МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
НАЦИОНАЛЬНАЯ МЕТАЛЛУРГИЧЕСКАЯ АКАДЕМИЯ УКРАИНЫ

Г.Г. ШВАЧИЧ, А.В. ОВСЯННИКОВ,
В.В. КУЗЬМЕНКО

ОСНОВЫ ЗАЩИТЫ ИНФОРМАЦИИ

Утверждено на заседании Учёного совета академии
в качестве конспекта лекций

Днепропетровск НМетАУ 2008

УДК 004 (075.8)
Швачич Г.Г., Овсянников А.В., Кузьменко В.В. Основы защиты информации: Конспект лекций. – Днепропетровск: НМетАУ, 2008. – 75 с.

Изложены основные положения методов защиты информации.

Предназначен для студентов специальности 6.020100 – документоведение и информационная деятельность, а также для студентов всех специальностей и иностранных студентов.

Илл. 9. Библиогр.: 7 наим.

Издается в авторской редакции.

Ответственный за выпуск Г.Г. Швачич, канд. техн. наук, проф.

Рецензенты: Б.И. Мороз, д-р техн. наук, проф. (АТСУ)
Д.Г. Зеленцов, д-р. техн. наук, доц. (УГХТУ)

© Национальная металлургическая академия Украины, 2008

Тема 1. Введение в информационную безопасность. Компьютеры: преступления, признаки уязвимости, меры защиты

Работа с критической (конфиденциальной, секретной) информацией требует правильного обращения с документами. Правильное обращение означает соблюдение одинаковых правил работы с документами, независимо от того, используются они в автоматизированной системе или нет. Правила работы могут включать работу в безопасном помещении, учет документов в журналах, гарантии того, что только люди, имеющие соответствующий допуск, могут ознакомиться с этими документами.

Информационная Эра привела к значительным изменениям в способе выполнения своих обязанностей для большого числа профессий. Теперь нетехнический специалист среднего уровня может выполнять работу, которую раньше делал высококвалифицированный программист. Служащий имеет в своем распоряжении столько точной и оперативной информации, сколько никогда не имел.

Использование компьютеров и автоматизированных технологий приводит к появлению ряда проблем для руководства организацией. Компьютеры, часто объединенные в сети, могут предоставлять доступ к колоссальному количеству самых разнообразных данных. Поэтому люди беспокоятся о безопасности информации и наличии рисков, связанных с автоматизацией и предоставлением гораздо большего доступа к конфиденциальным, персональным или другим критическим данным. Всё более увеличивается число компьютерных преступлений, что может привести в конечном счете к подрыву экономики. Исходя из изложенного становится ясно, что информация – это ресурс, который надо защищать.

Ответственность за защиту информации лежит на низшем звене руководства. Но также кто-то должен осуществлять общее руководство этой деятельностью, поэтому в организации должно иметься лицо в верхнем звене руководства, отвечающее за поддержание работоспособности информационных систем.

Так, как автоматизация привела к тому, что теперь операции с вычислительной техникой выполняются простыми служащими организации, а не специально подготовленным техническим персоналом, нужно, чтобы конечные пользователи знали о своей ответственности за защиту информации.

Число компьютерных преступлений (несанкционированных действий) растет, также увеличиваются масштабы компьютерных злоупотреблений. Шансов быть пойманным у компьютерного преступника гораздо меньше, чем у грабителя. Умышленные компьютерные преступления составляют заметную часть преступлений. Но злоупотреблений компьютерами и ошибок персонала еще больше. Как выразился один эксперт: «мы теряем из-за ошибок больше денег, чем могли бы украсть». Эти потери подчеркивают важность и серьезность убытков, связанных с компьютерными технологиями.

Основной причиной наличия потерь, связанных с компьютерными технологиями, является недостаточная образованность в области безопасности. Только наличие некоторых знаний в области безопасности может прекратить инциденты и ошибки, обеспечить эффективное применение мер защиты, предотвратить преступление или своевременно обнаружить подозреваемого. Осведомленность конечного пользователя о мерах безопасности обеспечивает четыре уровня защиты компьютерных и информационных ресурсов.

Четыре уровня защиты информации

Предотвращение – только авторизованный персонал имеет доступ к информации и технологии **Обнаружение** – обеспечивается раннее обнаружение преступлений и злоупотреблений, даже если механизмы защиты были обойдены. **Ограничение** – уменьшается размер потерь, если преступление все-таки произошло, несмотря на меры по его предотвращению и обнаружению.

Восстановление – обеспечивается эффективное восстановление информации при наличии документированных и проверенных планов по восстановлению.

Ранее контроль над технологией работы был заботой технических администраторов. Сегодня контроль над информацией стал обязанностью каждого нетехнического конечного пользователя. Контроль над информацией требует новых знаний и навыков для группы нетехнических служащих. Хороший контроль над информацией требует понимания возможностей совершения компьютерных преступлений и злоупотреблений, чтобы можно было в дальнейшем предпринять контрмеры против них.

Число служащих в организации, имеющих доступ к компьютерному оборудованию и информационной технологии, постоянно растет. Доступ к информации больше не ограничивается только узким кругом лиц из верхнего руководства организации. Этот процесс привел к тому, что произошла «демократизация преступления». Чем больше людей получало доступ к информационной технологии и компьютерному оборудованию, тем больше возникало возможностей для совершения компьютерных преступлений.

Трудно обобщать, но теперь компьютерным преступником может быть:

- конечный пользователь, не технический служащий и не хакер;
- тот, кто не находится на руководящей должности;
- тот, у кого нет судимостей;
- умный, талантливый сотрудник;
- тот, кто много работает;
- тот, кто не разбирается в компьютерах;
- тот, кого вы подозревали бы в последнюю очередь;
- именно тот, кого вы взяли бы на работу.

Некоторые причины, по которым люди совершают компьютерные преступления

Помимо запланированных атак на сети с целью несанкционированного доступа к конфиденциальной информации, фальсификации или разрушения информации, выполняемой профессионалами, наиболее часто встречаются другие виды несанкционированных действий осуществляемых непосредственно персоналом предприятия:

- личная или финансовая выгода;
- развлечение;
- месть;
- попытка добиться расположения кого-либо к себе;
- самовыражение;
- случайность;
- вандализм.

Значительно больший ущерб, около 60 процентов всех потерь, наносят ошибки людей и инциденты. Предотвращение компьютерных потерь, как из-за умышленных преступлений, так и из-за неумышленных ошибок, требует знаний в области безопасности.

Признаки компьютерных преступлений

- неавторизованное использование компьютерного времени;
- неавторизованные попытки доступа к файлам данных;
- кражи частей компьютеров и носителей информации;
- кражи программ;
- физическое и программное разрушение сетей;
- уничтожение данных или программ;
- неавторизованное владение носителями информации или копиями документов.

Это только самые очевидные признаки, на которые следует обратить внимание при выявлении компьютерных преступлений.

Меры защиты – это меры, вводимые руководством, для обеспечения безопасности информации – административные руководящие документы (приказы, положения, инструкции), аппаратные устройства или дополнительные программы – основной целью которых является предотвратить преступления и злоупотребления, не позволив им произойти. Меры защиты могут также выполнять функцию ограничения, уменьшая размер ущерба от преступления.

Информационная безопасность

То, что в шестидесятые годы называлось компьютерной безопасностью, а в семидесятые – безопасностью данных, сейчас более правильно

именуется информационной безопасностью. Информационная безопасность подчеркивает важность информации в современном обществе - понимание того, что информация – это ценный ресурс, нечто большее, чем отдельные элементы данных.

Информационной безопасностью называют меры по защите информации от неавторизованного доступа, разрушения, модификации, раскрытия и задержек в доступе.

Информационная безопасность включает в себя меры по защите процессов создания данных, их ввода, обработки и вывода. Целью информационной безопасности является обезопасить ценности системы, защитить и гарантировать точность и целостность информации, и минимизировать разрушения, которые могут иметь место, если информация будет модифицирована или разрушена. Информационная безопасность требует учета всех событий, в ходе которых информация создается, модифицируется, к ней обеспечивается доступ или она распространяется.

Информационная безопасность дает гарантию того, что достигаются следующие цели:

- конфиденциальность критической информации;
- целостность информации и связанных с ней процессов (создания, ввода, обработки и вывода);
- доступность информации, когда она необходима;
- учет всех процессов, связанных с информацией.

Некоторые технологии по защите системы и обеспечению учета всех событий могут быть встроены в сам компьютер. Другие могут быть встроены в программы. Некоторые же выполняются людьми и являются реализацией указаний руководства, содержащихся в соответствующих руководящих документах. Принятие решения о выборе уровня сложности технологий для защиты системы требует установления критичности информации и последующего определения адекватного уровня безопасности.

Что же такое критические данные? **Под критическими данными будем понимать данные, которые требуют защиты из-за вероятности нанесения (риска) ущерба и его величины в том случае, если произойдет случайное или умышленное раскрытие, изменение или разруше-**

ние данных. Этот термин включает в себя данные, чье неправильное использование или раскрытие может отрицательно отразиться на способности организации решать свои задачи, персональные данные и другие данные, защита которых регламентируется указами Президента, законами и другими подзаконными документами.

Основные технологии совершения компьютерных преступлений

1. **Обман с данными.** Самый распространенный метод при совершении компьютерных преступлений, так как он не требует технических знаний и относительно безопасен. Информация меняется в процессе ее ввода в компьютер или во время вывода. Например, при вводе документы могут быть заменены фальшивыми, вместо рабочих дискет подсунуты чужие, и данные могут быть сфальсифицированы.
2. **Сканирование.** Другой распространенный метод получения информации, который может привести к преступлению. Служащие, читающие файлы других, могут обнаружить там персональную информацию о своих коллегах. Информация, позволяющая получить доступ к компьютерным файлам или изменить их, может быть найдена после просмотра мусорных корзин. Дискеты, оставленные на столе, могут быть прочитаны, скопированы и украдены. Современные программы - сканеры могут просматривать остаточную информацию, оставшуюся на компьютере или на носителе информации после выполнения сотрудником задания и удаления своих файлов.
3. **Троянский конь.** Этот метод предполагает, что пользователь не заметил, что компьютерная программа была изменена таким образом, что включает в себя дополнительные функции. Программа, выполняющая полезные функции, пишется таким образом, что содержит дополнительные скрытые функции, которые будут использовать особенности механизмов защиты системы (возможности пользователя, запустившего программу, по доступу к файлам).
4. **Люк.** Этот метод основан на использовании скрытого программного или аппаратного механизма, позволяющего обойти методы защиты в системе. Этот механизм активируется некоторым неочевидным обра-

зом. Иногда программа пишется таким образом, что специфическое событие, например, число транзакций, обработанных в определенный день, вызовет запуск неавторизованного механизма.

5. **Технология салями.** Названа так из-за того, что преступление совершается понемногу, небольшими частями, настолько маленькими, что они незаметны. Обычно эта технология сопровождается изменением компьютерной программы. Например, платежи могут округляться до нескольких копеек и разница между реальной и округленной суммой поступать на специально открытый счет злоумышленника.
6. **Суперотключение**. Названа по имени программы, использовавшейся в ряде компьютерных центров, обходившей системные меры защиты и использовавшейся при аварийных ситуациях. Владение этим «мастер-ключом» дает возможность в любое время получить доступ к компьютеру и информации, находящейся в нем.

Причины уязвимости системы доступа к информации

Следующие признаки могут свидетельствовать о наличии уязвимых мест в информационной безопасности.

1. Не разработано положений о защите информации или они не соблюдаются. Не назначен ответственный за информационную безопасность.
2. Пароли пишутся на компьютерных терминалах, помещаются в общедоступные места, ими делятся с другими или они появляются на компьютерном экране при их вводе.
3. Удаленные терминалы и компьютеры оставляются без присмотра в рабочие и нерабочие часы. Данные отображаются на компьютерных экранах, оставленных без присмотра.
4. Не существует ограничений на доступ к информации или на характер ее использования. Все пользователи имеют доступ ко всей информации и могут использовать все функции системы.
5. Не ведутся записи в системных журналах и не хранится информация о том, кто и для чего использует компьютер.
6. Изменения в программы могут вноситься без их предварительного утверждения руководством.

7. Отсутствует документация или она не позволяет делать следующее: понимать получаемые отчеты и формулы, по которым получаются результаты, модифицировать программы, готовить данные для ввода, исправлять ошибки, производить оценку мер защиты и понимать сами данные - их источники, формат хранения, взаимосвязи между ними.
8. Делаются многочисленные попытки войти в систему с неправильными паролями.
9. Вводимые данные не проверяются на корректность и точность или при их проверке много данных отвергается из-за ошибок в них, требуется сделать много исправлений в данных, не делается записей в журналах об отвергнутых транзакциях.
10. Имеют место выходы из строя системы, приносящие большие убытки.
11. Не производился анализ информации, обрабатываемой в компьютере, с целью определения необходимого для нее уровня безопасности.
12. Мало внимания уделяется информационной безопасности. Хотя политика безопасности и существует, большинство людей считает, что на самом деле она не нужна.

Мероприятия по обеспечению информационной безопасности

1. Контролируйте доступ как к информации в компьютере, так и к прикладным программам. Не оставляйте на рабочем столе важные документы. Архивируйте с использованием пароля или криптографируйте конфиденциальные документы либо папки, базы данных, содержащие секретную информацию. Вы должны иметь гарантии того, что только авторизованные пользователи имеют доступ к информации и приложениям.

Идентификация пользователей

Обеспечьте, чтобы пользователи выполняли процедуры входа в компьютер, и используйте это как средство для идентификации в начале работы.

Аутентификация пользователей

Используйте уникальные пароли для каждого пользователя, которые не являются комбинациями личных данных пользователей, для аутентификации личности пользователя. Внедрите меры защиты при администриро-

вании паролей и ознакомьте пользователей с наиболее общими ошибками, позволяющими совершиться компьютерному преступлению.

Другие меры защиты

Пароли – только один из типов идентификации – что-то, что знает только пользователь. Двумя другими типами идентификации, которые тоже эффективны, являются что-то, чем владеет пользователь (например, магнитная карта, отпечаток пальца), или уникальные характеристики пользователя (его голос).

Если в компьютере имеется встроенный стандартный пароль (пароль, который встроен в программы и позволяет обойти меры по управлению доступом), обязательно измените его.

Сделайте так, чтобы программы в компьютере после входа пользователя в систему сообщали ему время его последнего сеанса и число неудачных попыток установления сеанса после этого. Это позволит сделать пользователя составной частью системы проверки журналов.

Защищайте ваш пароль:

- не делитесь своим паролем ни с кем;
- выбирайте пароль трудно угадываемым;
- попробуйте использовать строчные и прописные буквы, цифры, или выберите знаменитое изречение и возьмите оттуда каждую четвертую букву. А еще лучше позвольте компьютеру самому сгенерировать ваш пароль;
- не используйте пароль, который является вашим адресом, псевдонимом, именем жены, мужа, телефонным номером или чем-либо очевидным;
- используйте длинные пароли, так как они более безопасны;
- обеспечьте неотображаемость пароля на экране компьютера при его вводе;
- обеспечьте отсутствие паролей в распечатках документов;
- не записывайте пароли на столе, стене или терминале. Держите его в памяти.

Серьезно относитесь к администрированию паролей:

- периодически меняйте пароли и делайте это не по графику;

- шифруйте или скрывайте иным образом файлы паролей, хранящихся в компьютере, для защиты их от неавторизованного доступа;
- назначайте на должность администратора паролей только самого надежного человека;
- не используйте один и тот же пароль для всех сотрудников в группе;
- меняйте пароли, когда человек увольняется;
- заставляйте людей расписываться за получение паролей;
- установите и внедрите правила работы с паролями и обеспечьте, чтобы все знали их.

Процедуры авторизации

Разработайте процедуры авторизации, которые определяют, кто из пользователей должен иметь доступ к той или иной информации и приложениям и используйте соответствующие меры по внедрению этих процедур в организации.

Установите порядок в организации, при котором для использования компьютерных ресурсов, получения разрешения доступа к информации и приложениям и получения пароля требуется разрешение тех или иных ответственных работников.

Защита файлов

Помимо идентификации пользователей и процедур авторизации разработайте процедуры по ограничению доступа к файлам с данными:

- используйте внешние и внутренние метки файлов для указания типа информации, который они содержат, и требуемого уровня безопасности;
- ограничьте доступ в помещения, в которых хранятся файлы данных, такие как архивы и библиотеки данных;
- используйте организационные меры и программно-аппаратные средства для ограничения доступа к файлам только авторизованных пользователей.

Предосторожности при работе

- отключайте неиспользуемые терминалы;
- закрывайте комнаты, где находятся терминалы;
- разворачивайте экраны компьютеров так, чтобы они не были видны со стороны двери, окон и тех мест в помещениях, которые не контролируются;

- установите специальное оборудование, такое как устройства, ограничивающие число неудачных попыток доступа, или делающие обратный звонок для проверки личности пользователей, использующих телефоны для доступа к компьютеру;
- программируйте терминал отключаться после определенного периода не использования;
- если возможно, выключайте систему в нерабочие часы.

2. Защищайте целостность информации. Вводимая информация должна быть авторизована, полна, точна и должна подвергаться проверкам на ошибки.

Целостность информации

Проверяйте точность информации с помощью процедур сравнения результатов обработки с предполагаемыми результатами обработки. Например, можно сравнивать суммы или проверять последовательные номера.

Проверяйте точность вводимых данных, требуя от служащих выполнять проверки на корректность, такие как:

- проверки на нахождение символов в допустимом диапазоне символов (числовом или буквенном);
- проверки на нахождение числовых данных в допустимом диапазоне чисел;
- проверки на корректность связей с другими данными, сравнивающими входные данные с данными в других файлах;
- проверки на разумность, сравнивающие входные данные с ожидаемыми стандартными значениями;
- ограничения на транзакции, сравнивающие входные данные с административно установленными ограничениями на конкретные транзакции;

Делайте перекрестные проверки содержимого файлов с помощью сопоставления числа записей или контроля суммы значений поля записи.

3. Защищайте системные программы. Если ПО используется совместно, защищайте его от скрытой модификации при помощи политики безопасности, мер защиты при его разработке и контроле за ним в его жизненном цикле, а также обучения пользователей в области безопасности.

Меры защиты при разработке программ и соответствующие политики должны включать процедуры внесения изменений в программу, ее приемки и тестирования до ввода в эксплуатацию. Политики должны требовать разрешения ответственного лица из руководства для внесения изменений в программы, ограничения списка лиц, кому разрешено вносить изменения и явно описывать обязанности сотрудников по ведению документации.

Должен быть разработан и поддерживаться каталог прикладных программ.

Должны быть внедрены меры защиты по предотвращению получения, изменения или добавления программ неавторизованными людьми через удаленные терминалы.

4. Сделайте меры защиты более адекватными с помощью привлечения организаций, занимающихся тестированием информационной безопасности, при разработке мер защиты в прикладных программах и консультируйтесь с ними при определении необходимости тестов и проверок при обработке критических данных. Контрольные журналы, встроенные в компьютерные программы, могут предотвратить или выявить компьютерное мошенничество и злоупотребление.

Должны иметься контрольные журналы для наблюдения за тем, кто из пользователей обновлял критические информационные файлы Если критичность информации, хранимой в компьютерах, требует контрольных журналов, то важны как меры физической защиты, так и меры по управлению доступом.

Контрольные журналы не должны отключаться для повышения скорости работы.

Распечатки контрольных журналов должны просматриваться достаточно часто и регулярно.

5. Рассмотрите вопрос о коммуникационной безопасности. Данные, передаваемые по незащищенным линиям, могут быть перехвачены.

Физическая безопасность

Физическая безопасность связана с внедрением мер защиты, которые защищают от стихийных бедствий (пожаров, наводнений, и землетрясений), а также всяких случайных инцидентов. Меры физической безопасности определяют, каким будет окружение компьютера, вводимые данные, и результаты обработки информации. Помимо помещений, где размещено компьютерное оборудование, окружение включает в себя библиотеки программ, журналы, носители, помещения для архивов, и помещения для ремонта техники.

Меры физической защиты должны отвечать требованиям современной действительности и сочетать эффективность с невысокой ценой. Например, установка дорогой противопожарной системы может быть необходимой для защиты большого числа компьютеров, обрабатывающих критические данные, но оказаться неоправданно дорогой при защите одной персональной ЭВМ.

Преступления и злоупотребления

Компьютеры могут быть повреждены, украдены и специально выведены из строя с помощью короткого замыкания. Диски и другие носители могут быть разрушены разлитыми способами, а компьютеры залиты водой. Также компьютеры могут быть серьезно повреждены пожаром, скачками напряжения, стихийными бедствиями и другими инцидентами. Информация может быть перехвачена, украдена, продана и использоваться в корыстных целях отдельным человеком или целой компанией.

Следующие признаки могут указывать на наличие уязвимых мест в физической безопасности:

- разрешено курить, принимать пищу рядом с компьютерами;
- компьютерное оборудование оставляется в незапертых комнатах или является незащищенным по какой-либо другой причине;
- не установлена пожарная сигнализация;
- диски оставляются в ящиках столов, не делается архивных копий дисков;
- посетителям не задается вопросов о причине их нахождения в помещениях, где установлены компьютеры;

- реестр компьютерного оборудования и программ отсутствует, неполон, не обновляется или не проверяется после его заполнения;
- распечатки, диски, содержащие критические данные выбрасываются в обычное мусорное ведро;
- замки на входах в помещения, где находится компьютерное оборудование, никогда не менялись;
- не производилось аттестации автоматизированной системы организации, то есть анализа насколько она уязвима к доступу неавторизованных людей, пожару или наводнению.

Меры физической безопасности

1. Необходимо разработать мероприятия по предотвращению злонамеренных разрушений и неавторизованного использования или краж.
2. Стихийные бедствия могут нанести большой ущерб как большим, так и маленьким компаниям.
3. Защищайте все носители информации (исходные документы, диски, распечатки).

- ведите, контролируйте и проверяйте реестры носителей информации;
- обучайте пользователей правильным методам очищения и уничтожения носителей информации;
- делайте метки на носителях информации, отражающие уровень критичности информации, которая в них содержится;
- уничтожайте носители информации в соответствии с планом организации;
- удостоверьтесь, что доступ к носителям информации для их хранения, передачи, нанесения меток, и уничтожения предоставлен только авторизованным людям;
- доведите все руководящие документы до сотрудников.

Тема 2. Основы криптографии

Введение

Под безопасностью (в широком смысле) понимается способность информационной системы сохранять свою целостность и работоспособность при случайных или преднамеренных внешних воздействиях.

Широкое использование информационных технологий привело к бурному развитию различных методов защиты информации, из которых основными можно, пожалуй, назвать, помехоустойчивое кодирование и криптографию.

Простейшие способы шифрования появились очень давно, однако, научный подход к исследованию и разработке криптографических методов появился только в прошлом (XX) веке.

К настоящему времени криптография содержит множество теорем и алгоритмов, как фундаментальных, так и прикладных. Применение криптографии невозможно без серьезной математической подготовки. Особенно необходимы знания в области дискретной математики, теории чисел, абстрактной алгебры и теории алгоритмов.

Вместе с тем не следует забывать, что криптографические методы предназначены в первую очередь для практического применения, а теоретически стойкие алгоритмы могут оказаться незащищенными перед атаками, не предусмотренными математической моделью. Поэтому после анализа абстрактной математической модели всегда необходим анализ полученного алгоритма с учетом ситуации, в которой он будет использоваться на практике.

Математическая криптография возникла как наука о шифровании информации, т.е. как наука о криптосистемах. В классической модели системы секретной связи имеют два полностью доверяющих друг – другу участника, которым необходимо передавать между собой информацию, не предназначенную для третьих лиц. Такая информация называется конфиденциальной или секретной. Отсюда возникает задача обеспечения конфиденциальности, т.е. защита секретной информации от противника. Эта за-

дача, по крайней мере, исторически, – первая задача криптографии. Она традиционно решается с помощью криптосистем.

При обмене информацией между участниками часто возникает ситуация, когда информация не является конфиденциальной, но важен факт поступления сообщений в неискаженном виде, т.е. наличие гарантии, что никто сумеет не подделать сообщение. Такая гарантия называется обеспечением целостности информации и составляет вторую задачу криптографии.

Для предотвращения угрозы, контроля над источниками информации, (откуда пересылаются сообщения) необходима система контроля над доступом к ресурсам, которая должна удовлетворять двум, казалось бы, взаимно исключающим требованиям. Во-первых, всякий желающий должен иметь возможность обратиться к этой системе анонимно; а во-вторых, при этом все же доказать свое право на доступ к ресурсам. Примером могут служить бумажные купюры. Если ресурсом является некоторый товар, то наличие у покупателя достаточного количества купюр является доказательством его права на доступ к ресурсу. С другой стороны, хотя каждая бумажная купюра и имеет уникальный номер, отслеживать купюры по номерам практически невозможно, т.е. определить, кто ее использовал и в каких платежах, практически невозможно. Аналог этого свойства в криптографии называется неотслеживаемостью. Обеспечение неотслеживаемости – третья задача криптографии.

Если задача обеспечения конфиденциальности решается с помощью криптосистем, то для обеспечения целостности и неотслеживаемости разрабатываются криптографические протоколы.

История криптографии

Термин криптография (тайнопись) ввел Д. Валлис. Потребность шифровать и передавать шифрованные сообщения возникла очень давно.

Так, еще в V-IV вв. до н. э. греки применяли специальное шифрующее устройство. По описанию Плутарха, оно состояло из двух палок одинаковой длины и толщины. Одну оставляли себе, а другую отдавали отъезжающему. Эти палки называли скиталами. Когда правителям нужно было сообщить какую-нибудь важную тайну, они вырезали длинную и узкую,

вроде ремня, полосу папируса, наматывали ее на свою скиталу, не оставляя на ней никакого промежутка, так чтобы вся поверхность палки была охвачена этой полосой. Затем, оставляя папирус на скитале в том виде, как он есть, писали на нем все, что нужно, а, написав, снимали полосу и без палки отправляли адресату. Так как буквы на ней разбросаны в беспорядке, то прочитать написанное можно только при помощи соответствующей скиталы, намотав на нее без пропусков полосу папируса.

Аристотелю принадлежит способ дешифрования этого шифра. Надо изготовить длинный конус и, начиная с основания, обертывать его лентой с шифрованным сообщением, постепенно сдвигая ее к вершине. В какой-либо момент начнут просматриваться куски сообщения. Так можно определить диаметр скиталы.

В Древней Греции (II в. до н. э.) был известен шифр, называемый квадрат Полибия. Это устройство представляло собой квадрат 5 * 5, столбцы и строки которого нумеровали цифрами от 1 до 5. В каждую клетку этого квадрата записывалась одна буква. (В греческом варианте одна клетка оставалась пустой, в латинском – в одну клетку помещали две буквы **i** и j.) В результате каждой букве отвечала пара чисел, и шифрованное сообщение превращалось в последовательность пар чисел.

Пример: 13 34 22 24 44 34 15 42 22 34 43 45 32

Это сообщение записано при использовании латинского варианта квадрата Полибия, в котором буквы расположены в алфавитном порядке. («Cogito, ergo sum» – лат, «Я мыслю, следовательно, существую»).

В 1 в. н. э. Ю. Цезарь во время войны с галлами, переписываясь со своими друзьями в Риме, заменял в сообщении первую букву латинского алфавита (A) на четвертую (D), вторую (B) – на пятую (E), наконец, последнюю – на третью:

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

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

Пример: Донесение Ю. Цезаря Сенату об одержанной им победе над Понтийским царем выглядело так:

YHQL YLGL YLFL («Veni, vidi, vici» – лат. «Пришел, увидел, победил»).

Император Август (1 в. н. э.) в своей переписке заменял первую букву на вторую, вторую – на третью и т. д., наконец, последнюю – на первую:

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

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

Пример: Любимое изречение императора Августа выглядело так:

GFTUJOB MFOUF («Festina lente» – лат. «Торопись медленно»).

Квадрат Полибия, шифр Цезаря входят в класс шифров, называемых подстановка или простая замена, т.е., это такой шифр, в котором каждой букве алфавита соответствует буква, цифра, символ или какая-нибудь их комбинация.

В известных рассказах «Пляшущие человечки» Конан Дойля и «Золотой жук» Эдгара По используемые шифры относятся к указанному классу шифров. В другом классе шифров – перестановка – буквы сообщения, каким-нибудь способом переставляются между собой. К этому классу принадлежит шифр скитала.

Предмет криптографии

Криптография – это наука о способах преобразования информации с целью ее защиты от незаконных пользователей. Методы решения противоположной задачи (взлом криптографической защиты) составляют предмет другой науки – криптоанализа.

Вместе с тем, было бы неправильным разделять криптографию и криптоанализ. И криптография и криптоанализ изучают одни и те же объекты, но с разных точек зрения. Поэтому они скорее являются двумя частями одной и той же науки (она называется «криптология»), а не независимыми дисциплинами.

Изучать их тоже надо совместно, потому что невозможно серьезно заниматься криптографией (например, разрабатывать шифры), не изучив криптоанализ.

Основные задачи криптографии

Криптография возникла как наука о методах шифрования, и долгое время именно шифрование (т.е. защита передаваемых или хранимых дан-

ных от несанкционированного чтения) оставалась единственной проблемой, изучаемой криптографией.

Однако в последнее время, в связи с бурным развитием информационных технологий, возникло множество новых применений, напрямую не связанных с сокрытием секретной информации. Необходимость применения криптографических методов вытекает из условий, в которых происходит хранение и обмен информацией. В современных информационных системах очень часто происходит обмен данными в коллективах, члены которых не доверяют друг другу. В качестве примеров можно привести подписание контрактов или других документов, финансовые операции, совместное принятие решений и т.п. В таких ситуациях необходимы средства, гарантирующие, что в процессе обмена или хранения информация не будет подвергнута искажениям, или не будет подменена целиком. Такую гарантию может дать только применение научно обоснованных криптографических методов.

Итак, целью применения криптографических методов является защита информационной системы от целенаправленных разрушающих воздействий (атак) со стороны противника. Способы защиты существенно зависят от ситуации: от какого рода угрозы необходимо защищаться, какими возможностями обладает противник.

Основные цели криптографии:

- Обеспечение **конфиденциальности** данных (предотвращение несанкционированного доступа к данным). Это одна из основных задач криптографии, для ее решения применяется шифрование данных, т.е. такое их преобразование, при котором прочитать их могут только законные пользователи, обладающие соответствующим ключом.
- Обеспечение **целостности** данных – гарантии того, что при передаче или хранении данные не были модифицированы пользователем, не имеющим на это права. Под модификацией понимается вставка, удаление или подмена информации, а также повторная пересылка перехваченного ранее текста.

- Обеспечение **аутентификации**. Под аутентификацией понимается проверка подлинности субъектов (сторон при обмене данными, автора документов, и т. д.) или подлинности самой информации. Частным случаем аутентификации является идентификация – процедура доказательства субъектом того, что он действительно является именно тем, за кого себя выдает. Во многих случаях субъект (X) должен не просто доказать свои права, но сделать это так, чтобы проверяющий субъект (Y) не смог впоследствии сам использовать полученную информацию для того, чтобы выдать себя за (X). Подобные доказательства называются «доказательствами с нулевым разглашением».
- Обеспечение **невозможности отказа от авторства** – предотвращение возможности отказа субъектов от совершенных ими действий (обычно – невозможности отказа от подписи под документом). Эта задача неотделима от двойственной — обеспечение невозможности приписывания авторства. Наиболее яркий пример ситуации, в которой стоит такая задача – подписание договора двумя или большим количеством лиц, не доверяющих друг другу. В такой ситуации все подписывающие стороны должны быть уверены в том, что в будущем, во-первых, ни один из подписавших не сможет отказаться от своей подписи; во-вторых, никто не сможет модифицировать, подменить или создать новый документ (договор) и утверждать, что именно этот документ был подписан. Основным способом решения данной проблемы является использование **цифровой подписи**.

Помимо перечисленных основных задач можно назвать также электронное голосование, жеребьевку, разделение секрета (распределение секретной информации между несколькими субъектами таким образом, чтобы воспользоваться ей они могли только все вместе) и многое другое.

Примечание. Чем шифрование отличается от кодирования? Слова «кодирование » и «шифрование » часто используются как синонимы. Однако в современной прикладной математике (к которой можно отнести и криптографию) эти термины разделяются. Под шифрованием понимается

такое преобразование текста (сообщения), в результате которого прочитать преобразованный текст может только тот, кто обладает специальным ключом. Кодированием называется любое преобразование данных из одной формы представления в другую. Таким образом, кроме шифрования, термин «кодирование » включает в себя также так называемое «помехоустойчивое кодирование» (преобразование текста, позволяющее восстанавливать его в случае сбоя при передаче или хранении), сжатие данных и т.п. В широком смысле, кодированием можно назвать также сканирование текста или изображения, при котором информация преобразуется из визуального представления в цифровое.

ТЕМА 3. МОДЕЛЬ КРИПТОГРАФИЧЕСКОЙ СИСТЕМЫ

Простейшую модель криптографической системы можно изобразить так, как показано на рисунке (рис. 3.1). Таким образом, имеется некая информационная система, включающая двух или более абонентов (законных пользователей) и канал (или каналы), по которым абоненты могут обмениваться сообщениями. Имеется также возможность появления противника, т.е. незаконного пользователя. Противник может перехватывать сообщения, передаваемые абонентами друг другу.

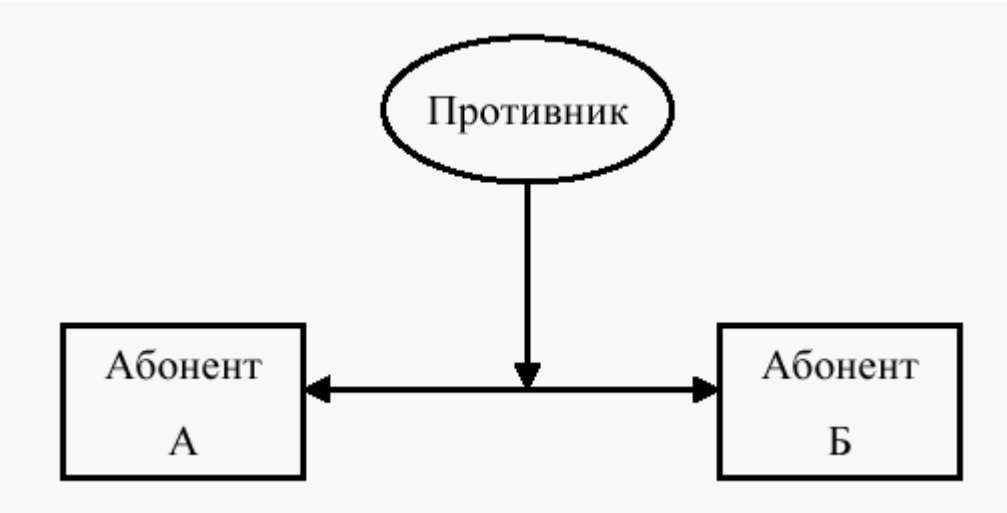

Рис. 3.1. Простейшая модель криптосистемы

Здесь необходимы следующие пояснения. Во-первых, противник может быть как внешним (т.е. не входить в число абонентов системы), так и внутренним (быть абонентом системы). В последнем случае этот абонент считается незаконным пользователем, если он пытается получить доступ к сообщениям, на которые не имеет права (например, конфиденциальные сообщения, которыми обмениваются другие абоненты). Во-вторых, противник может перехватывать сообщения с разными целями – например, с целью разглашения перехватываемой информации (использование этой информации в своих целях или передача информации другому лицу), подмены или имитации сообщения и т.д.

Подобные цели называются угрозами. Для защиты от различных видов угроз необходимо применять различные криптографические методы.

Рассматриваемая нами задача обеспечения конфиденциальности информации представляет собой задачу защиты от угрозы разглашения.

Следует иметь в виду, что описанная модель может применяться и в случаях, внешне отличных от обмена сообщениями. Например, при защите данных, хранящихся на компьютере, можно считать, что абонент А и абонент Б – одно и то же лицо, работающее с данными в разные моменты времени. В этом случае «каналом» является жесткий диск компьютера, на котором хранятся данные.

Итак, рассматривается модель, в которой противник имеет доступ к каналу передачи сообщений. Поэтому абонент, передающий сообщение (отправитель) должен перед отправкой преобразовать исходную информацию (открытый текст) в закрытый текст (который называется шифртекстом, зашифрованным текстом или криптограммой). Преобразование открытого текста в шифртекст называется шифрованием (часто используется также термин зашифрование). Абонент, получивший такой зашифрованный текст (получатель), с помощью обратного преобразования (расшифрования, расшифровки) восстанавливает исходный открытый текст.

Процедуры шифрования и расшифрования используют некоторые секретные данные, называемые ключами, причем в некоторых криптосистемах ключ шифрования и ключ расшифрования совпадают, а в других – различаются. Ключи известны только абонентам криптосистемы, причем для обмена

данными с различными пользователями один и тот же абонент может использовать различные ключи. Противник не знает ключ расшифрования, но может попытаться вскрыть шифр, т.е. либо подобрать ключ, либо преобразовать зашифрованный текст в открытый каким-либо другим способом.

Методы вскрытия шифров называются криптоанализом, а противник, применяющий эти методы – криптоаналитиком. Успех криптоанализа зависит как от свойств криптографической системы, так и от имеющихся у противника ресурсов (время, мощность вычислительных машин и т.п.).

Способность шифра (криптосистемы) противостоять попыткам взлома (атакам) называется стойкостью шифра.

Существуют абсолютно стойкие системы шифрования, однако они очень не удобны и требуют больших затрат при использовании. Ни одна из широко используемых на практике систем шифрования не является абсолютно стойкой. Это означает, что если противник обладает неограниченными ресурсами и достаточно широкими возможностями для атаки (например, имеет доступ к некоторым открытым текстам и соответствующим им шифртекстам, полученным с использованием одного и того же ключа), то рано или поздно он сможет взломать шифр. Однако если выгода от использования полученной информации будет меньше, чем затраты на взлом, противник вряд ли будет этим заниматься. Поэтому при выборе алгоритма шифрования необходимо точно оценить соотношение ценности защищаемой информации, стойкости шифра и удобства его использования – иначе затраты на защиту информации могут превысить стоимость самой информации.

Формальная модель и классификация шифров

Введем формальное определение шифра и его составных частей. Пусть T, C и K – конечные множества возможных открытых текстов, шифртекстов и ключей. Обычно каждое из этих множеств, представляет собой множество слов в некотором алфавите, причем алфавиты открытых текстов, шифртекстов и ключей могут различаться.

Для большинства современных систем шифрования открытые тексты, шифртексты и ключи представляют собой слова в алфавите $\{0,1\}$, т.е. последовательности нулей и единиц.

Процедура шифрования задает функцию Ek: T $\rightarrow$ C, которая отображает множество открытых текстов во множество шифртекстов в зависимости от некоторого ключа k $\in$ K. Аналогично, процедура расшифрования Dk : C $\rightarrow$T также зависит от ключа k и отображает множество шифртекстов во множество открытых текстов. Так как получатель всегда должен иметь возможность по шифртексту восстановить исходный текст, то при любом k из K функции Ek и Dk должны удовлетворять условию:

Dk °Ek =I, где I – тождественное отображение T в T.

Примечание 1. Во многих криптографических системах предполагается, что открытый текст, шифртекст и ключ – это целые числа. Такое предположение удобно для построения и обоснования алгоритмов шифрования и расшифрования, поскольку числовые функции хорошо изучены. Вместе с тем, это не ограничивает область применения таких алгоритмов, потому что любой текст, записанный с помощью букв, например, русского алфавита, всегда можно представить в виде целого числа. Обычно для этого каждый символ алфавита кодируют набором нулей и единиц (например, в соответствии с таблицей ASCII), и текст представляют в виде последовательности кодов соответствующих символов, записанных друг за другом. Получившаяся последовательность нулей и единиц является числовым представлением текста.

Примечание 2. Часто при реализации алгоритмов шифрования и расшифрования бывает удобно считать, что длина ключа, используемого для преобразования текста, равна длине самого текста или зависит от длины текста каким – то определенным образом. Очевидно, что если ключ используется для шифрования нескольких текстов, то его длина не может зависеть от длины каждого конкретного текста. В этом случае перед шифрованием текста поступают так: на основе данного секретного ключа фиксированной длины с помощью определенного алгоритма формируют ключ шифрования, имеющий необходимую длину. Полученный ключ шифрования используют для преобразования текста. Простейшим способом сформировать ключ шифрования нужной длины является периодическое повторение символов секретного ключа.

Например, из секретного ключа k = (k1, k2, …, k n) можно получить ключ шифрования (k1, k2, …, kn , k1, k2, …, kn , k1, k2, …, kn , k1…) произвольной длины.

Современные системы шифрования можно разделить на два больших класса:

- Симметричные (одноключевые) системы – в них для шифрования и расшифрования текста используется один и тот же ключ к.
- Асимметричные (двухключевые) системы используют различные ключи для шифрования и расшифрования текста.

Для таких систем ключ k можно представить в виде пары (ke, kd), где часть ke используется для шифрования, а kd – для расшифрования.

Тема 4. Симметричные системы шифрования

К симметричным системам шифрования относятся такие системы, в которых для шифрования и для расшифрования используется один и тот же ключ. Поэтому такие системы называют также одноключевыми.

В зависимости от типа преобразования, выполняемого над открытым текстом при шифровании, симметричные системы шифрования можно разделить на шифры замены, шифры перестановки и композиционные шифры.

К шифрам замены относятся преобразования, при которых фрагменты открытого текста (отдельные символы или группы символов – блоки) заменяются некоторыми символами или группами символов в шифртексте.

Метод шифрования – гаммирование, в принципе, также является разновидностью шифров замены. Обычно гаммирование выделяют в отдельный тип шифрования, поскольку по многим практически важным параметрам он отличается от «обычных» шифров замены.

Шифры перестановки для получения шифртекста лишь переставляют символы открытого текста местами.

Наиболее распространены композиционные шифры, представляющие собой последовательное применение нескольких процедур шифрования разных типов.

Шифры перестановки

Ключом шифра перестановки является перестановка номеров символов открытого текста. Это, в частности, означает, что длина ключа шифрования должна быть равна длине преобразуемого текста. Для того чтобы из секретного ключа получить ключ шифрования, удобный для использования в шифрах перестановки, предложен ряд методов.

С помощью одного из таких методов формируются так называемые маршрутные перестановки. Открытый текст записывают в некоторую геометрическую фигуру (чаще всего — прямоугольник) по некоторой траектории, а затем, выписывая символы из этой фигуры по другой траектории, получают шифртекст.

Пример. Запишем фразу «это маршрутная перестановка» в прямоугольную таблицу размером 3 ×9, двигаясь по строкам, слева направо и пропуская пробелы (рис. 4.1).

э	т	о	м	а	р	ш	р	у
т	н	а	я	п	е	р	е	с
т	а	н	о	в	к	а		

Рис. 4.1. Пример маршрутной перестановки

Для зашифрования текста выпишем из этой таблицы буквы, двигаясь по столбцам сверху вниз: эттт наоанмяоапврекршареус.

Из-за своей низкой стойкости, в системах шифры перестановки используются только как составная часть композиционных шифров.

Шифры замены

Простейшим из шифров замены является одноалфавитная подстановка, называемая также шифром простой замены.

Ключом такого шифра является взаимно однозначное отображение (подстановка) F алфавита открытого текста (X) в шифртекста (Y): F: X $\leftrightarrow$ Y. Зафиксируем нумерацию символов в алфавитах X и Y: X = { x_1, x_2 , … x_n }, Y = { y_1, y_2, … y_n }.

Тогда отображение F фактически задается перестановкой p порядка n = |X| = |Y|: при шифровании символ xi открытого текста заменяется на символ $y_{p(i)}$ шифртекста.

Эта перестановка может быть задана либо таблицей, либо с помощью формулы. При задании с помощью формулы значение $p(i)$ представляется в виде выражения, зависящего от i.

Пример. Типичным примером шифра замены является шифр Цезаря. Этот шифр реализует следующее преобразование текста, записанного с помощью латинского алфавита: каждая буква открытого текста заменяется буквой, стоящей на три позиции позже нее в алфавите (при этом алфавит считается записанным по кругу, то есть после буквы ‘z’ идет буква ‘a’).

Открытый текст ‘secret’ будет преобразован в ‘vhfuhw’. Ключ для шифра Цезаря можно задать в виде следующей таблицы (рис. 4.2.). В первой строке записаны буквы открытого текста, во второй – соответствующие им буквы шифртекста.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

Рис. 4.2. Ключ для шифра Цезаря

Шифр Цезаря можно описать и в виде формулы. Для этого пронумеруем буквы латинского алфавита числами от 0 до 25: a = 0, b = 1, …, z = 25. Тогда правило замены можно описать следующим образом: буква с номе-

ром i заменяется на букву с номером i+3 (mod 26), где операция ‘mod 26’ означает вычисление остатка от деления на 26.

Возможен обобщенный вариант шифра Цезаря, при котором буква с номером i заменяется на букву с номером i+k (mod 26). В этом случае ключом шифра является число k.

Еще больше обобщив этот метод, придем к семейству аффинных шифров. Для алфавита из n символов $\{a_1, a_2, \ldots, a_n\}$ аффинным шифром называется процедура, заменяющая входной символ a_i на символ a_j, где j = k ·i+l (mod n).

Шифры простой замены в настоящее время не используются, поскольку их стойкость невелика. Методы взлома таких шифров основаны на анализе частотности отдельных символов и их комбинаций. Дело в том, что в любом языке различные буквы и комбинации из двух, трех или большего количества букв имеют характерные частоты повторений в текстах. Например, в текстах на русском языке чаще всего встречается буква ‘О’, затем, в порядке убывания частоты, идут буквы ‘Е’ (считая, что ‘Е’ и ‘Ё’ – одна и та же буква), ‘А’, ‘И’, ‘Т’ и т. д. Для английского языка аналогичная последовательность самых частых букв: ‘E’, ‘T’, ‘A’, ‘I’, ‘N’. Самым частым символом в текстах является, однако, не буква, а символ пробела.

Становится ясно, что при использовании шифра простой замены частота повторений зашифрованных символов в шифртексте совпадает с частотой повторений соответствующих исходных символов в открытом тексте. Это позволяет достаточно легко вскрыть такой шифр. Более тонкие характеристики (учет сочетаемости различных букв) позволяют даже автоматизировать процесс взлома.

Для того чтобы увеличить стойкость шифров замены, применяют многоалфавитную замену.

Процедура шифрования для многоалфавитной замены включает набор подстановок $\{p_1, p_2, \ldots, p_m\}$ и функцию – распределитель y (k, i), задающую последовательность применения подстановок p_i.

При шифровании i-го символа открытого текста применяется подстановка с номером p (k,i), где k – ключ шифрования.

Частным случаем многоалфавитной замены является шифр Виженера. Формально этот шифр можно описать следующим образом.

В качестве ключа шифрования выберем набор из m целых чисел:

k = (k_1, k_2, …, k_m). Процедуру преобразования открытого текста t = (t_1, t_2, …) в шифртекст c = (c_1, c_2, …) построим на основе обобщенного шифра Цезаря: $c_1 = t_1 + k_1 \pmod{26}$, $c_2 = t_2 + k_2 \pmod{26}$, и т.д. Когда будут использованы все m компонент ключа k, для шифрования (m+1)-й буквы снова возьмем k_1, и т.д. Фактически, в качестве ключа шифрования используется бесконечная последовательность, образованная периодическим повторением исходного набора: k_1, k_2, …, k_m, k_1, k_2, …, k_m, k_1, k_2, … Такую последовательность принято называть гаммой.

Взломать шифр многоалфавитной замены немного сложнее, чем шифры простой замены, но тоже достаточно легко. Такой шифр на самом деле представляет собой одновременное применение m шифров простой замены (обобщенный шифр Цезаря), причем часть исходного текста, состоящая из букв t_i, t_{m+1}, t_{2m+i}, … шифруются с использованием «ключа» ki (i=1, …, m).

Если известен период гаммы (т.е. число m), то к каждой такой части можно применить любой из методов взлома шифров простой замены. Если период гаммы не известен, то задача усложняется. Но и для этих случаев разработаны эффективные методы взлома. Эти методы позволяют с достаточной вероятностью определить период гаммы, после чего задача сводится к взлому шифра гаммирования с известным периодом.

Как было указано выше, основой для атак на шифры замены является анализ частот вхождений символов в шифртекст. Для того чтобы затруднить взлом шифра замены, можно попытаться скрыть частотные свойства исходного текста. Для необходимо, чтобы частоты появления разных символов тексте совпадали.

Такие шифры замены называются гомофоническими.

Простейшим вариантом гомофонического шифра является следующий. Предположим, что нам известны частоты вхождений символов в открытый текст. Пусть fi – частота появления i–го символа в открытом тек-

ста (i – номер буквы в алфавите). Каждой t_i исходного алфавита (т.е. алфавита, с помощью которого записывается открытое сообщение) сопоставим подмножество F_i, содержащее f_i символов выходного алфавита (т.е. алфавита, с помощью которого записывается шифртекст). Причем никакие два подмножества F_i и F_j не пересекаются. При шифровании будем заменять каждое вхождение символа t_i на случайный символ из множества F_i.

Очевидно, что средняя частота появления в шифртексте любого из символов выходного алфавита одинакова, что существенно затрудняет криптоанализ.

Гаммирование

Формально гаммирование можно отнести к классу шифров многоалфавитной замены. Однако, благодаря удобству реализации и формального описания, шифры гаммирования широко используются, и обычно их выделяются в отдельный класс.

Суть метода гаммирования заключается в следующем. С помощью секретного ключа k генерируется последовательность символов $g = g_1 g_2 \ldots g_i \ldots$, эта последовательность называется гаммой.

При шифровании гамма накладывается на открытый текст $t = t_1 t_2 \ldots t_i \ldots$, т.е. символы шифртекста получаются из соответствующих символов открытого текста и гаммы с помощью некоторой обратимой операции: $c_i = t_i \bullet g_i$, i=1,2,…

Примечание. Знак (•) – некоторая обратимая операция, например XOR.

В качестве обратимой операции обычно используется либо сложение по модулю количества букв в алфавите N: $c_i = t_i + g_i \pmod N$, либо, при представлении символов открытого текста в виде двоичного кода, операция поразрядного суммирования по модулю 2 (операция ‘побитовый XOR’): $c_i = t_i \oplus g_i$.

Расшифрование осуществляется применением к символам шифртекста и гаммы обратной операции: $t_i = c_i - g_i \pmod N$ или $t_i = c_i \oplus g_i$ (операция XOR является обратной к самой себе).

Стойкость систем шифрования, основанных на гаммировании, зависит от характеристик гаммы – ее длины и равномерности распределения вероятностей появления знаков гаммы.

Наиболее стойким является гаммирование с бесконечной равновероятной случайной гаммой, т.е. процедура шифрования, удовлетворяющая следующим трем условиям, каждое из которых является необходимым:

1) все символы гаммы полностью случайны и появляются в гамме с равными вероятностями;
2) длина гаммы равна длине открытого текста или превышает ее;
3) каждый ключ (гамма) используется для шифрования только одного текста, а потом уничтожается.

Такой шифр не может быть взломан в принципе, то есть является абсолютно стойким. Однако абсолютно стойкие шифры очень не удобны в использовании, и поэтому почти не применяются на практике. Обычно гамма либо получается периодическим повторением ключевой последовательности фиксированного размера, либо генерируется по некоторому правилу. Для генерации гаммы удобно использовать так называемые генераторы псевдослучайных чисел. Такие генераторы обычно основаны на рекуррентных математических формулах, использующих несколько ключевых (секретных) параметров.

Простейший генератор псевдослучайных чисел задается рекуррентной формулой:

$$g_i = ag_{i-1} + b(\mathrm{mod}\, m) \qquad (1)$$

где: g_i – i-й член последовательности псевдослучайных чисел;

a, b, m и g_0 — ключевые параметры.

Данная последовательность состоит из целых чисел от 0 до m–1, и если элементы g_i и g_j совпадут, то последующие участки последовательности также совпадут: $g_{i+1} = g_{j+1}$, $g_{i+2} = g_{j+2}$, и т.д.

Поэтому последовательность $\{g_i\}$ является периодической, и ее период не превышает m.

Для того чтобы период последовательности псевдослучайных чисел, сгенерированной по формуле (1), был максимальным (равным m), параметры формулы (1) должны удовлетворять следующим условиям:

- b и m —взаимно простые числа;
- a–1 делится на любой простой делитель числа m;
- a–1 кратно 4, если m кратно 4.

ТЕМА 5. АСИММЕТРИЧНЫЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ

Одной из основных проблем в практическом использовании рассмотренных систем шифрования является проблема распределения секретных ключей между абонентами и проблема хранения этих ключей.

Если в системе имеется N абонентов, то для обеспечения секретного обмена информацией между любыми двумя абонентами потребуется сгенерировать и распределить N(N–1)/2 секретных ключей, причем каждый абонент вынужден будет хранить N–1 секретный ключ для обмена информацией с остальными абонентами.

Основы асимметричных систем

Решить проблему распределения ключей помогает использование асимметричных криптографических систем. В этих системах для обмена данными используются два ключа, один из которых является секретным, а другой — открытым, т.е. общедоступным.

По этой причине асимметричные системы называются также двухключевыми.

Все асимметричные криптографические системы основаны на использовании односторонних функций с секретом.

Определение. Функция $F:X \to Y$ называется односторонней, если выполняются следующие два условия:

1) существует эффективный алгоритм, вычисляющий F(x) для любого $x \in X$;
2) не существует эффективного алгоритма инвертирования функции F, т.е. алгоритма, позволяющего определить значение x по значению F(x).

В данном определении «эффективным» называется полиномиальный алгоритм, т.е. алгоритм, который для получения результата для входа длины n тратит не более P(n) шагов, где P – некоторый полином.

В настоящее время теория алгоритмов не позволяет доказать не существование эффективных алгоритмов решения той или иной задачи. Поэтому, строго говоря, не известна ни одна односторонняя функция.

Однако предложено несколько функций, которые могут оказаться односторонними – для этих функций в настоящее время, несмотря на интенсивные исследования, не известны эффективные алгоритмы инвертирования.

Наиболее часто используются «односторонние» функции, заимствованные из теории чисел:

- функция F(a, b) = ab, то есть произведение двух чисел. Если a и b – простые числа, то по известному c = ab можно однозначно определить a и b. Эта задача называется задачей факторизации числа. До сих пор не известен ни один полиномиальный алгоритм для решения задачи факторизации, хотя для вычисления произведения чисел (т.е. самой функции F) такие алгоритмы известны;
- функция F(a, n) = $a^n \pmod M$, где a, n и M — целые числа. При известных a, n и M значение b = $a^n \pmod M$ может быть вычислено за полиномиальное количество шагов. Однако для обратной задачи — определить n по известным a, b и M (задача дискретного логарифмирования) — полиномиальные алгоритмы, в общем случае, пока не известны.

Не любая односторонняя функция может быть использована для шифрования. Действительно, если преобразовать открытый текст t с помощью односторонней функции: c = F(t), то расшифровать полученный текст, то есть по **c** восстановить t, не сможет уже никто, в том числе и законный получатель. Для использования в криптографии необходимо, чтобы задача инвертирования шифрующего преобразования (т.е. вычисления t по F(t)) была разрешима за приемлемое время, но сделать это мог только тот, кто знает секретный ключ.

Такие функции называются односторонними функциями с секретом (или с потайным ходом).

Определение. Односторонняя функция с секретом – это функция F_k :X $\rightarrow$Y, зависящая от параметра k $\in$ K (этот параметр называется секретом), для которой выполняются следующие условия:

1) при любом k $\in$ K существует эффективный алгоритм, вычисляющий Fk(x) для любого x$\in$ X;
2) при неизвестном k не существует эффективного алгоритма инвертирования функции F_k;
3) при известном k существует эффективный алгоритм инвертирования функции F_k.

Существование односторонних функций с секретом, как и существование односторонних функций, пока не доказано. Однако известны функции, которые могут оказаться односторонними функциями с секретом, если будет доказано, что инвертирование этих функций действительно является сложной задачей.

Рассмотрим в общем виде принцип использования односторонних функций с секретом для шифрования сообщений. Каждый абонент криптосистемы выбирает некоторую одностороннюю функцию E_k с секретом k. Функции E_k всех абонентов заносятся в общедоступный справочник, но значение секрета k каждый абонент, как и следует из названия, держит в секрете.

Если абонент B хочет переслать сообщение t абоненту A, он извлекает из справочника функцию E_k абонента A и с ее помощью вычисляет c = $E_k(t)$. Шифртекст c пересылается абоненту A, который по нему вычисляет исходное сообщение t, инвертировав функцию E_k с помощью секрета k. Расшифроватьсообщение может только абонент A, поскольку кроме него никто не знает секрет k.

Примечание. Обычно функции шифрования E_k для разных абонентов вычисляются по одному и тому же заранее установленному алгоритму, но в зависимости от некоторого параметра **p**. У каждого абонента параметр **p** собственный. Этот параметр называется открытым ключом данного абонента, поэтому асимметричные криптосистемы называют также криптосистемами с открытым ключом.

В качестве примера рассмотрим процедуру открытого распределения ключей и криптосистему RSA.

Процедура открытого распределения ключей

Как указывалось, одной из наиболее сложных проблем в применении криптографических систем является распределение секретных ключей между абонентами. При использовании классических, т.е. симметричных, систем шифрования, для распределения ключей необходимо устанавливать специальный канал, полностью защищенный от возможной атаки противника.

Асимметричные криптографические системы позволяют распределять секретные ключи по открытым каналам, т.е. каналам, которые потенциально могут быть прослушаны противником. Такая процедура открытого распределения ключей была впервые опубликована в 1976 году в работе У. Диффи и Э. Хеллмана «Новые направления в криптографии».

В основе процедуры Диффи-Хеллмана лежит использование односторонней функции дискретного возведения в степень:

$$F(x) = g^x \pmod p, \qquad (2)$$

где x — целое число ($1 \le x \le p-1$), p – простое число, g – первообразный корень по модулю p.

Определение. Первообразным корнем по модулю p называется такое целое число g ($g < p$), для которого:

1) все степени $g^1 \pmod p$, $g^2 \pmod p$, …, $g^{p-1} \pmod p$ различны;

2) для любого целого числа a, такого что $1 \le a \le p-1$, найдется n, при котором $a = g^n \pmod p$.

Возводя число g в степени 1, 2, …, p–1(по модулю p), получим все числа от 1 до p–1, образующие Zp * (мультипликативную группу кольца Zp). Поэтому такое число g называется также генератором группы Zp *.

Процедура Диффи-Хеллмана для открытого распределения ключей заключается в следующем. Для начала выбирается большое простое число p и число g — первообразный корень по модулю p.

Для обеспечения стойкости число p должно иметь длину, большую или равную 512 бит, и разложение числа p – 1 на множители должно содержать хотя бы один большой простой множитель (например, p – 1 = 2q,

где q — простое число). Здесь и далее длиной целого числа будем называть количество бит в двоичной записи этого числа.

При таком выборе числа p в настоящее время не существует эффективного алгоритма для решения задачи инвертирования функции (2).

Каждый абонент в качестве своего секретного ключа выбирает некоторое случайное число x, по которому вычисляет свой открытый ключ $y = g^x \pmod p$. Все абоненты помещают свои открытые ключи в общедоступный справочник.

После этого, если два абонента, A и B, захотят обменяться секретным сообщением, они берут из общедоступного справочника открытые ключи друг друга (соответственно, Y_A и Y_B) и вычисляют общий секретный ключ:

1) абонент A вычисляет

$$z_A = (y_B)^{x_A} = (g^{x_B})^{x_A} \pmod p = g^{x_A x_B} \pmod p;$$

2) абонент B вычисляет

$$z_B = (y_A)^{x_B} = (g^{x_A})^{x_B} \pmod p = g^{x_A x_B} \pmod p.$$

Таким образом, после выполнения описанной процедуры у абонентов A и B есть общее число $Z_A = Z_B$. Это число они при обмене сообщениями могут использовать в качестве ключа для шифрования (например, методом гаммирования).

Противник знает числа $Y_a = g^{x_A} \pmod p$ и $Y_b = g^{x_B} \pmod p$, но для того чтобы определить секретный ключ, ему необходимо решить задачу дискретного логарифмирования (по известным Y_A и Y_B вычислить X_A и X_B). Как уже отмечалось раньше, для этой задачи в настоящее время не существует эффективного алгоритма.

ТЕМА 6. КРИПТОСИСТЕМА RSA

Криптосистема RSA названа по первым буквам фамилий разработавших ее специалистов – Ривеста (R. Rivest), Эдлмана (L. Adleman). Эта система используется как для шифрования данных, так и для формирования цифровой подписи.

Опишем процедуру шифрования с помощью системы RSA, а затем дадим ее математическое обоснование. В системе RSA каждый абонент формирует для себя секретный и открытый ключ следующим образом:

1. выбирает два больших, неравных между собой, простых числа n = pq и m = (p–1)(q–1);
2. выбирает целое число e, такое , что e<m и НОД (e, m) = 1 (то есть число e должно быть взаимно просто с m);
3. выбирает число d, удовлетворяющее условию: ed = 1 (mod m);
4. секретным ключом абонента является тройка чисел (p, q, d), а открытым ключем пара чисел (n, e);
5. открытые ключи всех абонентов помещаются в общедоступный справочник.

Примечание. Существование числа d, удовлетворяющего условию шага 3, что следует из теоремы Евклида: “Для b найдутся целые числа x и y, такие, что ax + by = НОД (a,b)”. Расширенный алгоритм Евклида позволяет эффективно вычислить это число d.

Алгоритмы для построения больших (длиной 512 и более бит) простых чисел для нахождения числа e, удовлетворяющего условию сложны для понимания.

Функция шифрования сообщения, представленного в виде числа t (t<n) в системе RSA определяется формулой : $E(t) = t^e \pmod n$.

Функция расшифрования (зависящая от секретного ключа) задается формулой: $D(c) = c^d \pmod n$.

Длинное сообщение разбивается на блоки длиной $\log_2 n$ (чтобы каждый блок представлял собой число, меньшее n). Каждый блок шифруется, и затем расшифровывается, отдельно.

Проверим, что функция E(t) действительно является односторонней функцией с секретом.

Свойство 1 из определения односторонней функции с секретом выполняется, поскольку для возведения в степень (в том числе и по определенному модулю) существуют эффективные алгоритмы.

Свойство 2 пока строго не доказано. Считается, что для инвертирования функции E необходимо определить число d, а для этого надо вычислить m, что невозможно без разложения числа n на простые множители p и q.

Для доказательства свойства 3 надо убедиться, что функция D действительно является обратной к функции E, то есть что для любого числа t выполняется D(E(t)) = t.

Доказательство основано на теореме Эйлера из теории чисел: «Для любых взаимно простых целых чисел a и n выполняется соотношение»:

$a^{j(n)} = 1 \pmod n$, где:

$f(n)$ – функция Эйлера, равная количеству целых чисел, больших 0 и меньших n, и взаимно простых с n. Для n=pq, где p и q — простые, j (n) = (p–1)(q–1), то есть j (n) в точности равно выбранному нами параметру m. Поскольку n равно произведению двух больших простых чисел, любое произвольно выбранное число t взаимно просто с n с вероятностью, практически равной 1.

Докажем , что функция D обратна к функции E:

$$D(E(t)) = D(t^e \pmod n) = (t^e)^d \pmod n.$$

Числа e и d выбраны так, что выполняется условие $ed = 1 \pmod{j(n)}$. Это равносильно тому, что существует целое число r, такое , что $ed = r\, j(n) + 1$. Поэтому

$$(t^e)^d \pmod n = t^{r\,\varphi(n)+1} \pmod n = \left(t^{\varphi(n)}\right)^r \pmod n \cdot t \pmod n.$$

Воспользовавшись теоремой Эйлера, получим:

$$\left(t^{\varphi(n)}\right)^r \pmod n = 1^r \pmod n = 1.$$

Следовательно, мы доказали, что для любого t, меньшего n, выполняется: D(E(t)) = t (mod n) = t.

Особенности использования асимметричных криптосистем на практике

По эффективности и стойкости асимметричные (двухключевые) системы проигрывают симметричным (одноключевым) — при одинаковой длине ключа лучшие асимметричные процедуры шифрования работают медленнее и обеспечивают меньшую секретность, чем лучшие симметричные шифры. Поэтому обычно асимметричные системы используют для шифрования не самостоятельно, а в комплексе с симметричными системами. Например, шифрование данных с помощью симметричного (одноключевого) алгоритма A_1 и асимметричного (двухключевого) алгоритма A_2 выполняется в следующем порядке:

1) генерируется случайный ключ k_1 для алгоритма A_1;

2) с помощью этого ключа производится шифрование данных: c’ = $A_1(t, k_1)$;

3) Ключ k_1 шифруется с помощью алгоритма A_2 на открытом ключе

$$k_p : c'' = A_2(k_1, k_p)$$

4) Шифртекст представляет собой пару c=(c', c'').

Для того чтобы расшифровать полученное сообщение (c', c''), получатель по c'' восстанавливает ключ k_1, с помощью которого затем по c' расшифровывает исходный текст.

Поскольку длина ключа k_1 невелика по сравнению с длиной текста, подобная схема дает значительный выигрыш в скорости.

Примечание. Стойкость асимметричных систем основана на предположении о существовании односторонних функций. Это предположение пока не доказано, хотя и не опровергнуто. В результате развития теории алгоритмов могут быть получены эффективные методы решения задач, использующихся в асимметричных криптосистемах (например, задачи дискретного логарифмирования), и тогда такие системы перестанут быть стойкими, а все зашифрованные с их помощью документы смогут быть расшифрованы. Но даже если будет доказано отсутствие эффективных алгоритмов для решения таких задач, асимметричные системы все равно ос-

танутся только вычислительно стойкими, т.е. их взлом будет теоретически возможным, хотя и будет требовать больших временных и вычислительных ресурсов. Абсолютно стойкие системы шифрования есть только в классе симметричных систем.

Асимметричные системы нашли также широкое применение в криптографических протоколах, позволяя решать задачи, не сводящиеся к «классическому» шифрованию: цифровая подпись, аутентификация и т.д.

Тема 7. Способы увеличения стойкости шифров

Стойкость большинства из рассмотренных алгоритмов шифрования можно существенно повысить, модифицировав сам алгоритм применения. Рассмотрим некоторые способы повышения стойкости шифров.

Сцепление блоков

Определение. Система шифрования называется поточной, если при шифровании символы исходного текста последовательно заменяются на символы шифртекста в соответствии с некоторым алгоритмом: $t_i = E_k(c_i)$. При блочном шифровании исходный текст разбивается на блоки, и алгоритм шифрования преобразует одновременно все символы каждого блока.

Из рассмотренных ранее систем шифры замены и гаммирования относятся к поточным системам, а шифры перестановки и шифр RSA являются блочными.

Одним из существенных недостатков блочных шифров является тот факт, что одинаковые блоки открытого сообщения они преобразуют в одинаковые блоки шифртекста. Очевидно, что это понижает стойкость шифра — если к противнику попадет образец исходного текста вместе с соответствующим шифртекстом, то он сможет частично расшифровывать другие шифртексты, при условии что в них будут встречаться такие же блоки. Одним из способов избавления от подобного недостатка является использование блочных шифров в режиме сцепления блоков. В этом режиме при шифровании очередного блока используются также предыдущие блоки от-

крытого текста. Например, текущий блок открытого текста (T_i) суммируется побитово по модулю два с предыдущим блоком шифртекста (C_{i-1}), и к результату применяется алгоритм шифрования:

$$C_i = E_k(T_i \oplus C_{i-1}).$$

В качестве начального блока C_0 используется либо блок, состоящий только из нулей, либо произвольный случайный блок (в этом случае он включается в шифртекст).

Возможна также двойственная схема, при которой алгоритм шифрования применяется к предыдущему блоку шифртекста, а затем берется побитовая сумма по модулю два с текущим блоком:

$$C_i = T_i \oplus E_k(C_{i-1}).$$

Применение описанных схем обеспечивает зависимость всех последующих блоков шифртекста от всех предыдущих блоков открытого текста. Поэтому изменение какого-то блока открытого текста приводит к изменению не только соответствующего блока шифртекста, но и всех последующих блоков шифртекста.

Добавление случайных данных

Еще одним эффективным способом затруднить криптоанализ шифра является добавление случайных данных к шифруемому тексту. Этот способ заключается в следующем. Перед началом шифрования текста T необходимо сгенерировать случайный блок данных R заранее определенной длины, и дописать его к тексту. Получившийся блок данных R|T, где знак ‘|’ означает конкатенацию (сцепление) двоичных наборов данных, преобразуется в шифртекст с помощью процедуры шифрования по ключу k: C = E k (R|T). Для того чтобы расшифровать сообщение, получатель применяет к шифртексту процедуру расшифрования, получая екоторый набор данных B = D k (C). Этот набор данных представляет собой конкатенацию R и T, и,

поскольку длина блока R известна, исходный текст T однозначно восстанавливается из B.

Достоинством такого метода является то, что при шифровании одного и того же блока данных в разные моменты времени получаются различные блоки шифртекста. А это сильно затрудняет атаку на шифр.

Недетерминированные шифры

При оценке стойкости шифра обычно предполагается, что алгоритм шифрования известен лицу, пытающемуся взломать шифр. Это предположение основывается на том факте, что, с одной стороны, практически нереально удержать этот алгоритм в секрете, не ограничивая распространение программных средств шифрования, а с другой – оценка реальной стойкости шифра возможна только после открытого изучения алгоритма шифрования экспертами.

Очевидно, что знание алгоритма преобразования данных существенно облегчает криптоанализ. Для того чтобы этого избежать, используются недетерминированные (гибкие) шифры.

В простейшем случае такие шифры включают в себя набор процедур, $F_1, F_2, ..., F_n$ и алгоритм, который по секретному ключу k формирует последовательность $a_{(1)}, a_{(2)}, ..., a_{(i)}$. Процедура шифрования текста T по ключу k заключается в применении к этому тексту процедур F_i в порядке, определяемом последовательностью a(i):

$$C = E_k(T) = F_{a(i)}(\ldots(F_{a(2)}(F_{a(1)}(T))\ldots).$$

Таким образом, недетерминированный алгоритм шифрования состоит из известных процедур, что позволяет научно оценить стойкость шифра, но порядок применения этих процедур определяется секретным ключом и поэтому неизвестен криптоаналитику.

Тема 8. Система конфиденциального обмена информацией PGP. Основы шифрования. Анализ стойкости системы

Разработанная Филипом Циммерманном программа PGP (Pretty Good Privacy «Почти полная приватная безопасность») относится к классу систем с двумя ключами, публичным и секретным. Это означает, что пользователь может сообщить о своем публичном ключе всему миру, при этом пользователи программы смогут отправлять вам зашифрованные сообщения, которые никто, кроме вас, расшифровать не сможет. Вы же их расшифровываете с помощью вашего второго, секретного ключа, который держится в тайне. Публичный ключ выглядит как небольшой текстовый блок, и его можно разместить на своей Web странице или послать его электронной почтой своему партнеру.

Рассмотрим функционирование системы. Предположим, вы желаете отправить сообщение своей коллеге (назовем ее Алис), и вы хотите, чтобы никто, кроме Алис, не смог его прочитать. Как показано на рис.8.1., вы можете зашифровать (или закодировать), то есть преобразовать сообщение безнадежно сложным образом, зная, что никто, кроме вас и Алис не сможет его прочитать. Вы применяете для шифрования криптографический ключ, а Алис должна использовать тот же ключ для его расшифровки (или раскодирования). По крайней мере, так это выглядит при применении обычной криптографии с «секретным ключом».

Один и тот же ключ используется как для зашифровки, так и для расшифровки сообщения. Это означает, что этот ключ должен быть сначала передан по надежному каналу, с тем, чтобы обе стороны знали его до того, как передавать зашифрованное сообщение по ненадежному каналу. Но если у вас есть надежный канал, которым вы можете воспользоваться для обмена ключами, спрашивается, зачем вам вообще нужна криптография?

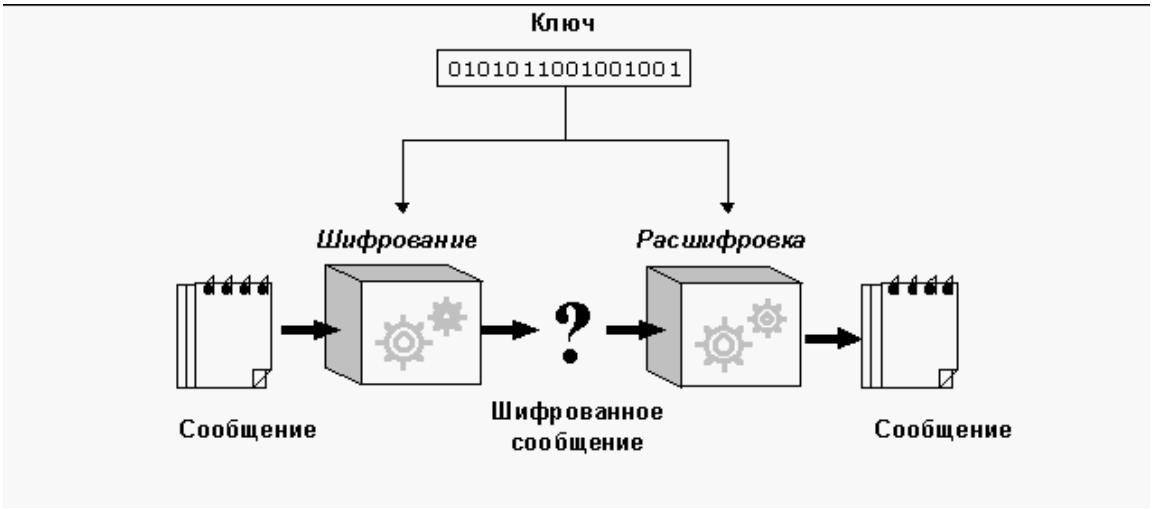

Рис. 8.1. Общая схема шифрования

Как работает криптография с открытым ключом

Как показано на рис. 8.1., при использовании криптографии с открытым ключом, каждый обладает двумя дополняющими друг друга ключами: открытым и закрытым. Каждый из этих ключей подходит для расшифровки сообщения, зашифрованного с применением другого ключа. Открытый ключ может быть опубликован и широко распространен по сетям коммуникаций.

Такой протокол обеспечивает приватность без необходимости обладания надежным каналом, которого требует обычная криптография с секретным ключом.

Кто угодно может использовать открытый ключ получателя для того, чтобы зашифровать отправляемое тому сообщение. Получатель затем использует соответствующий закрытый ключ для его расшифровки. Никто, кроме получателя, не может расшифровать сообщение, так как никто больше не имеет доступа к этому закрытому ключу. Даже тот, кто зашифровал сообщение с помощью открытого ключа, не сможет его расшифровать.

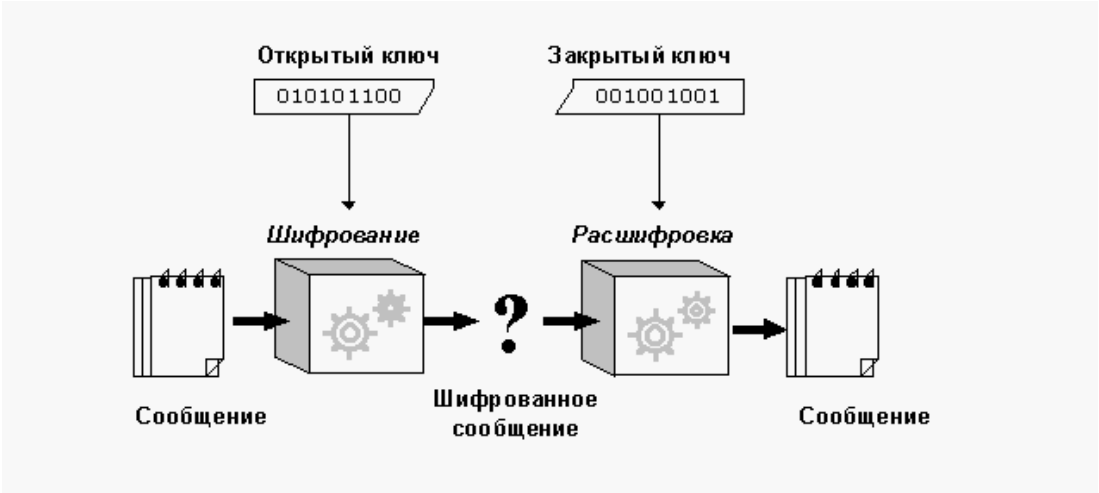

Рис. 8.2. Схема шифрования с открытым ключом

Как шифруются файлы и сообщения

Поскольку алгоритм шифрования с открытым ключом значительно медленнее алгоритма обычного шифрования, использующего один ключ, шифрование лучше всего выполнять, используя процесс, показанный на рис.3.

Для шифрования сообщения используется качественный и быстрый алгоритм обычного шифрования с секретным ключом. Это сообщение в оригинальной, незашифрованной форме называется "открытым текстом". В ходе процесса, невидимого для пользователя, для обычного шифрования открытого текста используется временный случайный ключ, сгенерированный специально для этого «сеанса». Затем этот случайный ключ шифруется с помощью открытого ключа получателя. Этот, зашифрованный с использованием открытого ключа «сеансовый ключ», отправляется получателю вместе с зашифрованным текстом («шифровкой»).

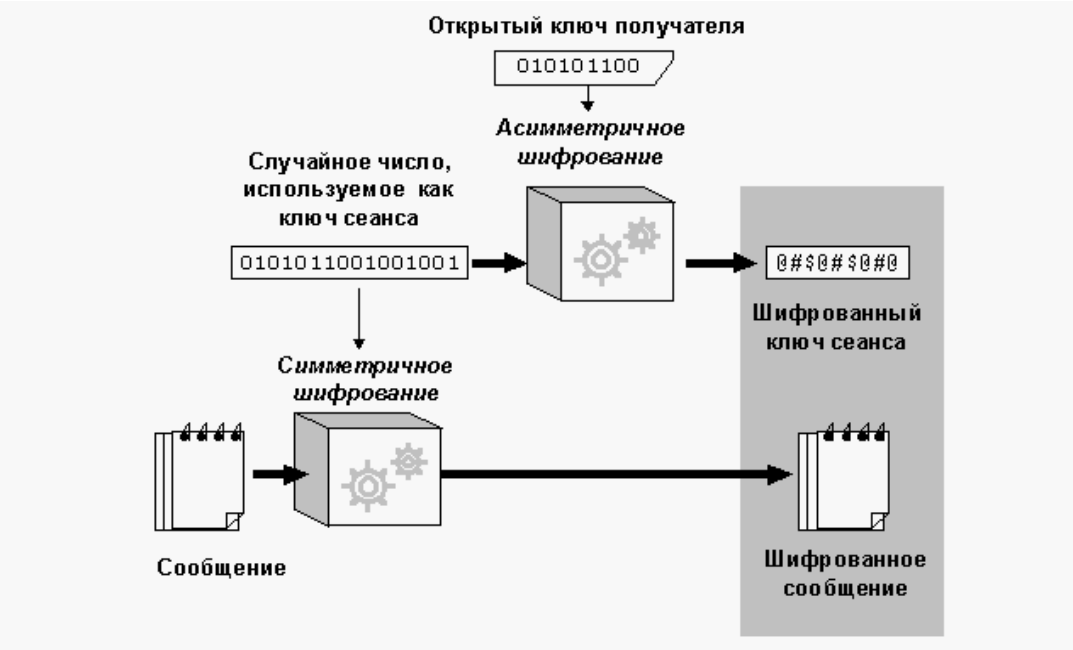

Рис. 8.3. Процесс шифрования

Симметричные алгоритмы PGP

PGP предоставляет выбор из ряда различных алгоритмов с секретным ключом, используемых для шифрования тела сообщения. Под алгоритмом с секретным ключом мы понимаем обычный, или симметричный, блочный шифр, который использует для шифрования и расшифровки один и тот же ключ. PGP предоставляет три симметричных блочных шифра, включая CAST, тройной DES и IDEA. Эти алгоритмы не являются «домашними поделками»; все они разработаны командами криптографов с выдающейся репутацией.

Все три шифра оперируют 64-битными блоками открытого текста и шифровки. CAST и IDEA работают со 128-битным ключом, а тройной DES – с ключом длиной 168 бит. Как и Стандарт шифрования данных (DES), любой из этих шифров может использоваться в режимах CFB (контекстно-зависимого шифрования) и CBC (односторонне-контекстно-зависимого шифрования). PGP использует их в режиме CFB с размером блока 64 бит.

Алгоритм CAST включен в PGP, потому что он является многообещающим в качестве хорошего блочного шифра с 128-битной длиной ключа, потому что он очень быстрый, и потому что он может быть использован бесплатно. Его название состоит из инициалов разработчиков, Карлис-

ла Адамса и Стаффорда Тавареса из Northern Telecom (Nortel). Nortel подал патентную заявку на CAST, но разработчики сделали письменное заявление о том, что CAST может использоваться всеми на бесплатной основе. Специалистами с хорошей репутацией в области криптографии CAST признан исключительно хорошо построенным алгоритмом. Он основан на очень формальном подходе, с использованием ряда математически доказуемых положений. Это позволяет предположить, что для взлома его 128-битного ключа требуется исчерпывающий перебор вариантов. Существуют сильные аргументы в пользу того, что CAST полностью иммунен как к линейному, так и к дифференциальному криптоанализу (двум самым мощным из опубликованных схем криптоанализа, обе из которых оказались достаточно эффективными для взлома DES).

Блочный шифр IDEA (Международный алгоритм шифрования данных) основан на понятии «смешения операций, принадлежащих различным алгебраическим группам». Он был разработан в ETH в Цюрихе Джеймсом Л. Мэссли и Ксуэйджа Лайем, и опубликован в 1990 г. До сих пор IDEA оказывался устойчивым к криптографическим атакам в большей степени, чем другие шифры, такие, как FEAL, REDOC-II, LOKI, Snefru и Khafre. IDEA более устойчив, чем DES, к очень успешной криптографической атаке Бихама и Шамира, использующей дифференциальный криптоанализ, так же, как и к атакам с применением линейного криптоанализа. Поскольку этот шифр продолжает быть мишенью для атак со стороны наиболее выдающихся представителей мира криптоанализа, уверенность в стойкости IDEA растет со временем.

В репертуар блочных шифров PGP включает также тройной DES, использующий три ключа. Алгоритм DES был разработан в IBM в середине 1970-х гг. При хорошем дизайне, 56-битный ключ является по сегодняшним стандартом слишком коротким. Тройной DES очень стоек, и изучался многие годы, так что ставка на его использование может оказаться более верной, чем использование таких шифров, как CAST и DES. Тройной DES – это DES, примененный к одному и тому же блоку данных три раза с тремя разными ключами, причем второй раз DES запускается в режиме расшифровки. Хотя тройной DES много медленнее, чем CAST и

IDEA, скорость обычно не является критичной для применения в электронной почте. Тройной DES обладает ключом длиной 168 бит, но, эффективная приведенная длина ключа, вероятно, составляет 112 бит при атаке, когда атакующий располагает невероятно большим ресурсом для хранения данных. Открытые ключи, генерируемые PGP версий 5.0 или более ранних, содержат информацию, которая сообщает отправителю, какие из блочных шифров поддерживаются программным обеспечением получателя, так что программное обеспечение отправителя знает, какие из шифров могут быть использованы. С открытыми ключами DSS/DH могут использоваться блочные шифры CAST, IDEA и тройной DES, причем CAST является выбором по умолчанию. С открытыми ключами RSA в настоящее время может использоваться только IDEA, так как ранние версии PGP поддерживают лишь RSA и IDEA.

Сжатие данных

Обычно PGP сжимает данные до того, как шифровать их, так как сжимать их после шифрования слишком поздно: зашифрованные данные несжимаемы. Сжатие данных экономит время передачи данных по модему, дисковое пространство и, что более важно, усиливает криптографическую безопасность. Большинство приемов криптоанализа используют для взлома шифра избыточность исходного открытого текста. Сжатие данных снижает избыточность, значительно увеличивая, таким образом, устойчивость к криптоанализу. На сжатие исходного текста требуется дополнительное время, но с точки зрения безопасности это оправдано.

О случайных числах, используемых в качестве сеансовых ключей

В PGP для генерации временных сеансовых ключей использован криптографически стойкий генератор псевдослучайных чисел. Если файл с исходным для генерации этих чисел числом не существует, он автоматически генерируется с использованием строго случайных событий, в качестве источника которых используются параметры нажатий клавиш и движений мыши.

Этот генератор записывает файл с исходным числом каждый раз при его использовании, смешивая его содержимое с данными, получаемыми из

значения даты и других строго случайных источников. В качестве средства генерации случайных чисел использован алгоритм обычного шифрования. Этот файл содержит как исходный материал для генерации случайных чисел, так и исходный материал для ключа, используемого для генерации следующего случайного числа.

Этот файл с исходным случайным числом должен предохраняться от несанкционированного доступа для снижения риска того, что атакующему станет, доступен ваш следующий или предыдущий сеансовый ключ. Хотя атакующему будет крайне сложно извлечь какую-либо пользу от захвата этого файла, криптографически очищаемого до и после каждого использования, благоразумным будет попытаться предохранить его от попадания в чужие руки. Если это возможно, сделайте этот файл доступным для чтения только себе. Если же это невозможно, не позволяйте другим бесконтрольно копировать данные с вашего компьютера.

Как осуществляется расшифровка

Как показано на Рис. 8. 4., процесс расшифровки обратен по отношению к шифрованию. Закрытый ключ получателя используется для восстановления временного сеансового ключа, который, в свою очередь, используется при запуске быстрого обычного алгоритма с секретным ключом для расшифровки основного тела сообщения.

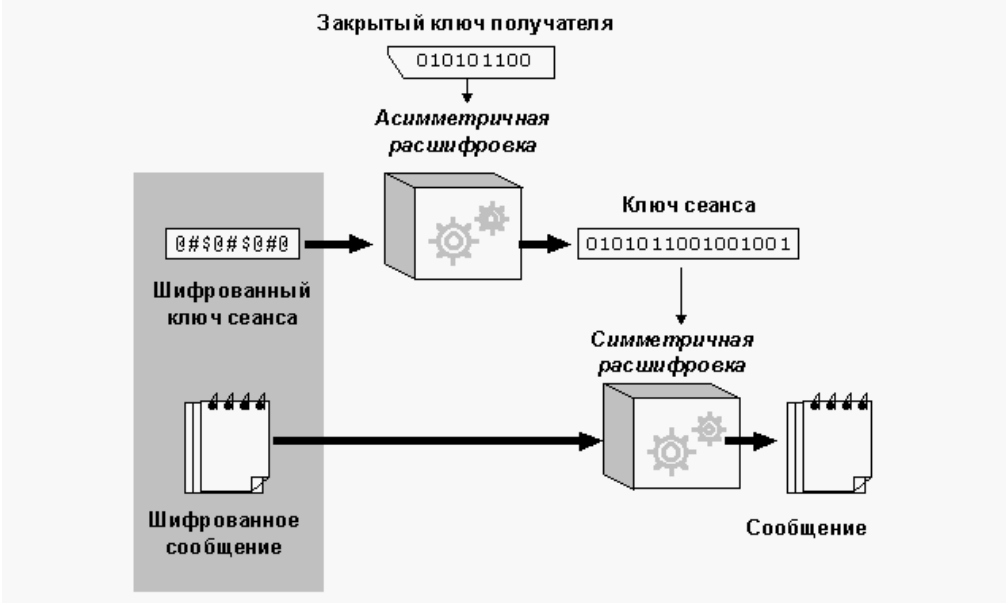

Рис. 8.4. Процесс расшифрования

Как осуществляется электронная подпись

PGP накладывает цифровую подпись для обеспечения аутентификации сообщения. Закрытый ключ отправителя используется для зашифровки дайджеста сообщения, таким образом «подписывая» сообщение. Дайджест сообщения – это 160- или 128-битная криптографически стойкая односторонняя хэш-функция. В чем-то она похожа на «контрольную сумму», или код проверки ошибок CRC, который компактно представляет сообщение и используется для проверки сообщения на наличие изменений. В отличие от CRC, дайджест сообщения формируется таким образом, что злоумышленник не может сгенерировать поддельное сообщение с аналогичным дайджестом. Дайджест сообщения передается в зашифрованном закрытым ключом отправителя виде, составляя цифровую подпись сообщения.

Получатель может проверить правильность цифровой подписи, используя открытый ключ отправителя для расшифровки дайджеста сообщения. Это доказывает, что тот, кто указан в качестве отправителя сообщения, является его создателем и что сообщение не было впоследствии изменено другим человеком, так как только отправитель владеет своим закрытым ключом, использованным для формирования цифровой подписи. Подделка цифровой подписи невозможна, и отправитель не может впоследствии отрицать ее подлинность.

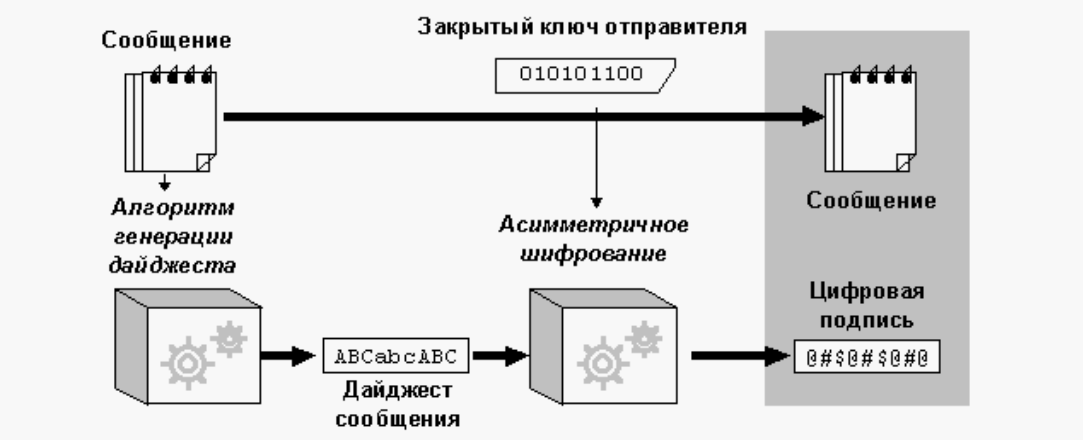

Рис. 8. 5. Схема создания дайджеста сообщения

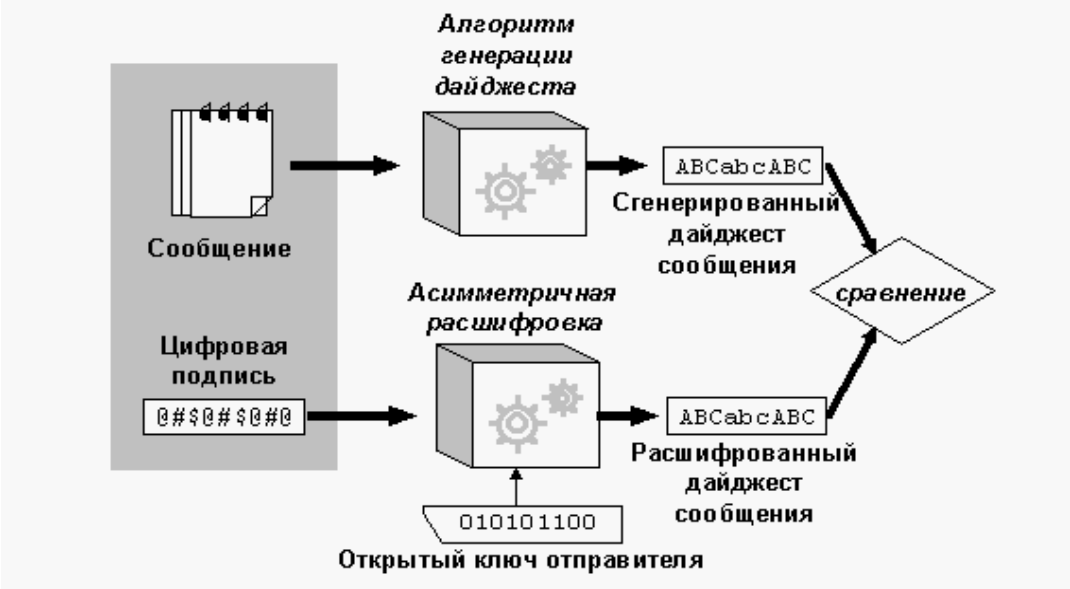

Рис. 8.6. Схема проверки подлинности документа

О дайджесте сообщения

Дайджест сообщения – это компактная (160- или 128-битная) «выжимка» вашего сообщения, или контрольная сумма файла. Ее можно также сравнить с отпечатком вашего пальца на сообщении или файле. Дайджест сообщения «представляет» ваше сообщение таким образом, что если сообщение подвергнется какому-либо изменению, ему будет соответствовать другой дайджест. Это позволяет обнаружить любое изменение, внесенное в сообщение злоумышленником. Дайджест сообщения вычисляется с использованием криптографически стойкой односторонней хэш-функции сообщения. Для атакующего должно быть вычислительно невозможным изобрести подложное сообщение, которому соответствовал бы идентичный дайджест. В этом отношении, дайджест сообщения гораздо лучше контрольной суммы, потому что сгенерировать другое сообщение, дающее ту же контрольную сумму, достаточно просто. Но, как и из контрольной суммы, из дайджеста сообщения невозможно восстановить само сообщение.

В PGP алгоритм получения дайджеста сообщения называется SHA (Алгоритм защищенного хеширования), он разработан в АНБ для Национального института стандартов и технологий (NIST). SHA является 160-битным алгоритмом хеширования. Некоторые относятся ко всему, исходящему от АНБ, с подозрением, поскольку АНБ занимается перехватом

коммуникации и взломом шифров. Но следует иметь в виду, что АНБ не заинтересовано в подделке подписей, и что правительство только выиграет от внедрения стандарта цифровой подписи, которую невозможно подделать, поскольку это помешает кому-либо отрицать подлинность своей подписи. Это несет очевидные преимущества для поддержания законности и сбора разведывательных данных. Кроме того, SHA был опубликован в открытой литературе. Он внимательно изучен большинством криптографов во всем мире, которые специализируются на хэш-функциях. Они единодушно заявляют об исключительно хорошей проработанности SHA. Во всех новых версиях PGP в качестве алгоритма генерации дайджестов сообщений используется SHA, а для наложения цифровой подписи – новые ключи DSS, соответствующие Стандарту цифровой подписи NIST. Из соображений совместимости, новые версии PGP продолжают поддерживать алгоритм MD5 в сочетании с RSA, поскольку такой была технология подписи в прежних версиях PGP.

Использовавшийся в ранних версиях PGP для создания дайджестов сообщений алгоритм MD5, предоставленный в общее пользование RSA Data Security, Inc., является 128-битным хеш-алгоритмом. Он был почти взломан в 1996 г. немецким криптографом Хансом Доббертином. Хотя к настоящему времени MD5 и не взломан окончательно, в нем были обнаружены настолько серьезные слабые места, что никто не должен продолжать использовать его для генерации цифровой подписи. Дальнейшие разработки в этой области могут окончательно взломать его, позволив, таким образом, подделывать подписи. Если вы не хотите однажды увидеть свою цифровую подпись PGP на каком-либо подложном документе, примите совет перейти к новым ключам DSS в качестве основного метода наложения цифровой подписи, так как DSS использует в качестве алгоритма защищенного хеширования SHA.

Как защищать открытые ключи от подмены

В криптосистемах с открытыми ключами вам не нужно защищать открытые ключи от несанкционированного доступа. На самом деле, чем шире они распространяются, тем лучше. Однако важно защитить открытые

ключи от подделки, чтобы быть уверенным в том, что ключ действительно принадлежит тому, чье имя он несет. Давайте сначала взглянем на потенциальную опасность такой подмены, а затем опишем, как ее избежать при использовании PGP.

Предположим, вы хотите отправить приватное сообщение Алис. Вы подгружаете открытый ключ Алис с какой-нибудь электронной доски объявлений. Вы шифруете свое письмо Алис ее открытым ключом и отправляете его через систему электронной почты.

К несчастью, незаметно для Вас или Алис, другой пользователь, по имени Виктор, проникает на электронную доску объявлений и генерирует открытый ключ, несущий идентификатор пользователя Алис. Он тайно подменяет своим фальшивым ключом настоящий открытый ключ Алис. Вы неосторожно используете этот фальшивый ключ, принадлежащий Виктору, вместо открытого ключа Алис. Все выглядит нормально, потому что фальшивый ключ несет идентификатор пользователя Алис. Теперь Виктор может расшифровать сообщение, предназначенное Алис, поскольку обладает секретным ключом из фальшивой пары. Он даже может затем снова зашифровать расшифрованное им сообщение настоящим ключом Алис и отправить его ей, так что никто ничего не заметит. Более того, он даже сможет потом накладывать от имени Алис подпись, которая будет казаться подлинной, так как все будут использовать для ее проверки фальшивый ключ.

Единственный способ предотвратить такую неприятность – это исключить возможность подделки открытых ключей. Если вы получили открытый ключ Алис непосредственно от нее, проблем не возникает. Но это может быть затруднительным, если Алис находится на расстоянии тысячи км, или по другим причинам с ней невозможно встретиться лично.

Возможно, открытый ключ Алис может передать вам ваш общий друг Генри, которому вы оба доверяете, и который знает, что обладает подлинным ключом Алис. Генри может подписать открытый ключ Алис, ручаясь, таким образом, за его целостность. Для подписи он должен использовать всей собственный закрытый ключ.

Эта процедура создает подписанный сертификат открытого ключа, который подтверждает, что ключ Алис не был подделан. Конечно, для того,

чтобы вы могли проверить правильность подписи Генри, необходимо, чтобы у вас была заведомо правильная копия его открытого ключа. Возможно, Генри может также передать Алис подписанную копию вашего ключа. Генри, таким образом, будет служить «посредником» между вами и Алис.

Этот подписанный сертификат открытого ключа Алис или Генри могут подгрузить на электронную доску объявлений, откуда вы можете его позднее скопировать. Так как вы в состоянии проверить подпись Генри с помощью его открытого ключа, вы можете быть уверены, что это – действительно ключ Алис. Никакой злодей не сможет обмануть вас, заставив поверить, что изготовленный им фальшивый ключ принадлежит Алис, поскольку никто не может подделать подпись Генри.

Централизованный доверенный сертификатор особенно подходит для больших централизованно управляемых организаций, правительственных или корпоративных. Некоторые организационные среды используют иерархии доверенных сертификаторов.

Для более децентрализованных сред, вероятно, более походящим, чем создание централизованного доверенного сертификатора, будет предоставление всем пользователям возможности действовать в качестве «представителей».

Одним из наиболее привлекательных свойств PGP остается то, что она в равной мере успешно может работать как в централизованной среде с доверенным сертификатором, так и в децентрализованной среде, в которой индивидуумы свободно обмениваются своими ключами.

Задача защиты открытых ключей от подделки как таковая составляет единственную серьезную проблему практического приложения криптографии с открытыми ключами. Она является "ахиллесовой пятой" этой технологии, и сложность программного обеспечения в основном связана с решением именно этой задачи.

Как защищать закрытые ключи от раскрытия

Свой закрытый ключ и пароль следует сохранять очень тщательно. Если же закрытый ключ окажется скомпрометированным, следует быстро оповестить об этом все заинтересованные стороны, пока кто-нибудь не использовал его для фальшивой подписи от вашего имени. Например, укра-

денный ключ может быть использован для создания фальшивых сертификатов открытых ключей, что создаст проблемы для массы людей, особенно если ваша подпись пользуется широким доверием. И, разумеется, компрометация вашего ключа может привести к тому, что все зашифрованные сообщения, адресованные вам, смогут быть расшифрованы.

Защиту закрытого ключа следует начать с того, что вы должны всегда сохранять над ним физический контроль. Держать его на домашнем персональном компьютере или на переносном компьютере, который вы носите с собой, приемлемо. Если вы вынуждены использовать служебный компьютер, над которым вы не всегда сохраняете физический контроль, держите связки закрытых и открытых ключей на защищенном от записи флоппи-диске, и забирайте его с собой, когда выходите из офиса.

Стойкость криптографических программ

При оценке пакета криптографического программного обеспечения всегда остается вопрос: «Почему вы должны доверять этому продукту?» Он остается даже в случае, когда вы сами изучили исходный текст программ – ведь не каждый обладает криптографическим опытом, чтобы оценить уровень безопасности. И даже если вы опытный криптограф, от Вас могут ускользнуть неочевидные слабые места в алгоритмах.

Когда в начале семидесятых автор PGP учился в колледже, он изобрел схему шифрования, которая казалась ему блестящей. Для создания шифровки к открытому тексту добавлялась простая последовательность псевдослучайных чисел. Казалось бы, это должно противостоять любому частотному анализу шифровки и сделать ее нераскрываемой даже правительственными разведывательными службами с их огромными ресурсами. Я так гордился своим достижением!

Годами позже, автор обнаружил ту же самую схему в нескольких текстах введения в криптографию и учебниках. Как мило: о ней думали и другие криптографы. К несчастью, эта схема приводилась как задание для простой домашней работы на применение элементарных приемов криптоанализа для тривиального ее взлома.

Из этого унизительного опыта автор узнал, как просто впасть в ложное чувство безопасности при разработке алгоритма шифрования. Большинство людей просто не представляет, как немыслимо сложно придумать алгоритм шифрования, который выдержит продолжительную и целеустремленную атаку со стороны хорошо оснащенного противника. Многие разработчики обычного программного обеспечения используют столь же наивные схемы шифрования (а иногда – и ту же самую схему), а некоторые из этих схем оказываются внедренными в коммерческие программные пакеты шифрования и продаются за немалые деньги ничего не подозревающим пользователям.

Некоторые коммерческие пакеты используют Федеральный стандарт шифрования данных (DES), действительно неплохой алгоритм обычного шифрования, рекомендованный правительством для коммерческого использования (но не для защиты правительственной секретной информации – достаточно странно…). Существует несколько «режимов использования» DES, некоторые из которых лучше, чем другие. Правительство не рекомендует использовать для шифрования сообщений самый слабый из них, ECB («электронная кодовая книга»), а рекомендует более стойкие, но и более сложные режимы «шифрования с обратной связью» (CFB) и «цепочки шифрованных блоков» (CBC) режимы.

Качество коммерческого криптографического программного обеспечения в США подрывается тремя факторами:

- первым из них является практически повсеместная некомпетентность разработчиков коммерческого криптографического программного обеспечения (впрочем, с публикацией PGP это начало меняться). Каждый программист воображает себя криптографом, что ведет к распространению исключительно плохого криптообеспечения;
- второй – жесткое и систематическое подавление хороших коммерческих технологий шифрования со стороны АНБ посредством установления юридических ограничений и экономического давления. Частично, это давление производится путем строгих ограничений на экспорт, что, в свою очередь, благодаря законам рынка программного обеспечения,

ведет к подавлению криптографического программного обеспечения для внутреннего применения;

- другим основным методом подавления служит предоставление всех патентных прав на алгоритмы шифрования с открытым ключом единственной компании, что приводит к замыканию проблемы предотвращения распространения этой технологии на одну фирму.

Уязвимые места

Никакая система защиты данных не является неуязвимой. PGP можно обойти целым рядом способов. Защищая данные, вы должны задать себе вопрос: является ли информация, которую вы пытаетесь защитить, более ценной для атакующего, чем стоимость атаки? Ответ на этот вопрос приведет вас к тому, чтобы защититься от дешевых способов атаки и не беспокоиться о возможности более дорогой атаки.

Нижеследующее обсуждение местами может показаться маниакальным, но такой подход уместен при обсуждении уязвимых мест. «Если все персональные компьютеры мира (260 миллионов штук) заставить работать с единственным сообщением, зашифрованным PGP, расшифровка такого сообщения в среднем потребует времени, в 12 миллионов раз превышающего возраст Вселенной». Уильям Кроуэлл, заместитель директора Агентства национальной безопасности, 20 марта 1997 г.

Скомпрометированные пароль и закрытый ключ

Наверное, самую простую атаку можно осуществить, если вы оставите записанный где-нибудь пароль, защищающий ваш закрытый ключ. Если кто-нибудь получит его, а затем получит доступ к файлу с вашим закрытым ключом, он сможет читать адресованные вам зашифрованные сообщения и ставить от вашего имени цифровую подпись.

Вот некоторые рекомендации по защите пароля:

1. Не используйте очевидные фразы, которые легко угадать, например, имена своих детей или супруги.
2. Используйте в пароле пробелы и комбинации цифр и букв. Если ваш пароль будет состоять из одного слова, его очень просто отгадать, за-

ставив компьютер перебрать все слова в словаре. Именно поэтому фраза в качестве пароля гораздо лучше, чем слово. Более изощренный злоумышленник может заставить свой компьютер в поисках пароля перебрать словарь известных цитат.

3. Используйте творческий подход. Придумайте фразу, которую легко запомнить, но трудно угадать: такая фраза может быть составлена из бессмысленных выражений или очень редких литературных цитат.

Подделка открытых ключей

Самое уязвимое место – это возможность подделки открытых ключей. Вероятно, это самое серьезное слабое место любой криптосистемы с открытыми ключами, в частности, потому, что большинство новичков не в состоянии немедленно обнаружить такую подделку. О том, почему это важно, и какие против этого следует предпринимать контрмеры, подробно написано выше, в разделе «Как защитить открытые ключи от подделки».

Вкратце: когда вы используете чей-то открытый ключ, удостоверьтесь, что он не был подделан. Целостности нового чужого открытого ключа следует доверять только если он получен непосредственно от его владельца или подписан кем-то, кому вы доверяете. Обеспечьте невозможность подделки открытых ключей на вашей связке. Сохраняйте физический контроль, как над связкой открытых ключей, так и над своим закрытым ключом, при возможности сохраняйте их на своем персональном компьютере, а не на удаленной системе с разделением доступа. Сохраняйте резервную копию обеих связок.

Не до конца удаленные файлы

Еще одна потенциальная проблема безопасности связана со способом, которым большинство операционных систем удаляет файлы. Когда вы шифруете файл и затем удаляете файл с исходным открытым текстом, операционная система не стирает данные физически. Она просто помечает соответствующие блоки на диске, как свободные, допуская тем самым повторное использование этого пространства. Это похоже на то, как если бы ненужные секретные документы выбрасывались в мусорную корзину вме-

сто того, чтобы отправить их в шредер. Блоки диска все еще сохраняют исходные секретные данные, которые вы хотели стереть, и лишь со временем будут заняты новыми данными. Если злоумышленник прочитает эти блоки данных вскоре после того, как они помечены как свободные, он сможет восстановить ваш исходный открытый текст.

Это может произойти и случайно: если из-за какого-нибудь сбоя будут уничтожены или испорчены другие файлы, для их восстановления запустят программу восстановления, а она восстановит также и некоторые из ранее стертых файлов. Может случиться так, что среди последних окажутся и ваши конфиденциальные файлы, которые вы намеревались уничтожить без следа, но они могут попасться на глаза тому, кто восстанавливает поврежденный диск. Даже когда вы создаете исходное сообщение с использованием текстового редактора или Word-процессора, программа может оставить множество промежуточных временных файлов, просто потому, что она так работает. Эти временные файлы обычно удаляются редактором при его закрытии, но фрагменты вашего секретного текста остаются где-то на диске.

Единственный способ предотвратить восстановление открытого текста – это каким-либо образом обеспечить перезапись места, занимаемого удаленными файлами. Если вы не уверены, что все блоки, занимаемые на диске удаленными файлами, будут вскоре использованы, нужно предпринять активные шаги для перезаписи места, занятого исходным открытым текстом и временными файлами, создаваемыми Word-процессором. Это можно осуществить с использованием любой утилиты, которая способна перезаписать все неиспользованные блоки на диске.

Вирусы и закладки

Другая атака может быть предпринята с помощью специально разработанного компьютерного вируса или червя, который инфицирует PGP или операционную систему. Такой гипотетический вирус может перехватывать пароль, закрытый ключ или расшифрованное сообщение, а затем тайно сохранять их в файле или передавать по сети своему создателю. Вирус также может модифицировать PGP таким образом, чтобы она перестала надле-

жащим образом проверять подписи. Такая атака обойдется дешевле, чем криптоаналитическая.

Защита от подобных нападений подпадает под категорию общих мер защиты от вирусных инфекций. Существует ряд коммерчески доступных антивирусных программ с неплохими возможностями, а также набор гигиенических процедур, следование которым серьезно снижает риск заражения вирусами. PGP не содержит никакой защиты от вирусов, и ее использование предполагает, что ваш персональный компьютер является надежной средой. Если такой вирус или червь действительно появится, будем надеяться, что сообщение об этом достигнет ушей каждого.

Нарушение режима физической безопасности

Нарушение режима физического доступа может позволить постороннему захватить ваши файлы с исходным текстом или отпечатанные сообщения. Серьезно настроенный противник может выполнить это посредством ограбления, роясь в мусоре, спровоцировав необоснованный обыск и изъятие, с помощью шантажа или инфильтрации в ряды ваших сотрудников. Применение некоторых из этих методов особенно подходит против самодеятельных политических организаций, использующих в основном труд неоплачиваемых добровольцев.

Не стоит впадать в ложное чувство безопасности только потому, что у вас есть криптографическое средство. Приемы криптографии защищают данные только пока те зашифрованы, и не могут воспрепятствовать нарушению режима физической безопасности, при котором скомпрометированными могут оказаться исходные тексты, письменная или звуковая информация.

Этот вид атаки также дешевле, чем криптоаналитическая атака на PGP.

Радиоатака

Хорошо оснащенным противником может быть предпринята атака еще одного вида, предполагающая удаленный перехват электромагнитного излучения, испускаемого вашим компьютером. Эта дорогая и часто трудоемкая атака, вероятно, также является более дешевой, чем криптоанализ.

Соответствующим образом оборудованный фургон может припарковаться рядом с вашим офисом и издалека перехватывать нажатия клавиш и сообщения, отображаемые на мониторе. Это скомпрометирует все ваши пароли, сообщения и т.п. Такая атака может быть предотвращена соответствующим экранированием всего компьютерного оборудования и сетевых кабелей с тем, чтобы они не испускали излучения. Технология такого экранирования известна под названием «Tempest» и используется рядом правительственных служб и фирм, выполняющих оборонные заказы. Существуют поставщики оборудования, которые продают Tempest.

Защита от фальшивых дат подписей

Несколько менее очевидным слабым местом PGP является возможность того, что нечестный пользователь создаст электронную подпись на сообщении или сертификате ключа, снабженную фальшивой датой. Если вы пользуетесь PGP от случая к случаю, вы можете пропустить этот раздел и не погружаться в дебри сложных протоколов криптографии с открытыми ключами.

Ничто не помешает нечестному пользователю изменить системную дату и время на своем компьютере и создать сертификат своего открытого ключа или подпись, содержащие другую дату. Он может создать видимость того, что подписал что-то раньше или позже того времени, когда он это действительно сделал, или что его пара ключей была создана раньше или позже. Из этого могут проистекать различные юридические или финансовые выгоды, например, путем создания некоего оправдания, позволяющего ему затем отрицать свою подпись.

Утечка данных в многопользовательских системах

PGP была разработана для использования на однопользовательском персональном компьютере, находящимся под физическим контролем пользователя. Если вы запускаете PGP дома на своем собственном PC, ваши зашифрованные файлы находятся в безопасности, пока никто не ворвался в ваш дом, не украл компьютер и не заставил вас открыть ему свой пароль (или если пароль достаточно прост для того, чтобы его можно было угадать).

PGP не предназначена для защиты исходных открытых данных на скомпрометированной системе. Она также не может предотвратить использование злоумышленником изощренных способов доступа к закрытому ключу во время его использования. Вы должны просто знать о существовании этих опасностей при использовании PGP в многопользовательской среде и соответствующим образом изменить свои ожидания и свое поведение. Возможно ваши обстоятельства таковы, что вы должны рассмотреть возможность использования PGP только на изолированной однопользовательской машине, находящейся под вашим непосредственным физическим контролем.

Анализ активности

Даже если атакующий не сможет прочитать содержимое вашей зашифрованной корреспонденции, он может извлечь, по крайней мере, некоторую полезную информацию, наблюдая, откуда приходят, и куда уходят сообщения, отмечая их размер и время дня, когда они отправляются. Это похоже на то, как если бы злоумышленник смог взглянуть на счет за междугородные телефонные переговоры, чтобы узнать, кому вы звонили, когда, и сколько времени разговаривали, даже если содержание телефонных разговоров остается ему неизвестно. Это называется «анализом активности». Решение этой проблемы требует введения специальных коммуникационных протоколов, разработанных для повышения сопротивления анализу активности в вашей коммуникационной среде. Возможно, при этом потребуется применение ряда криптографических приемов.

Криптоанализ PGP

Возможно, кто-то, обладающий суперкомпьютерными ресурсами (например, правительственная разведывательная служба) предпримет дорогостоящую и чудовищную критоаналитическую атаку. Возможно, ему удастся сломать ваш ключ RSA, используя новые засекреченные знания в области разложения чисел на множители. Но гражданские ученые интенсивно и безуспешно атакуют этот алгоритм с 1978 г.

Возможно, правительство обладает каким-либо секретным методом взлома обычного шифра IDEA, использованного в PGP. Это – самый

страшный кошмар для криптографа. Но абсолютных гарантий безопасности в практическом приложении криптографии не бывает.

И все же, осторожный оптимизм кажется оправданным. Разработчики алгоритма IDEA - одни из самых сильных криптографов Европы. Он подвергался интенсивной проверке на безопасность и экспортировался лучшими гражданскими криптографами мира. В том, что касается устойчивости к дифференциальному криптоанализу, он, вероятно, лучше DES.

Кроме этого, даже если этот алгоритм обладает какими-то до сих пор не замеченными слабыми местами, опасность сильно уменьшается из-за того, что PGP сжимает открытый текст до шифрования. Стоимость необходимых для взлома вычислений, скорее, всего будет больше ценности любого сообщения.

Если обстоятельства, в которых вы находитесь, оправдывают предположения о том, что вы можете подвергнуться столь чудовищной атаке, возможно, вам следует обратиться к консультанту по вопросам безопасности данных для выработке особого подхода, соответствующего вашим чрезвычайным требованиям.

Без надежной криптографической защиты ваших данных, от противника не требуется практически никаких усилий для перехвата ваших сообщений, и он может делать это на повседневной основе, особенно если они передаются по модему или электронной почтой. Если вы используете PGP и соблюдаете разумные меры предосторожности, атакующему потребуется затратить намного больше усилий и средств для нарушения вашей приватности.

Тема 9. Сокрытие передачи и хранения информации. Стеганография

Когда в V веке до н.э. тиран Гистий, находясь под надзором царя Дария в Сузах, должен был послать секретное сообщение своему родственнику в анатолийский город Милет, он побрил наголо своего раба и вытатуировал послание на его голове. Когда волосы снова отросли, раб отпра-

вился в путь. Так Геродот описывает один из первых случаев применения в древнем мире стеганографии – искусства скрытого письма.

Искусство развивалось, превратившись в науку, помогавшую людям на протяжении многих веков скрывать от посторонних глаз сам факт передачи информации. Еще древние римляне писали между строк невидимыми чернилами, в качестве которых использовались фруктовые соки, молоко и некоторых другие натуральные вещества. При нагревании невидимый текст проявлялся. Во время второй мировой войны немцами применялась «микроточка», представлявшая из себя микрофотографию размером с типографскую точку, которая при увеличении давала четкое изображение печатной страницы стандартного размера. Такая точка или несколько точек вклеивались в обыкновенное письмо, и, помимо сложности обнаружения, обладали способностью передавать большие объемы информации, включая чертежи.

Распространение стеганографии во время «холодной войны» и тотальная шпиономания вызвали появление многих цензурных ограничений, которые сегодня могут вызвать лишь улыбку. В США были запрещены к международной почтовой пересылке шахматные партии, инструкции по вязанию и шитью, вырезки из газет, детские рисунки. Запрещалось посылать телеграммы с указанием доставить определенный сорт цветов к определенной дате, а впоследствии американским и английским правительствами были запрещены вообще все международные телеграммы, касающиеся доставки и заказа цветов.

Развитие компьютерной технологии и средств коммуникации сделали бесполезными подобные ограничения. Сегодня каждый может воспользоваться теми преимуществами, которые дает стеганография как в области скрытой передачи информации, так и в области защиты авторских прав.

Стеганографические программные продукты

По сути, компьютерная стеганография базируется на двух принципах. Первый заключается в том, что файлы, содержащие оцифрованное изображение или звук, могут быть до некоторой степени видоизменены без потери функциональности, в отличие от других типов данных, требующих абсо-

лютной точности. Второй принцип состоит в неспособности органов чувств человека различить незначительные изменения в цвете изображения или качестве звука, что особенно легко использовать применительно объекту, несущему избыточную информацию, будь то 16-битный звук, 8-битное или еще лучше 24-битное изображение. Если речь идет об изображении, то изменение значений наименее важных битов, отвечающих за цвет пиксела, не приводит к сколь-нибудь заметному для человека изменению цвета.

Один из лучших и самых распространенных продуктов в этой области для платформы Windows9x/NT – это S-Tools (имеет статус freeware). Программа позволяет прятать любые файлы как в изображениях формата gif и bmp, так и в аудио файлах формата wav. При этом S-Tools – это стеганография и криптография «в одном флаконе», потому что файл, подлежащий сокрытию, еще и шифруется с помощью одного из криптографических алгоритмов с симметричным ключом: DES, тройной DES или IDEA – два последних на сегодня вполне заслуживают доверия. Файл-носитель перетаскивается в окно программы, затем в этот файл перетаскивается файл с данными любого формата, вводится пароль, выбирается алгоритм шифрования, и перед вами результат, который впечатляет! Внешне графический файл остается практически неизменным, меняются лишь кое-где оттенки цвета. Звуковой файл также не претерпевает заметных изменений. Для большей безопасности следует использовать неизвестные широкой публике изображения, изменения в которых не бросятся в глаза с первого взгляда, а также изображения с большим количеством полутонов и оттенков. Использовать картину *Танец* Матисса – идея плохая, т.к. все знают, как она выглядит, и, кроме того, она содержит большие зоны одного цвета. А вот фотография вашего песика вполне подойдет. Рассмотрим примеры:

В первом ряду левое изображение (8.9K) не содержит зашифрованной информации, правое же (11.2K) содержит текст этой главы. Соотношение между размером файла с изображением или звуком и размером текстового файла, который можно спрятать, зависит от конкретного случая.

Иногда размер текстового файла даже превышает размер графического. Впрочем, даже если подозрения у кого-то и возникнут, то их придется оставить при себе: не зная пароля, сам факт использования S-Tools установить и доказать нельзя.

Другая распространенная стеганографическая программа – Steganos for Windows 95 (shareware). Она обладает практически теми же возможностями, что и S-Tools, но использует другой криптографический алгоритм (HWY1), и, кроме того, способна прятать данные не только в файлах формата bmp и wav, но и в обычных текстовых и HTML файлах, причем весьма оригинальным способом - в конце каждой строки добавляется определенное число пробелов. Везете вы на дискете «Опыты» Монтеня, а в них – чертежи секретной макаронной фабрики … А еще Steganos добавляет в меню Отправить (то, которое появляется при правом щелчке мышью на файле) опцию отправки в шредер, что позволяет удалить файл с диска без возможности его последующего восстановления.

Цифровые водяные знаки

Если рассматривать коммерческие приложения стеганографии, то одним из наиболее перспективных направлений ее развития видится digital watermarking, т.е. создание невидимых глазу водяных знаков для защиты авторских прав на графические и аудио файлы. Такие помещенные в файл цифровые водяные знаки могут быть распознаны специальными программами, которые извлекут из файла много полезной информации: когда создан файл, кто владеет авторскими правами, как вступить в контакт с автором.

Сегодня на рынке существует довольно много фирм, предлагающих продукты для создания и детектирования водяных знаков. Один из лидеров – фирма Digimarc, программы которой, если верить предоставленной самой фирмой информации, установили себе более миллиона пользователей. Фирма предлагает сгрузить с сайта PictureMarc, подключаемый модуль для Photoshop и CorelDraw, или отдельно стоящий ReadMarc.

Несмотря на все заверения создателей соответствующих продуктов, цифровые водяные знаки оказались нестойкими. Они могут перенести многое – изменение яркости и контраста, использование спецэффектов,

даже печать и последующее сканирование, но они не могут перенести хитрое воздействие специальных программ-стирателей, таких как UnZign и StirMarK, которые вскоре появились в Интернете, причем очевидно не с целью навредить фирме Digimarc Signum Tehnologies и другим, а для того, чтобы дать пользователям возможность сделать правильный выбор, основываясь на независимой оценке стойкости водяных знаков. А оценка эта на сегодняшний день малоутешительна – водяные знаки всех производителей уничтожаются без заметного ухудшения качества изображения.

Компьютерные вирусы и их классификация

Деструктивные действия вирусов

Что из себя представляет вирус, что он может либо не может выполнять, откуда он взялся, кто его написал и зачем? Чрезвычайно много вопросов возникает при размышлении на тему вирусов.

Понятие вируса

Вирус – это программа или кодовый сегмент, который при получении управления стремиться выполнить скрытое само копирование в различные области выполняемых кодов других программ, максимально защищается от обнаружения и по истечении инкубационного периода заявляет о себе тем или иным действием.

Существует мнение, что вирусы способны физически разрушать компьютер, уничтожать мониторы, сжигать модемы и т.д.

В реальности деструктивные действия вируса, как правило, ограничиваются удалением или порчей информации. Хотя конечно существует возможность испортить жесткий диск или посадить монитор, но действия настолько явны, что вряд ли кто-нибудь даст добро на их продолжение. С другой стороны деструкция не является обязательным выражением вируса, ибо в теорию вируса заложено свойство распространения, а не нанесения вреда. Так существует множество безобидных вирусов, где максимальная неприятность – это какой либо видео или звуковой эффект.

На данный момент можно выдвинуть 3 ситуации, побуждающие к написанию вируса:

4. написание в качестве самоутверждения начинающих программистов;
5. простая тяга к программированию. Вирус в данном случае является просто компьютерной задачкой. Возможность распространения здесь сводится к нулю (по данному пути проходит подавляющее большинство программистов);
6. злоба или месть. В данном случае вирусы являются опасными и оснащаются «боевыми» действиями. Такие вирусы предназначаются обычно конкретным людям, какой либо фирме или в качестве «разминки» всему миру.

Вирусы можно разделить на классы по следующим признакам:

- по среде обитания вируса;
- по способу заражения среды обитания;
- по деструктивным возможностям;
- по особенностям алгоритма вируса.

По среде обитания вирусы можно разделить на сетевые, файловые и загрузочные. Сетевые вирусы распространяются по компьютерной сети, файловые внедряются в выполняемые файлы, загрузочные - в загрузочный сектор диска (boot-сектор) либо в сектор, содержащий системный загрузчик винчестера (Master Boot Record). Существуют сочетания – например, файлово-загрузочные вирусы, заражающие как файлы, так и загрузочные сектора дисков. Такие вирусы, как правило, имеют довольно сложный алгоритм работы, часто применяют оригинальные методы проникновения в систему, используют стелс и полиформик – технологии.

Способы заражения делятся на резидентный и нерезидентный. Резидентный вирус при инфицировании компьютера оставляет в оперативной памяти свою резидентную часть, которая затем перехватывает обращение операционной системы к объектам заражения и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера. Нерезидентные вирусы не заражают память компьютера и являются активными ограниченное время. Некоторые вирусы оставляют в оперативной памяти небольшие резидентные программы, которые не распространяют вирус. Такие вирусы считаются нерезидентными.

По деструктивным возможностям вирусы можно разделить на:

- безвредные, т.е. никак не влияющие на работу компьютера (кроме уменьшения свободной памяти на диске в результате своего распространения);
- неопасные, влияние которых ограничивается уменьшением свободной памяти на диске и графическими, звуковыми и пр. эффектами;
- опасные вирусы, которые могут привести к серьезным сбоям в работе компьютера;
- очень опасные, которые могут привести к потере программ, уничтожить данные, стереть необходимую для работы компьютера информацию, записанную в системных областях памяти, и, как гласит одна из компьютерных легенд, способствовать быстрому износу движущихся частей механизмов – вводить в резонанс и разрушать головки винчестера. Но даже если в алгоритме вируса не найдено ветвей, наносящих ущерб системе, этот вирус нельзя с полной уверенностью назвать безвредным, так как проникновение его в компьютер может вызвать непредсказуемые и порой катастрофические последствия. Ведь вирус, как и всякая программа, имеет ошибки, в результате которых могут быть испорчены как файлы, так и сектора дисков.

По особенностям алгоритма можно выделить следующие группы вирусов:

- компаньен-вирусы (companion) – это вирусы, не изменяющие файлы. Алгоритм работы этих вирусов состоит в том, что они создают для EXE-файлов файлы-спутники, имеющие то же самое имя, но с расширением .COM, например, для файла XCOPY.EXE создается файл XCOPY.COM. Вирус записывается в COM-файл и никак не изменяет EXE-файл. При запуске такого файла первым обнаружит и выполнит COM-файл, т.е. вирус, который затем запустит и EXE-файл;
- вирусы-«черви» (worm) – вирусы, которые распространяются в компьютерной сети и, так же как и компаньон вирусы, не изменяют файлы или сектора на дисках. Они проникают в память компьютера из компьютерной сети, вычисляют сетевые адреса других компьютеров и рассылают по этим адресам свои копии. Такие вирусы иногда создают рабочие файлы на дисках системы, но могут вообще не обращаться к ресурсам компьютера (за исключением оперативной памяти);

- «паразитические» – все вирусы, которые при распространении своих копий обязательно изменяют содержимое дисковых секторов или файлов. В эту группу относятся все вирусы, которые не являются «червями» или «компаньон»;
- «студенческие» – крайне примитивные вирусы, часто нерезидентные и содержащие большое число ошибок;
- «стелс» – вирусы (вирусы-невидимки, stealth), представляющие собой весьма совершенные программы, которые перехватывают обращения OS к пораженным файлам или секторам дисков и «подставляют» вместо себя незараженные участок информации. Кроме этого такие вирусы при обращении к файлам используют достаточно оригинальные алгоритмы, позволяющие «обманывать» резидентные антивирусные мониторы. Один из первых «стелс» – вирусов – вирус «Frodo»;
- «полиморфик» – вирусы (polymorphic) – достаточно труднообнаруживаемые вирусы, не имеющие сигнатур, т.е. не содержащие ни одного постоянного участка кода. В большинстве случаев два образца одного и того же полиморфик-вируса не будут иметь ни одного совпадения. Это достигается шифрованием основного тела вируса и модификациями программы-расшифровщика. Некоторые вирусы (например, вирусы семейства «Eddie», «Murphy») используют часть функций полноценного стелс-вируса.

К особенностям алгоритма вируса можно отнести и скорость его распространения. Скорость распространения файловых вирусов, заражающих файлы только при их запуске на выполнение, будет ниже, чем у вирусов, заражающих файлы и при их открытии, переименовании, изменении атрибутов файла и т.д.

Иногда к вирусам относят «троянских коней». Это не совсем корректно. «Троянский конь» представляет собой программу – шпион, основная цель которой сбор и передача информации получателю. В редких случаях «троянский конь» фальсифицирует либо модифицирует определенную информацию. С точки зрения информационной безопасности такие программы относятся к особо опасным.

Средства защиты от вирусов

Для защиты от вирусов обычно используются:

- общие средства защиты информации, которые полезна также как страховка от физической порчи дисков, неправильно работающих программ или ошибочных действий пользователей;
- профилактические меры позволяющие уменьшить вероятность заражения вирусом;
- специализированные программы для защиты от вирусов.

Среди программ защиты от вирусов можно выделить следующие:

- **программы-детекторы** позволяют обнаруживать файлы, зараженные одним из известных вирусов;
- **программы-доктора**, или **фаги** лечат зараженные программы или диски, выкусывая из зараженных программ тело вируса, т.е. восстанавливая программу в то состояние, в котором она находилась до заражения вируса;
- **программы-ревизоры** сначала запоминают сведения о состоянии программ и системных областей дисков, а затем сравнивают их состояние с исходными. При выявлении несоответствий об этом сообщается пользователю;
- **доктора-ревизоры** это гибриды ревизоров и докторов, т.е. программ, которые не только обнаруживают изменения в файлах и системных областях дисков, но и могут в случае изменения вернуть их в исходное состояние;
- **программы-фильтры** располагаются резидентно в оперативной памяти компьютера и перехватывают те обращения к оперативной системе, которые используются вирусами для размножения и нанесения вреда, и сообщают о них пользователю. Пользователь может разрешить или запретить выполнение соответствующей операции;
- **программы-вакцины** или **иммунизаторы** модифицируют программы и диски таким образом, что это не отражается на работе программ, но тот вирус, против которого производится вакцинация, считает эти программы или диски уже зараженными. На данный момент эти программы крайне не эффективны.

Литература

1. Айгер М. Комбинаторная теория. – М.: Мир, 1982. – 556 с.
2. Акимов О. Е. Дискретная математика. Логика, группы, графы. – М.: Лаборатория базовых знаний, 2001. – 376 с.
3. Ершов Ю. Л. Теория нумераций. – М.: Наука, 1977. – 416 с.
4. Новиков Ф. А. Дискретная математика для программистов. – СПб.: – Москва – Харьков, – Минск. 2002. – 301 с.
5. Developer’s Guide for Delphi 3, Borland Inprise Corporation, 100 Enterprise Way, Scotts Valley, CA 95066-3249
6. Developer’s Guide for Delphi 5, Borland Inprise Corporation, 100 Enterprise Way, Scotts Valley, CA 95066-3249
7. Object Pascal Language Guide, Borland Inprise Corporation, 100 Enterprise Way, Scotts Valley, CA 95066-3249

Содержание

Учебное издание

Швачич Геннадий Григорьевич
Овсянников Александр Васильевич
Кузьменко Вячеслав Витальевич

Основы защиты информации

Конспект лекций

Тем. план 2008, поз.

Подписано к печати _ __ Формат 60x84 1/16. Бумага офсетная. Печать Times.
Уч.-изд. лист. Усл.-печ. лист. Тираж экз. заказ №

Национальная металлургическая академи Украины
49600, Днепропетровск – 5, пр. Гагарина, 4

Редакционно-издательский отдел НМетАУ