Тема 1. Основные понятия

1.1. Введение

Благодаря современным достижениям компьютерных и информационных технологий автоматизированные системы (АС) обработки информации стали играть существенную роль в производстве, научных исследованиях, обеспечении потребностей общества и отдельного человека. Круг пользователей АС постоянно расширяется, наряду с этим происходит объединение АС в единую глобальную сеть Internet, дающую возможность миллионам людей получать доступ к ресурсам АС всего мира и воздействовать на порядок их работы.

Автоматизированным системам поручается решение важных для безопасности государства, общества и отдельного человека задач, например охрана государственных секретов, управление атомными станциями, электронные банковские расчеты. В связи с этим нет необходимости доказывать, что без решения комплекса задач защиты АС будут нести в себе постоянную угрозу.

Проблема защиты информации в АС с момента формулирования в середине 70-х годов до современного состояния прошла длительный и во многом противоречивый путь. Первоначально существовали два направления решения задачи поддержания конфиденциальности: использование криптографических методов защиты информации в средах передачи и хранения данных и программно-техническое разграничение доступа к данным и ресурсам вычислительных систем. Заметим, что в начале 80-х годов АС были слабо распределенными, технологии глобальных и локальных вычислительных сетей находились на начальной стадии своего развития, и указанные задачи удавалось достаточно успешно решать.

Обращаясь к работам того времени, можно отметить, что они оперируют рядом терминов, которые авторы считают "интуитивно" понятными (противник, ресурс, данные и т.д.). Надо отметить, что такое положение сохранилось и до сегодняшнего дня.

Позднее с появлением тенденции к распределенной обработке информации (проект ARPANET и последующие разработки) классический подход к организации разделения ресурсов и классические криптографические протоколы начинают постепенно исчерпывать себя и эволюционировать. На лидирующее место выходят проблемы аутентификации взаимодействующих элементов АС, а также способы управления криптографическими механизмами в распределенных системах. При этом в различных работах начинает

складываться мнение о том, что функции криптографической защиты являются равноправными для АС и должны быть рассмотрены вместе с другими функциями. Данный тезис послужил отправной точкой для разделения проблематики собственно средств защиты (включая криптографические средства, средства контроля доступа и др.) и средств обеспечения их корректной работы.

С другой стороны, остро встает проблема реализации технических решений по защите для конкретной программно-аппаратной среды. Ситуация осложняется тем, что качественные криптографические механизмы считаются свойственными только закрытым государственным сетям. Кроме того, идеология аутентификации пользователей, сформировавшаяся в конце 70-х годов и действующая до настоящего времени, подразумевает применение символьных пользовательских паролей.

Проблематика защиты информации в середине 80-х все более явно разделяется на несколько направлений: формулирование и изучение свойств теоретических моделей безопасности АС, анализ моделей безопасного взаимодействия, рассматривающих различные аспекты криптографической защиты, теория создания качественных программных продуктов. На сегодняшний день такое положение продолжает сохраняться, а "лавинообразное" появление новых программных продуктов порождает определенный кризис в решении практических вопросов при проектировании систем защиты. Так, новые технологические решения в АС (в первую очередь связанные с распределенностью), например механизм удаленного вызова процедур или технология типа "клиент-сервер", в теоретических работах пока отражены недостаточно.

В начале 80-х годов возникает ряд моделей защиты, основанных на декомпозиции автоматизированной системы обработки информации на субъекты и объекты,-модели Белла-Лападула (Bell-LaPadula), модель Take-Grant и т.д. В данных моделях ставятся и исследуются вопросы взаимодействия элементов системы с заданными свойствами. Целью анализа и последующей реализации модели является именно достижение таких свойств системы, как конфиденциальность и доступность. Например, описывается дискреционный механизм безопасности, разделяющий доступ поименованных субъектов к поименованным объектам или полномочное управление доступом, моделирующее систему категорий и грифов доступа. Как правило, та или иная модель безопасности исходит из априорной технологии работы с объектами (так, полномочное управление моделирует структуру секретного делопроизводства), которая может быть формализована и обоснована. При практической реализации данных моделей в конкретных АС встал вопрос о

гарантиях выполнения их свойств (фактически это выполнение условий тех или иных утверждений, обосновывающих свойства формализованной модели). В связи с этим в зарубежной литературе формулируется понятие доверенной (достоверной) вычислительной базы (в английской транскрипции TCB), гарантирующей свойства системы.

Необходимо также упомянуть о том, что существующая методология проектирования защищенной системы представляет собой итеративный процесс устранения найденных слабостей, некорректностей и неисправностей. Причем зачастую ряд злоумышленных действий не блокируется принципиально – противодействие данным угрозам выводится в область организационно-технических мер, что означает отказ от рассмотрения как конкретных угроз, так и целых их классов.

С середины 80-х годов несовершенство западной методологии было замечено российскими специалистами и отражено в ряде работ. С этого времени намечается тенденция к появлению комплексных решений в области проектирования и реализации механизмов защиты АС (по крайней мере, в теории).

В 1991 г. В.А. Герасименко предложил модель системно- концептуального подхода к безопасности, которая описывает методологию анализа и синтеза системы безопасности, исходя из комплексного взаимодействия ее компонентов, рассматриваемых как система. Результатом изучения является также совокупность системно-связанных рекомендаций по решению проблемы.

В 1996 г. в классической работе. Грушо А.А. и Тимониной Е.Е. "Теоретические основы защиты информации" высказан и обоснован тезис о том, что гарантированную защищенность в автоматизированной системе следует понимать как гарантированное выполнение априорно заданной политики безопасности. В указанной работе также приведены примеры гарантированных политик. С другой стороны, в моделях АС, как правило, редуцируется порождение субъектов, которому в реальных системах соответствует порождение процессов и запуск программ. Очевидно, что данное допущение в определенной степени снижает достоверность модели, поскольку порождение субъектов существенно влияет на свойства защищенности.

В настоящее время часто акцентируют внимание на проблеме компьютерных вирусов. Действительно, компьютерные вирусы представляют собой угрозу безопасности практически всех компьютерных систем и при этом формально не укладываются в рамки непосредственного влияния на систему. В ряде работ они рассматриваются как некая отдельная сущность или даже форма жизни. Впоследствии проблематика компьютерных вирусов трансформировалась в рассмотрение целого класса так называемых

разрушающих программных воздействий (информационного оружия) . (В этой с вязи нельзя не отмети ть интересные работы С. П. Расторгуева.)

Ранее условия гарантий политики безопасности формулировались в виде стандартов (без доказательства). Именно такой подход применили американские специалисты по компьютерной безопасности, опубликовав с 1983 г. несколько книг стандарта так называемой "радужной серии". В России аналогичные документы были приняты Государственной технической комиссией в 1992 г. и дополнены в 1995 г. Р 50739-95 (СВТ. Защита от несанкционированного доступа к информации. Общие технические требования).

Можно отметить конструктивность такого подхода к формулированию гарантий политики безопасности, поскольку проверка наличия заданного набора свойств достаточно легко проводится или может быть заложена при проектировании. Однако, налицо определенная деструктуризация и рассмотрение формально не связанных между собой требований. Качественное описание свойств системы (например "идентификация и аутентификация" или "контроль целостности") не касается взаимосвязи данных механизмов и их количественные характеристики.

Математическая модель политики безопасности рассматривает систему защиты в некотором стационарном состоянии, когда действуют защитные механизмы, а описание разрешенных или неразрешенных действий не меняется. На практике АС обработки информации проходит путь от отсутствия защиты к полному оснащению защитными механизмами; при этом система управляется, т.е. разрешенные и неразрешенные действия в ней динамически изменяются.

Упоминавшиеся выше методики оценки защищенности представляют собой в какой-то мере необходимые условия. Выполнение заданного набора качественных показателей, с одной стороны, не позволяет оценить количественно каждый показатель, а с другой – как было указано выше, препятствует системному подходу.

Таким образом, основной особенностью информационной безопасности АС является ее практическая направленность. Большинство положений сначала реализовывалось в виде конкретных схем и рекомендаций, а уж затем обобщалось и фиксировалось в виде теоретических положений или методических рекомендаций. Другой особенностью информационной безопасности АС была на первых этапах развития значительная зависимость теоретических разработок от конкретных способов реализации АС, определявшихся проектными программными или аппаратными решениями. Как можно заметить, данная особенность связана с предыдущей: особенности

реализации той или иной АС определяют возможные виды атак, а следовательно, те или иные необходимые защитные меры. На настоящий момент эти две особенности АС в определенной степени нивелированы, что позволяет перейти к разработке системонезависимых теоретических положений, на основании которых будут реализовываться проекты различных АС.

Еще одной особенностью информационной безопасности АС является многоаспектность, т.е. обеспечение безопасности ведется по широкому кругу направлений. С этим связано возникновение определенных коллизий – когда некая организация объявляла рекламный лозунг "Мы защитим вашу информацию!", речь могла идти о чем угодно – от продажи блоков бесперебойного питания до средств шифрования. Для обеспечения безопасности АС важны все направления.

Несмотря на все многообразие систем, для которых необходимо решать задачу защиты информации (это могут быть операционные системы (ОС), системы управления базами данных (СУБД), локальные или глобальные вычислительные сети), имеются общие теоретические подходы к ее решению. Комплексному, структурированному, методологически обоснованному рассмотрению теории компьютерной безопасности и посвящены эти лекции.

Как естественно-научная дисциплина теория компьютерной безопасности постепенно эволюционирует в направлении формализации и математизации своих положений, выработки единых комплексных подходов к решению задач защиты информации. В то же время на настоящий момент нельзя сказать, что этот процесс близок к завершению. Некоторые подходы носят характер описания применяемых методов и механизмов защиты и представляют их механическое объединение. Кроме того, в связи с развитием АС и информационных технологий постоянно возникают новые задачи по обеспечению безопасности информации, подходы к решению которых чаще всего в начале также имеют описательный характер.

Таким образом, налицо два основных подхода (неформальный или описательный и формальный) к рассмотрению вопросов теории компьютерной безопасности.

1.2. Основные понятия и определения

В настоящее время понятие "информация" трактуется, пожалуй, наиболее широко – от философски обобщенного до бытового, в зависимости от целей, которые ставит перед собой автор. Так, В. И. Шаповалов дает такое определение: "Информация об объекте есть изменение параметра наблюдателя,

вызванное взаимодействием наблюдателя с объектом". С другой стороны, к определению информации существуют подходы, предложенные Шенноном и связанные с ее количественным измерением. Эти подходы излагаются в теории информации, где "информация определяется только вероятностными свойствами сообщений. Все другие их свойства, например, полезность для тех или других действий, принадлежность тому или иному автору и др. игнорируются".

Далее мы под информацией будем понимать сведения: о фактах, событиях, процессах и явлениях, о состоянии объектов (их свойствах, характеристиках) в некоторой предметной области, воспринимаемые человеком или специальным устройством и используемые (необходимые) для оптимизации принимаемых решений в процессе управления данными объектами. Информация может существовать в различных формах в виде совокупностей некоторых знаков (символов, сигналов и т. п.) на носителях различных типов. В связи с развивающимся процессом информатизации общества все большие объемы информации накапливаются, хранятся и обрабатываются в автоматизированных системах, построенных на основе современных средств вычислительной техники и связи. В дальнейшем будут рассматриваться только те формы представления информации, которые используются при ее автоматизированной обработке.

Под автоматизированной системой обработки информации (АС) будем понимать организационно-техническую систему, представляющую собой совокупность следующих взаимосвязанных компонентов:

– технических средств обработки и передачи данных (средств вычислительной техники и связи);

– методов и алгоритмов обработки в виде соответствующего программного обеспечения;

– информации (массивов, наборов, баз данных) на различных носителях;

– персонала и пользователей системы, объединенных по организационно-структурному, тематическому, технологическому или другим при знакам для выполнения автоматизированной обработки информации (данных) с целью удовлетворения информационных потребностей субъектов информационных отношений.

Одной из особенностей обеспечения информационной безопасности в АС является то, что абстрактным понятиям, таким как "субъект доступа", "информация" и т.п. (см. главу 3), ставятся в соответствие физические представления в среде вычислительной техники:

– для представления "субъектов доступа" – активные программы и процессы,

– для представления информации – машинные носители информации в виде внешних устройств компьютерных систем (терминалов, печатающих устройств, различных накопителей, линий и каналов связи), томов, разделов и подразделов томов, файлов, записей, полей записей, оперативной памяти и т.д.

Информационная безопасность АС-состояние рассматриваемой автоматизированной системы, при котором она, с одной стороны, способна противостоять дестабилизирующему воздействию внешних и внутренних информационных угроз, а с другой – ее наличие и функционирование не создает информационных угроз для элементов самой системы и внешней среды.

Тема 2. Анализ угроз информационной безопасности

Под угрозой (вообще) обычно понимают потенциально возможное событие, действие (воздействие), процесс или явление, которое может привести к нанесению ущерба чьим-либо интересам. В дальнейшем изложении угрозой информационной безопасности АС будем называть возможность реализации воздействия на информацию, обрабатываемую в АС, приводящего к искажению, уничтожению, копированию, блокированию доступа к информации, а также возможность воздействия на компоненты АС, приводящего к утрате, уничтожению или сбою функционирования носителя информации, средства взаимодействия с носителем или средства его управления.

В настоящее время рассматривается достаточно обширный перечень угроз информационной безопасности АС, насчитывающий сотни пунктов. Перечислим наиболее характерные и часто реализуемые из них:

- несанкционированное копирование носителей информации;
- неосторожные действия, приводящие к разглашению конфиденциальной информации, или делающие ее общедоступной;
- игнорирование организационных ограничений (установленных правил) при определении ранга системы.

Задание возможных угроз информационной безопасности проводится с целью определения полного перечня требований к разрабатываемой системе защиты. Перечень угроз, оценки вероятностей их реализации, а также модель нарушителя служат основой для анализа риска реализации угроз и формулирования требований к системе защиты АС. Кроме выявления возможных угроз должен быть проведен анализ этих угроз на основе их классификации по ряду признаков. Каждый из признаков классификации отражает одно из обобщенных требований к системе защиты. При этом угрозы, соответствующие каждому признаку классификации, позволяют детализировать отражаемое этим признаком требование.

Необходимость классификации угроз информационной безопасности АС обусловлена тем, что архитектура современных средств автоматизированной обработки информации, организационное, структурное и функциональное построение информационно-вычислительных систем и сетей, технологии и условия автоматизированной обработки информации такие, что накапливаемая, хранимая и обрабатываемая информация подвержена случайным влияниям чрезвычайно большого числа факторов. В силу этого становится невозможным формализовать задачу описания полного множества угроз. Как следствие, для защищаемой системы определяют не полный перечень угроз, а перечень классов угроз.

Классификация всех возможных угроз информационной безопасности АС может быть проведена по ряду базовых признаков.

1. По природе возникновения:
    1. Естественные угрозы-угрозы, вызванные воздействиями на АС и ее компоненты объективных физических процессов или стихийных природных явлений, независящих от человека.
    2. Искусственные угрозы – угрозы информационной безопасности АС, вызванные деятельностью человека.
2. По степени преднамеренности проявления:
    1. Угрозы случайного действия и/или угрозы, вызванные ошибками или халатностью персонала. Например:
        - проявление ошибок программно-аппаратных средств АС;
        - некомпетентное использование, настройка или неправомерное отключение средств защиты персоналом службы безопасности;
        - неумышленные действия, приводящие к частичному или полному отказу системы или разрушению аппаратных, программных, информационных ресурсов системы (неумышленная порча оборудования, удаление, искажение файлов с важной информацией или программ, в том числе системных и т.п.);
        - неправомерное включение оборудования или изменение режимов работы устройств и программ;
        - неумышленная порча носителей информации;
        - пересылка данных по ошибочному адресу абонента (устройства);
        - ввод ошибочных данных;
        - неумышленное повреждение каналов связи.
    2. Угрозы преднамеренного действия (например, угрозы действий злоумышленника для хищения информации).
3. По непосредственному источнику угроз:
    1. Угрозы, непосредственным источником которых является природная среда (стихийные бедствия, магнитные бури, радиоактивное излучение и т. п.).
    2. Угрозы, непосредственным источником которых является человек. Например:

- • внедрение агентов в число персонала системы (в том числе, возможно, и в административную группу, отвечающую за безопасность);
- • вербовка (путем подкупа, шантажа и т.п.) персонала или отдельных пользователей, имеющих определенные полномочия;
- • угроза несанкционированного копирования секретных данных пользователем АС;
- • разглашение, передача или утрата атрибутов разграничения доступа(паролей, ключей шифрования, идентификационных карточек, пропусков и т. п.).

3. Угрозы, непосредственным источником которых являются санкционированные программно-аппаратные средства. Например:
   - • запуск технологических программ, способных при некомпетентном использовании вызывать потерю работоспособности системы (зависания или зацикливания) или необратимые изменения в системе (форматирование или реструктуризацию носителей информации, удаление данных и т. п.);
   - • возникновение отказа в работе операционной системы.
4. Угрозы, непосредственным источником которых являются несанкционированные программно-аппаратные средства. Например:
   - • нелегальное внедрение и использование неучтенных программ (игровых, обучающих, технологических и др., не являющихся необходимыми для выполнения нарушителем своих служебных обязанностей) с последующим необоснованным расходованием ресурсов (загрузка процессора, захват оперативной памяти и памяти на внешних носителях);
   - • заражение компьютера вирусами с деструктивными функциями.

4. По положению источника угроз:
   1. Угрозы, источник которых расположен вне контролируемой зоны территории (помещения), на которой находится АС. Например:
      - • перехват побочных электромагнитных, акустических и других излучений устройств и линий связи, а также наводок активных излучений на вспомогательные технические средства, непосредственно не участвующие в обработке

информации (телефонные линии, сети питания, отопления и т. п.);

- • перехват данных, передаваемых по каналам связи, и их анализ с целью выяснения протоколов обмена, правил вхождения в связь и авторизации пользователя и последующих попыток их имитации для проникновения в систему;
- • дистанционная фото- и видеосъемка.

2. Угрозы, источник которых расположен в пределах контролируемой зоны территории (помещения), на которой находится АС. Например:

- • хищение производственных отходов (распечаток, записей, списанных носителей информации и т.п.);
- • отключение или вывод из строя подсистем обеспечения функционирования вычислительных систем (электропитания, охлаждения и вентиляции, линий связи и т.д.);
- • применение подслушивающих устройств.

3. Угрозы, источник которых имеет доступ к периферийным устройствам АС (терминалам).

4. Угрозы, источник которых расположен в АС. Например:

- • проектирование архитектуры системы и технологии обработки данных, разработка прикладных программ, которые представляют опасность для работоспособности системы и безопасности информации;
- • некорректное использование ресурсов АС.

5. По степени зависимости от активности АС:

1. Угрозы, которые могут проявляться независимо от активности АС. Например:

- • вскрытие шифров криптозащиты информации;
- • хищение носителей информации (магнитных дисков, лент, микросхем памяти, запоминающих устройств и компьютерных систем).

2. Угрозы, которые могут проявляться только в процессе автоматизированной обработки данных (например, угрозы выполнения и распространения программных вирусов).

6. По степени воздействия на АС:

1. Пассивные угрозы, которые при реализации ничего не меняют в структуре и содержании АС (например, угроза копирования секретных данных).

2. Активные угрозы, которые при воздействии вносят изменения в структуру и содержание АС. Например:

- • внедрение аппаратных спецвложений, программных "закладок" и "вирусов" ("троянских коней" и "жучков"), т.е. таких участков программ, которые не нужны для выполнения заявленных функций, но позволяют преодолеть систему защиты, скрытно и незаконно осуществить доступ к системным ресурсам с целью регистрации и передачи критической информации или дезорганизации функционирования системы;
- • действия по дезорганизации функционирования системы (изменение режимов работы устройств или программ, забастовка, саботаж персонала, постановка мощных активных радиопомех на частотах работы устройств системы и т.п.);
- • угроза умышленной модификации информации.

7. По этапам доступа пользователей или программ к ресурсам АС:

1. Угрозы, которые могут проявляться на этапе доступа к ресурсам АС(например, угрозы несанкционированного доступа в АС).

2. Угрозы, которые могут проявляться после разрешения доступа к ресурсам АС (например, угрозы несанкционированного или некорректного использования ресурсов АС).

8. По способу доступа к ресурсам АС:

1. Угрозы, направленные на использование прямого стандартного пути доступа к ресурсам АС. Например:

- • незаконное получение паролей и других реквизитов разграничения доступа (агентурным путем, используя халатность пользователей, подбором, имитацией интерфейса системы и т.д.) с последующей маскировкой под зарегистрированного пользователя ("маскарад");
- • несанкционированное использование терминалов пользователей, имеющих уникальные физические характеристики, такие как номер рабочей станции в сети, физический адрес, адрес в системе связи, аппаратный блок кодирования и т.п.

2. Угрозы, направленные на использование скрытого нестандартного пути доступа к ресурсам АС. Например:

• • вход в систему в обход средств защиты (загрузка посторонней операционной системы со сменных магнитных носителей и т.п.);

• • угроза несанкционированного доступа к ресурсам АС путем использования недокументированных возможностей ОС.

9. По текущему месту расположения информации, хранимой и обрабатываемой в АС:

1. Угрозы доступа к информации на внешних запоминающих устройствах(например, угроза несанкционированного копирования секретной информации с жесткого диска).

2. Угрозы доступа к информации в оперативной памяти. Например:

• • чтение остаточной информации из оперативной памяти;

• • чтение информации из областей оперативной памяти, используемых операционной системой (в том числе подсистемой защиты) или другими пользователями, в асинхронном режиме, используя недостатки мультизадачных АС и систем программирования;

• • угроза доступа к системной области оперативной памяти со стороны прикладных программ.

3. Угрозы доступа к информации, циркулирующей в линиях связи. Например:

• • незаконное подключение к линиям связи с целью работы "между строк", с использованием пауз в действиях законного пользователя от его имени с последующим вводом ложных сообщений или модификацией передаваемых сообщений;

• • незаконное подключение к линиям связи с целью прямой подмены законного пользователя путем его физического отключения после входа в систему и успешной аутентификации с последующим вводом дезинформации и навязыванием ложных сообщений;

• • перехват всего потока данных с целью дальнейшего анализа не в реальном масштабе времени.

4. Угрозы доступа к информации, отображаемой на терминале или печатаемой на принтере (например, угроза записи отображаемой информации на скрытую видеокамеру).

Вне зависимости от конкретных видов угроз или их проблемно-ориентированной классификации АС удовлетворяет потребности эксплуатирующих ее лиц, если обеспечиваются следующие свойства информации и систем ее обработки.

Конфиденциальность информации – субъективно определяемая (приписываемая) характеристика (свойство) информации, указывающая на необходимость введения ограничений на круг субъектов, имеющих доступ к данной информации, и обеспечиваемая способностью системы (среды) сохранять указанную информацию в тайне от субъектов, не имеющих полномочий доступа к ней.

Объективные предпосылки подобного ограничения доступности информации для одних субъектов заключены в необходимости защиты их законных интересов от других субъектов информационных отношений.

Целостность информации – существование информации в неискаженном виде (неизменном по отношению к некоторому фиксированному ее состоянию).

Точнее говоря, субъектов интересует обеспечение более широкого свойства-достоверности информации, которое складывается из адекватности (полноты и точности) отображения состояния предметной области и непосредственно целостности информации, т.е. ее неискаженное™. Однако мы ограничимся только рассмотрением вопросов обеспечения целостности информации, так как вопросы обеспечения адекватности отображения выходят далеко за рамки проблемы обеспечения информационной безопасности.

Доступность информации – свойство системы (среды, средств и технологии обработки), в которой циркулирует информация, характеризующееся способностью обеспечивать своевременный беспрепятственный доступ субъектов к интересующей их информации и готовность соответствующих автоматизированных служб к обслуживанию поступающих от субъектов запросов всегда, когда в обращении к ним возникает необходимость.

Таким образом, в соответствии с существующими подходами, принято считать, что информационная безопасность АС обеспечена в случае, если для любых информационных ресурсов в системе поддерживается определенный уровень конфиденциальности (невозможности несанкционированного получения какой-либо информации), целостности (невозможности несанкционированной или случайной ее модификации) и доступности

(возможности за разумное время получить требуемую информацию). Соответственно для автоматизированных систем было предложено рассматривать три основных вида угроз.

Угроза нарушения конфиденциальности заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней. В терминах компьютерной безопасности угроза нарушения конфиденциальности имеет место всякий раз, когда получен доступ к некоторой секретной информации, хранящейся в вычислительной системе или передаваемой от одной системы к другой. Иногда, в связи с угрозой нарушения конфиденциальности, используется термин "утечка".

Угроза нарушения целостности включает в себя любое умышленное изменение информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую. Когда злоумышленники преднамеренно изменяют информацию, говорится, что целостность информации нарушена. Целостность также будет нарушена, если к несанкционированному изменению приводит случайная ошибка программного или аппаратного обеспечения. Санкционированными изменениями являются те, которые сделаны уполномоченными лицами с обоснованной целью (например, санкционированным изменением является периодическая запланированная коррекция некоторой базы данных).

Угроза отказа служб возникает всякий раз, когда в результате преднамеренных действий, предпринимаемых другим пользователем или злоумышленником, блокируется доступ к некоторому ресурсу вычислительной системы. Реально блокирование может быть постоянным – запрашиваемый ресурс никогда не будет получен, или оно может вызывать только задержку запрашиваемого ресурса, достаточно долгую для того, чтобы он стал бесполезным. В этих случаях говорят, что ресурс исчерпан.

Данные виды угроз можно считать первичными или непосредственными, так как если рассматривать понятие угрозы как некоторой потенциальной опасности, реализация которой наносит ущерб информационной системе, то реализация вышеперечисленных угроз приведет к непосредственному воздействию на защищаемую информацию. В то же время непосредственное воздействие на информацию возможно для атакующей стороны в том случае, если система, в которой циркулирует информация, для нее "прозрачна", т. е. не существует никаких систем защиты или других препятствий. Описанные выше угрозы были сформулированы в 60-х годах для открытых UNIX-подобных систем, где не предпринимались меры по защите информации.

На современном этапе развития информационных технологий подсистемы или функции защиты являются неотъемлемой частью комплексов

по обработке информации. Информация не представляется "в чистом виде", на пути к ней имеется хотя бы какая-нибудь система защиты, и поэтому чтобы угрожать, скажем, нарушением конфиденциальности, атакующая сторона должна преодолеть эту систему. Однако не существует абсолютно стойкой системы защиты, вопрос лишь во времени и средствах, требующихся на ее преодоление. Исходя из данных условий, примем следующую модель: защита информационной системы считается преодоленной, если в ходе ее исследования определены все уязвимости системы. Поскольку преодоление защиты также представляет собой угрозу, для защищенных систем будем рассматривать ее четвертый вид-угрозу раскрытия параметров АС, включающей в себя систему защиты. С точки зрения практики любое проводимое мероприятие предваряется этапом разведки, в ходе которого определяются основные параметры системы, ее характеристики и т. п. Результатом этого этапа является уточнение поставленной задачи, а также выбор наиболее оптимального технического средства.

Угрозу раскрытия можно рассматривать как опосредованную. Последствия ее реализации не причиняют какой-либо ущерб обрабатываемой информации, но дают возможность реализоваться первичным или непосредственным угрозам, перечисленным выше. Введение данного вида угроз позволяет описывать отличия защищенных информационных систем от открытых. Для последних угроза разведки параметров системы считается реализованной.

Для научного обоснования введения четвертого вида угроз вкратце рассмотрим следующую модель. Пусть существует информационная система W. Под информационной системой будем понимать систему, осуществляющую получение входных данных; обработку этих данных и/или изменение собственного внутреннего состояния (внутренних связей/отношений); выдачу результата либо изменение своего внешнего состояния (внешних связей/отношений)

Для функционирования система W использует собственную модель M_W, оптимизируя по соотношению затрат на управление системой и его качества. Одним из параметров модели является объем информационных ресурсов. Под информационными ресурсами будем понимать фактические сведения, отражающие восприятие как самих себя, так и окружающего мира, хранимые информационной системой. Говоря другими словами, информационный ресурс-это база знаний информационной системы.

Рассматриваемой системе W противодействует аналогичная система T (противник), также функционирующая на основе некоторой собственной модели M_T. Между системами осуществляется информационное

противоборство. Отличие информационного противоборства от остальных его видов заключается в том, что оно влияет на информационные ресурсы противоборствующих сторон. При информационном противоборстве системы помимо собственных моделей строят модели противоборствующих сторон: $M_T(W)$ и $M_W(T)$ для систем W и T соответственно.

При информационном взаимодействии отсутствует непосредственный контакт двух систем. Вся получаемая и отдаваемая информация передается от одной системы к другой по информационному каналу. Информационный канал является еще одной составляющей модели информационного взаимодействия двух систем и описывает совокупность средств и сред, осуществляющих передачу информационных ресурсов. Схема взаимодействия двух систем с использованием информационного канала показана на рисунке 2.1.

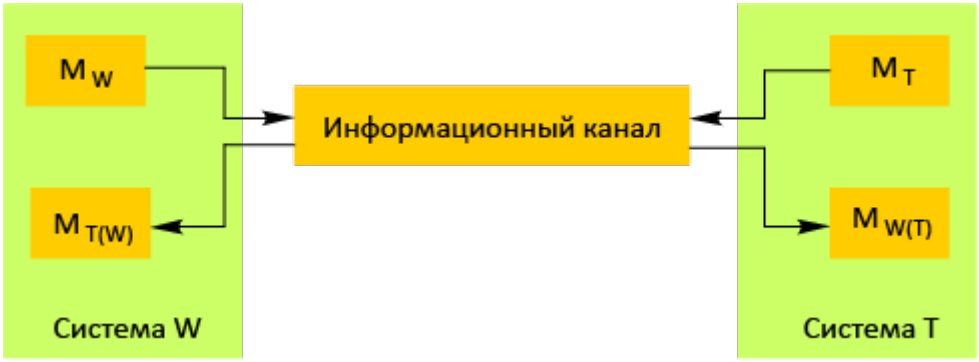

Рисунок 2.1 – Взаимодействие информационных систем W и T через информационный канал

Сопоставим вышеперечисленные виды угроз с моделью информационного противоборства двух систем. Исходя из симметричности модели сопоставление можно вести относительно любой из систем. Пусть это будет система W. Угрозе нарушения конфиденциальности системы W соответствует возможность системы T добавлять информационные ресурсы системы W к собственным информационным ресурсам, используя для этого передачу по информационному каналу. Угрозе нарушения целостности системы W соответствует возможность системы T внедрять собственные информационные ресурсы в информационные ресурсы системы W, используя для этого передачу по информационному каналу. Угрозе отказа служб системы W соответствует возможность системы T разорвать существующий информационный канал. Угрозе разведки параметров системы W соответствует возможность системы T организовать информационный канал с целью реализации угрозы нарушения конфиденциальности и нарушения целостности.

Таким образом, существование угрозы разведки параметров системы получает свое подтверждение с формальной точки зрения на основании модели информационного противоборства двух информационных систем.

Тема 3. Структуризация методов обеспечения информационной безопасности

3.1. Обзор

При рассмотрении вопросов защиты АС часто используют четырехуровневую градацию доступа к хранимой, обрабатываемой и защищаемой АС информации. Она систематизирует как возможные угрозы, так и меры по их нейтрализации и парированию. Эти уровни следующие:

- уровень носителей информации;
- уровень средств взаимодействия с носителем;
- уровень представления информации;
- уровень содержания информации.

Данные уровни были введены исходя из того, что, во-первых, информация для удобства манипулирования чаще всего фиксируется на некотором материальном носителе, которым может быть бумага, дискета или что-нибудь в этом роде. Во-вторых, если способ представления информации таков, что она не может быть непосредственно воспринята человеком, возникает необходимость в преобразователях информации в доступный для человека способ представления. Например, для чтения информации с дискеты необходим компьютер, оборудованный дисководом соответствующего типа. В-третьих, как уже было отмечено, информация может быть охарактеризована способом своего представления или тем, что еще называется языком в обиходном смысле. Язык жестов, язык символов и т.п.-все это способы представления информации. В-четвертых, человеку должен быть доступен смысл представленной информации, ее семантика.

Защита носителей информации должна обеспечивать парирование всех возможных угроз, направленных как на сами носители, так и на зафиксированную на них информацию, представленную в виде изменения состояний отдельных участков, блоков, полей носителя. Применительно к АС защита носителей информации в первую очередь подразумевает защиту машинных носителей. Вместе с тем, необходимо учитывать, что носителями информации являются также каналы связи, документальные материалы, получаемые в ходе эксплуатации АС, и т.п. Защита средств взаимодействия с носителем охватывает спектр методов защиты программно-аппаратных средств, входящих в состав АС, таких как средства вычислительной техники, операционная система, прикладные программы. В основном защита на данном уровне рассматривается как защита от несанкционированного доступа,

обеспечивающая разграничение доступа пользователей к ресурсам системы. Защита представления информации, т.е. некоторой последовательности символов, обеспечивается средствами криптографической защиты. Защита содержания информации обеспечивается семантической защитой данных.

3.2. Основные методы реализации угроз информационной безопасности

К основным направлениям реализации злоумышленником информационных угроз относятся:

- непосредственное обращение к объектам доступа;
- создание программных и технических средств, выполняющих обращение к объектам доступа в обход средств защиты;
- модификация средств защиты, позволяющая реализовать угрозы информационной безопасности;
- внедрение в технические средства АС программных или технических механизмов, нарушающих предполагаемую структуру и функции АС.

К числу основных методов реализации угроз информационной безопасности АС относятся:

- определение злоумышленником типа и параметров носителей информации;
- получение злоумышленником информации о программно-аппаратной среде, типе и параметрах средств вычислительной техники, типе и версии операционной системы, составе прикладного программного обеспечения;
- получение злоумышленником детальной информации о функциях, выполняемых АС;
- получение злоумышленником данных о применяемых системах защиты;
- определение способа представления информации.

3.3. Основные принципы обеспечения информационной безопасности в автоматизированных системах

Для защиты АС на основании руководящих документов Гостехкомисии могут быть сформулированы следующие положения.

1. Информационная безопасность АС основывается на положениях и требованиях существующих законов, стандартов и нормативно-методических документов.
2. Информационная безопасность АС обеспечивается комплексом программно-технических средств и поддерживающих их организационных мер.
3. Информационная безопасность АС должна обеспечиваться на всех технологических этапах обработки информации и во всех режимах функционирования, в том числе при проведении ремонтных и регламентных работ.
4. Программно-технические средства защиты не должны существенно ухудшать основные функциональные характеристики АС (надежность, быстродействие, возможность изменения конфигурации АС).
5. Неотъемлемой частью работ по информационной безопасности является оценка эффективности средств защиты, осуществляемая по методике, учитывающей всю совокупность технических характеристик оцениваемого объекта, включая технические решения и практическую реализацию средств защиты.
6. Защита АС должна предусматривать контроль эффективности средств защиты. Этот контроль может быть периодическим либо инициироваться по мере необходимости пользователем АС или контролирующим органом.

Рассмотренные подходы могут быть реализованы при обеспечении следующих основных принципов:

- системности;
- комплексности;
- непрерывности защиты;
- разумной достаточности;
- гибкости управления и применения;
- открытости алгоритмов и механизмов защиты;
- простоты применения защитных мер и средств.

Принцип системности

Системный подход к защите компьютерных систем предполагает необходимость учета всех взаимосвязанных, взаимодействующих и изменяющихся во времени элементов, условий и факторов:

- при всех видах информационной деятельности и информационного проявления;
- во всех структурных элементах;
- при всех режимах функционирования;
- на всех этапах жизненного цикла;
- с учетом взаимодействия объекта защиты с внешней средой.

При обеспечении информационной безопасности АС необходимо учитывать все слабые, наиболее уязвимые места системы обработки информации, а также характер, возможные объекты и направления атак на систему со стороны нарушителей (особенно высококвалифицированных злоумышленников), пути проникновения в распределенные системы и несанкционированного доступа (НСД) к информации. Система защиты должна строиться не только с учетом всех известных каналов проникновения, но и с учетом возможности появления принципиально новых путей реализации угроз безопасности.

Принцип комплексности

В распоряжении специалистов по компьютерной безопасности имеется широкий спектр мер, методов и средств защиты компьютерных систем. В частности, современные средства вычислительной техники, операционные системы, инструментальные и прикладные программные средства обладают теми или иными встроенными элементами защиты. Комплексное их использование предполагает согласование разнородных средств при построении целостной системы защиты, перекрывающей все существенные каналы реализации угроз и не содержащей слабых мест на стыках отдельных ее компонентов.

Принцип непрерывности защиты

Защита информации – это не разовое мероприятие и даже не конкретная совокупность уже проведенных мероприятий и установленных средств защиты, а непрерывный целенаправленный процесс, предполагающий принятие соответствующих мер на всех этапах жизненного цикла АС (начиная с самых ранних стадий проектирования, а не только на этапе ее эксплуатации). Разработка системы защиты должна вестись параллельно с разработкой самой защищаемой системы. Это позволит учесть требования безопасности при проектировании архитектуры и, в конечном счете, позволит создать более эффективные (как по затратам ресурсов, так и по стойкости) защищенные

системы. Большинству физических и технических средств защиты для эффективного выполнения своих функций необходима постоянная организационная (административная) поддержка (своевременная смена и обеспечение правильного хранения и применения имен, паролей, ключей шифрования, переопределение полномочий и т.п.). Перерывы в работе средств защиты могут быть использованы злоумышленниками для анализа применяемых методов и средств защиты, внедрения специальных программных и аппаратных "закладок" и других средств преодоления системы защиты после восстановления ее функционирования.

Разумная достаточность

Создать абсолютно непреодолимую систему защиты принципиально невозможно: при достаточных времени и средствах можно преодолеть любую защиту. Например, средства криптографической защиты в большинстве случаев не гарантируют абсолютную стойкость, а обеспечивают конфиденциальность информации при использовании для дешифрования современных вычислительных средств в течение приемлемого для защищающейся стороны времени. Поэтому имеет смысл вести речь только о некотором приемлемом уровне безопасности. Высокоэффективная система защиты стоит дорого, использует при работе существенную часть мощности и ресурсов компьютерной системы и может создавать ощутимые дополнительные неудобства пользователям. Важно правильно выбрать тот достаточный уровень зашиты, при котором затраты, риск и размер возможного ущерба были бы приемлемыми (задача анализа риска).

Гибкость системы защиты

Часто приходится создавать систему защиты в условиях большой неопределенности. Поэтому принятые меры и установленные средства защиты, особенно в начальный период их эксплуатации, могут обеспечивать как чрезмерный, так и недостаточный уровень защиты. Естественно, что для обеспечения возможности варьирования уровнем защищенности средства защиты должны обладать определенной гибкостью. Особенно важно это свойство в тех случаях, когда средства защиты необходимо устанавливать на работающую систему, не нарушая процесс ее нормального функционирования. Кроме того, внешние условия и требования с течением времени меняются. В таких ситуациях свойство гибкости спасает владельцев АС от необходимости принятия кардинальных мер по полной замене средств защиты на новые.

Открытость алгоритмов и механизмов защиты

Суть принципа открытости алгоритмов и механизмов защиты состоит в том, что защита не должна обеспечиваться только за счет секретности структурной организации и алгоритмов функционирования ее подсистем.

Знание алгоритмов работы системы защиты не должно давать возможности ее преодоления (даже автору). Однако это вовсе не означает, что информация о конкретной системе защиты должна быть общедоступна – необходимо обеспечивать защиту от угрозы раскрытия параметров системы.

Принцип простоты применения средств защиты

Механизмы защиты должны быть интуитивно понятны и просты в использовании. Применение средств защиты не должно быть связано со знанием специальных языков или с выполнением действий, требующих значительных дополнительных трудозатрат при обычной работе законных пользователей, а также не должно требовать от пользователя выполнения рутинных малопонятных ему операций (ввод нескольких паролей и имен и т.д.).

3.4. Причины, виды и каналы утечки информации

Основными причинами утечки информации являются:

- несоблюдение персоналом норм, требований, правил эксплуатации
- ошибки в проектировании АС и систем защиты АС;
- ведение противостоящей стороной технической и агентурной разведок.

Несоблюдение персоналом норм, требований, правил эксплуатации АС может быть как умышленным, так и непреднамеренным. От ведения противостоящей стороной агентурной разведки этот случай отличает то, что в данном случае лицом, совершающим несанкционированные действия, двигают личные побудительные мотивы. Причины утечки информации достаточно тесно связаны с видами утечки информации.

В соответствии с ГОСТ Р 50922-96 рассматриваются три вида утечки информации:

1. разглашение;
2. несанкционированный доступ к информации;
3. получение защищаемой информации разведками (как отечественными, так и иностранными).

Разглашение информации – несанкционированное доведение защищаемой информации до потребителей, не имеющих права доступа к защищаемой информации.

Несанкционированный доступ – получение защищаемой информации заинтересованным субъектом с нарушением установленных правовыми документами или собственником, владельцем информации прав или правил доступа к защищаемой информации.

При этом заинтересованным субъектом, осуществляющим несанкционированный доступ к информации, может быть: государство,

юридическое лицо, группа физических лиц, в том числе общественная организация, отдельное физическое лицо.

Получение защищаемой информации разведками может осуществляться с помощью технических средств (техническая разведка) или агентурными методами (агентурная разведка).

Канал утечки информации – совокупность источника информации, материального носителя или среды распространения несущего указанную информацию сигнала и средства выделения информации из сигнала или носителя.

Одним из основных свойств канала является месторасположение средства выделения информации из сигнала или носителя, которое может располагаться в пределах контролируемой зоны, охватывающей АС, или вне ее.

Применительно к АС выделяют следующие каналы утечки:

1. Электромагнитный канал. Причиной его возникновения является электромагнитное поле, связанное с протеканием электрического тока в аппаратных компонентах АС. Электромагнитное поле может индуцировать токи в близко расположенных проводных линиях (наводки). Электромагнитный канал в свою очередь делится на следующие каналы:
   • радиоканал (высокочастотное излучение);
   • низкочастотный канал;
   • сетевой канал (наводки на сеть электропитания);
   • канал заземления (наводки на провода заземления);
   • линейный канал (наводки на линии связи между компьютерными системами).
2. Акустический (виброакустический) канал. Связан с распространением звуковых волн в воздухе или упругих колебаний в других средах, возникающих при работе устройств отображения информации АС.
3. Визуальный канал. Связан с возможностью визуального наблюдения злоумышленником за работой устройств отображения информации АС без проникновения в помещения, где расположены компоненты системы. В качестве средства выделения информации в данном случае могут рассматриваться фото-, видеокамеры и т.п.
4. Информационный канал. Связан с доступом (непосредственным и телекоммуникационным) к элементам АС, к носителям информации, к самой вводимой и выводимой информации (и результатам), к программному обеспечению (в том числе к операционным системам), а также с подключением к линиям связи. Информационный канал может быть разделен на следующие каналы:

- канал коммутируемых линий связи;

- канал выделенных линий связи;
- канал локальной сети;
- канал машинных носителей информации;
- канал терминальных и периферийных устройств.

Тема 4. Вопросы безопасности информации в компьютерных системах

4.1. Понятие политики безопасности

Рассматривая вопросы безопасности информации в компьютерных системах можно говорить о наличии некоторых "желательных" состояний данных систем. Эти желательные состояния (описанные в терминах модели собственно компьютерной системы, например, в терминах субъектно-объектной модели – см. ниже) описывают "защищенность" системы. Понятие "защищенности" принципиально не отличается от любых других свойств технической системы, например, "надежной работы", и является для системы внешним, априорно заданным. Особенностью понятия "защищенность" является его тесная связь с понятиями "злоумышленник" (как обозначение внешней причины для вывода системы из состояния "защищенности") или "угроза" (понятие, обезличивающее причину вывода системы из защищенного состояния из-за действий злоумышленника).

При рассмотрении понятия "злоумышленник" практически всегда выделяется объект его воздействия – часть системы, связанная с теми или иными действиями злоумышленника ("объект атаки"). Следовательно, можно выделить три компоненты, связанные с нарушением безопасности системы: "злоумышленник" – внешний по отношению к системе источник нарушения свойства "безопасность", "объект атаки" – часть, принадлежащая системе, на которую злоумышленник производит воздействие, "канал воздействия" – среда переноса злоумышленного воздействия.

Интегральной характеристикой, описывающей свойства защищаемой системы, является политика безопасности.

Политика безопасности – качественное (или качественно-количественное) описание свойств защищенности, выраженное в терминах, описывающих систему.

Описание политики безопасности может включать или учитывать свойства злоумышленника b объекта атаки.

Приведем пример описания политики безопасности. Наиболее час то рассматриваются политики безопасности, связанные с понятием, "доступ".

Доступ – категория субъектно-объектной модели (субъект – активная компонента системы, активность понимается как возможность выполнять операции над объектами – пассивной компонентой) описывающая процесс выполнения операций субъектов над объектами.

Описание политики безопасности включает:

1. Множество возможных операций над объектами.

2. Для каждой пары "субъект-объект" (S_i, O_j) назначение множества разрешенных операций, являющегося подмножеством всего множества возможных операций. Операции связаны обычно с целевой функций защищаемой системы (т.е. с категорией, описывающей назначение системы и решаемые задачи), например, операциями "создание объекта", "удаление объекта", "перенос информации от произвольного объекта к предопределенному – чтение" и т. д.

Можно сформулировать две аксиомы защищенных компьютерных систем (АС):

Аксиома 1

В защищенной АС всегда присутствует активная компонента (субъект), выполняющая контроль операций субъектов над объектами. Данная компонента фактически отвечает за реализации некоторой политики безопасности.

Аксиома 2

Для выполнения в защищенной АС операций над объектами необходима дополнительная информация (и наличие содержащего ее объекта) о разрешенных и запрещенных операциях субъектов с объектами.

В данном случае мы оперируем качественными понятиями "контроль", "разрешенная и запрещенная операция", данные понятия будут раскрыты и проиллюстрированы ниже.

Далее сформулирована дополнительная аксиома.

Аксиома 3

Все вопросы безопасности информации описываются доступами субъектов к объектам.

Важно заметить, что политика безопасности описывает в общем случае нестационарное состояние защищенности. Защищаемая система может изменяться, дополняться новыми компонентами (субъектами объектами, операциями субъектов над объектами). Очевидно, что политика безопасности должна быть поддержана во времени, следовательно, в процесс изучения свойств защищаемой системы должны быть добавлены процедуры управления безопасностью.

С другой стороны, нестационарность защищаемой АС, а также вопросы реализации политики безопасности в конкретных конструкциях защищаемой системы (например, программирование контролирующего субъекта в командах конкретного процессора компьютера) предопределяют необходимость рассмотрения задачи гарантирования заданной политики безопасности.

Итак, резюмируя, можно сказать, что компьютерная безопасность решает четыре класса взаимосвязанных задач:

1. Формулирование и изучение политик безопасности.
2. Реализация политик безопасности.
3. Гарантирование заданной политики безопасности. Управление безопасностью.

Типовой жизненный цикл АС состоит из следующих стадий:

1. Проектирование АС и проектирование политики безопасности.
2. Моделирование ПБ и анализ корректности ПБ, включающий установление адекватности политики безопасности и целевой функции АС.
3. Реализация ПБ и механизмов ее гарантирования, а также процедур и механизмов управления безопасностью.
4. Эксплуатация защищенной системы.

Безопасность АС достаточно часто описывается в категориях "достоверность", "конфиденциальность", "целостность" и "доступность".

Свойство достоверности понимается как сохранение информацией своих семантических свойств в любой момент времени от момента ввода в систему. Свойство доступности понимается как возможность пользования некоторый ресурсом АС и информацией в произвольный момент времени. Свойство целостности (связанное со свойством достоверности) подразумевает неизменность свойств информации и ресурсов в любой момент времени от момента их порождения или ввода в систему. Свойство конфиденциальности понимается как недоступность информации или сервисов для пользователей, который априорно не задана возможность использования указанных сервисов или информации (данных). Иногда выделяют также свойство актуальности информации, связанное со свойством доступности.

4.2. Понятие доступа и монитора безопасности

В теории компьютерной безопасности практически всегда рассматривается модель произвольной АС в виде конечного множества элементов. Сказанное множество можно разделить на два подмножества: множество объектов и множество субъектов. Данное разделение основано на свойстве элемента "быть активным" или "получать управление" (применяются также термины "использовать ресурсы" или "пользоваться вычислительной мощностью"). Оно исторически сложилось на основе модели вычислительной системы, принадлежащей фон Нейману, согласно которой последовательность

исполняемых инструкций (программа, соответствующая понятию "субъект") находится в единой среде с данными (соответствующими понятию "объект").

Модели, связанные с реализацией ПБ, не учитывают возможности субъектов по изменению АС, которые могут привести к изменению ее свойств и как предельный случай к полной неприменимости той или иной модели к описанию отношений "субъект-объект" в измененной АС.

Этот факт не является недостатком политики безопасности. Достоверность работы механизмов реализации политики безопасности считается априорно заданной, поскольку в противном случае невозможна формализация и анализ моделей. Однако вопрос гарантий политики безопасности является ключевым как в теории, так и в практике.

Рассматривая активную роль субъектов в АС, необходимо упомянуть о ряде важнейших их свойств, на которых базируется излагаемая ниже модель.

Во-первых, необходимо заметить, что человек-пользователь воспринимает объекты и получает информацию о состоянии АС через субъекты, которыми он управляет и которые производят отображение информации в воспринимаемом человеком виде.

Во-вторых, угрозы компонентам АС (АС рассматривается в модели потоков или состояний) исходят от субъектов как активной компоненты, порождающей потоки и изменяющей состояние объектов в АС.

В-третьих, субъекты могут влиять друг на друга через изменяемые ими объекты, связанные с другими субъектами, порождая в конечном итоге в системе субъекты (или состояния системы), которые представляют угрозу для безопасности информации или для работоспособности самой системы.

Будем считать разделение АС на субъекты и объекты априорным. Будем считать также, что существует априорный безошибочный критерий различения субъектов и объектов в АС (по свойству активности). Кроме того, считаем в условиях всех утверждений, что декомпозиция АС на субъекты и объекты фиксирована.

Подчеркнем отличие понятия субъекта компьютерной системы от человека-пользователя следующим определением.

Пользователь – лицо (физическое лицо), аутентифицируемое некоторой информацией и управляющее субъектом компьютерной системы через органы управления ЭВМ.

Пользователь АС является, таким образом, внешним фактором, управляющим состоянием субъектов. В связи с этим далее будем считать пользовательское управляющее воздействие таким, что свойства субъектов, сформулированные в ниже приводимых определениях, не зависят от него (т. е. свойства субъектов не изменяемы внешним управлением). Смысл данного

условия состоит в предположении того факта, что пользователь, управляющий программой, не может через органы управления изменить ее свойства (условие неверно для систем типа компиляторов, средств разработки, отладчиков и др.).

Будем также полагать, что в любой дискретный момент времени множество субъектов АС не пусто (в противном случае соответствующие моменты времени исключаются из рассмотрения и рассматриваются отрезки с ненулевой мощностью множества субъектов).

Аксиома 4

Субъекты в АС могут быть порождены только активной компонентой (субъектами) из объектов.

Специфицируем механизм порождения новых субъектов следующим определением.

Определение 1

Объект O_i, называется источником для субъекта S_m, если существует субъект S_j, в результате воздействия которого на объект O_i, в компьютерной системе возникает субъект S_m.

Субъект S_j, порождающий новый субъект из объекта O_i, в свою очередь, называется активизирующим субъектом для субъекта S_m, S_m назовем порожденным объектом.

Введем обозначение: $Create(S_j, O_i) \rightarrow S_k$ – из объекта O_i порожден субъект S_k при активизирующем воздействии субъекта S_j. *Create* назовем операцией порождения субъектов (см. рисунок 4.1).

Операция *Create* задает отображение декартова произведения множеств субъектов и объектов на объединение множества субъектов с пустым множеством. Заметим также, что в АС действует дискретное время и фактически новый субъект S_k порождается в момент времени $t+1$ относительно момента t, в который произошло воздействие порождающего субъекта на объект-источник.

Очевидно, что операция порождения субъектов зависит как от свойств активизирующего субъекта, так и от содержания объекта-источника.

Считаем, что если $Create(S_j,O_i) \rightarrow NULL$ (конструкция *NULL* далее обозначает пустое множество), то порождение нового субъекта из объекта O_i при активизирующем воздействии S_j невозможно. Так, практически во всех операционных средах существует понятие исполняемого файла – объекта, могущего быть источником для порождения субъект та. Например, для MS DOS файл edit.com является объектом-источником для порождения субъекта-программы текстового редактора, а порождающим субъектом является, как правило, командный интерпретатор shell (объект-источник – command.com).

Из архитектуры фон Неймана следует также, что с любым субъектом связан (или ассоциирован) некоторый объект (объекты), отображающий его состояние, – например, для активной программы (субъекта) ассоциированным объектом будет содержание участка оперативной памяти с исполняемым кодом данной программы.

Определение 2

Объект O_i в момент времени t ассоциирован с субъектом S_m, если состояние объекта O_i повлияло на состояние субъекта в следующий момент времени (т.е. субъект S_m использует информацию, содержащуюся в объекте O_i).

Введем обозначение «множество объектов $\{O_m\}t$ ассоциировано с субъектом S_i в момент времени t»: $S_i(\{O_m\}\ t)$.

В данном случае определение не в полной мере является формально строгим, поскольку состояние субъекта описывается упорядоченной совокупностью ассоциированных с ним объектов, а ассоциированный объект выделяется по принципу влияния на состояние субъекта, т. е. в определении прослеживается некая рекурсия. С другой стороны, известны рекурсивные определения различных объектов (например, дерева). Зависимость от времени позволяет однозначно выделять ассоциированные объекты в том случае, если в начальный момент ассоциированный объект можно определить однозначно (как правило, это вектор исполняемого кода и начальные состояния ряда переменных программы).

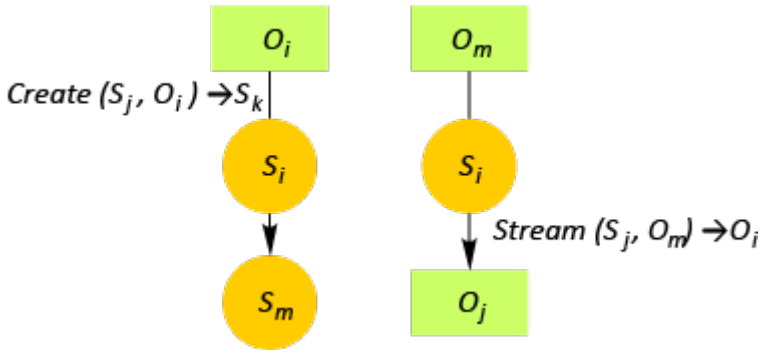

Рисунок 4.1 – Порождения субъекта и понятие потока

Субъект в общем случае реализует некоторое отображение множества ассоциированных объектов в t-ный момент времени на множество ассоциированных объектов в $t+1$-ный момент времени. В связи с этим можно выделить ассоциированные объекты, изменение которых изменяет вид отображения ассоциированных объектов (объекты, содержащие, как правило, код программы – функционально ассоциированные), и ассоциированные объекты-данные (являющиеся аргументом операции, но не изменяющие вида отображения). Далее под ассоциированными объектами понимаются функционально ассоциированные объекты, в иных случаях делаются уточнения.

Следствие (к определению 2)

В момент порождения субъекта S_m из объекта O_i он является ассоциированным объектом для субъекта S_m. Необходимо заметить, что объект-источник может быть ассоциированным для активизирующего субъекта, тогда порождение является автономным (т.е. не зависящим от свойств остальных субъектов и объектов). Если же объект-источник является не ассоциированным (внешним) для активизирующего субъекта, то порождение не является автономный и зависит от свойств объекта- источника.

Свойство субъекта «быть активным» реализуется и в возможности выполнения действий над объектами. При этом необходимо отметить, что пассивный статус объекта необходимо требует существования потоков информации от объекта к объекту (в противном случае невозможно говорить об изменении объектов), причем данный поток инициируется субъектом.

Определение 3

Потоком информации между объектом O_m и объектом O_j называется произвольная операция над объектом O_j, реализуемая в субъекте S_i и зависящая от O_m.

Заметим, что как O_j, так и O_m, могут быть ассоциированными или неассоциированными объектами, а также "пустыми" объектами (*NULL*).

Обозначения: $Stream(S_i \rightarrow O_m)\ O_j$ – поток информации от объекта O_m, к объекту O_j. При этом будем выделять источник (O_m) получатель (приемник) потока (O_j). В определении подчеркнуто, что поток информации рассматривается не между субъектом и объектом, а между объектами, например, объектом и ассоциированными объектами субъекта (либо между двумя объектами), а активная роль субъекта выражается в реализации данного потока (это означает, что операция порождения потока локализована в субъекте и отображается состоянием его функционально ассоциированных объектов). Отметим, что операция *Stream* может создавать новый объект или уничтожать его.

На рисунке 4.2 схематически изображены различные виды потоков.

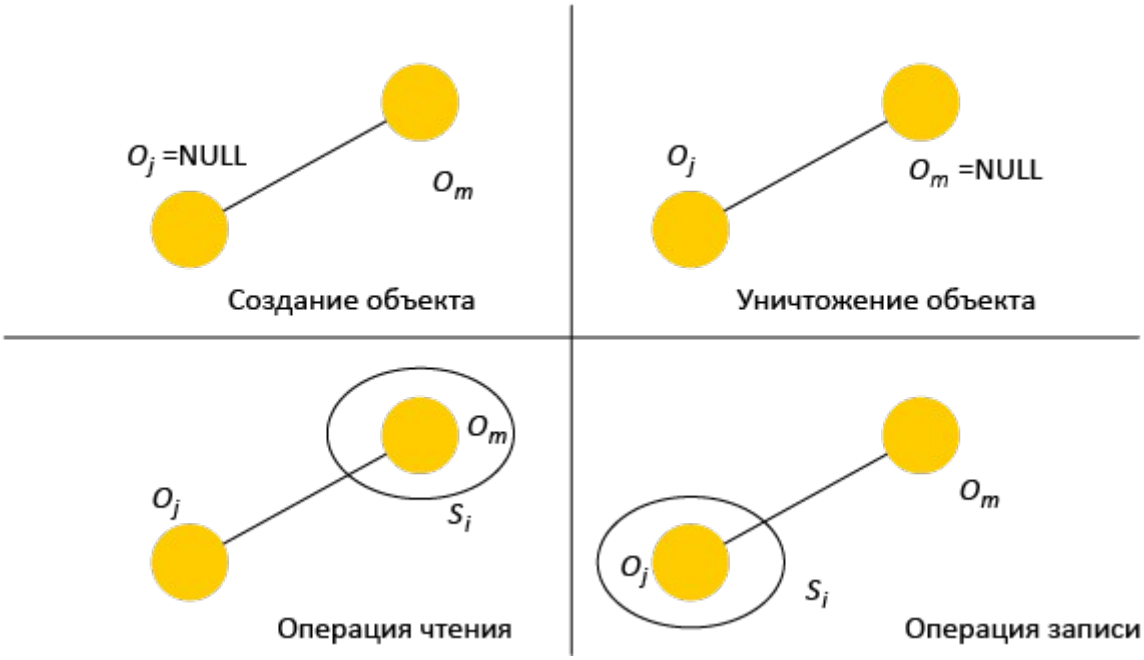

Рисунок 4.2 – Примеры потоков в АС

Далее будем для краткости говорить о потоке, подразумевая введенное понятие потока информации.

Понятие ассоциированных с субъектом объектов, как легко видеть из вышеизложенного, не является искусственной конструкцией. Корректно говорить о потоках информации можно лишь между одинаковыми сущностями, т. е. объектами. Кроме того, в ассоциированных объектах отображается текущее состояние субъекта. Отображениями *Stream* и *Create* описываются с точки зрения разделения на субъекты и объекты все события (изменения субъектов и объектов), происходящие в АС.

Из данного определения также следует, что поток всегда инициируется (порождается) субъектом.

Определение 4

Доступом субъекта S_i к объекту O_j будем называть порождение потока информации между некоторым объектом (например, ассоциированным с субъектом объектами $S_j(O_m)$) и объектом O_j.

Выделим все множество потоков P для фиксированной декомпозиции АС на субъекты и объекты во все моменты времени (все множество потоков является объединением потоков по всем моментам дискретного времени) и произвольным образом разобьем его на два не пересекающихся подмножества: N и L, $P – N\ U\ L$. Обозначим:

N – подмножество потоков, характеризующее несанкционированный доступ;

L – подмножество потоков, характеризующих легальный доступ.

Дадим некоторые пояснения к разделению множеств L и N. Понятие "безопасности" подразумевает наличие и некоторого состояния "опасности" – нежелательных состояний какой-либо системы (в данном случае АС). Будем считать парные категории типа "опасный-безопасный" априорно заданными для АС и описываемыми политикой безопасности, а результатом применения политики безопасности к АС – разделение на множество "опасных" потоков N и множество "безопасных" L. Деление на L и N может описывать как свойство целостности (потоки из N нарушают целостность АС) или свойство конфиденциальности (потоки из N нарушают конфиденциальность АС), так и любое другое произвольное свойство.

Определение 5

Правила разграничения доступа субъектов к объектам есть формально описанные потоки, принадлежащие подмножеству L.

В предлагаемой субъектно-ориентированной модели не производится уточнений известных моделей политик безопасности (политика безопасности описывает только критерии разбиения на множества L и N), но формулируются условия корректного существования элементов АС, обеспечивающих реализацию той или иной политики безопасности. Поскольку критерий разбиения на множества L и N не связан со следующими далее утверждениями (постулируется лишь наличие субъекта, реализующего фильтрацию потоков), то можно говорить об инвариантности субъектно-ориентированной модели относительно любой принятой в АС политики безопасности (не противоречащей условиям утверждений).

Определение 6

Объекты O_i и O_j тождественны в момент времени t, если они совпадают как слова, записанные в одном языке.

Например, при представлении в виде байтовых последовательностей объекты $O_1 = (O_{11}, O_{12}, \dots, O_{1m})$ и $O_2 = (O_{2i}, O_{22}, \dots, O_{2k})$ одинаковы, если $m = k$ и $O_{11} = O_{2i}$ для всех i от 1 до k (O_{ij} – байты).

Для введения понятия тождественности субъектов условимся о наличии процедуры сортировки ассоциированных объектов, которая позволяет говорить о возможности попарного сравнения. На практике всегда существует алгоритм, обеспечивающий возможность попарного сравнения и зависящий от конкретной архитектуры АС. Например, достаточно легко выделить и попарно сравнивать, например, участки оперативной памяти, отвечающие коду программ (отличающиеся абсолютным адресом загрузки в оперативную память) или содержанию переменных и массивов.

Определение 7

Субъекты S_j и S, тождественны в момент времени t, если попарно тождественны все ассоциированные с ними объекты.

Следствие (из определений 6 и 7)

Порожденные субъекты тождественны, если тождественны порождающие субъекты и объекты-источники.

Верность данного следствия вытекает из тождества функционально ассоциированных объектов в порождающих субъектах, которые отвечают за порождение нового субъекта, а также из тождества аргументов (ассоциированных объектов-данных), которые отвечают объектам-источникам.

Для разделения всего множества потоков в АС на подмножества L и N необходимо существование активной компоненты (субъекта), который:

1. активизировался бы при возникновении любого потока;
2. производил бы фильтрацию потоков в соответствии с принадлежностью множествам L или N.

Заметим, что если существует $Stream(S_j, O_j) \rightarrow O_m$ и существует $Stream(S_k, O_m) \rightarrow O_b$ то существует и $Stream((S_b S_k), O_j) \rightarrow O_b$, т.е. отношение "между объектами существует поток" является транзитивным (относительно пары субъектов). Именно в этом смысле будем говорить об участии субъекта (S_k) в потоке (если O_m, является ассоциированным объектом субъекта, не тождественного S_i). Введем несколько определений.

Определение 8

Монитор обращений (МО) – субъект, активизирующийся при возникновении потока от любого субъекта к любому объекту.

Можно выделить два вида МО:

Индикаторный МО – устанавливающий только факт обращения субъекта к объекту.

Содержательный МО – субъект, функционирующий таким образом, что при возникновении потока от ассоциированного объекта O_m любого субъекта S; ($S_i,(O_m)$) к объекту O_j и обратно существует ассоциированный с МО объект O_{mo} (в данном случае речь идет об ассоциированных объектах-данных), тождественный объекту O_m, или $S_j(O_m)$. Содержательный МО полностью участвует в потоке от субъекта к объекту (в том смысле, что информация проходит через его ассоциированные объекты-данные и существует тождественное отображение объекта на какой-либо ассоциированный объект МО).

Теперь сформулируем понятие монитора безопасности (в литературе также применяется понятие монитора ссылок). Это понятие связано с упоминаемой выше задачей фильтрации потоков. Поскольку целью является

обеспечение безопасности АС, то и целевая функция монитора – фильтрация с целью обеспечения безопасности (отметим еще раз, что разделение на N и L задано априорно).

Определение 9

Монитор безопасности объектов (МБО) – монитор обращений, который разрешает поток, принадлежащий только множеству легального доступа L. Разрешение потока в данном случае понимается как выполнение операции над объектом – получателем потока, а запрещение – как невыполнение (т.е. неизменность объекта – получателя потока).

Монитор безопасности объектов фактически является механизмом реализации политики безопасности в АС. Обратимся теперь к основным моделям работы МБО.

Тема 5. Разработка и реализация политики безопасности

5.1. Основные типы политики безопасности

Существуют два типа политики безопасности: дискреционная и мандатная.

Основой дискреционной (дискретной) политики безопасности является дискреционное управление доступом (Discretionary Access Control-DAC), которое определяется двумя свойствами:

- все субъекты и объекты должны быть идентифицированы;
- права доступа субъекта к объекту системы определяются на основании некоторого внешнего по отношению к системе правила.

К достоинствам дискреционной политики безопасности можно отнести относительно простую реализацию соответствующих механизмов защиты. Этим обусловлен тот факт, что большинство распространенных в настоящее время АС обеспечивают выполнение положений именно данной политики безопасности.

В качестве примера реализаций дискреционной политики безопасности в АС можно привести матрицу доступов, строки которой соответствуют субъектам системы, а столбцы – объектам; элементы матрицы характеризуют права доступа. К недостаткам относится статичность модели. Это означает, что данная политика безопасности не учитывает динамику изменений состояния АС, не накладывает ограничений на состояния системы.

Кроме этого, при использовании дискреционной политики безопасности возникает вопрос определения правил распространения прав доступа и анализа их влияния на безопасность АС. В общем случае при использовании данной политики безопасности перед монитором безопасности объектов (МБО), который при санкционировании доступа субъекта к объекту руководствуется некоторым набором правил, стоит алгоритмически неразрешимая задача: проверить приведут ли его действия к нарушению безопасности или нет.

В то же время имеются модели АС, реализующих дискреционную политику безопасности (например, модель Take-Grant), которые предоставляют алгоритмы проверки безопасности.

Так или иначе, матрица доступов не является тем механизмом, который бы позволил реализовать ясную и четкую систему защиты информации в АС. Этим обуславливается поиск других более совершенных политик безопасности.

Основу мандатной (полномочной) политики безопасности составляет мандатное управление доступом (Mandatory Access Control-MAC), которое подразумевает, что:

- все субъекты и объекты системы должны быть однозначно идентифицированы;
- задан линейно упорядоченный набор меток секретности;
- каждому объекту системы присвоена метка секретности, определяющая ценность содержащейся в нем информации – его уровень секретности в АС;
- каждому субъекту системы присвоена метка секретности, определяющая уровень доверия к нему в АС – максимальное значение метки секретности объектов, к которым субъект имеет доступ; метка секретности субъекта называется его уровнем доступа.

Основная цель мандатной политики безопасности – предотвращение утечки информации от объектов с высоким уровнем доступа к объектам с низким уровнем доступа, т.е. противодействие возникновению в АС информационных каналов сверху вниз.

Чаще всего мандатную политику безопасности описывают в терминах, понятиях и определениях свойств модели Белла-Лапалуда, которая будет рассмотрена в пункте 16.4. В рамках данной модели доказывается важное утверждение, указывающее на принципиальное отличие систем, реализующих мандатную защиту, от систем с дискреционной защитой: если начальное состояние системы безопасно, и все переводы системы из состояния в состояние не нарушают ограничений, сформулированных политикой безопасности, то любое состояние системы безопасно.

Кроме того, по сравнению с АС, построенными на основе дискреционной политики безопасности, для систем, реализующих мандатную политику, характерна более высокая степень надежности. Это связано с тем, что МБО такой системы должен отслеживать не только правила доступа субъектов системы к объектам, но и состояния самой АС. Таким образом, каналы утечки в системах данного типа не заложены в нее непосредственно (что мы наблюдаем в положениях предыдущей политики безопасности), а могут появиться только при практической реализации системы вследствие ошибок разработчика. В дополнении к этому правила мандатной политики безопасности более ясны и просты Для понимания разработчиками и пользователями АС, что также является фактором, положительно влияющим на уровень безопасности системы. С другой стороны, реализация систем с политикой безопасности данного типа довольно сложна и требует значительных ресурсов вычислительной системы.

5.2. Разработка и реализация политики безопасности

Воспользуемся определениями и обозначениями, данными при описании рассматриваемой в данной главе модели АС как совокупности взаимодействующих субъектов и объектов.

Представляется очевидным, что при изменении функционально ассоциированных с монитором безопасности объектов (МБО) могут измениться и свойства самого МБО, заключающиеся в фильтрации потоков, и, как следствие, могут возникнуть потоки, принадлежащие множеству N (см. рисунок 5.1). Введем в связи с этим понятие корректности субъектов.

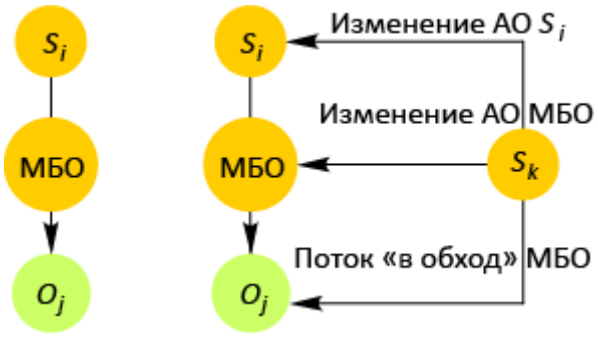

Рисунок 5.1 – Возможные пути нарушения политики безопасности (АО – ассоциированные объекты)

Определение 10

Субъекты S_i и S_j называются не влияющими друг на друга (или корректными относительно друг друга), если в любой момент времени отсутствует поток (изменяющий состояние объекта) между любыми объектами

O_i и O_j, ассоциированными соответственно с субъектами S_i и S_j. Причем O_j не является ассоциированным объектом S_i, а O_i – ассоциированным объектом S_j.

Дадим некоторые пояснения к определению: "изменение состояния объекта" трактуется в данном определении как не тождественность объектов в соответствующие моменты времени, но при этом подчеркнуто, что операция изменения объекта локализована в субъекте, с которым этот объект не ассоциирован. Смысл понятия корректности можно пояснить на примере: существующие в едином пространстве оперативной памяти (ОП) программы не должны иметь функциональных возможностей изменения "чужого" вектора кода и состояния переменных.

Вообще говоря, можно сформулировать более жесткое определение.

Определение 11

Субъекты S_i и S_j называются абсолютно не влияющими друг на друга (или абсолютно корректными относительно друг друга), если в условиях определения 10 множества ассоциированных объектов указанных субъектов не имеют пересечения.

Абсолютная корректность легко достижима в случае виртуального адресного пространства. Определение абсолютной корректности позволяет сформулировать достаточные условия гарантированного осуществления только легального доступа.

Утверждение 1 (достаточное условие 1 гарантированного выполнения политики безопасности в АС)

Монитор безопасности объектов разрешает порождение потоков только из множества L, если все существующие в системе субъекты абсолютно корректны относительно него и друг друга.

Доказательство

Условие абсолютной корректности (по определению 11) предполагает неизменность функционально ассоциированных объектов МБО (поскольку потоков, изменяющих ассоциированные объекты МБО, не существует). С другой стороны, такие потоки могут появиться при изменении ассоциированных объектов, принадлежащих другим субъектам АС (изменятся свойства субъекта, в том числе (возможно) и по порождению потоков к МБО). Условие корректности субъектов относительно друг друга делает это невозможным (по определению абсолютной корректности). Это в свою очередь означает, что МБО реализует только потоки из подмножества L Утверждение доказано.

Однако сформулированное утверждение накладывает весьма жесткие и трудноисполнимые условия на свойства субъектов в АС. Кроме того, невозможно гарантировать корректность любого субъекта, активизируемого в

АС, относительно МБО. В связи с этим логично ограничить множество порождаемых субъектов, которые априорно корректны относительно МБО. В связи с этим введем определение монитора порождения субъектов (по аналогии с монитором обращений) и монитора безопасности субъектов.

Определение 12

Монитор порождения субъектов (МПС) – субъект, активизирующийся при любом порождении субъектов.

Определение 13

Монитор безопасности субъектов (МБС) – субъект, который разрешает порождение субъектов только для фиксированного подмножества пар активизирующих субъектов и порождающих объектов.

Воздействие МБС выделяет во всем множестве субъектов S подмножество разрешенных *E*. Заметим также, что если в подмножество субъектов в момент времени *f* включается субъект МБС, то первым аргументом операции Create может быть только субъект, входящий во множество субъектов, а аргумент-объект, вообще говоря, любым.

Сформулируем теперь ряд базовых определений, которые в дальнейшем будут повсеместно использоваться.

Определение 14

АС называется замкнутой по порождению субъектов, если в ней действует МБС, разрешающий порождение только фиксированного конечного подмножества субъектов для любых объектов-источников, рассматриваемых для фиксированной декомпозиции АС на субъекты и объекты.

При обсуждении вопросов реализации защищенных сред будет использоваться термин "замкнутая программная среда", который по существу эквивалентен приведенному выше определению. Однако замкнутости АС по порождению субъектов недостаточно для описания свойств системы в части защищенности, поскольку необходимо обеспечить корректность порождаемых МБС субъектов относительно его самого и МБО. Механизм замкнутой программной среды сокращает множество возможных субъектов до некоторого множества фиксированной мощности, но при этом допускает существование некорректных субъектов, включенных в замкнутую среду.

Сформулируем определение изолированности АС.

Определение 15

Множество субъектов АС называется изолированным (абсолютно изолированным), если в ней действует МБС и субъекты из порождаемого множества корректны (абсолютно корректны) относительно друг друга и МБС.

Следствие 1: Любое подмножество субъектов изолированной (абсолютно изолированной АС), включающее МБС, также составляет изолированную (абсолютно изолированную) среду.

Следствие 2: Дополнение изолированной (абсолютно изолированной) АС субъектом, корректным (абсолютно корректным) относительно любого из числа входящих в изолированную (абсолютно изолированную) среду, оставляет ее изолированной (абсолютно изолированной).

Теперь возможно переформулировать достаточное условие гарантированного выполнения политики безопасности следующим образом.

Утверждение 2 (достаточное условие 2 гарантированного выполнения политики безопасности в АС)

Если в абсолютно изолированной АС существует МБО и порождаемые субъекты абсолютно корректны относительно МБО, а также МБС абсолютно корректен относительно МБО, то в такой АС реализуется только доступ, описанный политикой разграничения доступа (ПРД).

Доказательство

Из определения абсолютной изолированности следует возможность существования в АС только конечного множества субъектов, которые в свою очередь корректны относительно МБС (по определению 15 и следствиям из него). Далее, по условию утверждения (корректность МБО относительно любого из порождаемых субъектов и МБС) ассоциированные объекты могут изменяться только самим МБО, следовательно, в АС реализуются только потоки, принадлежащие множеству L Утверждение доказано.

Легко видеть, что данное утверждение является более конструктивным чем предыдущее достаточное условие гарантированной защищенности, поскольку ранее требовалась корректность МБО относительно произвольного субъекта, что практически невозможно. В данном же случае множество субъектов ограничено за счет применения механизма МБС, и возможно убедиться в попарной корректности порождаемых субъектов.

При рассмотрении технической реализации изолированности субъектов в АС будет употребляться термин "изолированная программная среда" (ИПС), который описывает механизм реализации изолированности для конкретной программно-аппаратной реализации АС и при соответствующей декомпозиции на субъекты и объекты.

При рассмотрении операции порождения субъекта возникает весьма важная проблема, связанная с тем, что в реальных АС одинаково поименованные объекты могут иметь различное состояние в пространстве (например, быть размещенными в различных каталогах) или во времени. Предположим, что зафиксировано состояние объекта O_m в некоторый момент

времени t_0. Будем обозначать состояние объекта O_m в момент времени t как $O_m[t]$

Определение 16

Операция порождения субъекта $Create(S_k, O_m) \rightarrow S_i$ называется порождением с контролем неизменности объекта, если для любого момента времени $t > t_0$, в который активизирована операция порождения объекта *Create*, порождение субъекта S_i возможно только при тождественности объектов $O_m[t_0]$ и $O_m[t]$.

Следствие: В условиях определения 16 порожденные субъекты $S_i[t_1]$ и $S_i[t_2]$ тождественны, если $t_1 > t_0$ и $t_2 > t_0$. При $t_1 = t_2$ порождается один и тот же субъект.

При порождении субъектов с контролем неизменности объекта в АС допустимы потоки от субъектов к объектам-источникам, участвующим в порождении субъектов, с изменением их состояния.

Утверждение 3 (базовая теорема ИПС)

Если в момент времени t_0 в изолированной АС действует только порождение субъектов с контролем неизменности объекта и существуют потоки от любого субъекта к любому объекту, не противоречащие условию корректности (абсолютной корректности) субъектов, то в любой момент времени $t > t_0$ АС также остается изолированной (абсолютно изолированной).

Доказательство

Ухо условию утверждения в АС возможно существование потоков, изменяющих состояние объектов, не ассоциированных в этот момент времени с каким-либо субъектом. Если объект с измененным состоянием не является источником для порождения субъекта, то множество субъектов изолированной среды нерасширяемо, в противном случае (измененный объект является источником для порождения субъекта) по условиям утверждения (порождение субъекта с контролем) порождение субъекта невозможно. Следовательно, мощность множества субъектов не может превышать той, которая была зафиксирована до изменения состояния любого объекта. По следствию из определения 16 (о замкнутости множества субъектов в ИПС с невозрастанием мощности множества субъектов) получим, что множество субъектов АС изолировано. Утверждение доказано,

Можно сформулировать методологию проектирования (разработки) гарантированно защищенных АС. Сущность данной методологии состоит в том, что при проектировании защитных механизмов АС необходимо опираться на совокупность приведенных выше (утверждения 1-3) достаточных условий, которые должны быть реализованы для субъектов, что гарантирует защитные

свойства, определенные при реализации МБО в АС (т.е. гарантированное выполнение заданной МБО политики безопасности).

Рассмотренная концепция изолированной программной среды является расширением зарубежных подходов к реализации ядра безопасности ста. Обычно модель функционирования ядра безопасности изображается в виде схемы, представленной на рисунке 5.2, где "база данных защиты" означает объект, содержащий информацию о потоках множества *L* (защита по "белому списку"-разрешения на потоки) или множества N (защита по "черному списку"-запрещение на потоки).

Рисунок 5.2 – Классическая схема ядра безопасности

Для учета влияния субъектов в АС необходимо рассматривать расширенную схему взаимодействия элементов системы реализации и гарантирования политики безопасности.

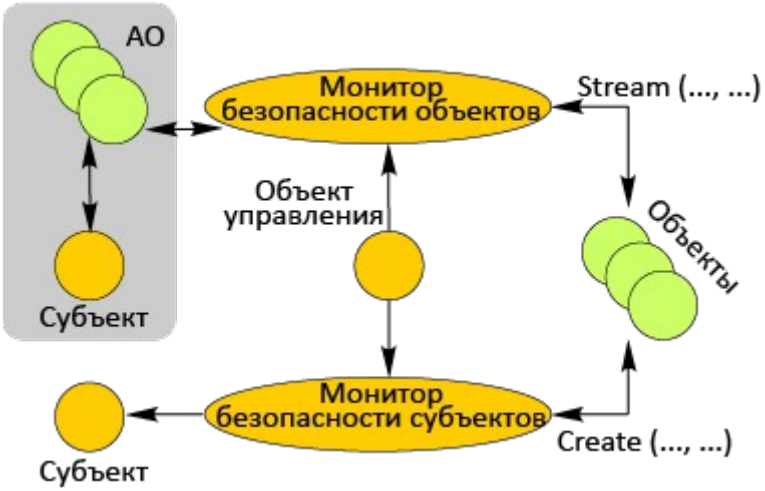

Рисунок 5.3 – Схема ядра безопасности с учетом контроля порождения субъектов

В рисунке 5.3 подчеркнута роль монитора безопасности субъектов при порождении субъектов из объектов. Взаимодействие субъектов и объектов при порождении потоков уточнено введением ассоциированных с субъектом объектов. Объект управления содержит информацию о разрешенных значениях отображения *Stream* (об элементах множества *L* или *N*) и *Create* (элементы множества *E*). Объект управления может быть ассоциирован (ассоциированный объект-данные) как с МБО, так и с МБС.

Перейдем к описанию практических методов построения ИПС. Целью рассмотрения практических подходов является иллюстрация тезиса о том, что

достаточные условия гарантированной защищенности могут быть практически выполнены в реальных АС.

Опираясь на утверждение 3 (базовую теорему ИПС), сформулированное и доказанное выше, опишем метод субъектно-объектного взаимодействия в рамках ИПС для более конкретной архитектуры АС.

Из утверждения 3 следует, что для создания гарантированно защищенной АС (в смысле выполнения заданной политики безопасности) необходимо:

1. убедиться в попарной корректности субъектов, замыкаемых в ИПС (либо убедиться в корректности любого субъекта относительно МБО и МБС);
2. спроектировать и реализовать программно (или программно-аппаратно) МБС так, чтобы;
   - субъекта и любого объекта производился контроль порождения субъектов (т.е. чтобы реализация МБС соответствовала его определению);
   - порождение любого субъекта происходило с контролем неизменности объекта-источника;
3. реализовать МБО в рамках априорно сформулированной политики безопасности.

Надо заметить, что приводимые выше утверждения верны только тогда, когда описанная и реализованная политика безопасности не нарушает их условий (проверка данного факта зависит от модели политики безопасности и является отдельной весьма важной задачей).

Кроме того, необходимо обратить внимание на следующее. Объект управления, который является ассоциированным объектом МБС (обычно ассоциированный объект-данные), играет решающую роль в проектировании ИПС. При изменении состояния объекта управления потенциально возможно "размыкание" программной среды, т.е. добавление к множеству разрешенных субъектов дополнительных, реализующих злоумышленные функции. С другой стороны, процесс управления безопасностью подразумевает способность изменения объекта управления. Возможность изменения объекта управления (реализация потока *Stream* (субъект управления, АО субъекта управления) → объект управления) должна присутствовать для выделенных субъектов (может быть, с дополнительным условием активизации этого субъекта выделенным пользователем или пользователями).

Важную роль при проектировании ИПС играет свойство АС, заключающееся в поэтапной активизации субъектов из объектов различного уровня представления информации.

Пусть в АС выделяется конечное число уровней представления объектов $U = (0,..., R)$, где R – максимальный уровень представления объекта. С точки

зрения выполнения условий утверждения 3 имело бы смысл говорить о некотором "стационарном" состоянии АС, когда в отображениях *Stream* и *Create* участвуют только объекты уровня R. Тогда реализация МБС может быть значительно упрощена (в том смысле, что все аргументы-объекты операции Create имеют тот же уровень). Необходимо обратить внимание на то, что такое требование, с одной стороны, может накладывать ограничительные условия на свойства прикладного ПО (невозможность инициирования потоков, включающих объекты уровня менее *R*, прикладными программами), с другой стороны, быть следствием проектировочных решений реализации субъекта, локализованного в ядре операционной системы (примером является ядро ОС Windows NT4.0, запрещающее операции ниже уровня "файл" со стороны субъектов прикладного уровня).

Практическая реализация всех операционных систем позволяет выделить две фазы их работы: активизация субъектов с ростом уровня представления объектов (фаза загрузки или начальная фаза) и фаза стационарного состояния (когда уровень представления объектов не увеличивается). Конечно, необходимо сделать оговорку, касающуюся возможности реализации потоков к объектам нижнего уровня (операционные системы типа DOS, в которых возможна операция с любым объектом нижнего уровня (сектор) из программ прикладного уровня).

Практическая реализация ИПС может состоять из двух этапов:

1. предопределенное выполнение начальной фазы, включающее в себя активизацию МБС (и МБО);
2. работа в стационарной фазе в режиме ИПС (возможно, с контролем неизменности объектов-источников).

Введем понятие последовательности активизации компонентов АС. Смысл вводимых понятий и формулируемых ниже утверждений состоит в необходимости приведения субъектов АС в одно и то же состояние после активизации первичного субъекта аппаратно-программного уровня или, иначе говоря, в задании предопределенной последовательности активизации субъектов АС.

Обозначим: Z_L – последовательность пар $(i, j)_t$, $(t = 0, 1, 2, ..., L-1$ - моменты времени) длиной L, таких, что $Create(S_i, O_j)\ [t] \rightarrow S_m[t+1]$; S_Z – множество всех субъектов, включенных в последовательность Z_L; O_Z – множество всех объектов, включенных в последовательность Z_L.

Для многопотоковых АС можно рассматривать несколько (возможно, зависимых друг от друга) последовательностей Z_L и соответственно множеств S_Z и O_Z.

Определение 17

Состоянием АС в момент времени t называется упорядоченная совокупность состояний субъектов.

Напомним, что каждый объект есть слово в априорно определенном языке, а понятие состояния субъекта сформулировано выше.

Утверждение 4 (условие одинакового состояния АС)

Состояние АС в моменты времени $t_x^{(1)}$ и $t_x^{(2)}$ ($t_x^{(1)}$ и $t_x^{(2)}$ исчисляются для двух отрезков активности АС от нулевых моментов $t_0^{(1)}$ и $t_0^{(2)}$ одинаково, если:

1. тождественны субъекты $S_i[t_0^{(1)}]$ и $S_j[t_0^{(2)}]$;
2. неизменны все объекты из множества O_Z;
3. неизменна последовательность Z_L

Доказательство

Используем принцип математической индукции. Верность утверждения при $t = 1$ следует из определения тождественности субъектов. Пусть утверждение верно для $t - k < 1$. Тогда в момент времени $k + 1$ могут быть порождены только тождественные субъекты, поскольку тождественны активизирующие субъекты (по предположению индукции) и по условию утверждения неизменны элементы множества O_Z.

Длина L последовательности Z_L определяется:

- по признаку невозможности управления субъектами, принадлежащими множеству S_Z со стороны пользователя (в противном случае последовательность активизации субъектов может быть изменена);
- по признаку доступности для контроля неизменности всех объектов из множества O_Z;
- по признаку не возрастания уровня представления информации (в данном случае имеется в виду, что существует такой момент времени t_x что для любого $t > t_x$ объект-аргумент O_j операции $Stream\ (S_i, O_j)_t$ при надлежит одному уровню представления).

Необходимо заметить, что последовательность Z_L локализуется в некотором объекте либо совокупности объектов (например, для DOS последовательность активизации субъектов предопределена содержанием файлов AUTOEXEC.BAT и CONFIG. SYS), и неизменность последовательности Z_L тождественна неизменности указанных объектов, для ОС Windows NT последовательность активизации компонентов определена содержанием соответствующих ключей реестра ресурсов (REGISTRY).

Пусть в последовательности Z_L можно выделить Z_i, такое, что для любого Z_k, $k>i$, отображения *Create* и *Stream* используют только объекты уровня R. Другими словами, с момента времени i наступает стационарная фаза

функционирования АС. В этих условиях, а также при попарной корректности субъектов и действии МБС с контролем неизменности объектов-источников на уровне R с момента времени $m>k$ верно.

Утверждение 5 (достаточное условие ИПС при ступенчатой загрузке)

При условии неизменности Z_L и неизменности объектов из O_Z в АС с момента времени установления неизменности Z_L и O_Z действует изолированная программная среда.

Доказательство

Необходимо заметить, что все условия утверждения 5 соответствуют утверждению 4. Уточнения касаются структуры последовательности Z_L. Согласно утверждению 4 с момента времени $t=0$ до момента $t=L$ действует изолированная (в рамках) S_Z программная среда.

Для доказательства утверждения необходимо убедиться в том, что:

- МБС в момент времени $t = m$ гарантировано активизируется;
- в любой момент $f > m$ программная среда изолирована.

Первое следует из утверждения 4 (при $t = m$ состояние программной среды всегда будет одинаково, следовательно, всегда будет активизирован субъект МБС); второе – из определения МБС и условия теоремы.

С момента времени $t = 0$ до момента времени L программная среда изолирована, с момента времени $t > m$ программная среда также изолирована, следовательно, АС изолирована при любом $t > 0$. Утверждение доказано.

Используя утверждения 3, 4 и 5, рассмотрим процесс практического проектирования защищенного фрагмента АС.

Первоначально необходимо убедиться в выполнении условий корректности или абсолютной корректности для субъектов, участвующих в порождении ИПС. Указанные субъекты в основном могут быть локализованы на уровне программно-аппаратного компонента компьютера (программы ПЗУ, загрузчики операционных сред), т.е. на аппаратном уровне, либо на уровне операционной среды. Доказательство корректности субъектов программно-аппаратного уровня значительно отличается от соответствующих доказательств для субъектов прикладного уровня. В связи с этим выделим проверку условий корректности субъектов за два шага.

Шагом 1 назовем доказательство корректности субъектов программно-аппаратного уровня. Понятие "модуль" обозначает реализацию объекта-источника. Совокупность субъекта, порожденного из объекта-источника и всего множества ассоциированных с ним объектов в течение всего времени его существования, называется, как правило, процессом (или задачей, заданием). Далее, необходимо определить состав программных средств базовой вычислительной среды, т.е. определить конкретную операци-онную среду,

дополнительные программные средства сервиса (например, программные оболочки или средства телекоммуникации) и программные средства поддержки дополнительного оборудования (программы управления принтером и др.).

Шаг 2 – самый трудоемкий, на нем необходимо убедиться в корректности субъектов базового набора программных средств. При этом важно заметить следующее. В составе ПО АС не должно быть целого класса возможностей – назовем их инструментальными. Прежде всего, это возможность изменения состояния одних субъектов другими (например, изменение содержимого оперативной памяти), возможность инициирования и прекращения выполнения процессов нестандартным образом (помимо механизмов операционной среды). Кроме того, при реализации МБС и МБО на стационарной фазе функционирования АС в любых субъектах, замкнутых в ИПС, должны отсутствовать операции порождения потоков *Stream* к объектам уровня $k<R$.

Обобщенно требования к базовому набору ПО можно сформулировать следующим утверждением.

Утверждение 6 (требования к субъектному наполнению изолированной программной среды)

Для поддержания ИПС в течение всего времени активности АС достаточно, чтобы в состав программного обеспечения, инициированного в ИПС, не входили функции порождения субъектов и прекращения их работы (кроме заранее предопределенных при реализации МБС) и не существовало возможности влияния на среду выполнения любого процесса, а также инициирования потоков к объектам уровня менее R.

Легко видеть, что данное утверждение есть собранные воедино условия выполнения приводимых выше утверждений.

Поясним требование невозможности прекращения функционирования субъекта каким-либо иным образом, кроме предопределенного. В данном случае необходимо учитывать, что во множестве субъектов, замкнутых в ИПС, выделены два особых субъекта – МБС и МБО. Прекращение существования МБС означает нарушение условия замкнутости среды, а прекращение существования МБО-допустимость потоков множества N, т.е. несанкционированный доступ.

Шаг 3 заключается в проектировании и разработке программных или программно-аппаратных средств защиты в АС, а затем и их тестировании. Он подразумевает проектирование и реализацию в заданном множестве субъектов МБС и МБО.

Практически шаги 1-3 могут быть выполнены на основе описанных в литературе методик разработки и тестирования ПО.

Шаг 4 заключается в "замыкании" всего комплекса программного обеспечения, включая и средства защиты, в изолированную программную среду.

Итак, показано, что основными элементами поддержания изолированности программной среды являются контроль целостности и контроль порождения процессов.

Выше мы уже сформулировали понятия МБС и порождения субъектов с контролем их неизменности. Заметим, что для достоверного контроля неизменности объекта (т.е. с вероятностью ошибки, равной нулю) необходимо убедиться в полном тождестве проверяемого объекта и образца. Из этого следует, что эталон должен содержать не менее информации, чем проверяемый объект. Из этого в свою очередь следует, что эталонный объект должен быть как минимум одинаковой длины с проверяемым. На практике такой подход может быть применен с ограничениями (например, для объектов небольшого объема типа программ ПЗУ или загрузчиков ОС).

В связи с этим для контроля целостности применяют объекты, содержащие информацию, зависящую от содержания объекта, но, тем не менее, значительно меньшего объема, вычисленную при помощи класса функций типа "хеш-функций". Очевидно, что в этом случае процесс установления неизменности объекта становится вероятностным.

Исходя из данного факта, невозможно говорить о гарантированных (детерминировано) свойствах системы (поскольку неизменность объекта гарантируется лишь с некоторой вероятностью, не равной единице). Следовательно, все условия утверждений выполняются с некоторой вероятностью, зависящей от свойств хеш-функций, применяемых для контроля целостности. Для подчеркивания изменившихся условий будем говорить далее не о контроле неизменности объекта, а о контроле целостности (КЦ) объекта.

Необходимо отметить также, что в процедуре контроля неизменности (которая теперь принимает вероятностный характер) участвует как минимум два объекта: объект контроля и эталонный объект (хеш-значение), а также субъект, реализующий хеш-функцию и производящий сравнение.

Поэтому для субъекта контроля целостности важно выполнение следующих условий:

1. качественный алгоритм контроля целостности (термин "качественный" будет пояснен ниже);
2. контроль реальных данных (т.е. отображение состояния контролируемого и эталонного объектов в ассоциированные объекты-данные субъекта КЦ, совпадающее с тождественным).

Поясним подробнее второй пункт. Контроль целостности всегда сопряжен с чтением данных, т.е. с инициированием потоков от объектов к ассоциированным объектам-данным субъекта контроля целостности, причем потоки могут соответствовать различному уровню представления информации (чтение по секторам, по файлам и т.д.). Например, встроенный в BIOS компьютера субъект (практически это программная закладка) может навязывать при чтении вместо одного сектора другой или редактировать непосредственно буфер, в который были считаны данные. Аналогичный эффект может быть вызван субъектами операционной среды, например субъектами, локализованными в ее первичных загрузчиках. С другой стороны, даже контроль BIOS может происходить "под наблюдением" какой-либо дополнительной аппаратуры и не показать его изменение. Аналогичные эффекты могут возникать и при обработке файла. Цель организации режима чтения реальных данных состоит в тождественном отображении параметров чтения на АО субъекта чтения (поток от АО субъекта КЦ к АО субъекта чтения) и тождественном отображении считываемого объекта (в соответствии с параметрами, переданными субъекту чтения) к ассоциированным объектам-данным субъекта КЦ.

Поясним теперь понятие качественного КЦ с точки зрения математических свойств функции КЦ. Предположим, что имеется некоторый объект F и некоторый алгоритм H, преобразующий объект F в некоторый объект M, который представляется словом того же языка, но меньшей длины. Этот алгоритм таков, что при случайном равновероятном выборе двух объектов F_1 и F_2 из множества возможных соответствующие им объекты $M_1=H(F_1)$ и $M_2 = H(F_2)$ с высокой вероятностью различны. Тогда проверка целостности данных строится так: рассматриваем объект F, по известному алгоритму H строим $K=H(F)$ и сравниваем M, заранее вычисленное как $M=H(F)$, с K. При совпадении считаем объект неизменным. Алгоритм H называют, как правило, хеш-функцией или реже – контрольной суммой, а число M – хеш-значением, содержащимся в некотором объекте.

Качество КЦ определяется в данном случае выполнением следующих условий:

1. по известному объекту $M=H(F)$ нахождение другого объекта G, не тождественного F, такого, что $M=H(G)$, является задачей с трудоемкостью не менее заданной T_H,
2. объект M должен быть недоступен для изменения;
3. длина объекта M должна обеспечивать условную вероятность $P(H(F_1) == H(F_2)|F_1$ не тождественны F_2) не более заданной P_H.

Поясним смысл этих условий. Пусть программа злоумышленника изменила объект *F* (статическое искажение). Тогда, вообще говоря, хеш-значение M объекта F изменится. Если субъекту злоумышленника доступен для изменения объект M (существует соответствующий поток), то он может по известному алгоритму H вычислить новое хеш-значение для измененного объекта и заместить им исходное.

Пусть хеш-значение недоступно, тогда можно попытаться так построить объект, чтобы хеш-значение его не изменилось (принципиально это возможно, поскольку отображение, задаваемое алгоритмом хеширования H, небиективно (неоднозначно)).

Таким образом, при условии недоступности хеш-значения для изменения и доступности для изменения объекта-источника трудоемкость нарушения ИПС с КЦ объектов-источников (т.е. возможность породить субъект из объекта-источника, нетождественного исходному объекту) совпадает с T_H. При однократной попытке инициировать субъект из случайно равновероятно выбранного объекта-источника вероятность нарушения ИПС (успешное порождение субъекта) не превышает P_H. Итак, "качество" ИПС определяется свойствами хеш-функции H, а именно: величинами T_H и P_H.

Обобщим приводимые выше рассуждения в методе "безопасной загрузки" или ступенчатого контроля. Он заключается в постепенном установлении неизменности компонентов программно-аппаратной среды.

1. Сначала проверяется неизменность программ ПЗУ, при положительном исходе через проверенные на целостность программы ПЗУ считывается загрузочный сектор и драйверы операционной системы (по секторам) и их неизменность также проверяется, кроме того, проверяется целостность объекта, определяющего последовательность активизации компонентов.
2. Через функции чтения проверенной ОС инициируется процесс контроля порождения процессов (реализация МБС).

Инициирование процесса контроля доступа к объектам завершает проектирование гарантировано защищенной АС

Рассматривая вопросы программно-технической реализации ИПС, необходимо заметить, что мощность множества субъектов в некотором сегменте АС (выделенном по признаку принадлежности одному компьютеру) с момента включения питания до момента запуска процессов пользователя увеличивается. Первоначально активизируются субъекты аппаратно-программного уровня (программы ПЗУ), затем указанные субъекты порождают из объектов-источников данного уровня (это, как правило, сектора внешних носителей информации) субъектов уровня операционной среды.

Субъекты уровня операционной среды, как уже отмечалось, также делятся на два подуровня: нижний уровень – субъекты-первичные загрузчики ОС (работающие с информацией уровня секторов) и верхний уровень – субъекты-драйверы (порождаемые субъектами-первичными загрузчиками из объектов-секторов), работающие с объектами уровня "файл" (последовательности секторов). На этапе перехода от субъектов-загрузчиков к субъектам-драйверам происходит переход и к другой декомпозиции АС на объекты (от секторов к файлам). Указанная иерархия действует в любой известной на сегодняшний день АС и естественным образом предопределяет архитектуру, в рамках которой формируется и функционирует ИПС.

Например, аппаратная архитектура IBM PC задает следующие этапы активизации различных субъектов АС. При включении питания происходит тестирование ОП, инициализация таблицы векторов прерываний и поиск расширений BIOS. При их наличии управление передается на них. После отработки расширений BIOS в память считывается первый сектор дискеты или винчестера, и управление передается на него (образуется код загрузчика), затем код загрузчика считывает драйверы операционной системы, далее интерпретируются файлы конфигурации, подгружается командный интерпретатор и выполняется файл автозапуска.

При реализации ИПС на нее должна быть возложена функция контроля запусков программ и контроля целостности. При описании методологии проектирования ИПС упоминалась проблема контроля реальных данных. Эта проблема состоит в том, что контролируемая на целостность информация может представляться по-разному на разных уровнях.

Внедренный в систему субъект может влиять на процесс чтения-записи данных на уровне файлов (или на уровне секторов) и предъявлять системе контроля некоторые другиэ данные вместо реально существующих данных. Этот механизм неоднократно реализовался в STELS-вирусах. Однако верно утверждение.

Утверждение 7 (достаточное условие чтения реальных данных)

Если субъект, обслуживающий процесс чтения данных (т.е. указанный субъект инициируется запрашивающим данные субъектом и участвует в потоке), содержал только функции тождественного отображения данных на ассоциированные объекты-данные любого субъекта, инициирующего поток чтения, и целостность объекта-источника для этого субъекта зафиксирована, то при его последующей неизменности чтение с использованием порожденного субъекта будет чтением реальных данных.

Доказательство

Верность утверждения следует из определения тождественности субъекта и из условия утверждения, гарантирующего неизменность объекта-источника Необходимо и здесь сделать оговорку о вероятностном характере установления неизменности и говорить, что чтение реальных данных возможно с вероятностью, определяемой алгоритмом КЦ.

Метод ступенчатого контроля не противоречит утверждениям 4 и 5 и предусматривает разделение последовательности активизации компонентов Z_L на подпоследовательности с одинаковым уровнем представления информации. Реализация метода ступенчатого контроля целостности должна удовлетворять условиям утверждения 4.

Выше было сказано о том, что субъект контроля неизменности объектов, входящих в процедуры активизации АС, и объектов, описывающих последовательность активизации компонентов, должен быть активен уже на этапе работы субъектов аппаратно-программного уровня, но его объект-источник технически не может быть проверен на неизменность. В связи с этим подчеркнем весьма важный факт для любых реализаций ИПС.

Аксиома 5

Генерация ИПС рассматривается в условиях неизменности конфигурации тех субъектов АС, которые активизируются до старта процедур контроля целостности объектов O_Z и последовательности Z_L. Неизменность данных субъектов обеспечивается внешними по отношению к АС методами и средствами. При анализе или синтезе защитных механизмов свойства указанных субъектов являются априорно заданными.

При решении практических вопросов генерации ИПС можно выделить три самостоятельных направления. Первое из них связано с использованием внешних по отношению к АС субъектов (как правило, размещенных на внешнем носителе), целостность которых гарантируется методами хранения или периодического контроля. Предопределенность активизации субъектов, локализованных на внешних носителях, обеспечивается свойствами субъектов аппаратно-программного уровня (например, возможно установить такую аппаратную конфигурацию компьютера, при которой будет происходить загрузка операционной системы с гибкого магнитного диска).

Второе направление связано с локализацией ИПС в рамках территориально ограниченного рабочего места (как правило, компьютера) и использует аппаратную поддержку для задания предопределенной последовательности активизации субъектов. Данное направление, как правило, включает и аппаратную поддержку аутентификации пользователей.

Третье направление связано с реализацией метода доверенной загрузки операционной среды с использованием уже имеющихся в ней механизмов реализации и гарантирования политики безопасности.

Необходимо заметить, что в различные интервалы активности АС субъектами могут управлять различные пользователи, для которых множество разрешенных субъектов E различно. В связи с этим будем говорить о множестве E_i для i-го пользователя АС. Будем также подразумевать, что перед установлением однозначного соответствия множества E_i пользователю i происходит процедура аутентификации.

Первый способ проектирования ИПС в рамках подхода с использованием внешнего носителя получил название "невидимой дискеты". Этот способ заключается в том, что все объекты, принадлежащие множеству O_Z, и объекты, описываюіцие последовательность Z_L, помещаются на внешний носитель, с которого может быть произведена загрузка операционной системы (обычно дискета). Неизменность объекта ДОМЕНЫ БЕЗОПАСНОСТИ физической защитой носителя от записи. Кроме того, специальная технология не позволяет использовать объекты (в том числе и обеспечить выполнение программ) без загрузки ОС именно с этой дискеты.

Как следует из утверждения 5, одним из важнейших условий поддержания ИПС является невозможность изменения последовательности активизации компонентов. В данном случае целостность объектов, содержащих последовательность активизации компонентов, гарантируется физическим запретом записи на дискету.

Важной проблемой является невозможность прерывания процесса активизации компонентов. В ряде операционных сред для этого имеются штатные возможности, предусмотренные для обеспечения защиты от ошибок пользователя, сформировавшего некорректную последовательность активизации компонентов ОС. В связи с этим должны быть приняты меры, гарантирующие пассивность органов управления в период отработки последовательности Z_L (например, аппаратная блокировка клавиатуры с момента активизации модифицированного BOOT до момента окончания активизации субъектов множества S_Z).

Описанный метод позже был реализован во внешних носителях типа CD-ROM, которые позволили значительно (на два порядка) увеличить информационную емкость носителя и загружать с него развитые операционные среды типа OS/2. Однако однократность записи существенно снимает гибкость построения ИПС таким методом.

Неудобство использования загрузочной дискеты и ее быстрый износ обусловили возникновение следующего способа проектирования ИПС.

Откажемся от рассмотрения загрузочной дискеты, и рассмотрим компьютер с загрузкой ОС с устройства локального хранения (винчестера) и дополнительным аппаратным устройством изолирования среды. Опишем два этапа – этап установки ИПС и этап эксплуатации ИПС.

Предположим существование N пользователей, каждый i-и из которых характеризуется некоторой персональной информацией K_i, не известной другим пользователям и хранящейся на некотором материальном носителе (например, устройстве сенсорной памяти типа Touch Memory). Администратор системы знает все K_i и единолично проводит этап установки. Пользователи (владельцы K_i) участвуют только в этапе эксплуатации.

Процесс установки ИПС состоит из следующих действий.

1. В компьютер устанавливают аппаратный модуль, включающий в себя устройство и программы ПЗУ данного устройства (субъекты аппаратно-программного уровня), реализующие:
   - операции сервиса аутентифицирующего носителя пользователя K_i (как минимум его чтение);
   - аутентификацию пользователя с номером i по введенному им K_i;
   - чтение массива данных, содержащего множество доступных для пользователя объектов-источников (исполняемых модулей) $F_{i1}, F_{i2},..., F_{im}$, составляющих O_Z, а также объект, содержащий Z_L;
   - вычисление информации $M_{i1}, M_{i2},..., M_{im}$, фиксирующей целостность объектов-источников $F_{i1}, F_{i2}, ..., F_{im}$ каждого объекта-источника (информация M_{ij} должна удовлетворять требованиям к хеш-значениям и, возможно, зависеть от K_i), $M_{ij}=H(K_i, F_{ij})$;
   - блокирование устройств управления и предотвращение загрузки операционной среды с внешнего носителя;
2. Администратор определяет для пользователя i набор потенциально возможных для активизации субъектов E_i, $E_i = \{P_{i1}...P_{imi}\}$ $i = 1, ..., N$, $Create\ (P_{ik},F_{ij})->P_{ij}$, где m_i-число разрешенных к запуску задач для i-го пользователя.
3. Администратор формирует (и заносит на носитель) или считывает с носителя для i-го пользователя его K, и вычисляет значения для последующего контроля целостности $M_{ijr} — H(K_i, F_{jr})$, где H-функция КЦ (хеш-функция).
4. Администратор проделывает действия 2 и 3 для всех N пользователей.
5. Администратор устанавливает в АС МБС с объектом-источником $F_{ИПС}$ и фиксирует его целостность. Установка модуля происходит с учетом условий утверждения 5.

6. Администратор фиксирует целостность объекта, содержащего Z_L роцесс эксплуатации состоит из следующих действий.

Включение питания и активизация аппаратного модуля:

а) идентификация пользователя *i* по K_i, при успехе выполняется п. б), при неудаче компьютер блокируется;

б) проверка целостности всех установленных ПЗУ; при положительном исходе выполняется п. в), при неудаче компьютер блокируется;

в) чтение по секторам файлов операционной среды и проверка их целостности;

г) чтение как файла $F_{ИПС}$ (с помощью функций операционной среды) и проверка его целостности; вариантом может быть чтение $F_{ИПС}$ по секторам;

д) активизация процесса контроля $P_{ИПС}$: $Create\ (S_x,\ F_{ИПС})\text{->}P_{ИПС}$; активизация МБО;

е) запуск избранной задачи *i*-го пользователя (может не выполняться).

Работа в ИПС. Запуск каждого процесса P_S сопровождается проверками:

а) принадлежит ли F_S к множеству разрешенных для $i(E_i)$, если да, то выполняется п. б), иначе запуск игнорируется;

б) совпадает ли $G=H(K_i,F_S)$ с $M=H(K_i,F_S)$, вычисленной администратором;

в) при положительном исходе п. б) задача запускается, иначе запуск игнорируется.

Легко видеть, что условия изолированности среды выполнены. Кроме того, в данном случае реализован механизм ступенчатого контроля, обеспечивающий чтение реальных данных.

Используя утверждение 4 об одинаковости состояний АС после активизации проверенных на неизменность субъектов в неизменной последовательности, можно описать метод доверенной загрузки компонентов операционной среды (кратко "метод доверенной загрузки").

Пусть предопределен порядок загрузки компонентов ОС некоторой АС (под загрузкой компонентов ОС понимается активизация различных субъектов ОС из соответствующих объектов-источников различного уровня иерархии). Процедуру загрузки ОС назовем доверенной, если:

- установлена неизменность компонентов ОС (объектов), участвующих в загрузке (иными словами объектов, принадлежащих множеству O_Z), причем неизменность установлена до порождения первого субъекта из Z_L.
- установлена неизменность объектов, определяющих последовательность активизации компонентов ОС (с учетом нескольких уровней иерархии), неизменность обеспечена в течение заданного интервала времени; состояние указанных объектов не может быть изменено никем, кроме

предопределенного пользователя (пользователей) ОС (это условие соответствует неизменности последовательности Z_L).

Легко видеть, что процедура доверенной загрузки обеспечивает одинаковое состояние ОС после выполнения загрузки (согласно утверждению 4).

Основная техническая проблема при реализации доверенной загрузки состоит в доступе к объектам высшего уровня иерархии ОС (файлам) до загрузки ядра данной ОС (загружаемую ОС далее будем называть пользовательской). Однако при возможности генерации ИПС для какой-либо иной ОС (далее будем называть ее базовой) можно предложить итеративную реализацию доверенной загрузки с использованием ресурсов указанной ОС.

Рассмотрим реализацию доверенной загрузки ОС на основе генерации ИПС для одной из операционных сред вычислительной системы.

Предположим, что имеется базовая операционная система, для которой возможна полноценная генерация ИПС. Пусть в АС существуют еще операционные среды OS_i, OS_2, ..., OS_n. Ставится задача доверенного запуска операционной среды OS_j. Пусть в базовой операционной среде имеется некоторое условно называемое "шлюзовое ПО" между базовой операционной средой и OS_j. Функции шлюзового ПО заключаются в обеспечении доступа к файловой системе операционной среды OS_j (т.е. объектам уровня R).

Пусть также пользователь i имеет физический доступ к комплекту технических средств (рабочему месту) сети T_m, на котором установлена операционная среда OS_j. При использовании комплекта T_m пользователем i происходят следующие действия:

- аутентификация пользователя i (по его индивидуальной информации);
- проверка прав пользователя по использованию аппаратного компонента комплекта T_m,
- контроль целостности (на основе информации пользователя K_i, либо без нее) всех объектов базовой ОС, размещенных на некотором носителе локально или удаленно (через технические средства ЛВС) связанном с T_m;
- загрузка базовой операционной системы и контроль целостности шлюзового ПО;
- загрузка шлюзового ПО (при этом становится доступным как минимум в режиме чтения файловая структура OS_j, размещенная локально на T_m);
- контроль целостности объектов уровней, меньших R_j (R_j – максимальный уровень представления объектов в OS_j) для OS_j;
- контроль целостности объектов уровня R_j (файлов) OS_j;

- контроль целостности объекта, задающего последовательность загрузки компонентов;
- принудительная загрузка (инициируется предопределенный в силу целостности объектов O_Z и последовательности Z_L порядок загрузки компонентов ОС) проверенной на целостность OS_j.

Утверждение 8 (условия генерации ИПС при реализации метода доверенной загрузки)

Пусть ядро ОС содержит МБО и МБС, инициируемые в ОС субъекты попарно корректны и их объекты-источники принадлежат множеству проверяемых на неизменность в ходе доверенной загрузки, МБО запрещает изменение любого объекта-источника и выполнена процедура доверенной загрузки ОС. Тогда после инициирования ядра ОС генерируется ИПС.

5.3. Домены безопасности

В рамках рассматриваемой модели АС как совокупности субъектов и объектов разграничение доступа субъектов к объектам может быть реализовано на основе таблицы, содержащей разрешенные типы доступа и называемой матрицей доступа. Как правило, матрица доступа (см. таблицу 5.1) имеет большие размеры (в системе присутствует множество различных субъектов и объектов) и является разреженной (субъектам необходим доступ только к небольшим подмножествам объектов).

Таблица 5.1

		Объекты			
		1	**2**	**...**	**m**
Субъекты	**1**	Чтение	Чтение	...	Исполнение
	2	Чтение	Чтение/Запись	...	Чтение/Запись
	...	...	...	...	...
	n	Запись	Исполнение	...	Нет доступа

Под доменом безопасности понимается совокупность объектов, к которым разрешен доступ конкретном у субъекту.

В таблице 5.1 домены безопасности представлены отдельными строками. В соответствии с обсуждаемым ниже принципом минимизации привилегий домен безопасности данного субъекта должен включать минимально возможный набор объектов и связанных с ними прав доступа, необходимый для работы субъекта. Тем самым снижается риск злоупотребления правами доступа со стороны субъекта и уменьшается разрушительный эффект от потенциального злоупотребления. Для реализации этого принципа необходимо, чтобы субъекты,

которым необходимо выполнять множество различных операций, могли поочередно работать в нескольких небольших (в смысле числа составляющих их объектов и назначенных прав доступа к этим объектам) доменах, переключаемых при необходимости. Следующие факторы определяют минимально возможные по практическим соображениям размеры доменов. гибкость и простота механизма переключения доменов:

1. размер защищаемых объектов;
2. наличие разных способов изменения матрицы доступа;
3. гибкость в определении произвольных типов доступа к объектам.

В АС переключение доменов безопасности может происходить, в частности, при вызове из основной программы некоторой процедуры или функции, или, говоря в терминах рассмотренной выше субъектно-объектной модели, в момент порождения одним субъектом (выполняющейся программы) нового субъекта (вызываемой процедуры) из некоторого объекта (области памяти, содержащей код процедуры). По завершению выполнения вызванной процедуры происходит обратное переключение домена безопасности.

Если с вызовом процедуры связано переключение доменов безопасности, процедура называется защищенной. Такая процедура фигурирует в матрице доступа и в качестве субъекта, и в качестве объекта. Первое объясняется тем, что процедура функционирует в собственном домене безопасности. Второе – тем, что по отношению к данной процедуре могут быть назначены права доступа, в частности право "исполнить".

Рассмотрим пример, где права доступа заданы согласно таблице 5.2.

Таблица 5.2

		Объекты		
		Файл программы редактора текстов	**Текстовый файл**	**Словарь**
Субъекты	**Пользователь**	Исполнить	Чтение/Запись	
	Программа редактор		Чтение/Запись	Чтение

Пользователь имеет доступ к текстовому файлу как при помощи текстового редактора, так и из собственного домена безопасности. Однако доступ к словарю для пользователя становится возможен только при переключении на домен безопасности редактора (путем запуска на исполнение программы редактора). При таком способе переключения доменов матрица доступа после переключения остается неизменной.

Более сложный случай переключения доменов связан с передачей прав доступа в качестве параметров вызываемой процедуре и сопровождается

изменением матрицы доступа. Предположим, что права доступа заданы так, как показано в таблице 5.3.

Таблица 5.3

		Объекты		
		Файл программы редактора текстов	**Текстовый файл**	**Словарь**
Субъекты	**Пользователь**	Исполнить	Чтение/Запись	
	Программа редактор			Чтение

В отличие от предыдущего случая, редактор не имеет права доступа к текстовому файлу пользователя. При вызове редактора это право должно быть ему передано, и в матрице доступа будет создана новая, временная строка (см. таблицу 5.4).

Таблица 5.4

		Объекты		
		Файл программы редактора текстов	**Текстовый файл**	**Словарь**
Субъекты	**Пользователь**	Исполнить	Чтение/Запись	
	Программа редактор			Чтение
	Редактор, действующий от имени пользователя		Чтение/Запись	Чтение

Созданный при этом временный домен безопасности описывает стандартное право текстового редактора на доступ к словарю и переданное ему при вызове право на доступ к файлу пользователя. Этот домен безопасности уничтожается по завершению работы редактора.

В рассмотренных примерах переключение доменов безопасности было связано либо только с потерей прав (например, пользователь теряет доступ к словарю, завершив работу с редактором), либо только с их приобретением (редактор при запуске получает доступ к текстовому файлу). Данная концепция доменов безопасности может быть расширена и на случай, когда потеря и приобретение различных прав доступа одним субъектом происходят при переключении домена безопасности одновременно, а также на случай, когда вызываемые процедуры являются реентерабельными.

Тема 6. Криптографические модели защиты информации

6.1. Роль криптографических методов защиты

При построении защищенных АС роль криптографических методов для решения различных задач информационной безопасности трудно переоценить. Криптографические методы в настоящее время являются базовыми для обеспечения надежной аутентификации сторон информационного обмена, защиты информации в транспортной подсистеме АС, подтверждения целостности объектов АС и т.д.

К средствам криптографической защиты информации (СКЗИ) относятся аппаратные, программно-аппаратные и программные средства, реализующие криптографические алгоритмы преобразования информации с целью:

- защиты информации при ее обработке, хранении и передаче по транспортной среде АС;
- обеспечения достоверности и целостности информации (в том числе с использованием алгоритмов цифровой подписи) при ее обработке, хранении и передаче по транспортной среде АС;
- выработки информации, используемой для идентификации и аутентификации субъектов, пользователей и устройств;
- выработки информации, используемой для защиты аутентифицирую- щих элементов защищенной АС при их выработке, хранении, обработке и передаче.

Предполагается, что СКЗИ используются в некоторой АС (в ряде источников – информационно-телекоммуникационной системе или сети связи), совместно с механизмами реализации и гарантирования политики безопасности.

Не останавливаясь детально на определении криптографического преобразования, отметим его несколько существенных особенностей:

- в СКЗИ реализован некоторый алгоритм преобразования информации (шифрование, электронная цифровая подпись, контроль целостности и д.р.);
- входные и выходные аргументы криптографического преобразования присутствуют в АС в некоторой материальной форме (объекты АС);
- СКЗИ для работы использует некоторую конфиденциальную информацию (ключи);

- алгоритм криптографического преобразования реализован в виде некоторого материального объекта, взаимодействующего с окружающей средой (в том числе с субъектами и объектами защищенной АС).

Таким образом, роль СКЗИ в защищенной АС – преобразование объектов. В каждом конкретном случае Сказанное преобразование имеет особенности. Так, процедура зашифрования использует как входные параметры объект-открытый текст и объект-ключ, результатом преобразования является объект-шифрованный текст; наоборот, процедура расшифрования использует как входные параметры шифрованный текст и ключ; процедура простановки цифровой подписи использует как входные параметры объект-сообщение и объект-секретный ключ подписи, результатом работы цифровой подписи является объект-подпись, как правило, интегрированный в объект-сообщение.

Можно говорить о том, что СКЗИ производит защиту объектов на семантическом уровне. В то же время объекты-параметры криптографического преобразования являются полноценными объектами АС и могут быть объектами некоторой политики безопасности (например, ключи шифрования могут и должны быть защищены от НСД, открытые ключи для проверки цифровой подписи – от изменений и т.д.).

Итак, СКЗИ в составе защищенных АС имеют конкретную реализацию-это может быть отдельное специализированное устройство, встраиваемое в компьютер, либо специализированная программа.

Существенно важными являются следующие моменты:

- СКЗИ обменивается информацией с внешней средой, а именно: в него вводятся ключи, открытый текст при шифровании;
- СКЗИ в случае аппаратной реализации использует элементную базу ограниченной надежности (т.е. в деталях, составляющих СКЗИ, возможны неисправности или отказы);
- СКЗИ в случае программной реализации выполняется на процессоре ограниченной надежности и в программной среде, содержащей посторонние программы, которые могут повлиять на различные этапы его работы;
- СКЗИ хранится на материальном носителе (в случае программной реализации) и может быть при хранении преднамеренно или случайно искажено;
- СКЗИ взаимодействует с внешней средой косвенным образом (питается от электросети, излучает электромагнитные поля и т.д.);

- СКЗИ изготавливает или/и использует человек, могущий допустить ошибки (преднамеренные или случайные) при разработке и эксплуатации.

Таким образом, можно выделить ряд основных причин нарушения безопасности информации при ее обработке СКЗИ.

Утечки информации no технические каналам:

- электромагнитному высокочастотному прямому (излучение электронно-лучевой трубки дисплея, несущее информацию о выводе на экран, высокочастотное излучение системного блока, модулированное информативным сигналом общей шины и т.д.);
- электромагнитному низкочастотному прямому (поле с сильной магнитной составляющей от магнитных элементов типа катушек или трансформаторов);
- электромагнитному косвенному (наводки на проводящие линии и поверхности, модуляция гетеродинов вспомогательной аппаратуры);
- акустическому (звуки и вибрации от нажатий клавиш и работы принтера, голоса оператора СКЗИ);
- визуальному (просмотр или фотографирование текстов на экране, принтере или иных устройствах отображения информации);
- акустоэлектрическому (преобразование звуковых и вибрационных сигналов в электрические с помощью вспомогательного оборудования (телефон, электрочасы, осветительные приборы и т.д.);
- сетевому (неравномерность потребляемого от сети тока, наводки на провода питания);
- по шине заземления или по линии связи компьютера – связное оборудование (модем) (наводки сигнала от СКЗИ в линии связи или заземлении).

Кроме того, возможен анализ вспомогательных материалов (красящих лент, неисправных дискет и винчестеров и т.д.).

Неисправности в элементах СКЗИ

Сбои и неисправности в элементах СКЗИ могут сказаться на виде шифрующего преобразования (можно показать, что в общем случае фиксация нулевых или единичных потенциалов приведет к упрощению реализации шифрующего преобразования), на протоколах взаимодействия аппаратуры или программ СКЗИ с прочим оборудованием и программами (например, ввод каждый раз фиксированного ключа) или на процедурах считывания ключа.

Работа совместно с другими программами

При этом речь может идти об их непреднамеренном и преднамеренном влиянии. Рассмотрим первое. Пусть программа зашифровывает файл и помещает шифртекст в тот же файл. Предположим, что в то же время работает программа запрета записи на диск. Тогда результатом шифрования будет исходный незашифрованный файл. В общем случае источником непреднамеренного взаимного влияния является, как правило, конкуренция из-за ресурсов вычислительной среды и некорректная обработка ошибочных ситуаций.

При рассмотрении второй ситуации применяют термин "программная закладка" (некоторые авторы используют термин "криптовирус", "троянский конь"). Речь идет о специализированном программном модуле, целенаправленно воздействующем на СКЗИ. Программная закладка может работать в следующих режимах:

1. пассивном (сохранение вводимых ключей или открытых текстов без влияния на информацию;
2. активном:
    - • влияние на процессы записи-считывания программ шифрования и цифровой подписи без изменения содержания информации (пример – программная закладка для системы цифровой подписи Pretty Good Privacy (PGP), выполняющая навязывание укороченных текстов для хеширования);
    - • влияние на процессы считывания и записи с изменением информации;
    - • изменение алгоритма шифрования путем редактирования исполняемого кода в файле или оперативной памяти.

Воздействие человека

Разработчик преднамеренно или непреднамеренно может внести в программу некоторые свойства (например, возможность переключения в отладочный режим с выводом части информации на экран или внешне носители). Эксплуатирующий программу защиты человек может решить, что программа для него "неудобна" и использовать ее неправильно (вводить короткие ключи либо повторять один и тот же ключ для шифрования разных сообщений). То же замечание относится и к аппаратным средствам защиты.

В связи с этим помимо встроенного контроля над пользователем необходимо отслеживать правильность разработки и использования средств защиты с применением организационных мер.

6.2. Требования к СКЗИ

Криптографические требования

Будем полагать, что для раскрытия шифрованной информации злоумышленник может в любой момент после получения криптографически защищенной информации применить любой алгоритм дешифрования (для цифровой подписи – получение секретного ключа подписи либо подбор текста) при максимальном использовании сведений и материалов, полученных при реализации вышеперечисленных угроз.

Эффективность применения злоумышленником алгоритмов определяется средней долей дешифрованной информации, являющейся средним значением отношения количества дешифрованной информации к общему количеству шифрованной информации, подлежащей дешифрованию, и трудоемкостью дешифрования единицы информации, измеряемой Q элементарными опробованиями. Под элементарным опробованием, как правило, понимается операция над двумя n-разрядными двоичными числами. При реализации алгоритма дешифрования может использоваться гипотетический вычислитель, объем памяти которого не превышает M двоичных разрядов. За одно обращение к памяти, таким образом, может быть записано по некоторому адресу или извлечено не более n бит информации. Обращение к памяти по трудоемкости приравнивается к элементарному опробованию.

За единицу информации принимаются общий объем информации, обработанной на одном СКЗИ в течение единицы времени (как правило, суток). Атака злоумышленника на конфиденциальность информации (дешифрование) успешна, если объем полученной открытой информации больше V.

Применение алгоритма считается неэффективным, если выполнено одно из условий:

$\pi<\pi_0$ или $Q>Q_0$.

Значение параметров π, Q, V, M и пороговые значения π_0 и Q_0 определяются для каждого СКЗИ отдельно.

Требования надежности

СКЗИ должны обеспечивать заданный уровень надежности применяемых криптографических преобразований информации, определяемый значением допустимой вероятности неисправностей или сбоев, приводящих к получению злоумышленником дополнительной информации о криптографическом преобразовании.

Эта криптографически опасная информация (КОИ) потенциально позволяет уменьшить фиксированные для конкретного СКЗИ параметры трудоемкости Q_0 при использовании некоторого алгоритма дешифрования.

При вычислении параметра Q учитываются затраты на определение только тех неисправностей, которые не выявляются до начала работы СКЗИ (например, если компьютер не загружается и СКЗИ не работает, то такой класс неисправностей не опасен).

Правильность функционирования технических средств АС, в рамках которых реализовано СКЗИ, определяется как соответствие выполнения элементарных инструкций (команд) описанному в документации. Ремонт и сервисное обслуживание СКЗИ также не должно приводить к ухудшению свойств СКЗИ в части параметров надежности.

Требования по защите от НСД для СКЗИ, реализованных в составе АС

В АС, для которых реализуются программные или программно-аппаратные СКЗИ, при хранении и обработке информации должны быть предусмотрены следующие основные механизмы защиты от НСД:

- идентификация и аутентификации пользователей и субъектов доступа (программ, процессов);
- управление доступом;
- обеспечение целостности;
- регистрация и учет.

Подсистема идентификации и аутентификации предназначена для выделения и распознания пользователей, допущенных к работе с СКЗИ, на основе их индивидуальных аутентифицирующих признаков (паролей, аппаратных носителей и т.д.). При осуществлении доступа пользователей к АС или компонентам СКЗИ вероятность P ложной аутентификации на одну попытку доступа должна быть не более вероятности PQ. В системе должно быть установлено ограничение на число следующих подряд неудачных попыток, достижение которого квалифицируется как факт НСД. Ложная аутентификация понимается как событие "принять незарегистрированного в системе пользователя за одного из легальных пользователей" при случайном равновероятном выборе без возвращения аутентифицирующего признака пользователя из множества возможных.

Подсистема управления доступом осуществляет контроль потоков информации между субъектами и объектами доступа и обеспечивает проверку выполнения правил доступа пользователей к компонентам СКЗИ.

Подсистема обеспечения целостности осуществляет контроль неизменности программных механизмов защиты от НСД (в том числе, алгоритма функционирования программного компонента СКЗИ) в соответствии с правилами управления доступом. При этом:

- вероятность P, с которой допускается при однократной попытке изменение закона функционирования СКЗИ или системы защиты от НСД, не должна превышать вероятности P_0;
- вероятность P, с которой допускается при однократной попытке несанкционированное чтение или изменение хранимой конфиденциальной информации или КОИ, не должна превышать вероятности P_0.

Подсистема регистрации и учета должна обеспечивать регистрацию параметров процесса идентификации и аутентификации пользователей, выдачи документов на внешний материальный носитель (дискету, твердую копию и др.), запуска (завершения) программ и процессов, предназначенных для обработки защищаемых файлов, попыток доступа программных средств к защищаемым файлам. Должен осуществляться автоматический учет создаваемых защищаемых файлов, защищаемых носителей информации. Подсистема регистрации и используемые в ней данные должны быть в числе объектов контроля доступа.

В системе защиты от НСД должен быть предусмотрен администратор (служба) защиты информации, ответственный за дополнение и исключение пользователей в системе с СКЗИ, установление правил доступа, нормальное функционирование и контроль работы механизмов защиты от НСД.

Требования к средам разработки, изготовления и функционирования СКЗИ

Аппаратные средства, на которых реализуются программные или программно-аппаратные СКЗИ, и программно-аппаратная среда (программно-аппаратное окружение), в которой разрабатываются, изготавливаются и эксплуатируются СКЗИ, не должны иметь явных и скрытых функциональных возможностей, позволяющих:

- модифицировать или изменять алгоритм работы СКЗИ в процессе их разработки, изготовления и эксплуатации;
- модифицировать или изменять информационные или управляющие потоки и процессы, связанные с функционированием СКЗИ;

- осуществлять доступ (чтение и модификацию) посторонних лиц (либо управляемых ими процессов) к ключам и идентификационной, и аутентификационной информации;
- получать доступ к конфиденциальной информации СКЗИ.

Состав и назначение программно-аппаратных средств должны быть фиксированы и неизменны в течение всего времени, определенного в заключении о возможности использования.

Тема 7. Криптографические модели защиты информации (продолжение)

7.1. Способы и особенности реализации криптографических подсистем

Возможны два подхода к процессу криптографической защиты (в основном к шифрованию) объектов АС: предварительное и динамическое ("прозрачное") шифрование (без существенного ограничения общности можно выводы, касающиеся шифрования, распространить и на алгоритмы цифровой подписи).

Предварительное шифрование

Предварительное шифрование состоит в зашифровании файла некой программой (субъектом), а затем расшифровании тем же или иным субъектом (для расшифрования может быть применена та же или другая (специально для расшифрования) программа). Далее расшифрованный массив непосредственно используется прикладной программой пользователя. Данный подход имеет ряд недостатков, хотя и применяется достаточно широко.

Принципиальные недостатки метода предварительного шифрования:

- необходимость дополнительного ресурса для работы с зашифрованным объектом (дискового пространства – в случае расшифрования в файл с другим именем, или времени);
- потенциальная возможность доступа со стороны активных субъектов АС к расшифрованному файлу (во время его существования);
- необходимость задачи гарантированного уничтожения расшифрованного файла после его использования.

Динамическое шифрование

В последнее время широко применяется динамическое шифрование. Сущность динамического шифрования объектов АС состоит в следующем. Происходит

зашифрование всего файла (аналогично предварительному шифрованию). Затем с использованием специальных механизмов, обеспечивающих модификацию функций ПО АС, выполняющего обращения к объектам, ведется работа с зашифрованным объектом. При этом расшифрованию подвергается только та часть объекта, которая в текущий момент времени используется прикладной программой. При записи со стороны прикладной программы происходит зашифрование записываемой части объекта.

Данный подход позволяет максимально экономично использовать вычислительные ресурсы АС, поскольку расшифровывается только та часть объекта, которая непосредственно нужна прикладной программе. Кроме того, на внешних носителях информация всегда хранится в зашифрованном виде, что исключительно ценно с точки зрения невозможности доступа к ней. Динамическое шифрование целесообразно, таким образом, применять для защиты разделяемых удаленных или распределенных объектов АС.

Динамическое шифрование файлов необходимо рассматривать в контексте защиты группового массива файлов – каталога или логического диска.

При необходимости обращения к удаленным файлам АС на рабочей станции активизируется сетевое программное обеспечение, которое переопределяет функции работы с файловой системой ОС и тем самым с точки зрения рабочей станции создает единое файловое пространство рабочей станции и файла-сервера. Поскольку работа с файлами происходит через функции установленной на рабочей станции ОС, сетевое программное обеспечение модифицирует эти функции так, что обращение к ним со стороны прикладного уровня АС происходит так же, как и обычным образом. Это позволяет обеспечить нормальную работу прикладного и пользовательского уровня программного обеспечения рабочей станции АС.

Функции работы с файлами АС встраиваются в цепочку обработки файловых операций так, как показано на рисунке 7.1. Необходимо заметить, что модули 1-4 физически локализованы в оперативной памяти рабочей станции АС.

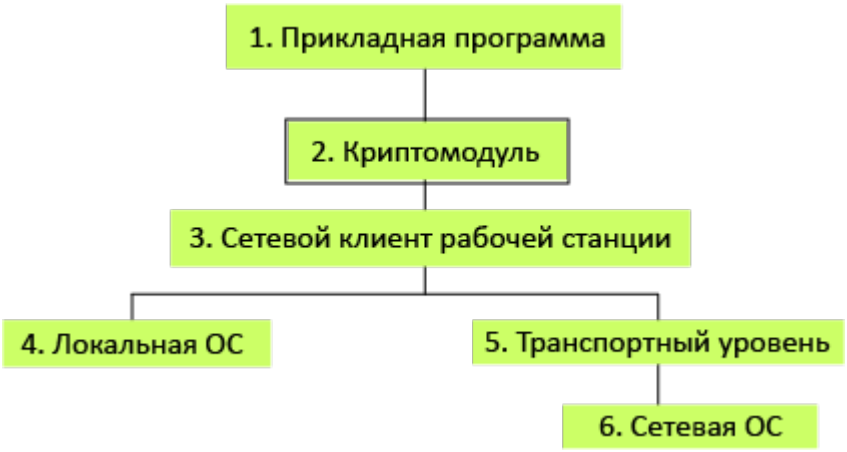

Рисунок 7.1 – Структура взаимодействия криптомодуля и ПО АС при файловом динамическом шифровании

Детализируем перечень обрабатываемых криптомодулем основных функций работы с файлами:

- создание файла;
- открытие файла;
- закрытие файла;
- чтение из открытого файла;
- запись в открытый файл.

Рассмотрим два основных потенциальных злоумышленных действия:

1. обращение к файлу на файл-сервере с рабочего места, не имею щего ключа расшифрования;
2. перехват информации в канале связи "рабочая станция-сервер".

Первое действие блокируется, поскольку шифрование информации происходит только в оперативной памяти рабочей станции АС и запись – считывание информации с диска файл-сервера или рабочей станции ведется только в шифрованном виде. По той же причине блокируется второе действие – обмен по транспортной системе "рабочая станция-сервер" проходит на уровнях 3-5, когда зашифрование уже закончено или расшифрование еще не произведено.

Можно показать, что метод динамического шифрования при условии инвариантности к прикладному программному обеспечению рабочей станции является оптимальным (обеспечивает минимальную вероятность доступа к незашифрованной информации) по сравнению с другими методами применения криптографических механизмов.

Принцип прикладного криптосервера

Некоторой модификацией описанного метода является принцип прикладного криптосервера. При этом методе выделяется активный аппаратный компонент АС (как правило, выделенная рабочая станция), которая имеет общий групповой ресурс со всеми субъектами, требующими исполнения криптографических функций. При создании файла, принадлежащего общему ресурсу, и записи в него автоматически происходит его зашифрование или фиксация целостности. Кроме того, в прикладном криптосервере может быть реализована функция изоляции защищенного объекта-файла, состоящая в его перемещении в выделенный групповой массив (директория "исходящих" файлов). Процесс обратного преобразования (или проверки целостности) происходит аналогичным образом в других выделенных массивах.

Для субъекта рабочей станции этот процесс выглядит как автоматическое зашифрование (или интеграция цифровой подписи в файл) при записи в некоторую заранее указанную директорию на файловом сервере и появление зашифрованного файла в другой директории.

Подход прикладного криптосервера широко применяется для криптографической защиты электронных файлов документов в гетерогенной АС или для сопряжения с телекоммуникационными системами.

7.2. Криптографическая защита транспортного уровня АС

При анализе защиты транспортного уровня АС необходимо учитывать свойство, следующее из иерархической модели взаимодействия открытых систем: передача информации от верхних уровней представления (файлов) к нижним уровням (пакетам) полностью и без изменения сохраняет содержательную часть информации. Отсюда следует, что шифрование файлов, сопряженное с файловыми операциями (рассмотренное выше), приводит к прохождению информационных частей пакета, полученного из зашифрованного файла, на транспортном уровне уже в зашифрованном виде. И наоборот, процедуры шифрования, локализованные на транспортном уровне, получают и передают информацию в верхние уровни представления в открытом виде.

Данный факт позволяет определить случаи применения шифрования транспортного уровня:

- в АС с прохождением кабельной системы по территории, доступной для злоумышленника;
- при невозможности взаимодействия прикладных задач пользователя с системой динамического файлового шифрования;
- при отсутствии необходимости шифрования локальных ресурсов.

Криптозащита транспортного уровня (см. рисунок 7.2) может быть реализована программно при встраивании в информационные потоки сетевых программных средств и аппаратно на стыке "рабочая станция-сетевые средства" (уровень 4 на рисунке 7.2) или "рабочая станция-кабельная система" (уровень 6 на рисунке 7.2), в этом случае можно говорить о наложенных криптосредствах). В зарубежной литературе подобного рода аппаратура именуется криптобоксом или модулем безопасности (SAM). Наложенные криптосредства понимаются в смысле полной независимости их функционирования от прикладного и системного программного наполнения АС.

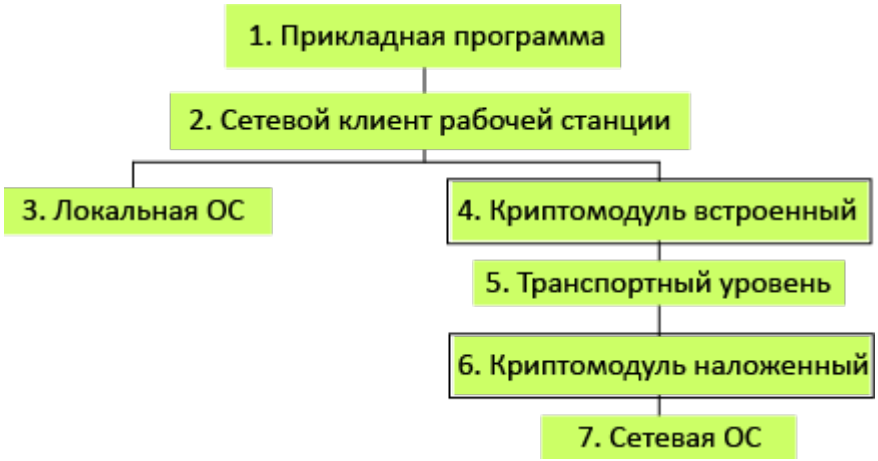

Рисунок 7.2 – Встраивание средств криптозащиты транспортного уровня

Криптографическая защита транспортного уровня, как было отмечено, прозрачно пропускает информацию прикладного уровня, в связи с чем не защищает от специфических угроз уровня взаимодействия операционной среды и прикладного взаимодействия (влияние прикладных программ на зашифрованные файлы и на программы шифрования).

7.3. Криптографическая защита на прикладном уровне АС

Криптографическая защита информации на прикладном уровне является наиболее предпочтительным вариантом защиты информации с точки зрения гибкости защиты, но наиболее сложным по программно-аппаратной реализации.

Криптографическая защита информации на прикладном уровне (или криптографическая защита прикладного уровня) – это такой порядок проектирования, реализации и использования криптографических средств, при котором входная и выходная информация и, возможно, ключевые параметры принадлежат потокам и объектам прикладного уровня (в модели ISO взаимодействия открытых систем).

Информация, находящаяся на нижестоящих иерархических уровнях модели ISO (далее используется термин "нижестоящие уровни") относительно объекта прикладного уровня представляет собой подобъекты данного объекта, рассматриваемые, как правило, изолированно друг от друга. В связи с этим на нижестоящих уровнях (сетевом и ниже) невозможно достоверно распознавать, а следовательно, и защищать криптографическими методами объекты сложной структуры типа "электронный документ" или поле базы данных. На нижестоящих уровнях данные объекты представляются последовательностью вмещающих подобъектов типа "пакет".

Кроме того, только на прикладном уровне возможна персонализация объекта, т.е. однозначное сопоставление созданного объекта породившему его субъекту (субъектом прикладного уровня является, как правило, 48 прикладная программа, управляемая человеком-пользователем). Субъекты нижестоящих уровней снабжают атрибутами порождаемую ими последовательность объектов (подобъектов объекта прикладного уровня) в основном адресом (информацией, характеризующей субъекта нижестоящего уровня – компьютер), который является лишь опосредованной характеристикой породившего информацию субъекта прикладного уровня.

В то же время необходимо отметить свойство наследования логической защиты вышестоящего уровня для нижестоящих. Поясним данное свойство примером. Предположим, на прикладном уровне зашифровано поле базы данных, при передаче информации по сети происходит ее преобразование на нижестоящий сетевой уровень. При этом поле будет передано в зашифрованном виде, разобщено (в смысле выполнения операции декомпозиции объекта на подобъекты) на последовательность пакетов, информационное поле каждого из которых также зашифровано, и затем передано по транспортной системе локальной или глобальной телекоммуникационной сети в виде датаграмм с зашифрованным информационным полем (адрес не будет закрыт, поскольку субъект нижестоящего уровня, который произвел декомпозицию, не имеет информации о функции преобразования объекта).

Следовательно, криптографическая защита объекта прикладного уровня действительна и для всех нижестоящих уровней.

Из указанного свойства следует:

Утверждение 1

При защите информации на прикладном уровне процедуры передачи, разборки на пакеты, маршрутизации и обратной сборки не могут нанести ущерба конфиденциальности информации.

Упомянув о понятии конфиденциальности, нельзя не отметить, что две классические задачи криптографической защиты: защита конфиденциальности и защита целостности инвариантны относительно любого уровня модели ISO (с учетом свойства наследования). Защита прикладного уровня также в основном решает две указанные задачи отдельно или в совокупности.

Особенностью существования субъектов-программ прикладного уровня, а также порождаемых ими объектов является отсутствие стандартизованных форматов представления объектов. Более того, можно утверждать, что такая стандартизация возможна лишь для отдельных структурных компонентов субъектов и объектов прикладного уровня (например, типизация данных в транслируемых и интерпретируемых языках программирования, форматы результирующего хранения для текстовых процессоров и др.). Субъекты и объекты прикладных систем создаются пользователем, и априорно задать их структуру не представляется возможным.

Можно рассмотреть два подхода к построению СКЗИ на прикладном уровне. Первый подход (наложенные СКЗИ) связан с реализацией функций криптографической защиты целиком в отдельном субъекте-программе (например, после подготовки электронного документа в файле активизируется программа цифровой подписи для подписания данного файла). Данный подход получил также название абонентской защиты (поскольку активизация программы производится оконечным пользователем-абонентом и локализуется в пределах рабочего места пользователя). Второй подход (встраивание СКЗИ) связан с вызовом функций субъекта СКЗИ непосредственно из программы порождения защищаемых объектов и встраиванием криптографических функций в прикладную программу.

Первый подход отличается простотой реализации и применения, но требует учета двух важных факторов. Во-первых, реализация субъекта СКЗИ должна быть в той же операционной среде либо операционной среде, связанной потоками информации с той, в которой существует прикладной субъект. Во-вторых, и прикладной субъект, и СКЗИ должны воспринимать объекты АС (т.е., декомпозиция компьютерной системы на объекты должна быть общей для обоих субъектов). Два вышеуказанных фактора предполагают раздельную реализацию СКЗИ в операционной среде и связь по данным. С другой стороны, в современных системах обработки и передачи информации достаточно сложно произвести пространственно-временную локализацию порождения конечного объекта, который должен подвергаться защите. В связи с этим современные информационные технологии предполагают более широкое использование

второго подхода (встраивание СКЗИ), используя для этого различные технические решения.

Сравним оба рассмотренных подхода построения СКЗИ на прикладном уровне в таблице 7.1.

Таблица 7.1

Свойства СКЗИ	**Первый подход**	**Второй подход**
Сопряжение с прикладной подсистемой	На этапе эксплуатации	На этапе проектирования и разработки
Зависимость от прикладной системы	Низкая	Высокая
Локализация защищаемого объекта	Внешняя (относительно защитного модуля и прикладной программы)	Внутренняя (защита внутреннего объекта прикладной программы)
Операционная зависимость	Полная	Низкая

Можно выделить несколько способов реализации криптографической защиты в отдельном субъекте.

1. Локальная реализация в виде выделенной прикладной программы.
    - 1.1. Локальная реализация в базовой АС.
    - 1.2. Локальная реализация в "гостевой" АС.
    - 1.3. Локальная реализация по принципу "копирование в защищенный объект хранения".
2. Распределенная реализация по технологии "создание и запись в защищенной области".

Подходы 1.3 и 2, как правило, называют реализацией в виде локального или распределенного прикладного криптосервера. Сущность их реализации была рассмотрена выше.

Встраивание криптографических функций в прикладную систему может осуществляться:

- по технологии "открытый интерфейс";
- по технологии "криптографический сервер";
- на основе интерпретируемого языка прикладного средства.

Основной проблемой встраивания является корректное использование вызываемых функций.

Субъекты АС, связанные с выполнением защитных функция (например, субъекты порождения изолированной программной среды могут использовать некоторое общее подмножество криптографических функций логического преобразования объектов (в частности, алгоритмы контроля целостности объектов). При проектировании АС исторически сложившийся подход относительно распределения общего ресурса связан с использованием разделяемых субъектов, выполняющих общие для других субъектов функции. Распространим данный подход на функции реализации логической защиты.

Разделяемая технология применения функций логической защиты -это такой порядок использования СКЗИ в защищенной АС, при котором:

- не требуются изменения в программном обеспечении при изменении криптографических алгоритмов;
- система защиты однозначно разделяема на две части: прикладная компонента и модуль реализации криптографических функций (МРКФ).

Открытый интерфейс (ОИ) МКРФ – детальное специфицирование функций, реализованных в МРКФ.

Относительно некоторого множества субъектов, использующих МРКФ, можно говорить о полноте функций МРКФ. Удобнее оперировать с формально описанными функциями ОИ, следовательно, далее будем говорить о полноте функций ОИ. Полнота функций ОИ может быть функциональной и параметрической.

Функциональная полнота ОИ – свойство, заключающееся в реализации всех функций класса функций защиты, инициируемых фиксированным набором субъектов АС.

Из данного определения следует, что функциональная полнота понимается относительно заданного множества программ, использующих функции ОИ.

Параметрическая полнота ОИ – свойство, заключающееся в возможности инициирования всех функций ОИ со стороны фиксированного множества субъектов с некоторым набором параметров, не приводящих к отказу в выполнении запрошенной функции.

Взаимодействие субъектов прикладного уровня с МРКФ есть взаимодействие типа "субъект-субъект". Следовательно, основной источник угроз системе состоит в некорректном взаимодействии субъекта с МРКФ.

Корректное использование МРКФ – такой порядок взаимодействия МРКФ с некоторым субъектом (далее будем называть его "вызывающим субъектом" или "использующим субъектом"), при котором выполняются следующие условия:

1. МРКФ и использующий его субъект корректны относительно друг друга;
2. результат выполнения функций МРКФ соответствует их описанию в ОИ;
3. поток информации от ассоциированных объектов вызывающего субъекта направлен только к ассоциированным объектам МРКФ и функция изменения ассоциированных объектов МРКФ, отвечающих передаче параметров, есть тождественное отображение соответствующих объектов (условие передачи параметров без изменения);
4. вышеуказанные свойства выполнены в любой момент времени существования МРКФ и вызывающего субъекта.

Предположим, что МРКФ протестирован и выполнение всех его функций соответствует описанию их в ОИ.

Утверждение 2

Условие 2 корректного использования МРКФ эквивалентно неизменности всех ассоциированных с ним объектов, не принадлежащих вызываемому субъекту.

Доказательство

Поскольку множество ассоциированных объектов, не принадлежащих вызываемому субъекту, описывает функции преобразования информации, реализуемые в МРКФ, то их неизменность предполагает и неизменность выполнения описанных в ОИ функций.

Утверждение 3

Для выполнения условий корректного использования достаточно:

- отсутствие потока от любого субъекта, отличного от вызывающего к ассоциированным объектам, принадлежащим как вызывающему субъекту, так и МРКФ, т.е. корректности всех существующих субъектов относительно как МРКФ, так и вызывающего субъекта;
- отсутствии потоков от вызывающего субъекта к другим субъектам.

Доказательство

Верность утверждения непосредственно следует из определения корректности субъектов относительно друг друга.

Утверждение 4

Достаточным условием корректности использования МРКФ является работа АС в условиях изолированной программной среды.

В настоящее время подход встраивания СКЗИ по технологии "открытого интерфейса" применен в операционных средах MS Windows NT4.0 в виде так называемого криптопровайдера (CryptoAPI 1.0 и 2.0).

Достаточно перспективным является подход к реализации криптографических функций на прикладном уровне при помощи интерпретируемых языков. Сущность данного подхода состоит в том, что с ассоциированными объектами прикладного субъекта, требующими выполнения криптографических преобразований производятся операции с использованием функций, реализованных в самом субъекте. Как правило, механизмы преобразования внутренних объектов реализованы на базе интерпретируемого языка (типа Basic). Преимуществом данного подхода является замкнутость относительно воздействия других субъектов, отсутствие необходимости использования внешнего субъекта (типа МРКФ), встроенных механизмов корректной реализации потоков информации в рамках субъекта прикладного уровня, а также потоков уровня межсубъектного взаимодействия. Основным недостатком является низкое быстродействие.

Практически идеальным языком программирования криптографических функций в субъекте прикладного уровня является язык JAVA. Данный язык имеет развитые встроенные средства работы с объектами прикладного уровня, но при этом широкие возможности для реализации криптографических преобразований (элементарные логические и арифметические операции с числами, работа с матрицами и т.д.). Однако необходимо обратить внимание на проблему реализации программного датчика случайных чисел, безусловно необходимого ряду СКЗИ (в частности, цифровой подписи).

7.4. Особенности сертификации и стандартизации криптографических средств

Процесс синтеза и анализа СКЗИ отличается высокой сложностью и трудоемкостью, поскольку необходим всесторонний учет влияния перечисленных выше угроз на надежность реализации СКЗИ. В связи с этим практически во всех странах, обладающих развитыми криптографическими технологиями, разработка СКЗИ относится к сфере государственного регулирования. Государственное регулирование включает, как правило, лицензирование деятельности, связанной с разработкой и эксплуатацией криптографических средств, сертификацию СКЗИ и стандартизацию алгоритмов криптографических преобразований.

В России в настоящее время организационно-правовые и научно-технические проблемы синтеза и анализа СКЗИ находятся в компетенции ФАПСИ при Президенте Российской Федерации.

Правовая сторона разработки и использования СКЗИ регламентируется в основном указом Президента Российской Федерации от 03.04.95 № 334 с учетом принятых ранее законодательных и нормативных актов РФ.

Дополнительно учитываемой законодательной базой являются законы: "О федеральных органах правительственной связи и информации", "О государственной тайне", "Об информации, информатизации и защите информации", "О сертификации продукции и услуг".

Лицензированию подлежат следующие виды деятельности:

- Разработка, производство, проведение сертификационных испытаний, реализация, эксплуатация шифровальных средств, предназначенных для криптографической защиты информации, содержащей сведения, составляющие государственную или иную охраняемую законом тайну, при ее обработке, хранении и передаче по каналам связи, а также предоставление услуг в области шифрования этой информации.
- Разработка, производство, проведение сертификационных испытаний, эксплуатация систем и комплексов телекоммуникаций высших органов государственной власти Российской Федерации.
- Разработка, производство, проведение сертификационных испытаний, реализация, эксплуатация закрытых систем и комплексов телекоммуникаций органов власти субъектов Российской Федерации, центральных органов федеральной исполнительной власти, организаций, предприятий, банков и иных учреждений, расположенных на территории Российской Федерации, независимо от их ведомственной принадлежности и форм собственности (далее – закрытых систем и комплексов телекоммуникаций), предназначенных для передачи информации, составляющей государственную или иную охраняемую законом тайну.
- Проведение сертификационных испытаний, реализация и эксплуатация шифровальных средств, закрытых систем и комплексов телекоммуникаций, предназначенных для обработки информации, не содержащей сведений, составляющих государственную или иную охраняемую законом тайну, при ее обработке, хранении и передаче по каналам связи, а также предоставление услуг в области шифрования этой информации. К шифровальным средствам относятся:

- Реализующие криптографические алгоритмы преобразования информации аппаратные, программные и аппаратно-программные средства, обеспечивающие безопасность информации при ее обработке, хранении и передаче по каналам связи, включая шифровальную технику.
- Реализующие криптографические алгоритмы преобразования информации аппаратные, программные и аппаратно-программные средства защиты от несанкционированного доступа к информации при ее обработке и хранении.
- Реализующие криптографические алгоритмы преобразования информации аппаратные, программные и аппаратно-программные средства защиты от навязывания ложной информации, включая средства имитозащиты и "цифровой подписи".
- Аппаратные, аппаратно-программные и программные средства для изготовления ключевых документов к шифровальным средствам независимо от вида носителя ключевой информации.

К закрытым системам и комплексам телекоммуникаций относятся системы и комплексы телекоммуникаций, в которых обеспечивается защита информации с использованием шифровальных средств, защищенного оборудования и организационных мер.

Дополнительно лицензированию подлежат следующие виды деятельности:

- эксплуатация шифровальных средств и/или средств цифровой подписи, а также шифровальных средств для защиты электронных платежей с использованием пластиковых кредитных карточек и смарт-карт;
- оказание услуг по защите (шифрованию) информации;
- монтаж, установка, наладка шифровальных средств и/или средств цифровой подписи, шифровальных средств для защиты электронных платежей с использованием пластиковых кредитных карточек и смарт-карт;
- разработка шифровальных средств и/или средств цифровой подписи, шифровальных средств для защиты электронных платежей с использованием пластиковых кредитных карточек и смарт-карт.

Порядок сертификации СКЗИ установлен "Системой сертификации средств криптографической защиты информации РОСС.RU.0001.030001 Госстандарта России.

Стандартизация алгоритмов криптографических преобразований включает всесторонние исследования и публикацию в виде стандартов элементов криптографических процедур с целью использования разработчиками СКЗИ

апробированных криптографически стойких преобразований, обеспечения возможности совместной работы различных СКЗИ, а также возможности тестирования и проверки соответствия реализации СКЗИ заданному стандартом алгоритму.

В России приняты следующие стандарты – алгоритм криптографического преобразования 28147-89, алгоритмы хеширования, простановки и проверки цифровой подписи Р34.10.94 и Р34.11.94. Из зарубежных стандартов широко известны и применяются алгоритмы шифрования DES, RC2, RC4, алгоритмы хеширования MD2, MD4 и MD5, алгоритмы простановки и проверки цифровой подписи DSS и RSA.

Подводя итоги, можно указать примерные пути развития информационных технологий в рамках защищенных АС, опирающиеся на применение криптографических механизмов.

1. Ориентация на шифрование группового массива данных и защиту целостности отдельных объектов АС типа исполняемые модули (из которых происходит порождение субъектов).
2. Полное управление защитой только со стороны администратора сети. Реализация процедур изоляции ключей пользователей от администратора программными мерами (существование ключей только внутри программ и их уничтожение сразу после использования, хранение ключей в объектах АС, недоступных обычным пользователям).
3. Работа администратора с выделенной рабочей станции с реализацией принципов централизованного управления СКЗИ.
4. Разделяемость клиентской части СКЗИ (рабочая станция) на отдельные модули и индивидуальные структуры данных для этих модулей.
5. Администрирование сетевых криптографически защищенных ресурсов преимущественно без участия владельцев индивидуальных ключей. Возможность установки защиты самим пользователем.
6. Локальная интероперабельность – реализация защитных механизмов СКЗИ для некоторого подмножества файлово-совместимых ОС на рабочей станции и любого ПО на сервере.
7. Существование у пользователя индивидуальных данных (возможность этого задается дополнительными атрибутами защиты-право порождения ключа для защиты локальных данных, возможность отключения администратора СКЗИ от своего криптографически защищенного объекта, возможность создания объекта индивидуального доступа, право допустить к объекту другого пользователя.

8. Использование открытого интерфейса криптографических функций. Организация универсальных структур данных под переменные длины ключей и имитовставок.
9. Реализация процедуры единого выхода во внешнюю сеть через почтовую СКЗИ-шлюз, либо шифрование объектов транспортной системы АС.
10. Снабжение клиента пользовательским интерфейсом (через функции открытого интерфейса), с помощью которого он может встраивать в свои приложения (в основном клиенты СУБД) криптографические функции.

На основании изложенных выше положений можно сделать вывод о том, что перспективные технологии применения криптографической защиты развиваются в направлении применения открытых интерфейсов криптографических функций, а также использования собственных вычислительных ресурсов прикладных средств.

Тема 8. Алгоритмы шифрования

8.1. Принципы криптографической защиты информации

Криптография – совокупность методов преобразования данных, направленных на то, чтобы сделать эти данные бесполезными для противника.

Такие преобразования позволяют решить две главные проблемы защиты данных: проблему конфиденциальности (путем лишения противника возможности извлечь информацию из канала связи) и проблему целостности (путем лишения противника возможности изменить сообщение так, чтобы изменился его смысл, или ввести ложную информацию в канал связи).

Проблемы конфиденциальности и целостности информации тесно связаны между собой, поэтому методы решения одной из них часто применимы для решения другой.

Обобщенная схема криптографической системы, обеспечивающей шифрование передаваемой информации, показана на рисунок 8.1. Отправитель генерирует открытый текст исходного сообщения M, которое должно быть передано законному получателю по незащищенному каналу. За каналом следит перехватчик с целью перехватить и раскрыть передаваемое сообщение. Для того чтобы перехватчик не смог узнать содержание сообщения M, отправитель шифрует его с помощью обратимого преобразования E_K и получает шифртекст (или криптограмму) $C = E_K(M)$, который отправляет получателю.

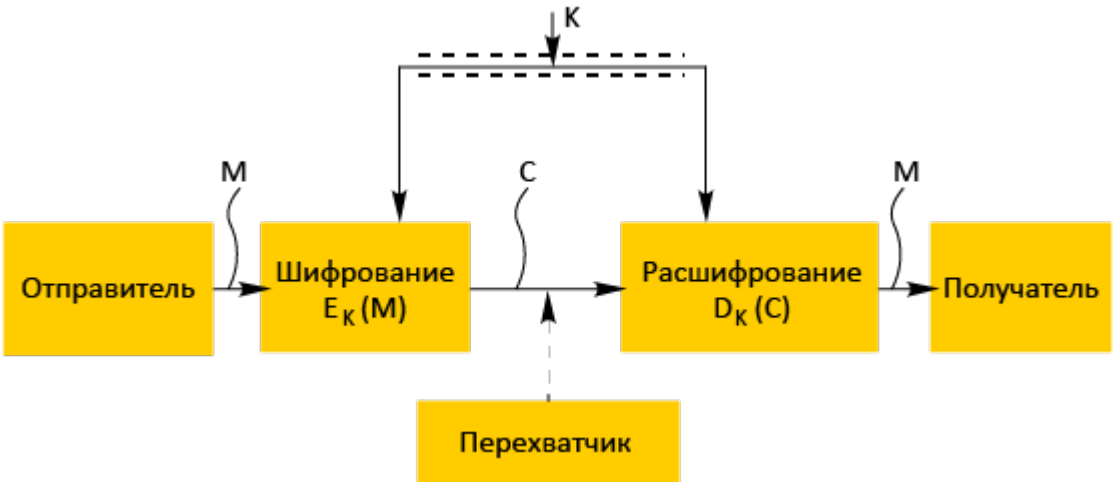

Рисунок 8.1 – Обобщенная схема криптосистемы

Законный получатель, приняв шифртекст C, расшифровывает его с помощью обратного преобразования $D = E_K^{-1}$ и получает исходное сообщение в виде открытого текста M:

$$D_K(C) = E_K^{-1}(E_K(M)) = M$$

Преобразование E_K выбирается из семейства криптографических преобразований, называемых криптоалгоритмами. Параметр, с помощью которого выбирается отдельное используемое преобразование, называется криптографическим ключом K. Криптосистема имеет разные варианты реализации: набор инструкций, аппаратные средства, комплекс программ компьютера, которые позволяют зашифровать открытый текст и расшифровать шифр-текст различными способами, один из которых выбирается с помощью конкретного ключа K.

Говоря более формально, криптографическая система – это однопараметрическое семейство $(E_K)_{K \in \overline{K}}$ обратимых преобразований

$$E_K : \overline{M} \to \overline{C}$$

из пространства M сообщений открытого текста в пространство C шифрованных текстов. Параметр K (ключ) выбирается из конечного множества K, называемого пространством ключей.

Вообще говоря, преобразование шифрования может быть симметричным или асимметричным относительно преобразования расшифрования. Это важное свойство функции преобразования определяет два класса криптосистем:

- симметричные (одноключевые) криптосистемы;
- асимметричные (двухключевые) криптосистемы (с открытым ключом).

Схема симметричной криптосистемы с одним секретным ключом была показана на рисунке 8.1. В ней используются одинаковые секретные ключи в блоке шифрования и блоке расшифрования.

Обобщенная схема асимметричной криптосистемы с двумя разными ключами K_1 и K_2 показана на рисунке 8.2. В этой криптосистеме один из ключей является открытым, а другой – секретным.

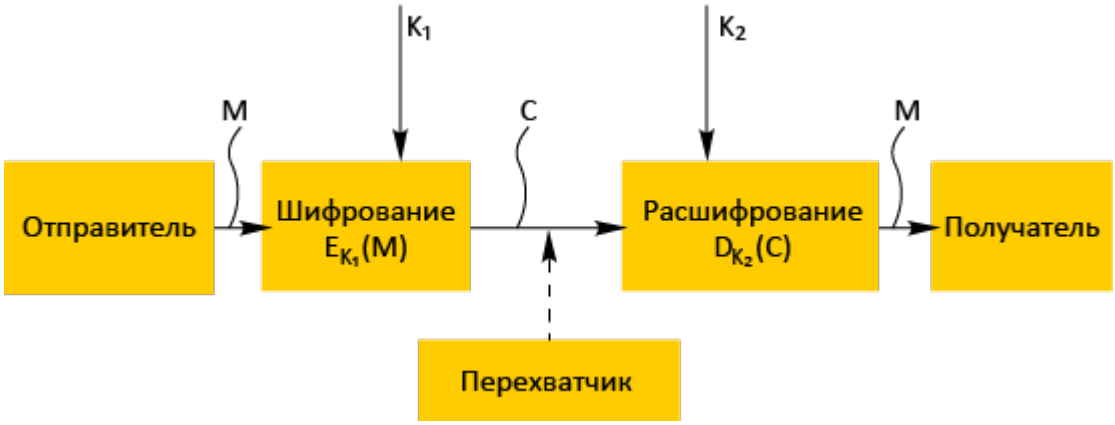

Рисунок 8.2 – Обобщенная схема асимметричной криптосистемы с открытым

В симметричной криптосистеме секретный ключ надо передавать отправителю и получателю по защищенному каналу распространения ключей, например такому, как курьерская служба. На рисунке 8.1 этот канал показан "экранированной" линией. Существуют и другие способы распределения секретных ключей, они будут рассмотрены позднее. В асимметричной криптосистеме передают по незащищенному каналу только открытый ключ, а секретный ключ сохраняют на месте его генерации.

На рисунке 8.3 показан поток информации в криптосистеме в случае активных действий перехватчика. Активный перехватчик не только считывает все шифртексты, передаваемые по каналу, но может также пытаться изменять их по своему усмотрению.

Любая попытка со стороны перехватчика расшифровать шифртекст C для получения открытого текста M или зашифровать свой собственный текст M' для получения правдоподобного шифртекста C, не имея подлинного ключа, называется крипто-аналитической атакой.

Если предпринятые криптоаналитические атаки не достигают поставленной цели и криптоаналитик не может, не имея подлинного ключа, вывести M из C или C' из M', то полагают, что такая криптосистема является криптостойкой.

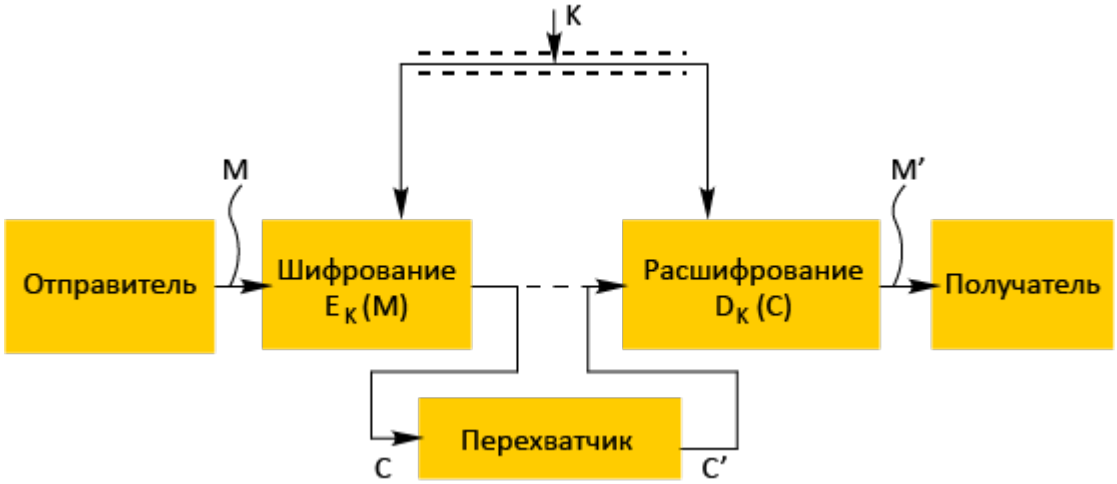

Рисунок 8.3 – Поток информации в криптосистеме при активном перехвате сообщений

Криптоанализ – это наука о раскрытии исходного текста зашифрованного сообщения без доступа к ключу.

Успешный анализ может раскрыть исходный текст или ключ. Он позволяет также обнаружить слабые места в криптосистеме, что, в конечном счете, ведет к тем же результатам.

Фундаментальное правило криптоанализа, впервые сформулированное голландцем А.Керкхоффом еще в XIX веке заключается в том, что стойкость шифра (криптосистемы) должна определяться только секретностью ключа. Иными словами, правило Керкхоффа состоит в том, что весь алгоритм шифрования, кроме значения секретного ключа, известен криптоаналитику противника. Это обусловлено тем, что криптосистема, реализующая семейство криптографических преобразований, обычно рассматривается как открытая система. Такой подход отражает очень важный принцип технологии защиты информации: защищенность системы не должна зависеть от секретности чего-либо такого, что невозможно быстро изменить в случае утечки секретной информации. Обычно криптосистема представляет собой совокупность аппаратных и программных средств, которую можно изменить только при значительных затратах времени и средств, тогда как ключ является легко изменяемым объектом. Именно поэтому стойкость криптосистемы определяется только секретностью ключа.

Другое почти общепринятое допущение в криптоанализе состоит в том, что криптоаналитик имеет в своем распоряжении шифртексты сообщений.

Существует четыре основных типа криптоаналитических атак. Конечно, все они формулируются в предположении, что криптоаналитику известны применяемый алгоритм шифрования и шифртексты сообщений.

Перечислим эти криптоаналитические атаки.

1. Криптоаналитическая атака при наличии только известного шифртекста.

Криптоаналитик имеет только шифртексты C_1, C_2, ..., C_iнескольких сообщений, причем все они зашифрованы с использованием одного и того же алгоритма шифрования E_K. Работа криптоаналитика заключается в том, чтобы раскрыть исходные тексты M_1, M_2, ..., M_i по возможности большинства сообщений или, еще лучше, вычислить ключ K, использованный для шифрования этих сообщений, с тем, чтобы расшифровать и другие сообщения, зашифрованные этим ключом.

2. Криптоаналитическая атака при наличии известного открытого текста

Криптоаналитик имеет доступ не только к шифртекстам C_1, C_2, ..., C_i нескольких сообщений, но также к открытым текстам M_1, M_2, ..., M_i этих сообщений. Его работа заключается в нахождении ключа K, используемого при шифровании этих сообщений, или алгоритма расшифрования D_K любых новых сообщений, зашифрованных тем же самым ключом.

3. Криптоаналитическая атака при возможности выбора открытого текста

Криптоаналитик не только имеет доступ к шифртекстам C_1, C_2, ..., C_i и связанным с ними открытым текстам M_1, M_2, ..., M_i нескольких сообщений, но и может по желанию выбирать открытые тексты, которые затем получает в зашифрованном виде. Такой криптоанализ получается более мощным по сравнению с криптоанализом с известным открытым текстом, потому что криптоаналитик может выбрать для шифрования такие блоки открытого текста, которые дадут больше информации о ключе. Работа криптоаналитика состоит в поиске ключа K, использованного для шифрования сообщений, или алгоритма расшифрования D_K новых сообщений, зашифрованных тем же ключом.

4. Криптоаналитическая атака с адаптивным выбором открытого текста

Это особый вариант атаки с выбором открытого текста. Криптоаналитик может не только выбирать открытый текст, который затем шифруется, но и изменять свой выбор в зависимости от результатов предыдущего шифрования. При криптоанализе с простым выбором открытого текста криптоаналитик Обычно может выбирать несколько крупных блоков открытого тек ста для их шифрования; при криптоанализе с адаптивным выбором открытого текста он имеет возможность выбрать сначала более мелкий пробный блок открытого текста, затем выбрать следующий блок в зависимости от результатов первого

выбора, и т.д. Эта атака предоставляет криптоаналитику еще больше возможностей, чем предыдущие типы атак.

Кроме перечисленных основных типов криптоаналитических атак, можно отметить, по крайней мере, еще два типа.

5. Криптоаналитическая атака с использованием выбранного шифртекста

Криптоаналитик может выбирать для расшифрования различные шифртексты C_1, C_2, ..., C_i и имеет доступ к расшифрованным открытым текстам M_1, M_2, ..., M_i. Например, криптоаналитик получил доступ к защищенному от несанкционированного вскрытия блоку, который выполняет автоматическое расшифрование. Работа криптоаналитика заключается в нахождении ключа. Этот тип криптоанализа представляет особый интерес для раскрытия алгоритмов с открытым ключом.

6. Криптоаналитическая атака методом полного перебора всех возможных ключей

Эта атака предполагает использование криптоаналитиком известного шифртекста и осуществляется посредством полного перебора всех возможных ключей с проверкой, является ли осмысленным получающийся открытый текст. Такой подход требует привлечения предельных вычислительных ресурсов и иногда называется силовой атакой.

Существуют и другие, менее распространенные, криптоаналитические атаки, некоторые из них будут описаны в соответствующих разделах книги.

8.2. Основные понятия и определения

Большинство средств защиты информации базируется на использовании криптографических шифров и процедур шифрования-расшифрования. В соответствии со стандартом ГОСТ 28147-89 под шифром понимают совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, задаваемых ключом и алгоритмом криптографического преобразования.

Ключ – это конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор только одного варианта из всех возможных для данного алгоритма.

Основной характеристикой шифра является криптостойкость, которая определяет его стойкость к раскрытию методами криптоанализа. Обычно эта характеристика определяется интервалом времени, необходимым для раскрытия шифра.

К шифрам, используемым для криптографической защиты информации, предъявляется ряд требований:

- достаточная криптостойкость (надежность закрытия данных);
- простота процедур шифрования и расшифрования;
- незначительная избыточность информации за счет шифрования;
- нечувствительность к небольшим ошибкам шифрования и др. В той или иной мере этим требованиям отвечают:
- шифры перестановок;
- шифры замены;
- шифры гаммирования;
- шифры, основанные на аналитических преобразованиях шифруемых данных.

Шифрование перестановкой заключается в том, что символы шифруемого текста переставляются по определенному правилу в пределах некоторого блока этого текста. При достаточной длине блока, в пределах которого осуществляется перестановка, и сложном неповторяющемся порядке перестановки можно достигнуть приемлемой для простых практических приложений стойкости шифра.

Шифрование заменой (подстановкой) заключается в том, что символы шифруемого текста заменяются символами того же или другого алфавита в соответствии с заранее обусловленной схемой замены.

Шифрование гаммированием заключается в том, что символы шифруемого текста складываются с символами некоторой случайной последовательности, именуемой гаммой шифра. Стойкость шифрования определяется в основном длиной (периодом) неповторяющейся части гаммы шифра. Поскольку с помощью ЭВМ можно генерировать практически бесконечную гамму шифра, то данный способ является одним из основных для шифрования информации в автоматизированных системах.

Шифрование аналитическим преобразованием заключается в том, что шифруемый текст преобразуется по некоторому аналитическому правилу (формуле).

Например: можно использовать правило умножения вектора на матрицу, причем умножаемая матрица является ключом шифрования (поэтому ее размер и содержание должны храниться в секрете), а символами умножаемого вектора последовательно служат символы шифруемого текста. Другим примером может служить использование так называемых однонаправленных функций для построения криптосистем с открытым ключом (см. главу 5).

Процессы шифрования и расшифрования осуществляются в рамках некоторой криптосистемы. Характерной особенностью симметричной криптосистемы является применение одного и того же секретного ключа как при шифровании, так и при расшифровании сообщений.

Как открытый текст, так и шифртекст образуются из букв, входящих в конечное множество символов, называемых алфавитом. Примерами алфавитов являются конечное множество всех заглавных букв, конечное множество всех заглавных и строчных букв и цифр и т. п. В общем виде некоторый алфавит можно представить так:

Объединяя по определенному правилу буквы из алфавита , можно создать новые алфавиты:

- алфавит , содержащий m^2 биграммаa_0a_0, a_0a_1, ..., $a_{m-1}a_{m-1}$;
- алфавит , содержащий m^3 триграмм $a_0a_0a_0$, $a_0a_0a_1$, ..., $a_{m-1}a_{m-1}a_{m-1}$.

В общем случае, объединяя по п букв, получаем алфавит , содержащий m^nn-грамм.

Например, английский алфавит объемом m = 26 букв позволяет сгенерировать посредством операции конкатенации алфавит из 26^2 = 676 биграмм AA, AB,...,XZ, ZZ, алфавит из 26^3 = 17576 триграмм AAA, AAB, ..., ZZX, ZZZ и т.д.

При выполнении криптографических преобразований полезно заменить буквы алфавита целыми числами 0, 1, 2, 3, ... Это позволяет упростить выполнение необходимых алгебраических манипуляций. Например, можно установить взаимно однозначное соответствие между русским алфавитом и множеством целых ; между английским алфавитом и множеством целых (см. таблицы 8.1 и 8.2). В дальнейшем будет обычно использоваться алфавит , содержащий m "букв" (в виде чисел).

Замена букв традиционного алфавита числами позволяет более четко сформулировать основные концепции и приемы криптографических

преобразований. В то же время в большинстве иллюстраций будет использоваться алфавит естественного языка.

Таблица 8.1 – Соответствие между русским алфавитом и множеством целых

Буква	**Число**	**Буква**	**Число**	**Буква**	**Число**	**Буква**	**Число**
А	0	И	8	Р	16	Ш	24
Б	1	И	9	С	17	Щ	25
В	2	К	10	Т	18	b	26
Г	3	Л	11	У	19	Ы	27
Д	4	М	12	Ф	20	Ъ	28
Е	5	Н	13	Х	21	Э	29
Ж	6	О	14	Ц	22	Ю	30
З	7	П	15	Ч	23	Я	31

Таблица 8.2 – Соответствие между английским алфавитом и множеством целых

Буква	**Число**	**Буква**	**Число**	**Буква**	**Число**
A	0	J	9	S	18
B	1	K	10	T	19
C	2	L	11	U	20
D	3	V	12	V	21
E	4	N	13	W	22
F	5	O	14	X	23
G	6	P	15	У	24
H	7	Q	16	Z	25
1	8	R	17		

Текст с n буквами из алфавита Z_m можно рассматривать как n-грамму:

где , для некоторого целого n = 1, 2, 3,...

Через будем обозначать множество n-грамм, образованных из букв множества .

Криптографическое преобразование E представляет собой совокупность преобразований:

(1)

Преобразование $E^{(n)}$ определяет, как каждая n-грамма открытого текстах $\bar{x} \in \overline{Z_{m,n}}$ заменяется n-граммой шифртекста $\bar{y}$, т.е. $\bar{y} = E^{(n)}(\bar{x})$, причем $\bar{x}, \bar{y} \in \overline{Z_{m,n}}$, при этом обязательным является требование взаимной однозначности преобразования $E^{(n)}$ на множестве $\overline{Z_{m,n}}$.

Криптографическая система может трактоваться как семейство криптографических преобразований:

$\overline{E} = \left\{ E_k : K \in \overline{K} \right\}$, (2)

помеченных параметром K, называемым ключом.

Множество значений ключа образует ключевое пространство $\overline{K}$.

Далее рассматриваются традиционные (классические) методы шифрования, отличающиеся симметричной функцией шифрования. К ним относятся шифры перестановки, шифры простой и сложной замены, а также некоторые их модификации и комбинации. Следует отметить, что комбинации шифров перестановок и шифров замены образуют все многообразие применяемых на практике симметричных шифров.

Приводимые сведения о шифрах каждой группы даются по возможности в хронологическом порядке, что позволяет постепенно вводить читателя в сферу криптографии. Как известно, довольно трудно понять концептуальную схему науки, ее модели и методы исследования, если не иметь хотя бы общего представления об истории развития этой науки.