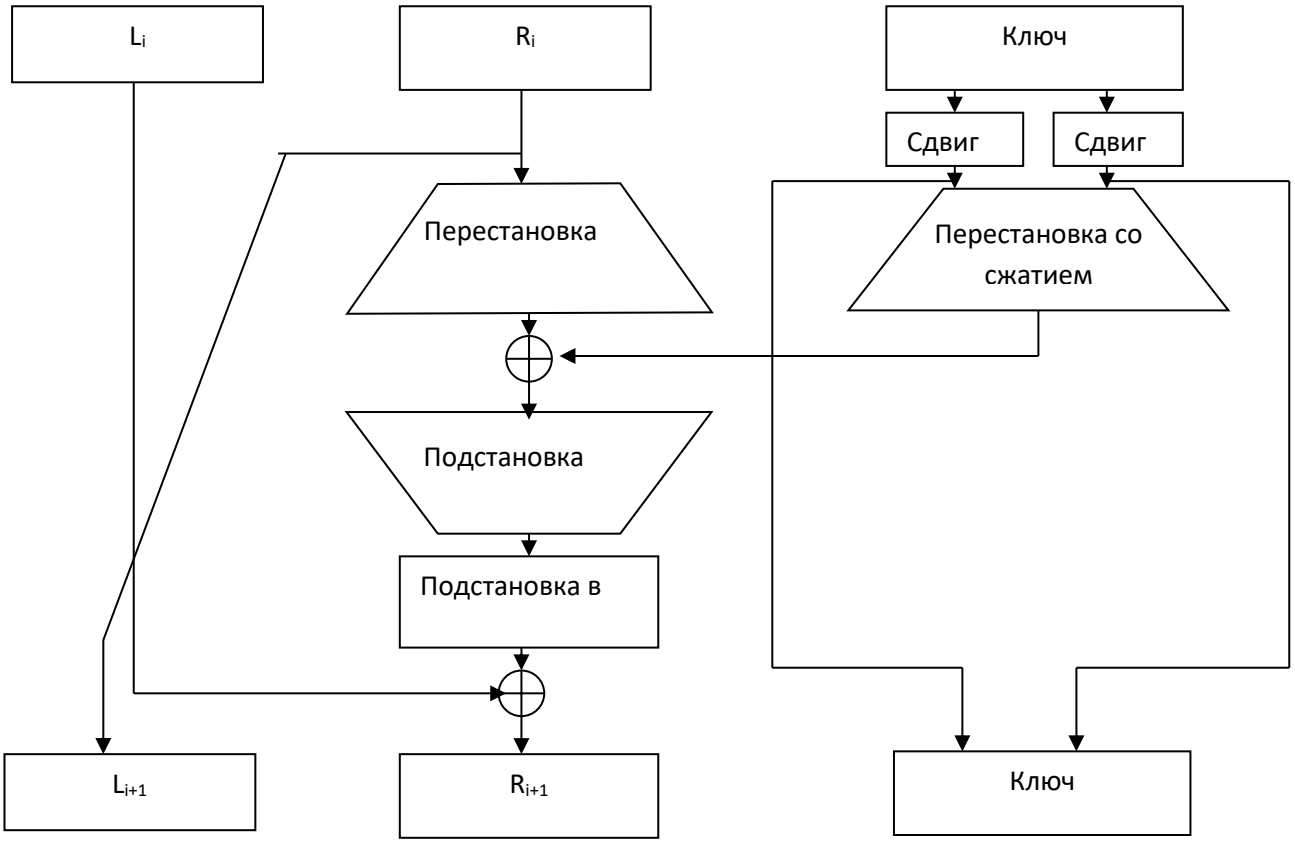


Рис. 3.2. Один раунд DES

Если обозначить B_i результат i-ой итерации, а L_i и R_i -левую и правую половины B_i , K_i – 48-битовый ключ для раунда i, а f –функцию, выполняющую все подстановки, перестановки и операции XOR с ключом, то раунд можно представить так:

$L_{i+1}=R_i$

$R_{i+1}=L_i$ XOR $f(R_i,K_{i+1})$

3.1.2. Начальная перестановка

Начальная перестановка (ее еще называют IP) выполняется еще до раунда 1, при этом входной блок переставляется (см. рис. 3.3.). Эту и все остальные таблицы этой главы, следует читать слева направо и сверху вниз. Например, начальная перестановка перемешает 58 бит в битовую позицию 2, бит 42 –в битовую позицию 3 и т.д.

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Рис. 3.3. Начальная перестановка

3.1.3. Преобразование ключа

Первоначально 64-битовый ключ DES сокращается до 56-битового отбрасыванием каждого восьмого бита. (см. рис. 3.4.) Эти биты используются только для контроля четности, позволяя проверять отсутствие ошибок.

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Рис. 3.4. Перестановка ключа.

После извлечения 56-битового ключа, для каждого из 16 раундов DES генерируется новый 48-битовый подключ. Эти подключи Ki определяются следующим образом:

- 56-битовый ключ делится на две 28-битовые половины
- две половины сдвигаются налево на 1 или 2 бита, в зависимости от раунда. Этот сдвиг показан в таблице 3.1.
- выбираются 48 из 56 битов. Поскольку при этом не только выбирается подмножество битов, но и изменяется их порядок, данная операция называется сжимающей перестановкой. В ее результате появляется набор из 48 битов. Сжимающая перестановка определена на рис. 3.5.

Таблица 3.1.

Число битов сдвига ключа в зависимости от раунда.

Раунд	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Число	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

14	17	11	24	1	5	3	28	15	6	21	10
23	19	11	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Рис. 3.5. Сжимающая перестановка

Благодаря сдвигу, в каждом подключе используется отличное от других подмножество битов ключа. Каждый бит используется приблизительно в 14 из 16 подключей, хотя невсе биты испольуются в точности одинаковое количество раз.

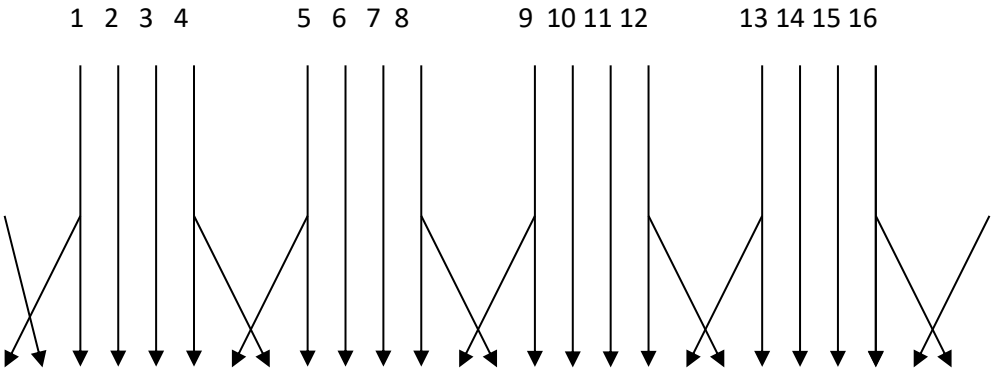
3.1.4 Расширающая перестановка

Эта операция расширяет правую половину данных Ri от 32 битов до 48 битов. Так как при этом не просто повторяются определенные биты**,** но и изменяется их порядок, эта опрерация называется расширяющей перестановкой. Она решает две задачи:

- Приведение размера правой половины в соответствии с ключом для операции XOR.
- Получение более длинного результата, который можно будет сжимать по ходу операции подстановки.

Однако основной криптографический смысл иной. За счет влияния одного бита на две подстановки зависимость би тов результата от битов исходных данных возрастает быстрее. Это явление называется лавинным эффектом. Алгоритм DES спроектирован так, чтобы как можно быстрее добиться в зависимости каждого бита шифротекста от каждого бита открытого текста ми каждого бита ключа.

Расширяющая перестановка показана на рис. 3.6. Иногда ее называют E-блоком или E/P. В каждом 4-битовом входном блоке, первый и четвертый бит выходного блока. На рис. 3.7. показано, какие позиции результата соответствуют разным позициям исходных данных. Хотя выходной блок больше входного, каждый входной блок генерирует уникальный выходной блок.



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 2122 23 24

Рис. 3.6. Расширяющаяся перестановка

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

Рис. 3.7. Позиции битов результата расширяющей перестановки

3.1.5. Подстановка с помощью S-блоков

После сложения сжатого ключа с расширенным блоком с помощью XOR, над 48-битовым результатом выполняется операция подстановки. Подстановки производится с помощью восьми блоков подстановки или S-блоков (от англ. Substitution boxes). У каждого s-блока 6-битовый вход и 4-битовый выход. 48 битов делятся на восемь. 6-битовых подблоков. Каждый отдельный подблок обрабатывается отдельным S-блоков: 1-й S-блоком 1, 2-й S-блоком 2 и т. д.(см. рис. 3.8.).

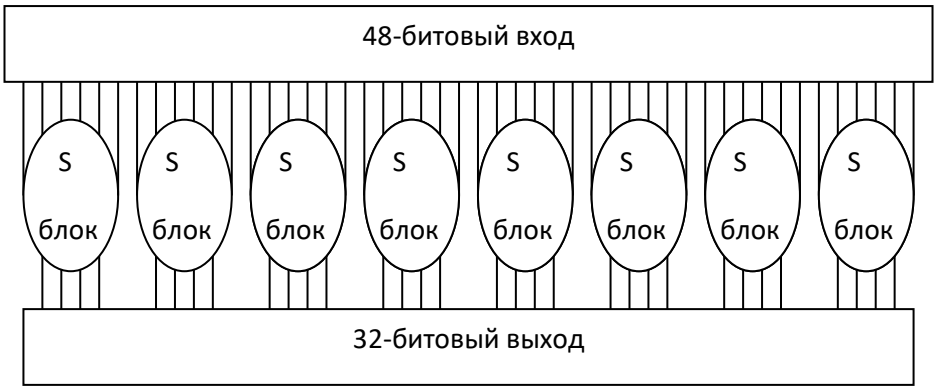


Рис. 3.8. Подстановка в s-блоках

Каждый S-блок представляет собой таблицу из 4 строк и 16 столбцов. Элементами блока сложат 4-битовые числа. По шести входным битам S-блока определяются номера столбцов и строк, под которыми следует искать выходное значение. Все 8 S-блоков показаны на рис. 3.9.

Входные биты определяют элемент s-блока особым образом. Рассмотрим 6-битовый вход S-блока: b1,b2,b3,b4,b5,b6. Биты b1 и b6 объединяются, образуя 2-битовое число от 0 до 3, соответствующее строке S-блока. Средние 4 бита, с b2 по b5 объединяются, образуя 4-битовое число от 0 до 15, соответствующее столбцу блока.

Например, пусть на вход шестого S-блока поступает 110011. Первый и последний бит, объединяясь образуют 11, что соответствует строке 3 шестого блока. Средние 4 бита образуют 1001, что соответствует столбцу 9 того же S-блока. Элемент S-блока 6, находящийся на пересечении строки 3 и столбца 9 –это 14.(Не забывайте, что строки и столбцы нумеруются, начиная с 0, а не с 1).

Подстановка с помощью S-блоков – ключевой шаг алгоритма DES. Другие операции алгоритма линейны и легко поддаются анализу. S-блоки нелинейны, и именно они в большей степени, чем все остальное, обеспечивают стойкость DES.

В результате этого шага подстановки получаются восемь 4-битовых блоков, которые вновь объединяются в единый 32-битовый блок. Этот блок поступает на вход перестановки с помощью P-блоков.

S-блок 1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-блок 2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S-блок 3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S-блок 4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S-блок 5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S-блок 6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S-блок 7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S-блок 8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8

2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Рис. 3.9. S-блоки

3.1.6. Перестановка с помощью P-блоков

32-битовый выход подстановки с помощью S-блоков перетасовывается в соответствии с P-блоком. Эта перестановка перемещает каждый входной бит в другую позицию, ни один бит не используется дважды, ни один бит не отбрасывается. Этот процесс называется прямой перестановкой или просто перестановкой.(см. таблицу 3.2.)

Таблица 3.2.

Перестановка с помощью P-блоков

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Наконец, результат перестановки с помощью P-блоков ксорится с левой половиной исходного 64-битового блока. Затем левая и правая половины меняются местами, и начинается следующий раунд.

3.1.7. Заключительная перестановка

Заключительная перестановка обратна начальной перестановке и показана на рис. 3.10. Обратите внимание: после последнего раунда DES левая и правая половины не меняются местами. Вместо этого объединеный блок $R_{16}L_{16}$ используется как вход заключительной перестановки.

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

Рис. 3.10. Заключительная перестановка

3.1.8. Расшифрование DES

После всех подстановок, перестановок, XOR и циклических сдвигов можно подумать, что алгоритм расшифрования резко отличается от алгоритма шифрования и точно так же запутан. Напротив, различные компоненты DES подобны так, чтобы выполнялось полезное свойство: для зашифрования и расшифрования используется один и тот же алгоритм.

Алгоритм DES позволяет использовать для зашифрования или расшифрования блока одну и ту же функцию. Единственное отличие состоит в том, что ключи должны использоваться в обратном порядке. Иными словами, если в раундах зашифрования использовались ключи К1,К2,К3…К16, то ключами расшифрования будут К16,К15,К14…К1. Алгоритм, который создает ключ для каждого раунда, тоже цикличен. Ключ сдвигается направо, а число позиций сдвига равно 0,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1.

3.2. Описание 3DES

В ряде реализаций DES используется тройной алгоритм DES (см. рис. 3.11). Так как преобразование DES не является группой, полученный шифротекст гораздо труднее вскрыть полным перебором: 2^{112} попыток вместо 2^{56}

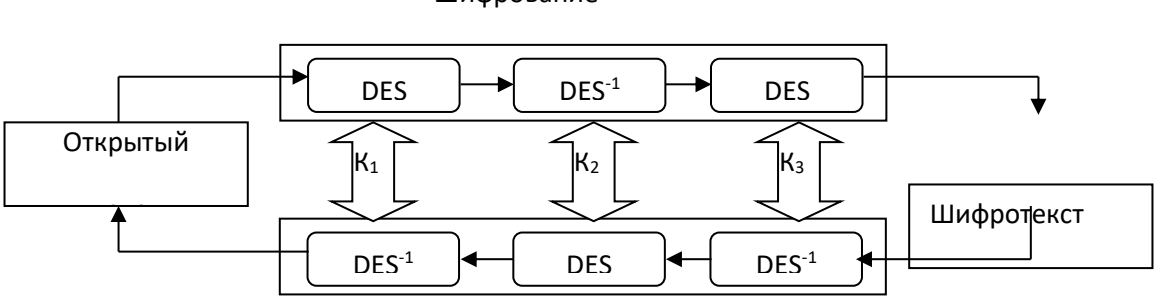


Рис. 3.11. Тройной DES

3.3. Описание S-DES

Упрощенный DES или S-DES – алгоритм шифрования, который носит скорее учебный, чем практический характер. В данной лабораторной работе S-DES описывается для лучшего

понимания DES-подобных симметричных шифров (основаных на « подстановке с расширением» и цепи Фейстеля). Основные характеристики S-DES :

- Алгоритм опрерирует 8-битными блоками открытого текста, на выходе 8-битный блок шифротекта
- Ключ 10-битный. На каждом раунде используется 8-битный подключ
- S-блоки имеют структуру 4*4. Имеют 4-битный вход и 2-битный выход

Для более подробного рассмотрения алгоритма S-DES в динамике можно воспользоваться файлом DEMO_SDES.exe.

3.4. Криптографические режимы DES

Криптографический режим обычно объединяет базовый шифр, обратную связь какого-то типа и нескольких простых операций. Опреации просты, поскольку стойкость определяется используемым алгоритмом шифрования, а не режимом.

Требования, предъявляемые к режимам:

- Режим не должен снижать стойкость используемого алгоритма
- Эффективность режима не должна быть ниже, чем у используемого алгоритма
- Отказоустойччивость(устойчивость к потерям синхронизации процессов шифрования и дешифрования).

3.4.1. Режим ECB

Самый очевидный метод использование шифров – использование режима электронной кодовой книги – ECB. В режиме ECB каждый блок открытого текста заменяется блоком шифротекста. Все блоки открытого текста шифруются независимо друг от друга.

Недостатки:

- Предоставление криптоаналитику более широких возможностей для криптоанализа по сравнению с другими криптографическими режимами.
- Противник, чтобы обмануть предполагаемого получателя, может изменять шифрованные сообщения, даже не зная ключа или алгоритма.

Достоинства:

- Возможность шифрования нескольких сообщений одним ключом без снижения надежности.

Распространение ошибки:

- При расшифровании ошибки в символах шифротекста ведут к некорректному расшифрованию соответствующего блока открытого текста, однако не затрагивают остальной открытый текст.
- При случайной потере или добавлении лишнего бита шифротекста весь последующий шифротекст будет расшифрован некорректно (если только для выравнивания границ блоков не используется какое-нибудь выравниевания по границам блока).

3.4.2. Режим CBC

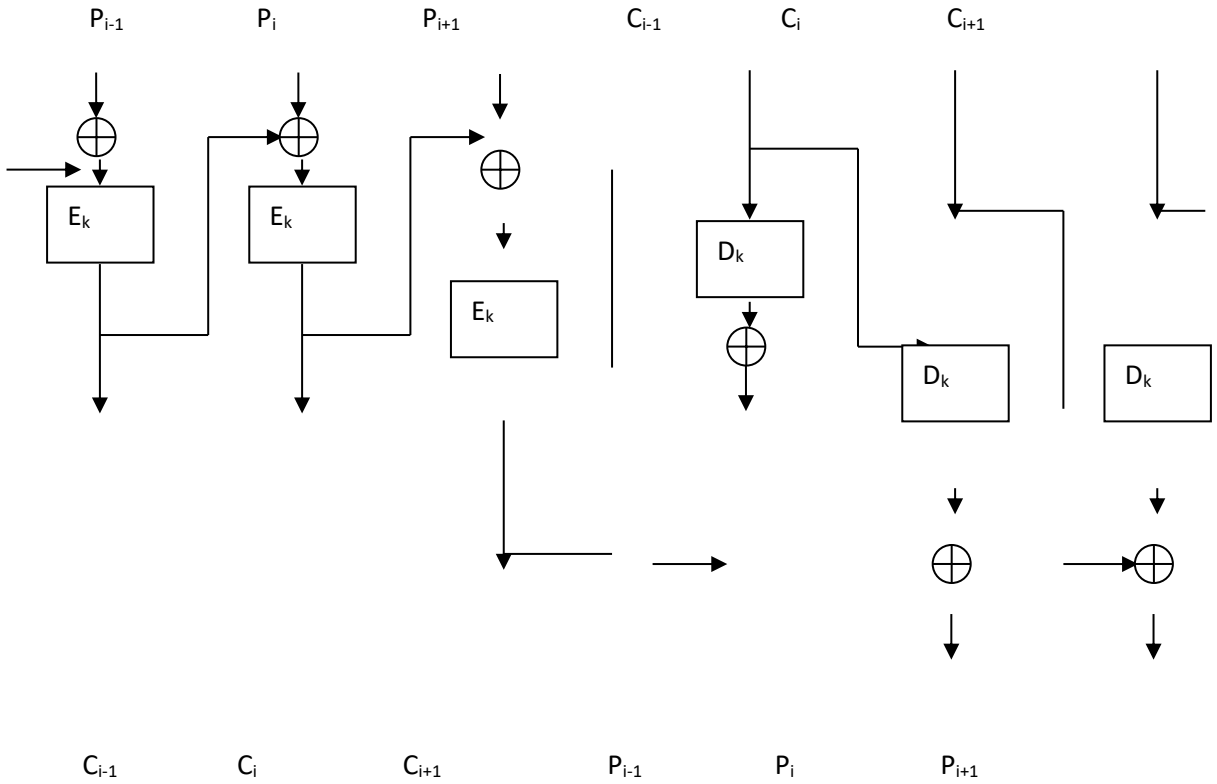
В режиме сцепления блоков шифротекста –CBC перед шифрованием над открытым текстом и предыдущим блоком шифротекста выполняется опреация XOR(см. рис. 3.12(а)). Когда блок открытого текста зашифрован, полученный шифротекст сохраняется в регистре обратной связи. Следующий блок открытого текста перед шифрованием подвергается опреации XOR с содеожимым регистра обратной связи. Результат этой операции используется как входные анные для следующего этапа процедуры шифрования и так до конца сообщения.

Расшифрование производится в обратном порядке(см. рис3.12.(б)).

Математически это выглядит следующим образом:

$C_i=E_k(P_i \text{ XOR } C_{i-1})$

$P_i=C_{i-1} \text{ XOR } D_k(C_i)$



(а) Шифрование в режиме CBC (б) Расшифрование в режиме CBC

Рис. 3.12. Режим сцепления блоков шифротекста.

Достоинства:

- Более надежный режим по сравнению с ECB за счет механизма сцепления.

Недостатки:

- При шифровании двух идентичных сообщений создается один и тотже шифротекст
- Так как блок шифротекста достаточно влияет на следующий блок, злоумышленик может незаметно добавляет блоки к концу зашифроанного сообщения. В некоторых ситуациях это нежелательно.
- Злоумышленник может изменить один бит шифротекста, тогда весь блок будет расшифрован нежелательно плюс один бит в следующем блоке. Возможны случаи, когда это нежелательно.

Распространение ошибки:

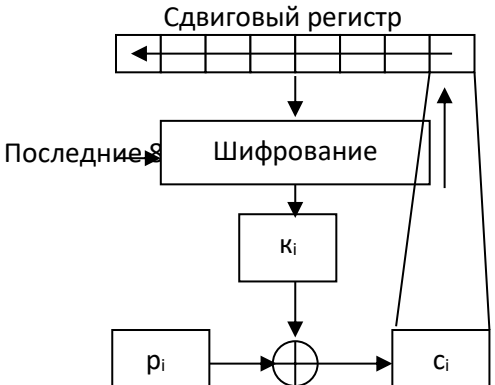
- Ошибка в одном бите блока шифротекста влияет на один блок и один бит восстановленного открытого текста – это блок после расшифрования искажается полностью. В следующем блоке искажается всего один бит, находящийся в позиции что и ошибочный бит. (Ошибка не влияет на последующие блоки)
- Если в потоке шифротекста теряется или добавляется бит, положения всех последующих блоков сдвигаются на один бит, и результатом расшифрования будет сплошная чепуха.

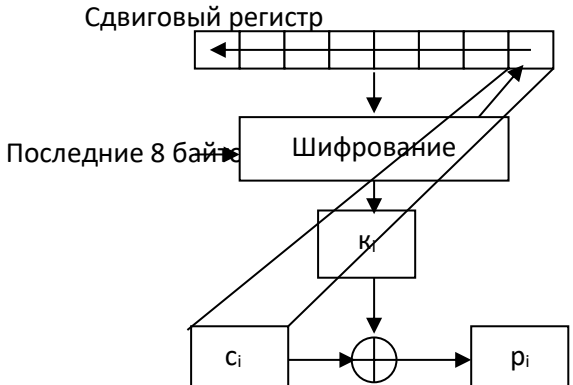
3.4.3. Режим CFB

Блочный шифр можно реализовывать, как самосинхронизующийся потоковый шифр. Такой режим называется режимом обратной связи по шифротексту CFB. В режиме CBC начать шифрование до поступления полного блока данных невозможно. Для некоторых сетевых приложений это создает проблемы. Например в защищенном сетевом окружении терминал должен иметь возможность передавать хосту каждый символ сразу после ввода. Если же данные нужно обрабатывать блоками в несколько байт, режим CBC не работает.

В режиме CFB можно шифровать единицы данных размером не больше блока. На рис.3.13.изображен 8-битовый CFB. Кроме того, можно использовать 64-битовый CFB, или любой n-битовый CFB, где n больше или равно размеру блока.

Блочный алгоритм в режиме CFB работает с очередью, регистром сдвига, размер которого равен размеру используемого блока. Как и в режиме CBC, превоначально очередь заполнена вектором инициализации ВИ (см.ниже). Очередь шифруется, затем выполняется операция XOR над восьмью старшими (крайними левыми) битами результата и первым 8-битовым символом открытого текста(см. рис. 3.13.). В результате появляется первый бит шифротекста. Теперь этот символ можно передать. Кроме того, полученные восемь битовпопадают в очередь на место восьми младших битов, а все остальные биты сдвигаются на 8 позиций влево. Предыдущин восемь старших битов отбрасываются. Затем точно также шифруется следующей символ открытого текста. Расшифрование выполняется в обратном порядке. Обе стороны – шифрующая и расшифровывающая-используют блочный алгоритм в режиме шифрования.





Ключ К Ключ К

(а) Шифрование в режиме CFB (б) Расшифрование в режиме CFB

Рис. 3.13. Режим 8-битовой обратной связи по шифротексту.

Если обозначить n размер блока алгоритма, то n-битовый CFB выглядит так, как показано на рис. 3.14.

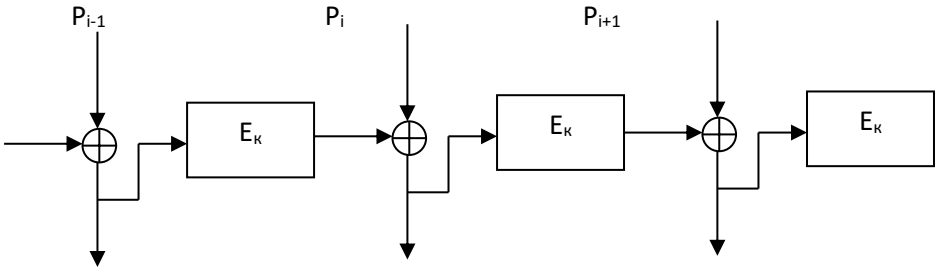


Рис. 3.14. Работа n-битового CFB с n-битовым алгоритмом.

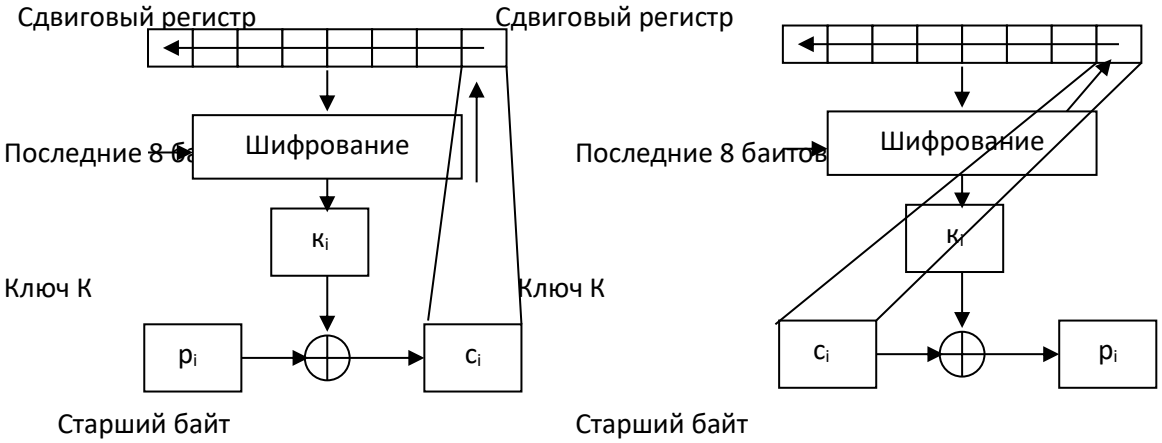
Распространение ошибки:

- Ошибка в шифротексте опасна. Первый результат сбоя бита шифротекста –сбой одного бита открытого текста. Затем ошибка попадает в сдвиговый регистр, и пока ошибочный бит не покинет регистра, формируемый шифротекст будет некорректен. В 8-битовом режиме CFB из-за сбоя 1 бита искажаются 9 байт дешифрованного открытого текста. Затем система восстанавливается, и весь последующий текст расшифровывается корректно. В общем случае, в n-битовом режиме CFB 1 ошибка шифротекста влияет на расшифрование текущего и следующих m/n-1 блоков, где m-размер блока.

- ❖ Режим CFB самостоятельно восстанавливается после ошибок синхронизации. Ошибка попадает в регистр сдвига и, пока она в нем находится, искажает 8 байт данных.

3.4.4. Режим OFB

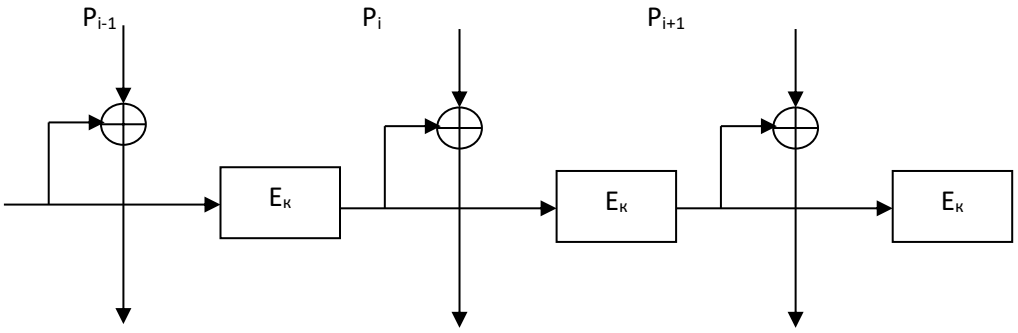
Режим обратной связи по выходу – OFB представляет собой метод использования блочного шифра в качестве синхронного потокового шифра. Этот режим подобен CFB за исключением того, что n-битов предыдущего выходного блока сдвигаются в крайние правые позиции очереди (см. рис. 3.15.). Расшифрование выполняется в обратном порядке. Такой режимом OFB. Блочный алгоритм работает в режиме шифрования как на шифрующей , так и на расшифровывающей сторонах. Такую обратную связь иногда называют внутренней, поскольку механизм обратной связи не зависит ни от потока открытого текста ни от потока шифротекста. Если обозначить как n размер блока алгоритма, то n-битовый алгоритм OFB выглядит, так как показано на рис. 3.16.



(а) Шифрование в режиме OFB (б) Расшифрование в режиме OFB

Рис. 3.13. Режим 8-битовой OFB

.



C_{i-1} C_i C_{i+1}

Рис. 3.16. n-битового OFB с n-битовым алгоритмом.

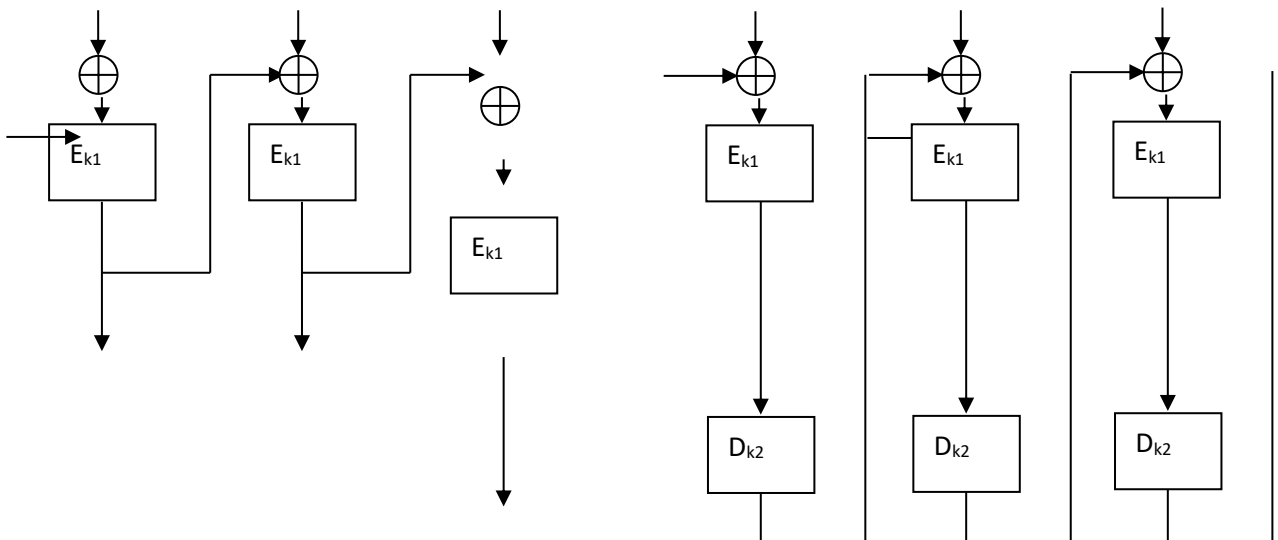
Распространение ошибки:

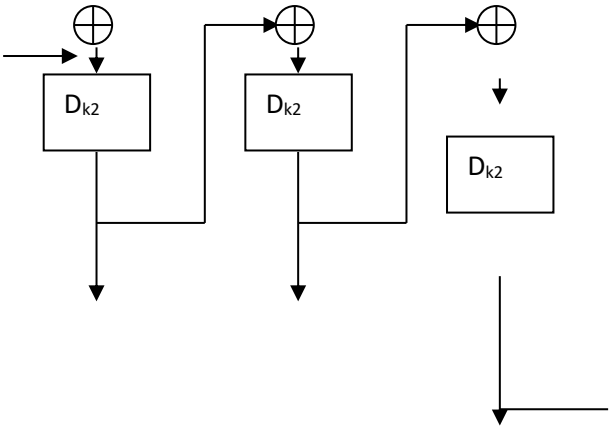
- В режиме OFB ошибки не распространяются. Некорректный бит в шифротексте ведет к некорректному биту в открытом тексте. Это может быть полезно при цифровой передаче аналоговых сигналов, например оцифрованных речевых сигналов или видеоизображений, когда случайный сбой бита терпим, но распростаранение ошибки недопустимо.
- С другой стороны потеря синхронизации фатальна. Если сдвиговый регистры пришифровании различаются, восстановленный открытый текст представляет собой бессмыслицу. В любой системе, использующей режим OFB, должен быть предусмотрен механизм обнаружения потери синхронизации и механизм заполнения обоих сдвиговых регистров новым (или тем же самим) вектором инициализации для восстановления синхронизации.

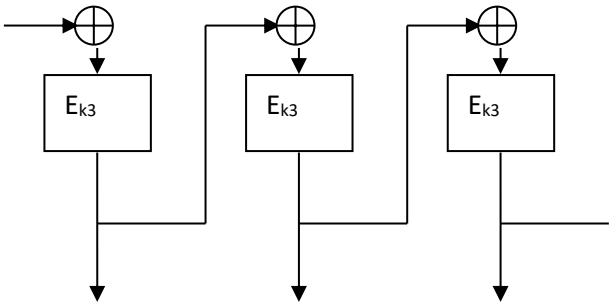
3.4.5. Режимы тройного шифрования

Помимо режима ECB, существует два возможных режима тройного шифрования:

- Внутренний CBC. Файл зашифровывается в режиме CBC три раза (см. рис. 3.17а). Для этого нужны 3 различных вектора инициализации.
- Внешний CBC. Файл шифруется с помощью тройного шифрования 1 раз в режиме CBC (см. рис. 3.17б). Для этого нужен 1 вектор инициализации.







(а) Внутрений CBC (б) Внешний CBC

Рис. 3.17. Тройное шифрование в режиме CBC

Оба режима требуют больше ресурсов, чем однократное шифрование: больше времени или больше аппаратуры.

3.4.6. Вектор инициализации

При шифровании в режимах CBC, CFB, OFB одинаковые блоки открытого текста превращаются в различающиеся друг от друга блоки шифротекста только в том случае, если

различались какие предшествующие блоки открытого текста. Однако при шифровании двух идентичных сообщений создается один и тот же шифротекст. Хуже того, два одинаково начинающихся сообщения будут шифроватьс одинаково вплоть до первого различия.

Чтобы избежать этого, можно зашифровать в первом блоке какие-то произвольные данные. Этот блок случайных данных называют векторм инициализации. Вектор не имеет какого-то смыслового значения, он используется только для того, чтобы сделать каждое сообщение уникальным. Когда получатель расшифровывает этот блок, он использует его только для заполнения регистра обратной связи. В качестве вектора удобно использовать метку времени, либо какие-то случайные данные.

Вектор необязательно хранить в секрете, его можно передаватьоткрыто – вместе с шифротекстом.

В режиме CBC вектор необязательно должен быть уникальным, хотя это желательно. В режимах же CFB и OFB уникальность вектора –обязательное условие.

3.4.7. Дополнение блоков

большинство сообщений не делятся точно на 64-битовые блоки шифрования – в конце обычно оказывается укороченный блок. Для решения этой проблемы используют дополнение.

Чтобы создать полный блок, последний блок дополняют некоторым стандартным шаблоном – нулями, единицами, чередованием нулей и единиц. Если после расшифрования дополнение необходимо удалить, запишите число байтов –заполнителей в последний байт последнего блока. Пусть ,например, размер блока составляет 64 бит, а размер последнего блока – 3 байта (24 бит). Таким образом, для дополнения блока до 64 бит необходимо добавить 5 байтов-заполнителей. Добавьте 4 байта нулей, а в последний байт запишите число 5. Чтобы этот метод работал корректно, нужно дополнять каждое сообщение. Даже если открытый текст заканчивается на границе блока, вам придется добавить один полный блок. Существуют еще другие альтернативные методы[1].

3.4.8. Выбор режима шифрования

Если вам необходима главным образом простота и скорость, режим ECB можно порекомендовать, как самый простой и быстрый режим блочного шифра. Помимо уязвимости к вскрытию с повторной передачей, алгоритм в режиме ECB проще всех для криптоаналитиков. Поэтому использовать режим ECB для шифрования сообщений не рекомендуется.

Режим ECB удобно использовать для случайных данных , например, ключей. Так как данные случайны и невелики по размеру, то недостатки режима ECB в данном случае несущественны.

В таблице 3.3 подведены итоги по различным показателям вышеперечисленных режимов.

Для шифрования файлов лучше всего подходит режим CBC. Надежность значительно возрастает, и хотя иногда появляюся ошибки в символах хранимых данных, почти никогда не бывает сбоев синхронизации.

Режим CFB - в частности, 8-битовый режим CFB –лучше всего подходит для шифрования потоков символов, когда каждый символ должен обрабатываться по отдельности, как, например, в линии связи между терминалом и хостом.

Режим OFB очень часто используют в высокоскоростных синхронных системах, где распространение ошибки недопустимо. Режим OFB предпочтительно использовать в среде, подтвержденной ошибкам, поскольку в этом режиме ошибка не распространяется, кроме того, режим OFB – лучший режим, если требуется предварительно обработка.

Режим	Стойкость	Эффективность	Помехоустойчивость
ECB	- структура открытого текста не скрыта - вход в блочный шифр не рандомизирован и совпадает с открытым ключом + одним ключом можно зашифровать несколько сообщений - открытым текстом нетрудно манипулировать, блоки можно перемещать, повторить или переставлять	+ скорость та же, что и у блочного шифра - предварительная обработка невозможна + параллельная обработка возможна	- ошибка шифротекста затрагивает один полный блок открытого текста - ошибка синхронизации невосстановима
CBC	+ структура открытого текста скрыта операцией XOR с предыдущим блоком шифротекста + ввод в блочный шифр рандомизирован опреацией XOR с предыдущим блоком шифротекста + одним ключом можно зашифровать несколько сообщений +/- манипулировать открытым текстом трудно; можно удалять блоки из начала и конца сообщения, изменять биты в последнем блоке, а повторенпие допускает некоторые контролируемые изменения	+ скорость та же, что и у блочного шифра - предварительная обработка невозможна +/- шифрование не может быть выполнено параллельно расшифрование может, и допускает произвольный доступ	- ошибка шифротекста затрагивает один полный блок открытого текста и соответствующий бит следующего блока - ошибка синхронизации невосстановима

CFB	+ структура открытого текста скрыта + ввод в блочный шифр рандомизирован + при использовании разных векторов инициализации одним ключом можно зашифровать несколько сообщений +/- манипулировать открытым текстом трудно; можно удалять блоки из начала и конца сообщения, изменять биты в последнем блоке, а повторенпие допускает некоторые контролируемые изменения	+ скорость та же, что и у блочного шифра +/- шифрование не может быть выполнено параллельно расшифрование может, и допускает произвольный доступ - прежде, чем блок станет видимым, допускается некоторая предварительная обработка; возможно шифрование предыдущего блока шифротекста	- ошибка шифротекста влияет на соответствующий бит открытого текста и весь последующий блок + ошибки синхронизации размером в полный блок восстановимы. 1-битовый режим может восстановиться после добавления или утраты единичных битов
OFB	+ структура открытого текста скрыта + ввод в блочный шифр рандомизирован + при использовании разных векторов инициализации одним ключом можно зашифровать несколько сообщений - открытым текстом очень легко манипулировать; любое изменение шифротекста напрямую изменяет открытый текст	+ скорость та же, что и у блочного шифра - прежде, чем блок станет видимым, допускается предварительная обработка - шифрование не может быть выполнено параллельно	+ ошибка шифротекста затрагивает только соответствующий бит открытого текста - ошибка синхронизации невосстановима

3.5. Дифференциальный криптоанализ

Дифференциальный криптоанализ – способ атаки DES- подобных алгоритмов с использованием подобного открытого текста, который является эффективнее атаки грубой силой (перебор всех ключей).

Дифференциальгный криптоанализ работает с парами шифротекстов, открытые тексты котрых имеют определенные разности. Метод анализирует эволюцию этих разностей в процессе прохождения открытых текстов через раунды DES при шифровании одним и те же ключом.

Просто-напросто выбирем два открытых текста с фиксированной разностью. Криптоаналитику даже не нужно знать их значения. – лишь бы удовлетворяли некоторому условию разности. (Для DES термин «разность» определяется с помощью операции XOR). Затем, используя разности полученных шифротекстов, присвоим различные вероятности различным ключам. В процессе дальнейшего анализа следующих пар шифротекстов один из ключей станет наиболее вероятным.

Подробности гораздо сложнее. На рис. 3.18.представлена функция одного раунда DES. Представьте себе пару входов, X и X^1 , с разностью Δ X . Выходы, У и $У^1$ известны, следовательно, известна их разность ΔУ. Известны и расширяющая перестановка, и P-блок, поэтому известны ΔA и ΔC. B и B^1 неизвестны, но ΔB известна и равна ΔA.($\Delta B=B \text{ xor } B^1=(A \text{ xor } K_i)(A^1 \text{ xor } K_i)=A \text{ xor } A^1=\Delta A$). Фокус вот в чем для любого заданного ΔA не все значения ΔC равновероятны. Так структура S-блоков известна, то комбинация ΔA и ΔC позволяет предположить значения битов для A xor K_i и A^1 xor K_i . А поскольку A и A^1 известны, это дает нам информацию о K_i.

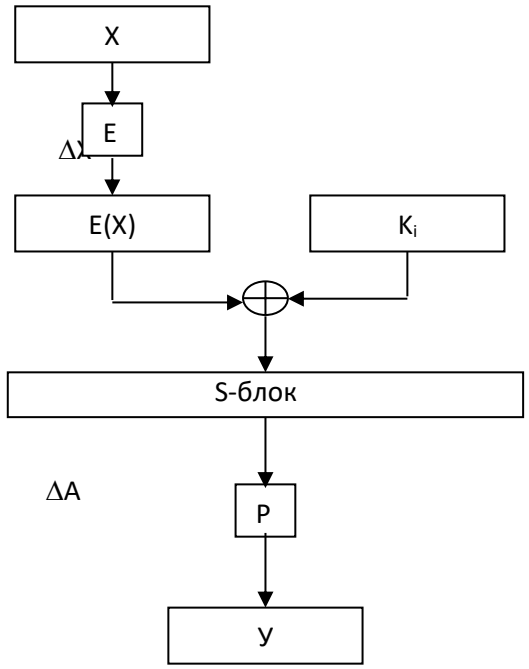


Рис. 3.18. Функция одного раунда DES

Взглянем на последний раунд DES. (В дифференциальном криптоанализе начальная и заключительная перестановки игнорируются. Они не влияют на атаку, а только затрудняют объяснение). Если мы сможем найти K_{16} , мы получим 48 битов ключа. Оставшиеся 8 битов мы можем получить грубой силой.

Далее рассмотрим пример атаки при помощи дифференциального криптоанализа на 3-раундовый S-DES. (описание S-DES вы можете наити в файле DEMO_DES.exe)

1. сначала мы должны поставить условие разности для пар открытого текста. Пусть оно будет равнятся 10000000_2 или 80_{16}. Открытые тексты различаются только первым битом.
2. Сгенерируем некоторое количество пар открытого текста, удовлетворяющих нашему условию разности. Например : 104,232,60,188,109,236,61,189,858,186,73,201,99,227,51,179
3. Выберем случайный 24-битовый ключ (алгоритм S-DES 3- раундовый). Пусть ключ будет – мяч. Далее зашифруем этим ключом наши пары открытого текста, получив пары шифротекста. В результат получим таблицу 3.4.

Таблица 3.4.

№ пары	Открытый текст	Шифротекст
1	104(01101000) 232(11101000)	67(0100001) 128(10000000)
2	60(00111100) 188(10111100)	199(11000111) 4(00000100)
3	108(01101100) 236(11101100)	97(01100001) 125(01111101)
4	61(00111101) 189(10111101)	22(00010110) 17(00010001)
5	58(00111010) 186(10111101)	127(01111111) 74(01001010)
6	73(01001001) 201(11001001)	212(11010100) 72(01001000)
7	99(01100011) 227(11100011)	107(01101011) 87(01010111)
8	51(00110011) 179(10110011)	64(01000000) 53(00110101)

4. Далее рассмотрим подробнее какую-то пару, например, первую.

a) Взглянем на последний раунд S-DES. Правая часть шифртекста (4 бита) являются входом в E/P последнего раунда. Соответственно 0011 для 1-го и 0000 для 2-го шифртекста. Так как известна расширяющая перестановка E/P, то мы можем определить A, A^1 и ΔA. $A=10010110_2=96_{16}$ $A=00000000_2$; $\Delta A=10010110_2$

b) Далее определим вход в S-блок или ΔB. Так как выше упоминалось, что: $\Delta B=B \text{ xor } B^1=(A \text{ xor } K_i)(A^1 \text{ xor } K_i)=A \text{ xor } A^1=\Delta A$, то $\Delta B=10010110_2=96_{16}$

c) Левая часть открытого текста ксорится с выходом первого раунда. Так как условие разности для открытого текста 10000000_2, то после перестановки между первым и вторым раундом разность станет 00001000_2. После второго раунда правая часть остается неизменной, и после перестановки становится левой частью. То есть в конце 3-го раунда разность на его выходе ксорится с разностью $1000_2=8_{16}$

d) Теперь определим разность $\Delta У$ у левых частей шифртекста первой пары. $\Delta У=1100_2=C_{16}$. Далее это разность поксорим с найденной в пункте с разностью. Получим $\Delta У=0100_2=4_{16}$

e) Так как структура P-блока известна, нетрудно сделать его инверсию. Получим $\Delta C=0001_2=1_{16}$

f) В результате для первой пары мы получили следующие данные : $A=10010110_2=96_{16}$ $A=00000000_2=0$; $\Delta A=\Delta B=10010110_2=96_{16}$; $\Delta C=0001_2=1_{16}$

Далее найдем эти данные для всех пар. Получим таблицу 3.5.:

Таблица 3.5.

№ пары	A	A^1	ΔA	ΔB	ΔC
1	10010110	00000000	10010110	10010110	0001
2	10111110	00101000	10010110	10010110	0001
3	10000010	11101011	01101001	01101001	1100
4	00111100	10000010	10111110	10111110	0100
5	11111111	01010101	10101010	10101010	1110
6	00101000	01000001	01101001	01101001	1000
7	11010111	10111110	01101001	01101001	1110
8	00000000	10101010	10101010	10101010	1111

5. Так как структура S-блоков нам известна (на рис), то мы можем установить зависимость, связывающую ΔC и ΔB. Для любой заданной ΔA, не все значения ΔC равновероятны. Комбинация ΔA и ΔC позволяет предполагать значения битов A xor K_i и A^1 xor K_i. А поскольку A и A^1 известны, это дает нам информацию о K_i. То есть сейчас мы проведем анализ пар шифртекстов, в процессе которого один из ключей будет встречаться чаще. Это и есть правильный ключ.

   a) Будем рассматривать отдельно S0-блок и S1-блок (см. рис. 3.19.). Для 1-ой пары для S0 $\Delta B=1001_2$. Это означает, что входы в S0 отличаются 1-м и 4- м битами.

	00	01	10	11
00	01	00	11	10
01	11	10	01	00
10	00	10	01	11
11	11	01	11	10

(а) S0

	00	01	10	11
00	00	01	10	11
01	10	00	01	11
10	11	00	01	00
11	10	01	00	11

(б) S1

Рис 3.19.Структура S-блоков S-DES

Исходя из этого, а также структуры S-DES, нужно рассматривать следующую комбинацию сток и столбцов S0-блока (см. рис. 3.20.) . Столбцы входов одинаковы.

	00	01	10	11
00	01	00	11	10
01	11	10	01	00
10	00	10	01	11
11	11	01	11	10

Рис. 3.20.

b) Для первой пары для S0 $\Delta C=00_2$, следовательно выходы S0 блока одинаковы. Поэтому нужно искать одинаковые элементы выделены на рис 3.21.

	00	01	10	11
00	01	00	***11***	***10***
01	11	***10***	***01***	00
10	00	***10***	***01***	11
11	11	01	***11***	***10***

Рис. 3.21.

c) Выпишем все возможные входы (неприобразованные), которым соответствует найденные элементы: 0101 и 1001 0010 и 1110 0110 и 1010 0011 и 1111
d) Преобразуем их, то есть запишем в виде, в котором они поступали на вход S-блока. Получим: 0011 и 1010 0100 и 1101 0101 и 1100 0110 и 1111
e) Далее данные входы мы ксорим с A и A^1. Для данной первой пары для S0-блока $A=1001_2$ $A^1=0000_2$. Однако, достаточно ксорить либо A, либо A^1 (можете сами проверить). Для простоты выберем A^1. Получим: 0011 и 1010 0100 и 1101 0101 и 1100 0110 и 1111 Запишем результат в 16-ной системе : 3 и A 4 и D 5 и C 6 и F

Мы получили на данный момент все возможные 4 бита ключа.

6. Далее по такому же принципу проанализируем другие пары. Получим табл.3.6

Таблица 3.6.

№ пары	Полученные варианты 4 битов ключа
1	3,A,4,D,5,C,6,F
2	1,8,6,F,7,E,4,D
3	8,E,A,C,9,F,B,D,0,6,2,4
4	9,2,C,7,E,5,F,4
5	F,5,9,3,A,0,8,2
6	F,9
7	B,D,9,F,A,C,8,E,3,5,1,7
8	0,A,6,C,5,F,7,D

7. Затем посмотрим, какой ключ чаще всего встречался в ходе анализа (таблица 3.7)

Таблица 3.7.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3	2	3	3	4	5	4	4	4	5	5	2	5	5	4	8

Таким образом, наиболее вероятными первыми четырьмя битами ключа, исходя из проанализированных 8 пар, является F_{16}

Проверка:

М использовали в процессе шифрования ключ мяч. Следовательно, на последнем раунде использовался подключ ч, который в кодировке ANSI имеет код $247_{10}=F7_{16}$

Вывод:

Первые 4 бита ключа найдены верно.

Примечание:

В данном примере атаки с помощью дифференциального криптоанализа на S-DES 8 пар хватило, чтобы выделить правильный подключ из шума. Однако, при увеличении числа раундов алгоритма число пар для анализа сильно возрастает.Для 16-раундового DES требует проанализировать 2^{47} пар.

8. По такому же принципу можно найти оставшиеся 4 бита ключа.

В пункте 4.с. мы нашли разность в конце последнего раунда. Определенные разности в паре открытыхтестов с высокой вероятностью вызывают определенные разности и получаемых шифртекстов. Эти разности называют характеристиками. Характеристики распространяются на на определенное количетво раундов. Существуют входная разность, разность на каждом раунде и выходная разность, каждая с определенной вероятностью.

Так как мы рассматривали всего 3 раунда, то нахождение характеристики на последнем раунде у нас не вызывало больших проблем. Однако, при рассмотрении алгоритма с числом раундов большим 3-х может произойти некоторая загвоздка.

Но эту проблему можно решить, создав таблицу, строки которой представляют возможные входные XOR (операции XOR двух различных наборов входных битов), столбцы – возможные выходные XOR, а элементы – сколько раз конкретный выход XOR встречается для заданого входа XOR. Такую таблицу можно сгенерировать для каждого используемого S-блока.

Например, в рассмотренном выше примере:

На входе 1-го раунда разность левой части –1000; правой части – 0000

На входе 2-го раунда (после перестановки) раз-ть лев. Части –0000; прав. Части –1000

На входе 3-го раунда (после перестановки) раз-ть лев. Части –0000; прав. Части –X

Как найти X?

На входе 2-го раунда разность правой части –1000. Проведем эту разность через расширяющую перестановку E/P, получим: 01000001. Это означает, что на вход S0-блока подается разность 0100, а на вход S1-блока разность 0001. Теперь нам надо найти разность таблицы, о которой указывалось выше.

Для S0-блока. Возьмем 8 пар, удовлетворяющих разности 0100:

0000 0100

0001 0101

0010 0110

0011 0111

0100 0000

0101 0001

0110 0010

0111 0011

1000 1100

1001 1101

1010 1110

1011 1111

1100 1000

1101 1001

1110 1010

1111 1011

Далее проведем все пары через S0-блок. Получим следующие выходы:

01 11

11 01

00 10

11 01

01 11

10 00

00 10

00 01

11 11

10 11

01 10

01 00

11 11

11 10

10 01

И соответственно из разности:

10

10

10

10

10

10

10

10

01

00

01

11

01

00

11

01

Таким образом, таблица распределения битовых разностей для S0-блока будет (таблицу 3.8.):

Таблица 3.8.

Выходная XOR Входная XOR	00	01	10	11
0000	.	.	.	.
0001	.	.	.	.
0010	.	.	.	.

0011	.	.	.	.
0100	2	4	8	2
.	.	.	.	.
.	.	.	.	.
1111	.	.	.	.

Остальные элементы таблицы можно найти таким же образом. В результате, исходя из таблицы распределенных вероятностей, можно сказать, что разность на выходе S0-блока, при входной разности равной 0100_2, будет 10_2 с вероятностью равной 8/16 =1/2. Примечание: такая таблица очень легко генерируется программно для любого S-блока.

Таким же способом получаем таблицу для S1-блока (см. таблицу 3.9):

Таблица 3.9.

Выходная XOR Входная XOR	00	01	10	11
0000	.	.	.	.
0001	.	.	.	.
0010	.	.	.	.
0011	.	.	.	.
0100	2	8	2	4
.	.	.	.	.
.	.	.	.	.
1111	.	.	.	.

В результате разность на выходе S1-блока, при входной разности равной 0001_2, будет 01_2 с вероятностью равной 8/116=1/2

Таким образом, исходя из того, чо вероятность одноцикловой характеристики вычисляется как произведение вероятностей всех одноблочных характеристик её составляющих, на вход P-блока с вероятностью ½* ½ = ¼ попадает разность 1001_2. На выходе P-блока получается разность 0101_2. Затем эта разность ксорится с0000_2 и в результате на выходе 2-го раунда получается разность 0101_2 с вероятностью 1/16.(См. рис 3.22.) Также различные характеристики можно

объединять. Более подробную инфомацию о дифференциальном криптоанализе можно найти в [1]и [2].

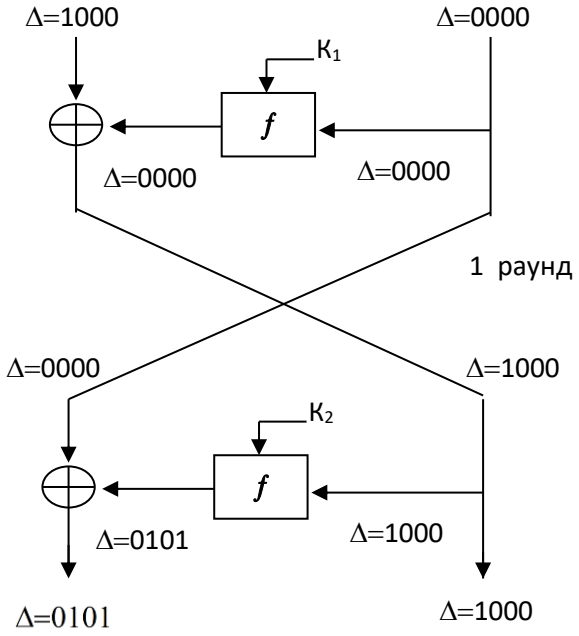


Рис. 3.22. 2-х раундовая характеристика S-DES