В. А. ГРИДНЕВ, Ю. А. ГУБСКОВ, А. С. ДЕРЯБИН, А. В. ЯКОВЛЕВ

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

Тамбов
Издательский центр ФГБОУ ВО «ТГТУ»
2023

Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тамбовский государственный технический университет»**

В. А. ГРИДНЕВ, Ю. А. ГУБСКОВ, А. С. ДЕРЯБИН, А. В. ЯКОВЛЕВ

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

В ТРЕХ ЧАСТЯХ

ЧАСТЬ 2

Утверждено Ученым советом университета в качестве учебного пособия для студентов 5 курса специальности 10.05.03 «Информационная безопасность автоматизированных систем» очной формы обучения и студентов 3 курса направления подготовки бакалавров 09.03.02 «Информационные системы и технологии» очной и заочной форм обучения

Учебное электронное издание

**Тамбов
Издательский центр ФГБОУ ВО «ТГТУ»
2023**

УДК 004.065
ББК 32.973.26
П78

Рецензенты:

Кандидат технических наук, доцент, преподаватель цикла
«Применение комплексов РЭБ и средств комплексного технического контроля»
ФКУ в/ч 61460
К. А. Малыков

Директор Центрально-Черноземного РУНЦ ИБ «ФГБОУ ВО «ТГТУ»
П. А. Щербинин

П78 **Программно-аппаратные средства** защиты информации [Электронный ресурс] : учебное пособие : в 3-х ч. / В. А. Гриднев, Ю. А. Губсков, А. С. Дерябин, А. В. Яковлев. – Тамбов : Издательский центр ФГБОУ ВО «ТГТУ».
ISBN 978-5-8265-2464-0

Ч. 2. – 2023. – 1 электрон. опт. диск (CD-ROM). – Системные требования : ПК не ниже класса Pentium II ; CD-ROM-дисковод ; 1,0 Mb ; RAM ; Windows 95/98/XP ; мышь. – Загл. с экрана.
ISBN 978-5-8265-2609-5

Посвящено вопросам построения и функционирования программно-аппаратных средств защиты информации в автоматизированных системах, являющихся основой комплексной системы информационной безопасности организации. Включает в себя полный теоретический курс учебной дисциплины «Программно-аппаратные средства обеспечения информационной безопасности».

Предназначено для студентов 5 курса специальности 10.05.03 «Информационная безопасность автоматизированных систем» очной формы обучения и студентов 3 курса направления подготовки бакалавров 09.03.02 «Информационные системы и технологии» очной и заочной форм обучения.

УДК 004.065
ББК 32.973.26

ISBN 978-5-8265-2464-0 (общ.)
ISBN 978-5-8265-2609-5 (ч. 2)

ВВЕДЕНИЕ

Учебное пособие состоит из трех частей и включает в себя полный теоретический курс учебной дисциплины «Программно-аппаратные средства защиты информации», преподаваемой в ФГБОУ ВО «Тамбовский государственный технический университет» студентам, обучающимся по специальности 10.05.03 «Информационная безопасность автоматизированных систем».

Данное учебное пособие способствует привитию студентам общепрофессиональной компетенции ОПК-15 (способен осуществлять администрирование и контроль функционирования средств и систем защиты информации автоматизированных систем, инструментальный мониторинг защищенности автоматизированных систем) [1].

Использование данного учебного пособия в учебном процессе должно:

– упростить подготовку студентов к практическим занятиям, лабораторным работам и экзамену по дисциплине «Программно-аппаратные средства защиты информации»;

– самостоятельно изучать материал лекционных занятий, пропущенный студентами по какой-либо причине;

– повысить эффективность изучения дисциплины «Программно-аппаратные средства защиты информации».

Учебное пособие может быть полезно также студентам, обучающимся по направлению подготовки бакалавров 09.03.02 «Информа-

ционные системы и технологии» при изучении дисциплины «Информационная безопасность и защита информации» для формирования общепрофессиональной компетенции ОПК-3 (способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности) [2].

Наличие контрольных вопросов в конце каждой главы учебного пособия позволит студентам самостоятельно проверить свои знания по изученному материалу.

1. СИСТЕМЫ АУТЕНТИФИКАЦИИ ЭЛЕКТРОННЫХ ДАННЫХ

1.1. ИМИТОВСТАВКА

Имитовставкой называется некий отрезок информации, имеющий определенную длину, который получен по определенному правилу, используя открытые данные и секретную ключевую информацию. Этот отрезок добавляется к зашифрованным данным для того, чтобы обеспечить имитозащиту, т.е. защиту от любых модификаций сообщений.

Факт того, что получатель обязан знать ключевую информацию, и в свою очередь эта ключевая информация дает возможность генерации сообщения с таким же значением имитовставки, что и у полученного сообщения, является потенциальной проблемой. В результате этого имитовставка, в основе которой лежит симметричный шифр, не позволяет определить, кто выработал эту имитовставку – получатель или отправитель. Таким образом, имитовставка на основе симметричного шифра позволяет только удостоверить целостность, но не аутентичность сообщения.

Формирование имитовставки может происходить следующими методами:

– перед шифрованием (после дешифрования) сообщения целиком;

– одновременно с шифрованием (дешифрованием) по блокам.

Первые блоки данных, находящихся в открытом доступе, принимают участие в формировании имитовставки и могут включать в себя служебную информацию, они не подлежат шифрованию.

Длина имитовставки (количество бит в ней) l определяется требованием к вероятности навязывания ложных данных P. Эта вероятность определяется выражением

$$P = 2^{-l}.$$

Полученные зашифрованные данные могут дешифроваться. В результате получаются открытые данные, из которых аналогично процессу шифрования формируется имитовставкаl'. Далее этот отрезок информации подвергается сравнению с имитовставкойl, которая была получена вместе с зашифрованными данными. Целостность удостоверяется только в том случае, если эта информация совпала. В противном случае полученные данные считаются неверными.

Важно иметь в виду, что имитовставка является инструментом обеспечения имитозащиты применительно к системам с корреспондентами, которые безусловно доверяют друг другу.

1.2. ХЭШ-ФУНКЦИЯ

При решении многих практических задач, требуется получение короткого «отпечатка» сообщения или электронного документа, имеющего большую длину. В таких случаях облегчить задачу позволяет хэш-функция. Хэш-функция – термин, который принадлежит теории сложности вычислений. В этой теории данный термин описывал некую функцию, целью которой было сжатие строки чисел любой длины в строку фиксированной длины. Такая функция применялась

при формировании методов, нацеленных на быстрый поиск. В лучшем случае это отображение должно соотносить любому сообщению произвольной длины свою уникальную последовательность определенной длины.

В алгоритмах, которые используются для аутентификации электронных, данных требования, которые предъявляются к хэш-функциям, могут быть описаны следующим образом:

1) хэш-функция должна являться отображением из множества сообщений различной длины в множество сообщений определенной длины, при этом результат должен быть явной функцией всех битов первоначального сообщения;

2) функция хэширования должна иметь такое же поведение, как и случайная функция, которая формирует результат как бы случайно, выбирая его из всего множества последовательностей фиксированной длины;

3) должна отсутствовать возможность подбора из всего множества сообщений такой пары сообщений, которые имели бы одинаковые образы;

4) хэш-функция должна быть односторонней.

Третье требование является самым важным применительно к криптографическим функциям хэширования. Оно позволяет избежать возможности создания сообщения, которое отличалось бы от данного, но при этом имело бы идентичное значение хэш-функции, а, следовательно, идентичную электронную подпись.

Криптографические функции хэширования могут разделяться на 2 класса:

– хэш-функции, не имеющие ключевой информации;

– хэш-функции, имеющие ключевую информацию (*MAC*-коды *MessageAuthenticationCode* – код аутентификации сообщения).

Бесключевые хэш-функции также могут разделяться на 2 класса:

– слабые хэш-функции;

– сильные хэш-функции.

Такая функция $H(x)$, которая является односторонней и удовлетворяет следующим требованиям, называется слабой хэш-функцией:

1) аргумент x может быть представлен в виде строки бит различной длины;

2) значение $H(x)$ должно быть представлено в виде строки бит определенной длины;

3) значение функции $H(x)$ должно быть легко вычисляемо;

4) для всякого определенного x не предоставляется возможным вычислить другой $x' \neq x$, для которого $H(x') = H(x)$.

Когда $H(x') = H(x)$, а $x' \neq x$, то такая пара называется коллизией хэш-функции.

Та функция, которая является односторонней, удовлетворяет первым трем свойствам, а также свойству 4′, называется сильной хэш-функцией.

4′) нельзя путем вычислений найти всякую пару $x' \neq x$, при которой $H(x') = H(x)$.

Первые два свойства позволяют сделать вывод, что все множество значений функции хэширования имеет ширину больше, чем у множества значений сообщений, а, значит, в этом случае должны существовать коллизии. Четвертое свойство говорит о том, что при

конкретном значении x почти не предоставляется возможности найти коллизии. Свойство 4′ гласит, что у функции хэширования, которая является сильной, нет возможности вычислить любую коллизию.

Слабая и сильная функции хэширования имеют существенные различия, несмотря на то, что их определения схожи. Анализ приведенных определений позволяет сделать вывод, что любая сильная хэш-функция является и слабой хэш-функцией, но не любая слабая хэш-функция будет сильной.

1.3. ЭЛЕКТРОННАЯ ПОДПИСЬ

В то время как появились электронные документы, возникли проблемы, относящиеся к их аутентификации, верификации их целостности, а также установления авторства. Также как и собственноручная подпись в бумажном документе, рассматриваемая ЭП предназначена для решения трех задач:

– произвести распознавание источника сообщения;

– констатировать целостность сообщения;

– создать такие условия, при которых нельзя отказаться от того, что человек подписал сообщение.

ЭП по форме представления является числом, зависящим от исходного сообщения, а также от ключевой информации, которую знает лишь тот человек, который подписывает документ. При этом легко осуществить проверку ЭП любым человеком, не имевшим доступа к засекреченной ключевой информации.

При возникновении конфликтных ситуаций, к которым может относиться отрицание того факта, что конкретный субъект подписы-

вал конкретный документ, либо попытка подделывания ЭП, третье лицо должно иметь возможность разрешить данную ситуацию.

Термин «подпись» в этом контексте применим в связи с тем, что ЭП во многом схожа с собственноручной подписью на бумажном документе. Обычная подпись, по аналогии с ЭП, решает те же самые три задачи. Несмотря на это, ЭП и собственноручная подпись имеют существенные различия. Основные различия этих подписей сведены в табл.1.1.

1.1. Основные различия между собственноручной подписью и ЭП

Собственноручная подпись	Электронная подпись
Не зависит от подписываемого текста, всегда одна и та же	Зависит от подписываемого текста и секретного ключа, практически всегда различная
Неразрывно связана с подписывающим лицом, однозначно определяется его психофизическими свойствами, не может быть утеряна	Определяется секретным ключом, принадлежащим подписывающему лицу, и поэтому может быть утеряна владельцем
Неотделима от носителя (бумаги), поэтому отдельно подписывается каждый экземпляр документа	Легко отделима от документа, поэтому верна для всех его копий
Не требует для реализации дополнительных механизмов	Требует дополнительных механизмов, реализующих алгоритмы ее вычисления и верификации
Не требует создания поддерживающей инфраструктуры	Требует создания доверенной структуры сертификатов открытых ключей

Для реализации электронной подписи, необходимы следующие алгоритмы:

– алгоритм вычисления (формирования) электронной подписи;

– алгоритм верификации электронной подписи.

Самыми важными требованиями к перечисленным алгоритмам являются следующие:

– невозможность вычисления подписи без использования засекреченной ключевой информации;

– гарантия возможности произвести проверку подписи без использования засекреченной ключевой информации.

Безопасность механизмов подписи определяется сложностью решения следующих задач:

– подделки ЭП, т.е. нахождения значения ЭП под документом таким субъектом, который не является владельцем засекреченной ключевой информации;

– создания подписанного сообщения, т.е. нахождение хотя бы одного текста с истинным значением ЭП;

– подмены текста, т.е. поиск двух разных текстов с идентичными значениями ЭП.

Сложность в принципиальном плане, которая может возникать при внедрении электронной подписи в решение практических задач, является проблемой формирования инфраструктуры открытых ключей. Для того чтобы работал алгоритм верификации ЭП, должна быть использована дополнительная информация из открытого доступа, которая нужна для того, чтобы обеспечить возможность верификации ЭП. Эта информация также должна быть зависима от засекреченной

ключевой информации человека, который является владельцем ЭП. Такая дополнительная информация называется «открытый ключ электронной подписи». Для того чтобы устранить всяческую возможность подделывания ключевой информации в открытом доступе теми лицами, которые хотят представиться легальными владельцами засекреченной ключевой информации, формируется такая специальная поддерживающая инфраструктура, которая включает в свой состав Центры сертификации открытых ключей или Удостоверяющие центры. Такая структура предоставляет возможность своевременно устанавливать достоверность того, что ключевая информация в открытом доступе принадлежит заявленному владельцу, а также позволяет распознать попытки подмены открытого ключа.

С технической стороны формирование центров сертификации не является сложным процессом. В основном такие ЦС строятся также, как и ЦС, которые участвуют в системах асимметричного шифрования (с открытыми ключами). Несмотря на это, с юридической стороны здесь имеются проблемы. Если наступает такая ситуация, в которой человек отрицает свое авторство или происходит подделка ЭП, то эти ЦС обязаны нести юридическую ответственность за оригинальность сертификатов, которые они выдают. В частных случаях ЦС обязаны компенсировать утраты в том случае, если алгоритм верификации ЭП устанавливает ее правильность и на основе этого возникают конфликтные ситуации. Из-за таких ситуаций на практике начали производить заключение гражданско-правовых договоров между участниками информационного воздействия с использованием

электронных подписей. В данном договоре должны быть предусмотрены следующие принципиальные аспекты:

– кто понесет ответственность в случае несостоявшихся подписанных сделок;

– кто понесет ответственность в том случае, если будет установлен факт подделки засекреченной ключевой информации (это будет являться следствием того, что система была ненадежной и, как следствие, была взломана);

– ответственность, которую обязан понести уполномоченный по сертификатам в том случае, если открытая ключевая информация будет подделана;

– ответственность, которую обязан понести владелец засекреченной ключевой информации в том случае, если он ее утратил;

– кто обязан нести ответственность в том случае, если реализация системы в ситуации повреждения или разглашения засекреченной ключевой информации будет скомпрометирована;

– порядок разрешения конфликтных ситуаций и так далее.

Так как перечисленные проблемы носят не технический, а юридический характер, то для решения этих проблем необходим договор, который юридически правильно заключен, а также оформлен документально в соответствии со стандартами.

Существует довольно большое количество разных подходов к формированию схем ЭП, которые отличаются по своему принципу. Такие подходы можно разделить на 3 группы:

1) схемы ЭП, в основе которых лежат системы шифрования с открытыми ключами;

2) схемы ЭП, использующие намеренно выработанные алгоритмы, которые направлены на вычисление, а также проверку ЭП;

3) схемы ЭП, в основе которых лежат симметричные системы шифрования.

Схемы ЭП, в основе которых лежат системы шифрования с открытыми ключами, являются наиболее распространенными. В связи с этим рассмотрим их подробнее.

Принцип применения систем шифрования, в основе которых лежат системы шифрования с ключевой информацией, находящейся в открытом доступе, для создания систем ЭП заложен в постановке задачи. Например, пусть существует некая пара преобразований (E, D), где E находится в зависимости от открытой ключевой информации, а D – в зависимости от засекреченной. Для вычисления ЭП S сообщения, обладатель засекреченной ключевой информации имеет возможность использовать второе преобразование D: $S = D(M)$ применительно к сообщению M. В этой ситуации лишь обладатель засекреченной ключевой информации имеет возможность вычислить ЭП, а осуществить проверку равенства $E(S) = M$ имеет возможность каждый. К преобразованиям E и D существует ряд требований. Основными из них являются:

– для любого сообщения M выполняется условие $M = E(D(M))$;

– без обладания засекреченной ключевой информации исключена возможность нахождения значения $D(M)$ для конкретного сообщения M.

Характерной особенностью этого способа формирования ЭП является то, что можно осуществить отказ от транспортирования исходного текста M, поскольку есть возможность восстановить этот текст, зная значение ЭП. По этой причине системы с подобным принципом называют механизмами ЭП с восстановлением сообщения.

Примечательно, что в случае, когда текст при передаче шифруется с использованием асимметричного шифра дополнительно, пара преобразований (E, D), которая участвует в схеме ЭП, должна быть отличной от используемой ЭП для шифрования текстов. В иной ситуации возникает возможность передачи в качестве шифрованных ранее подписанных сообщений. В этом случае рациональнее шифровать те данные, что подписаны, чем делать в инверсном порядке, т.е. подписывать те данные, что зашифрованы. Это рациональнее по той причине, что в первом случае до злоумышленника дойдет лишь шифртекст, а во втором случае и шифртекст и открытый текст.

Ясно, что рассмотренная выше схема, в основе которой лежит пара преобразований (E, D), подходит по требованию отсутствия возможности подделать ее. Но наряду с этим по требованию отсутствия возможности формирования подписанного сообщения она не подходит, так как для всякого значения S любой может найти значение $M = E(S)$, а это приведет к получению подписанного сообщения. Поскольку первое преобразование из пары (E, D) взаимно однозначно, из этого следует, что требование отсутствия возможности подменить сообщение заранее выполняется.

Для того, чтобы обеспечить защиту подписанного сообщения от его воссоздания нарушителем, можно использовать взаимно однозначное отображение R: $M \to \widetilde{M}$. Оно вносит избыточность в пред-

ставление изначального сообщения. К примеру, можно увеличить его длину, после чего вычислительно найти ЭП $S = D(\widetilde{M})$. В таком случае, пытаясь подобрать S и значение $\widetilde{M} = E(S)$, злоумышленник столкнется с сложностью поиска таких значений $\widetilde{M}$, для которых существует прообраз M. Если отображение R установлено таким образом, что количество всевозможных образов $\widetilde{M}$ гораздо уступает количеству всевозможных последовательностей такой же длины, то задача формирования подписанного сообщения будет вызывать затруднения.

Альтернативным подходом к конструированию механизмов подписей, в основе которых лежат системы шифрования, чьи ключи находятся в открытом доступе, заключается в том, чтобы применять бесключевые хэш-функции.

Изначально для конкретного сообщения M вычислительно происходит нахождение функции хэширования$h(M)$, а затем уже значение ЭП $S = D(h(M))$. В данной ситуации зная ЭП, сообщение не подлежит восстановлению. По этой причине ЭП нужно передавать вместе с сообщениями. Такие ЭП получили название *ЭП* с приложением к сообщению или с ЭП дополнением. Такие системы ЭП, которые построены на основе применения функций хэширования без ключевой информации, заведомо удовлетворяют всем тем требованиям, что предъявляются к электронным подписям.

К примеру, не предоставляется возможности сформировать сообщения с известным значением ЭП по той причине, что функция хэширования без ключевой информации представляет собой однонаправленную функцию.

1.4. ПРОГРАММНЫЙ КОМПЛЕКС *VCERT PKI*

В результате коллективной работы компаний ЗАО «МО ПНИЭИ» и ООО «ВАЛИДАТА» появился программный комплекс *VCERT PKI*. *VCERT PKI* называют такую систему управления сертификатами, которая включает в себя множество элементов, а также применяет структуру ключевой информации, находящейся в открытом доступе для того, чтобы обеспечивать тайность информации, контроль цельности и удостоверение авторства электронных документов на основе применения различных процедур криптографии, которые осуществлены согласованно с российскими стандартами и международными рекомендациями.

В общем случае *VCERT PKI* можно разделить на 2 компонента:

– система управления сертификатами –структура ключевой информации в открытом доступе (*PKI*);

– программный интерфейс к криптографическим функциям для *PKI*-приложений.

Система *VCERT PKI* гарантирует защиту информации, в основе которой лежит осуществление структуры ключевой информации в открытом доступе с применением международного стандарта *X*.509, реализована на *Windows NT*, *XP*, *Windows* 95/98.

Программное обеспечение VCERT PKI осуществлено по принципу модулей, оно включает в себя программные комплексы и модули, основными из которых являются:

– *VCA* (*VCERT CertificationAuthority*) – программный комплекс Сертификационный центр (ЦС), нацеленный на создание сертификатов ключевой информации в открытом доступе, списков сертификатов, которые были аннулированы и их бумажных копий, а также

сохранность эталонной базы сертификатов и списков аннулированных сертификатов на основе той информации, которую предоставляет Центр регистрации.

– *VRA* (*VCERT RegistrationAuthority*) – программный комплекс (Центр регистрации (ЦР), нацеленный на регистрацию пользователей и обеспечения взаимодействия пользователя с Центром сертификации);

– *VCS* (*VCERT CertificatesStore*) – программный комплекс Справочник сертификатов, который обеспечивает администрирование справочника сертификатов, создание служебных сообщений на рабочем месте пользователя, а также генерацию засекреченных и открытых ключей на рабочем месте пользователя и запись их на ключевые информационные носители;

– *VCrypt* – программный модуль осуществление криптографических функций и генерации ключевой информации.

Длиной открытой и засекреченной ключевой информации ЭП являются 256 или 512 бит (или 1024 бита). Засекреченная ключевая информация и ЭП могут храниться на ключевых информационных носителях – дискетах 3.5″, носителях *Touch-Memory* или смарт-картах.

Система *VCERT PKI* позволяет обеспечить:

– генерацию и подтверждение подписей под файлом или областью памяти;

– конфиденциальность и контроль целостности информации посредством ее шифрования и имитозащиты в соответствии с ГОСТ 28147–89;

– регистрацию электронных запросов пользователей на сертификаты ключевой информации ЭП в открытом доступе;

– формирование электронных сертификатов ключевой информации в открытом доступе ЭП пользователей.

Клиентское программное обеспечение *VCERT PKI* позволяет пользователям на своих рабочих местах формировать запросы на сертификаты ключевой информации в открытом доступе, генерировать засекреченную ключевую информацию и ключевую информацию в открытом доступе ЭП и шифрования, а также получать сообщения о компрометации засекреченной ключевой информации и информацию из базы сертификатов.

Инструментарий разработчика дает возможность встраивать в прикладное программное обеспечение криптографические функции генерации/верификации ЭП, а также шифрования/дешифрования информации.

Контрольные вопросы

1. Что такое имитовставка?
2. Какие требования предъявляются к хэш-функциям?
3. Для каких целей применяются хэш-функции?
4. В чем заключается отличие в применении ключевых и бесключевых хэш-функций?
5. Какие задачи решаются с помощью электронной подписи?
6. В чем сходства и отличия электронной подписи и собственноручной?
7. Какие существуют принципиальные подходы к построению схем электронной подписи?
8. Каковы назначения и основные характеристики программного комплекса *VCERT PKI*?

2. СРЕДСТВА УПРАВЛЕНИЯ КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ

2.1. КЛАССИФИКАЦИЯ КРИПТОГРАФИЧЕСКИХ КЛЮЧЕЙ

Криптографические ключи можно разделить на основные группы. По их характеристикам выделяют следующие группы ключей:

– по применению;

– по предназначению;

– по способу организации засекреченной связи или электронного взаимодействия;

– по типу носителя;

– по сроку действия.

По применению различают рабочие, аварийные и резервные криптографические ключи. Рабочие криптографические ключи в свою очередь делятся на действующие и очередные. Если же рабочие ключи были скомпрометированы, то вместо них используются аварийные. Резервные криптографические ключи предназначены для формирования новых сетей взаимодействия и засекреченных направлений.

По предназначению ключи можно делить на ключи шифрования и ключи электронной подписи.

По способу организации засекреченной связи или электронного взаимодействия криптографические ключи делят на сетевые (количество работающих корреспондентов на одинаковых ключах более

двух) и направленческие (количество работающих корреспондентов на одинаковых ключах равняется двум).

По типу носителя бывают электронные ключи, таблицы чисел, перфокарты, перфоленты.

По сроку действия криптографические ключи, в свою очередь, подразделяют на сеансовые, долговременные и разовые.

2.2. ГЕНЕРАЦИЯ КРИПТОГРАФИЧЕСКИХ КЛЮЧЕЙ

Лица, которые обладают доступом к носителям важных данных, несут персональную ответственность за них. Перечень лиц, получающих допуск к хранящим данную информацию дискетам, оформляет начальник подразделения информационной безопасности, а затем данный список закрепляется в распоряжении по организации.

Начальник организации, в свою очередь, приказом устанавливает несущих обязанность за формирование электронных взаимодействий с другими организациями. Несущими ответственность являются лица, отвечающие за подразделения. Также начальник организации предоставляет ответственным сотрудникам право постановки электронной подписи посылаемых бумаг.

Руководитель подразделения информационной безопасности определяет работников, которые в дальнейшем обладают ответственностью за организацию на рабочих станциях уполномоченных сотрудников необходимых методов, надлежащих для обеспечения средств криптографической защиты информации (СКЗИ) и электронных взаимодействий.

Сотрудникам подразделения информационной безопасности необходимо осуществить подготовку ответственных сотрудников подразделений, которые непосредственно участвуют в электронных согласованностях с другими предприятиями, а также в работе со средствами предоставления электронного документооборота и СКЗИ.

Реализацией контроля применения носителей ключевой информации (НКИ) и электронных взаимодействий занимаются сотрудники подразделения информационной безопасности и сферы внутреннего контроля.

Центр управления ключевыми системами (ЦУКС) занимается производством персональных ключевых носителей. При обслуживании ЦУКС какой-либо другой компанией, ключевые носители приобретаются назначенным приказом руководителем, уполномоченным пользователем СКЗИ либо сотрудником подразделения информационной безопасности.

Если же ключевые дискеты непосредственно производятся в самой организации, то в таком случае необходимо исполнение на базе заявки, которая в свою очередь, подписывается руководителем подразделения пользователя НКИ.

«Автоматизированное рабочее место генерации ключей (АРМ ГК)» – это программное обеспечение, которое помогает создать уникальную ключевую информацию, а затем сохраняет ее на дискету. Также АРМ ГК реализует функции, которые регулируются технологией создания ключей электронной подписи, ответственными сотрудниками при самом пользователе НКИ, указывается в «Журнале учета НКИ» и выдается ему под роспись. Оборудование АРМ ГК

обязано давать гарантию на внесение уникальной секретной ключевой информации исполнителя, которая производится только на его личный носитель.

Создание рабочей копии ключевой дискеты предотвращает потерю информации, если подлинник ключевой дискеты оказался неисправным. Создание копии происходит строго на АРМ ГК, так как АРМ ГК позволяет предотвратить создание копии НКИ на каких-либо промежуточных носителях.

НКИ должны иметь следующую информацию на этикетке:

– регистрационный номер (согласно «Журналу учета НКИ»);

– дата производства;

– вид ключевой информации (например, оригинал или копия);

– ФИО и подпись владельца-исполнителя;

– подпись ответственного сотрудника подразделения обеспечения информационной безопасности, производившего НКИ.

2.3. ПОРЯДОК ИСПОЛЬЗОВАНИЯ НОСИТЕЛЕЙ КЛЮЧЕВОЙ ИНФОРМАЦИИ

Всем сотрудникам (исполнителям), которым согласно с их функциональными обязанностями предоставляется право установления на электронный документ электронной подписи, выдается индивидуальный носитель ключевой информации (в частности, дискета), в содержании которого находится уникальная ключевая информация (собственно сам секретный ключ электронной подписи), относящийся к информации ограниченного распространения.

Работник должен содержать личные ключевые дискеты (рабочие копии) в специально предназначенном для этого опечатанном собственной печатью пенале.

В отделе учет и хранение личных дискет сотрудников возлагается на ответственного за информационную безопасность. Если же он отсутствует, то данная задача исполняется руководителем отдела или же лично исполнителем, если у него в наличии имеется сейф или металлический шкаф. Дискеты с ключевой информацией нужно хранить в сейфе у ответственного за информационную безопасность подразделения или же у работника в индивидуальных пеналах, которые опечатаны личными печатями исполнителей. Пеналы извлекают лишь на момент выдачи сотрудникам рабочих копий ключевых дискет. Подлинные ключевые дискеты исполнителей, хранящиеся в отделе информационной безопасности в опечатанном пенале, используются лишь для того, чтобы воссоздать рабочую копию ключевой дискеты при ее повреждении. Наличие подлинных ключевых дискет в пеналах контролируется назначенным сотрудником подразделения информационной безопасности каждый раз, когда происходит вскрытие и опечатывание пенала.

Контроль безопасности технологии обработки электронных документов, в том числе действий пользователей НКИ выполняется лицами, которые отвечают за информационную безопасность в рамках собственных полномочий и сотрудниками подразделения информационной безопасности.

Открытые ключи электронной подписи исполнителей заверяются специалистами центра управления ключевыми системами (ЦУКС)

по поставленным на них электронным подписям в справочнике открытых ключей.

2.4. ПРАВА И ОБЯЗАННОСТИ ПОЛЬЗОВАТЕЛЯ НКИ

Пользователь НКИ имеет право:

– обращаться за консультацией к лицу, ответственному за информационную безопасность своего отдела по вопросам организации информационной безопасности технологического процесса;

– информировать ответственного за информационную безопасность своего подразделения и своего непосредственного руководителя о несоответствии условий осуществления технологического процесса предъявляемым требованиям;

– выдвижение предложений по улучшению безопасности личного участка работы.

Пользователь НКИ несет персональную ответственность за сохранность личной ключевой информации и за ее правильное применение. Помимо этого, он несет ответственность за содержание документов, заверенных его электронной подписью.

Пользователь НКИ обязан:

1) лично присутствовать в момент создания личной ключевой информации (в случае, если ключи создаются в подразделении информационной безопасности организации) и удостовериться в том, что содержание ключевых дискет не подверглось компрометации;

2) приобрести рабочую копию ключевой дискеты под роспись в «Журнале учета НКИ» и удостовериться в правильной маркировке, наличии должной защиты от записи, зарегистрировать данную рабо-

чую копию ключевой дискеты у ответственного за информационную безопасность своего подразделения, уложить ее в пенал и опечатать его личной печатью (далее опечатанный пенал передается на хранение отвечающему за информационную безопасность структурного подразделения или храниться в личном сейфе пользователя НКИ):

3) использовать только свою копию ключевой дискеты;

4) при хранении НКИ у лица, ответственного за информационную безопасность подразделения, пользователю НКИ необходимо в начале рабочего дня получать собственную ключевую дискету, а в конце рабочего дня сдавать ее ответственному за информационную безопасность подразделения (при первоначальном вскрытии пенала каждая сторона должна убедиться в целостности и подлинности пенала до момента вскрытия, в случае, если целостность и(или) подлинность были нарушены, то ключ будет считаться скомпрометированным);

5) при хранении НКИ в личном сейфе работника, пользователю НКИ необходимо ее брать оттуда по мере необходимости и при вскрытии убеждаться в целостности и подлинности печати, если целостность и(или) подлинность были нарушены, то ключ будет считаться скомпрометированным;

6) после окончания рабочего дня ключевую дискету убирают в пенал, который в последующем необходимо опечатать и убрать в сейф;

7) рабочая копия НКИ, которая по каким-либо причинам стала неисправной, предоставляется его исполнителем ответственному сотруднику подразделения информационной безопасности, он обязан в присутствии пользователя НКИ или ответственного за информацион-

ную безопасность подразделения сделать новую копию ключевой дискеты с существующего подлинника, а затем выдать ее исполнителю НКИ вместо испорченной ключевой дискеты;

8) испорченную рабочую копию ключевой дискеты нужно обязательно уничтожать, как правило, в присутствии пользователя НКИ, выполненные действия регистрируются в «Журнале учета НКИ».

Пользователю НКИ запрещается:

– оставлять личную ключевую дискету без присмотра;

– передавать личную ключевую дискету кому-либо (исключение составляет только лицо, ответственное за информационную безопасность, которому дискета предоставляется в опечатанном пенале);

– создавать неучтенные копии ключевой дискеты, открывать или копировать с нее файлы на иные носители информации (в частности, жесткий диск ПЭВМ);

– вносить изменения в какие-либо файлы и(или) удалять защиту от записи на ключевой дискете;

– применять ключевую дискету при неисправностях дисковода и(или) ПЭВМ;

– подписывать личным персональным секретным ключом электронной подписи всевозможные электронные сообщения и документы, исключая регулируемые технологическим процессом;

– сообщать кому-либо вне нахождения рабочего места о том, что он является обладателем секретного ключа электронной подписи для данного технологического процесса.

2.5. ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРИ РАБОТЕ С НКИ

Порядок размещения, специального оборудования, охраны и режима в помещениях, в которых находятся средства криптографической защиты и носители ключевой информации:

– средства криптографической защиты для обслуживания носителей ключевой информации размещаются в помещениях серверной и подразделения информационной безопасности;

– размещение, специальное оборудование и режим в помещениях, в которых размещены средства криптографической защиты и носители ключевой информации, обеспечивают безопасность информации, средств криптографической защиты и ключевой информации, сведение к минимуму возможности неконтролируемого доступа к средствам криптографической защиты, просмотра процедур работы со средствами криптографической защиты посторонними лицами;

– порядок допуска в помещения определяется внутренней инструкцией, которая разработана с учетом специфики и условий функционирования организации;

– окна помещений оборудованы металлическими решетками и охранной сигнализацией, препятствующими несанкционированному доступу в помещения. В этих помещениях прочные входные двери, на которые установлены надежные замки;

– для хранения ключевых дискет, нормативной и эксплуатационной документации, инсталляционных дискет помещения должны быть обеспечены сейфами;

– установленный порядок охраны помещений предусматривает периодический контроль технического состояния средств охранной и пожарной сигнализации и соблюдения режима охраны;

– размещение и установка средств криптографической защиты осуществляется в соответствии с требованиями документации на средства криптографической защиты;

– системные блоки ЭВМ с установленными средствами криптографической защиты должны быть оборудованы средствами контроля их вскрытия.

Порядок обеспечения безопасности хранения НКИ:

– все ключи шифрования, ключи электронной подписи и инсталляционные дискеты берутся на поэкземплярный учет в выделенных для этих целей «Журнале учета НКИ» и «Журнале учета СКЗИ»;

– учет и хранение носителей ключей шифрования и инсталляционных дискет, непосредственная работа с ними поручается сотрудникам подразделения информационной безопасности или ответственным сотрудникам, назначенных приказом по организации, на этих сотрудников возлагается персональная ответственность за сохранность ключей шифрования;

– учет изготовленных для пользователей ключей шифрования, регистрация их выдачи для работы, возврата от пользователей и уничтожение ведется сотрудником подразделения информационной безопасности;

– хранение ключей шифрования, ключей электронной подписи, инсталляционных дискет допускается в одном хранилище с другими документами при условиях, исключающих их непреднамеренное

уничтожение или иное, непредусмотренное правилами пользования систем криптозащиты, применение;

– наряду с этим должна быть предусмотрена возможность безопасного раздельного хранения рабочих и резервных ключей, предназначенных для использования в случае компрометации рабочих ключей в соответствии с правилами пользования средств криптографической защиты;

– действующий закрытый ключ электронной подписи, записанный на магнитный носитель (дискету), должен хранится в личном, опечатываемом сейфе (контейнере) ответственного лица, возможность копирования и несанкционированного использования ключа подписи посторонним лицом должна быть исключена;

– резервный ключ электронной подписи должен хранится так же, как и действующий, но обязательно в отдельном опечатанном контейнере.

Минимальные квалификационные требования к сотрудникам, осуществляющим эксплуатацию и установку (инсталляцию) средств криптографической защиты и носителей ключевой информации:

– к работе со средствами криптографической защиты и носителям ключевой информации допускаются только сотрудники, знающие правила их эксплуатации, владеющие практическими навыками работы на ПЭВМ, изучившие правила пользования и эксплуатационную документацию по средствам криптографической защиты;

– сотрудник должен иметь представление о возможных угрозах информации при её обработке, передаче, хранении, методах и средствах защиты информации.

Уничтожение ключевой информации и дискет:

– уничтожение ключевой информации с дискет должно производиться путем двойного безусловного переформатирования дискеты командой «*FORMAT A:/U*»;

– уничтожение ключевой дискеты, пришедшей в негодность должно производиться путем расплавления на огне (сожжения) или измельчения гибкого магнитного диска, извлеченного из корпуса.

2.6. ПОРЯДОК ДЕЙСТВИЙ ПРИ КОМПРОМЕТАЦИИ КРИПТОГРАФИЧЕСКИХ КЛЮЧЕЙ

К событиям, связанным с компрометацией ключевой информации должны быть отнесены следующие события:

1) потеря НКИ;

2) потеря НКИ с последующей находкой;

3) увольнение работников, имевших доступ к ключевой информации;

4) возникновение подозрений на утечку информации или её искажения в системе связи;

5) нерасшифровывание у абонентов исходящих либо входящих сообщений;

6) нарушение целостности печати на пенале с НКИ или сейфе.

Первые три события должны трактоваться как безусловная компрометация действующих ключей. Три следующих события требуют специального рассмотрения в каждом конкретном случае.

При компрометации ключей электронной подписи и шифрования участника обмена электронными документами предусмотрены следующие мероприятия:

– участник обмена электронными документами немедленно прекращает передачу информации с использованием скомпрометированных ключей электронной подписи или шифрования;

– сообщает о факте компрометации в подразделение информационной безопасности;

– сотрудник подразделения информационной безопасности на основании извещения участника обмена электронными документами исключает из электронной базы открытых ключей скомпрометированный ключ электронной подписи или исключает криптографический номер участника обмена электронными документами из списка абонентов;

– в случае крайней необходимости участник обмена электронными документами после компрометации может продолжить работу на резервных ключах, о чем делается запись в «Журнале учета НКИ»;

– участник обмена электронными документами подает заявку на изготовление нового ключа и получает новые ключи электронной подписи и шифрования с регистрацией в «Журнале учета НКИ»;

– производится обмен тестовыми сообщениями на новых ключах электронной подписи и шифрования.

Контрольные вопросы

1. Организация работы с носителями ключевой информации в финансовых учреждениях.

2. Процедуры производства и учета носителей ключевой информации.

3. Процедуры использования ключевых носителей информации.

4. Права на использование носителей ключевой информации.

5. Что запрещено пользователям ключевых информационных носителей?

6. Порядок размещения специального оборудования, охраны и режимов в объектах с криптографическими средствами защиты и носителями ключевой информации.

7. Процедуры безопасного хранения ключевых дискет.

8. Требования к работникам, участвующим в эксплуатации и установке (установке) средств криптографической защиты и носителей ключевой информации.

9. Порядок действий при компрометации носителей ключевой информации.

3. СРЕДСТВА ЗАЩИТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОТ НЕСАНКЦИОНИРОВАННОГО ИСПОЛЬЗОВАНИЯ

3.1. ОБЩИЕ СВЕДЕНИЯ

Защита программного обеспечения (ПО) – комплекс мер, направленных на защиту программного обеспечения от несанкционированного приобретения, использования, распространения, модифицирования, изучения и воссоздания аналогов.

Защита от несанкционированного использования программ – система мер, направленных на противодействие нелегальному использованию программного обеспечения. При защите могут применяться организационные, юридические, программные и программно-аппаратные средства.

Защита от копирования к программному обеспечению применяется редко, в связи с необходимостью его распространения и установки на компьютеры пользователей. Однако, от копирования может защищаться лицензия на приложение (при распространении на физическом носителе) или его отдельные алгоритмы.

У разработчиков ПО может не хватать времени, финансов или квалификации на реализацию собственной (встроенной) стойкой защиты. Они вынуждены пользоваться сторонними автоматическими средствами защиты ПО. Эти средства пристыковывают к скомпилированной программе защитный модуль. Преимущество такой защиты в том, что её можно установить на любую программу (даже без до-

ступа к исходному коду программы). Недостаток в самом подходе – «шаблонности» метода. Стандартные (пристыковочные) защиты имеют большую вероятность быть взломанными, так как устанавливаются на несколько программ и тем самым обеспечивают спрос на рынке взлома.

Тем не менее, автоматические средства затрудняют взлом программы. Их иногда целесообразно использовать либо когда защиты нет вообще, либо в совокупности с реализацией собственной уникальной защиты.

3.2. МЕТОДЫ ВЗЛОМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Прежде чем говорить о защите ПО от несанкционированного использования, рассмотрим методы, применяемые злоумышленниками для несанкционированного использования проприетарного (являющегося частной собственностью авторов или правообладателей) программного обеспечения.

Практически любой взлом сводится к использованию одного из следующих способов:

– ввод серийного номера или регистрационного кода – взлом программы посредством введения правильного регистрационного ключа, полученного нелегальным способом (ключ может генерироваться на основе какой-либо информации: имени владельца ПО, характеристик аппаратной части компьютера, и тому подобной, либо иметь фиксированное значение, для генерации регистрационного ключа используется тот же алгоритм, что и в программе);

Примечание 1. Регистрационный код может распространяться в ключевом файле (файле лицензии, англ. *keyfile*), который обычно помещается в каталог с установленной программой.

Примечание 2. Для массового взлома, зачастую, создаётся (и в дальнейшем используется) генератор ключей – программа для генерации регистрационных ключей. Данный вид взлома наиболее востребован (особенно, когда программа часто обновляется или регистрационный ключ генерируется на основе какой-то информации и поэтому наиболее ценится. Как правило, требует бóльшей квалификации взломщика по сравнению с другими видами взлома, но не всегда.

– использование загрузчика (жарг. лоадер, англ. loader) – способ обходить некоторые виды защиты ПО, заключающиеся в использовании внешних систем защиты (состоит в изменении определённых фрагментов программы в оперативной памяти сразу после её загрузки в эту память, но перед её запуском, то есть перед выполнением кода в точке входа);

– применение бинарного патча – способ, похожий на «загрузчик», но модификация производится статически в файлах программы (как правило, это один из самых простых и быстрых способов взлома ПО);

– использование взломанной версии файла(ов) – способ заключается в подмене оригинальных файлов программы файлами, которые уже взломаны;

– использование эмулятора ключа (англ. *key emulator*) – способ используется для обмана защит, построенных на использовании элек-

тронного ключа, как правило, подключаемого к *LPT* или *USB* порту компьютера (заключается в снятии дампа внутренней памяти ключа: файл с содержимым этой памяти подаётся на вход специальной программы (эмулятора), которая подключает свой драйвер-фильтр в стек драйверов и обманывает защищённую программу, эмулируя работу с аппаратным ключом.

В случаях наличия в программе обращений к ключу для аппаратного шифрования участка памяти этот метод используется в связке с методом «бинарный патч». Современные аппаратные ключи настолько сложны, что, при грамотном их применении, возможно создать защиту, фактически не поддающуюся взлому.

При взломе сложных защит, а также при необходимости достичь максимального эффекта, применяется комбинация вышеперечисленных способов. В редких случаях, это происходит при недостаточной квалифицированности взломщика.

Этот список не является исчерпывающим, а лишь обозначает наиболее встречаемые способы взлома.

Вид взлома, обычно, обусловлен видом защиты. Для некоторых защит возможно использовать различные виды взлома, для других – способ может быть единственным.

Как правило, в основе работы программы-крэкера лежит исследование ассемблерного кода, полученного из машинных инструкций с помощью специально предназначенной для этого программы-дизассемблера. В зависимости от выбранного способа взлома, результат исследования может использоваться, например, для построения генератора ключей или для внесения необходимых изменений в ис-

полняемый файл. Последний способ в большинстве случаев наиболее лёгкий, так как не требует изучения алгоритма проверки правильности ключа: зачастую взлом сводится к поиску проверки нескольких условий (например, «введённое число равно эталонному числу?») и замене такого условия на безусловный переход (*goto, jmp*), или, реже, на противоположное (то есть, для данного примера, на «введённое число не равно эталонному числу?»).

Кроме того, внесение изменений в исполняемый файл (патч) может производиться с целью отключения нежелательных действий со стороны программы (например, напоминание о необходимости регистрации), сокращения функциональности программы. В этих случаях, часто, соответствующие команды процессору заменяются на байты со значением *90h* (в шестнадцатеричной системе счисления), что соответствует ассемблерной команде *nop* (*No Operation*), то есть «пустой команде», не выполняющей никаких действий. Если таких команд много, то применяется безусловный переход (перепрыгивание ненужного кода). Возможно также расширение возможностей программы написанием дополнительного кода, но, как правило, это слишком трудоёмкий процесс, не оправдывающий временных затрат.

Между тем, патч возможен, как правило, в том случае, когда исполняемый файл программы не защищён специальными «пакерами» и «протекторами» – программами, скрывающими реальный код исполняемого файла. Для последнего типа программ зачастую используется самая интеллектуальная часть обратной разработки (англ. *reverse engineering*) – исследование кода программы при помощи от-

ладчика и создание генератора ключей, но возможны и другие решения, например, создание загрузчика.

3.3. СПОСОБЫ ЗАЩИТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОТ НЕСАНКЦИОНИРОВАННОГО ИСПОЛЬЗОВАНИЯ

3.3.1. Локальная программная защита

Эта защита сводится к требованию ввода серийного номера (ключа) при установке/запуске программы. История этого метода началась тогда, когда приложения распространялись только на физических носителях (к примеру, компакт-дисках). На коробке с диском был напечатан серийный номер, подходящий только к данной копии программы.

С распространением сетей очевидным недостатком стала проблема распространения образов дисков и серийных номеров по сети. Поэтому в настоящий момент метод используется только в совокупности одним или более других методов (например, мер организационной защиты).

3.3.2. Сетевая защита

Сканирование сети исключает одновременный запуск двух программ с одним регистрационным ключом на двух компьютерах в пределах одной локальной сети.

Недостаток в том, что брандмауэр можно настроить так, чтобы он не пропускал пакеты, принадлежащие защищённой программе. Правда, настройка брандмауэра требует некоторых пользовательских навыков. Кроме того, приложения могут взаимодействовать по сети

(к примеру, при организации сетевой игры). В этом случае брандмауэр должен пропускать такой трафик.

Если программа работает с каким-то централизованным сервером и без него бесполезна (например, серверы онлайн-игр, серверы обновлений антивирусов), то она может передавать серверу свой серийный номер; если номер неправильный, сервер отказывает в услуге. Недостаток в том, что, существует возможность создать сервер, который не делает такой проверки. Например, существовал сервер *battle.da*, который по функциям был аналогичен *Battle.net*, но пускал пользователей неавторизованных копий игр. Сейчас этот сервер закрыт, но существует немалое количество *PvPGN*-серверов, которые также не проверяют регистрационные номера.

3.3.3. Защита при помощи компакт-дисков

Программа может требовать оригинальный компакт-диск. В частности, такой способ применяется в играх. Стойкость таких защит невелика, ввиду широкого набора инструментов снятия образов компакт-дисков.

Как правило, этот способ защиты применяется для защиты программ, записанных на этом же компакт-диске, являющимся одновременно ключевым.

Основные способы защиты от несанкционированного использования:

– запись информации в неиспользуемых секторах;

– проверка расположения и содержимого «сбойных» секторов;

– проверка скорости чтения отдельных секторов.

Первые два метода практически бесполезны из-за возможности снятия полного образа с диска с использованием соответствующего прикладного ПО. Третий метод считается более надежным (используется, в частности, в защите *StarForce*). Но существуют программы, которые могут эмулировать диски с учётом геометрии расположения данных, тем самым обходя и эту защиту. В *StarForce*, в числе прочих проверок, также выполняется проверка возможности записи на вставленный диск. Если она возможна, то диск считается не лицензионным. Однако, если образ будет записан на диск *CD-R*, то указанная проверка пройдет. Возможно также скрыть тип диска, чтобы *CD-R* или *CD-RW* был виден как обычный *CD-ROM*. Однако, в модуль защиты может быть включена проверка на наличие эмуляции.

В настоящее время наибольшую известность в мире имеют системы защиты от копирования *SecuROM, StarForce, SafeDisc, CD-RX* и *Tages*.

Для многих программ указанный метод защиты недоступен ввиду отличного способа распространения (например, *shareware*-программы).

3.3.4. Защита при помощи электронных ключей

Электронный ключ (донгл), вставленный в порт компьютера (с интерфейсом *COM*, *USB*, *LPT*) содержит в себе данные, называемые лицензией. Лицензионные данные записываются в него разработчиком данного защищенного программного обеспечения, а сама защита ПО основывается на известности для разработчика полного алгоритма работы ключа.

Возможные типы таких ключевых данных:

– информация для чтения или записи (в данный момент практически не используется, потому как после считывания ключ можно сэмулировать);

– ключи аппаратных криптографических алгоритмов (применяются наиболее часто);

– алгоритмы, которые созданы разработчиком программы (этот метод стал доступен сравнительно недавно, преимущественно благодаря появлению электронных ключей с микропроцессором, который имеет возможность исполнять произвольный код).

Достоинства защиты программ с применением электронных ключей:

– ключ можно использовать на любом компьютере, на котором необходимо запустить программу;

– ключ не требует наличия дисковода и тем самым не занимает его;

– электронный ключ может выполнять в том числе и преобразования криптографического типа;

– современные ключи имеют возможность исполнять любой код, помещаемый в них разработчиком системы защиты (пример – *Guardant Code*, *Senselock*).

Стойкость защиты программы сводится к тому, что ключевая информация защиты (загружаемый код, криптографические ключи) не покидает ключа во время работы с ним.

К основным недостаткам защиты программ с применением электронных ключей относятся прежде всего:

– низкая стоимость (15 – 30 долларов за единицу);

– необходимость транспортировки ключа крайнему пользователю.

Ранее к минусам их использования можно было также отнести низкое быстродействие ключа (в сравнении со значением *CPU* компьютера), но современные электронные ключи способны достигать производительности в 1,25 *DMIPS* (*Guardant*, *HASP*), а способ защиты с их помощью уже не предполагает непрерывного обмена с ключом.

Имевшие место ранее проблемы с установкой ключа на некоторые аппаратные платформы в данный момент решаются применением сетевых ключей, способных работать с одной или несколькими копиями защищенного приложения, при этом находясь с ним в одной локальной сети, или при помощи аппаратных или программных средств «проброса» по сети *USB*-устройств.

3.3.5. Привязка к параметрам компьютера и активация

Довольно часто применяется привязка проприетарного ПО к информации о пользователе, точнее о серийных номерах компонентов компьютера этого пользователя с последующей активацией установленного ПО (например, как в ОС *Windows*).

В процессе установки программа подсчитывает код активации – контрольное значение, однозначно соответствующее установленным комплектующим компьютера и параметрам установленной ОС. Это значение передается разработчику программы. На его основе разра-

ботчик генерирует ключ активации, подходящий для активации приложения только на указанной машине (копирование установленных исполняемых файлов на другой компьютер приведет к неработоспособности программы).

Достоинство привязки к параметрам компьютера заключается в отсутствии требования специфического аппаратного обеспечения и в том, что программу можно будет спокойно распространять посредством цифровой дистрибуции, т.е. в сети Интернет.

Основной недостаток такой привязки – если пользователь производит модернизацию компьютера (в случае привязки к «железу»), защита отказывает. Авторы многих программ в подобных случаях готовы дать новый регистрационный код. Например, *Microsoft* в *Windows XP* разрешает раз в 120 дней генерировать новый регистрационный код (но в исключительных случаях, позвонив в службу активации, можно получить новый код и после окончания этого срока).

Для привязки обычно используют серийный номер винчестера или номер *BIOS* материнской платы компьютера. Чтобы скрыть информацию о защите от конечного пользователя, ее могут расположить в неразмеченной области винчестера.

Совсем недавно такие виды защиты создавались и внедрялись разработчиками самой программы, но сейчас для этого обращаются к специальным средствам – *SDK*, которые применяются в работе с такими программными ключами, как *HASP SL* от компании Аладдин Р.Д. Все большее распространение также получают услуги, предлагающие в одно и то же время как серверы активации и лицензирования, так и функционал «навесной» защиты.

3.3.6. Защита программ от несанкционированного использования путем переноса их в онлайн

Другим трендом защиты программ является применение подхода *SaaS*, т.е. предоставление функционала этих программ в качестве сервиса. При таком подходе код программы хранится и выполняется на самом сервере, доступном в сети Интернет, а получение доступа к нему происходит по принципу тонкого клиента, т.е. с помощью браузера.

Это одна из немногих существующих ситуаций реализации защиты программы от копирования. Код программы в таком случае исполняется на «доверенной» стороне, откуда у злоумышленника не будет способа его скопировать.

Однако даже здесь возникают трудности, связанные с обеспечением безопасности, например, следующие:

– стойкость защиты такого типа зависит прежде всего от степени защищенности серверов, на которых он выполняется;

– важную роль играет обеспечение конфиденциальности аутентификации пользователей, запросов, а также целостности ресурса и доступности решения в целом.

Кроме перечисленного, важными становятся и вопросы доверия сервису, поскольку фактически ему в «открытом виде» предоставляется как само ПО, так и вся информация, которую оно обрабатывает (в том числе и персональные данные пользователей).

3.3.7. Защита кода от анализа

Злоумышленник может прибегнуть к анализу логики работы программы, чтобы затем ему было проще выделить блок, отвечающий за защиту, и провести его деактивацию. Этот способ применяется тогда, когда не получилось создать копию программы или же способ ее защиты остается неизвестным для злоумышленника.

Программное обеспечение обычно взламывается путем применения либо отладки, либо дизассемблирования.

Определение отладки просто для понимания, это некий режим пошагового выполнения программы, в процессе которого он одновременно управляет всей средой сразу, и сохраняет свою невидимость для защитных механизмов, прикрываясь ложным видом того, что программа работает только с самой системой. Подобного рода механизмы, отвечающие за отладку приложений, довольно распространены, поскольку представляют собой один из немногих, если не единственный существующий способ убедиться, по какой причине программа не отвечает (а следовательно и не работает в принципе), либо почему во время выполнения происходят те или иные ошибки.

Дизассемблирование кажется куда более сложным термином, нежели отладка, но на самом деле выглядит как определенный метод изменения на понятный разработчику язык – ассемблер – представленных в приложении исполняемых модулей. Пользующиеся дизассемблированием в результате получают распечатанный отчет, в котором подробно изложены все те действия, что выполняет программа.

Если говорить о том, какой механизм из двух представленных применяется чаще всего, то здесь нельзя выделить один конкретный – и отладка, и дизассемблирование оказываются полезными в разных случаях. Почему же так происходит? Ответ кроется в том, что использование какого-либо из этих способов с конкретно рассматриваемым приложением может значительно облегчить процесс взлома, тогда как с иными более выгодным окажется совершенно другой.

На данный момент выделяют множество различных отладчиков. Этот длинный список можно начать с тех, которые становятся одной из частей среды разработки ПО, а закончить – эмуляторами, перемещающими программу-цель в созданную для собирания статистических данных аналитическую среду. Во втором случае уровень опасности отладчиков становится весьма высок, ведь приложение будет считать, что работает четко в своей системе, а не в намеренно сымитированной. Примером подобного отладчика может служить *NuMega SoftIce*.

Найти управу на используемые отладчики не так и сложно, поскольку число ее вариантов нисколько им не уступает. Для противодействия отладчикам используются различные средства, такие, как замусоривание кода программы, использующее специальные функции. С помощью средств противодействия становится возможным предотвращение работы отладчика в принципе или такое максимальное ее осложнение. Если вернуться к теме замусоривания, то этот метод считается некой типичной «обманкой». Почему же? А потому как он отвлекает внимание взломщика сложными функциями или вызовами, внесенными в программу и периодически прояв-

ляющимися, а также может представлять собой слишком сложные для восприятия человеком операции. При применении замусоривания обман неизбежно раскроется через какой-то промежуток времени, но злоумышленник уже потеряет его в достатке, а может и вовсе упустит единичную возможность получения незаконного доступа.

Если приложения могут легко считываться отладчиком, при этом выполняясь линейно или разбивая процессы выполнения на несколько параллельных работающих потоков, то этот способ защиты будет более эффективен при данном раскладе. Для случая, когда процессы выполняются параллельно, характерно возрастание сложности анализа программы.

Кроме того, для борьбы с отладчиками используют такой механизм, который мог бы запретить проводить анализ программы только в заданный промежуток времени ее работы. Предполагается, что этот способ должен исключать возможность взлома злоумышленником путем установки неких контрольных отметок, впоследствии доступных для анализирования приложения. Для обхода этого хакеры применяют *Breakpoint* – это специальный вызов, который выполняет передачу всего управления в руки отладчику. Главный «минус» подобного метода заключается в том, что код взламываемого ПО будет прерван, делая доступным шанс внести в него изменения, а значит правильно настроенное приложение, которое способно автоматически проверять себя на наличие в нем контрольных отметок, будет сложнее взломать.

Что еще может представлять хоть какую-то преграду отладчикам? Например, создание такой защиты операционной среды, при ко-

торой невозможно было бы произвести какую-либо модификацию вообще. Не у каждого отладчика есть силы и способности, позволяющие имитировать условия, которые создает оригинальная системная среда, а потому неспособные это делать отсеются, выдав отчет об ошибке, и не смогут переместить программу в иную среду с лишними модификациями.

Модификации можно побороть ответными модификациями. Одними из таких являются, к примеру, вносимые в регистры процессора. Если разработчик будет согласен внести ряд специальных корректировок в регистры, то полезность отладчика для взломщиков значительно снизится, ведь он не сможет распознать и эмулировать определенные изменения в среде. Это происходит лишь потому, что как и любое другое ПО на компьютерах, отладчики всегда включают в свою работу использование операционной системы и процессора.

Шифрование – это, пожалуй, самый действенный и наиболее легкий к применению метод для создания препятствий отладчикам, ведь благодаря ему часть программы или даже она вся хранятся и передаются в зашифрованном виде, после чего расшифровываются только для непосредственного своего выполнения. В таком случае у хакеров попросту не получится получить зашифрованную часть кода, ведь она надежно защищена и дешифруется только тогда, когда приложение передает ему управление. Собственно, это намекает также на использование второго метода, идущего вместе с шифрованием, – дешифрования. Если помимо изначального шифрования кода после приведения в исполнение он будет обратно зашифровываться, то будни злоумышленника уже перестанут быть столь радужными

и ему придется вложить куда больше сил и времени, работая с отладчиком.

Для шифрования и дешифрования можно использовать еще и виртуальные машины, которые дополнительно повторяли бы эти действия, тем самым закрывая функционал программы от злоумышленников совсем и наиболее эффективно защищая код. В таком случае из программного он будет переводиться в машинный, а только потом уже приводиться в исполнение. Конечно, кажется, что достаточно это сделать и о нападках хакеров с отладчиками можно забыть навсегда, но, к сожалению, создание, поддержание и использование виртуальных машин более сложно, чем иные рассмотренные способы, но его можно применять, к примеру, только на критически важных участках кода. Говоря об еще одном недостатке этого метода борьбы, можно выявить снижение производительности приложения, потому как непрерывная работа виртуальной машины затребует значительных временных ресурсов.

Противодействие дамперам и дизассемблерам. По принципу работы дампер схож с дизассемблером, но отличается тем, что транслирует не файл, хранящийся на жестком диске в ассемблерный код, а содержание оперативной памяти компьютера в момент, когда началось корректное исполнение приложения и все защиты пройдены. Хакеру нет необходимости бороться с механизмами противодействия отладке, он ждет окончание проверок на санкционированный запуск приложением, а также проверяет метки на диске и начинает нормальную работу. В это время дампер снимает код без изменений. Но не все защиты могут так запросто раскрыть себя.

Дополнительно также применяются и другие методы, особенно в случаях, когда каждый модуль программы выполнен качественно, а в процессе разработки были взяты личные хитрости создателей ПО. Все методы, о которых говорилось выше, должны иметь место в любой системе.

Несомненно, необходимо использовать проверенные временем алгоритмы шифрования, надежность которых уже была доказана ранее.

Для хранения информации защиты можно использовать системные ресурсы (дополнительно выделяемую память) от *Windows*. В таком случае можно применить совершенно неожиданные решения в казалось бы, ожидаемом для них месте. Это как если бы ключи и парольные фразы хранили не в специально отведенном для этого месте, где взломщик думает их обнаружить, а в том, в которое он не удосужится даже заглянуть.

Можно применять методы стеганографии. При осуществлении функций защиты это выглядит как использование технологий цифрового водяного знака или скрытых цифровых маркеров.

У микропроцессора существуют подробные правила, известные обеим сторонам (и тому, кто защищает, и тому, кто взламывает), поэтому при разработке защиты можно подкорректировать стандартный системный ответ на те операции или инструкции, которые могли бы нестандартным образом обойти попытки взлома.

Определенно стоит исключить любые обращения к тем значениям, функциям и переменным, как-либо относящимся к защитным механизмам, особенно тем, что работают напрямую. Для этого можно

заменить их косвенными методами доступа в специально отведенные для этого обхода области.

Использовать «зеркалирование» – т.е. не ожидаемые хакером действия в ответ на стандартные запросы к системе и программам.

Выше были перечислены только общеизвестные подходы. В будущем, после взлома хакером очередной уникальной защиты какой-либо системы, мы сможем получить новые оригинальные разработки и методы.

3.3.8. Идентификация программных продуктов

Идентификация программы основана на сохранении всех копий программы, в которых скрыты данные управления правами. В случае возникновения конфликта между потенциальными пользователями, незаконно использующими программу, вы можете доказать свое авторство, раскрыв данные об управлении правами, заложенные в спорную копию ПО.

Закон устанавливает понятие информации об управлении правами.

«Информация об управлении правами – любая информация, которая идентифицирует автора, произведение … или информация об условиях использования произведения... и любые шифры или коды, в которых представлена такая информация, когда любой из этих элементов информации приложен к экземпляру произведения...

Устранение или изменение любой электронной информации об управлении правами без разрешения правообладателя является нарушением авторского права [ст. 39, п.5]».

Таким образом, включение информации об управлении правами в случае возникновения спора способствует защите авторских прав.

Информация об управлении правами, которая находится в экземпляре программы, должна быть скрыта. В противном случае она может быть легко изменена или удалена злоумышленником.

Поэтому для предварительной защиты программы автору приходится решать некоторые задачи:

– относительно организации информации об управлении авторскими правами;

– каким способом внести информацию в каждый экземпляр продукта;

– каким методом скрыть информацию об управлении правами.

Программный код, предоставляющий информацию об управлении правами, называется идентификационной меткой или авторской.

На практике разработчики программного обеспечения используют идентификаторы создателей в качестве идентификационных меток. Идентификатор создателя может быть текстовой информацией, содержащей имя разработчика и(или) код создателя, который является уникальным номером. Кроме того, идентификатор может включать адрес, номер телефона или адрес электронной почты. Программы обычно идентифицируются по таким характеристикам, как имя, дата выпуска или номер версии.

То, как реализуется информация об управлении правами, зависит от двух параметров. Во-первых, о назначении, формате и типе ПО. Во-вторых, от возможности скрытия данных в некоторых форматах программ.

Вы можете скрыть информацию об управлении правами и любую другую информацию. Есть два способа.

Первый заключается в сокрытии содержания информации, не зная факта ее существования.

Второй способ заключается в сокрытии факта существования или передачи информации при открытом содержании.

Методы и алгоритмы шифрования могут использоваться для сокрытия информации. Другими словами, чтобы скрыть содержание информации, ее необходимо зашифровать.

Способы сокрытия факта существования информации изучает наука стеганография. Компьютерная стеганография – это изучение способов включения скрытой информации в файлы.

Также обратите внимание на тот факт, что, если криптографические методы используются для сокрытия информации об управлении правами, злоумышленнику нужно только обнулить байты, содержащие идентификатор, и ему не нужно раскрывать ключ шифрования или алгоритм. Поэтому, чтобы надежно скрыть информацию об управлении правами, стоит скрыть факт эксплуатации такого рода защиты или запутать место, содержащее идентификатор. Итак, первоначально нужно использовать стеганографическую технику.

Данные управления правами можно легко закодировать в *HTML* и текстовые файлы. Например, для файлов, содержащих текст, вы можете использовать метод скрытых гарнитур шрифтов. Контуры

символов текста должны создавать невидимые искажения со смысловой нагрузкой. Файлы *HTML* кодируют скрытые сообщения или идентификаторы, добавляя определенное количество пробелов в конце каждой строки.

Чтобы защитить свои права на программу, разработчикам следует предпринять некоторые упреждающие действия, чтобы своевременно предоставить подтверждение авторства. Такие действия могут включать регистрацию программного обеспечения и обслуживание программного обеспечения с использованием данных управления правами. Остановимся на ключевых моментах сопровождения программ информацией об управлении правами разработчика.

Введение и сопровождение программы, содержащей информацию об управлении правами, называется идентификацией программы.

Поскольку программное обеспечение распространяется главным образом путем создания копий исполняемых файлов, а не исходного кода программы, информация об управлении правами может вводиться непосредственно в объектный код программного продукта или может быть привязана к объектному коду. Он должен сопровождаться конкретным механизмом, который может гарантировать взаимосвязь с опознавательными знаками.

Методы компьютерной стеганографии могут использоваться для идентификации программ. Они используются для внесения опознавательных знаков в объектный код, во-первых, из-за возможности изменения объектного кода программы без потери функциональности, а во-вторых, возможностью внести определенное множество байтов, которые будут нести в себе информационную нагрузку.

Рассмотрим методы компьютерной стеганографии, которые можно использовать для введения в программы кодов и чисел, представляющих данные управления правами.

В отдельное множество можно выделить способы, основанные на существовании незанятых участков в объектных программных кодах, содержащихся в исполнимых файлах:

– в файле есть частично или полностью свободные сектора;

– структуры заголовков файлов в формате *Portable Executable*, *New Executable*, *EXE* и содержат в себе зарезервированные поля;

– существует разрыв между сегментами исполняемого кода;

– и др.

Внимательное внесение информации об авторе в неиспользуемые области обеспечивает корректное поведение измененного объектного кода программы и не изменяет размер файла.

При введении идентификационных меток в незанятые участки в объектных программных кодах, следует не допускать удаление идентификационных меток способом, при котором хакер без изучения конкретных мест введения метки может «обнулить» все существующие свободные участки.

Другая группа методов компьютерной стеганографии, основанная на предположении, что объектный код содержит описательную информацию, может сочетать методы модификации объектного кода таким образом, чтобы модификации не приводили к потере правильной функциональности программы. Поэтому формат исполняемых файлов (*EXE*, *NewExecutable*, *Portable Executable*) таков, что изменение значений некоторых полей не влияет на выполнение программы.

Кроме того, объектный код содержит текстовую информацию, изменение которой не влияет на поведение программы.

К третьей группе методов относятся методы, основанные на вирусной технологии, внедряемой в исполняемые файлы. В частности, вы можете добавлять модули (фрагменты кода, наборы команд) в объектный код вашей программы, изменяя при этом соответствующие параметры файла. Например, после добавления информации об управлении правами в конец файла в его формате *EXE* нужно изменить значение поля длины файла в заголовке файла.

Поэтому идентификаторы должны быть защищены от несанкционированного преобразования или удаления, а также от потери доступа. Другими словами, идентификационная метка должна быть устойчива к модификации и(или) удалению посторонними лицами.

Стойкость идентификационных меток обычно характеризуется способом введения опознавательного знака в программу. Добавление маскировки может повысить долговечность этих меток. Для этого предварительно зашифруйте метку.

Подходы, используемые в технологии программной идентификации, могут быть использованы для повышения стойкости идентификационных меток. Технология идентификации программ была специально разработана, чтобы предоставить практический инструмент идентификации программ для введения идентификационных меток в объектный код существующих программ.

Контрольные вопросы

1. Что представляет собой встроенная и пристыковочная система защиты программного обеспечения.

2. Поясните сущность локальной и сетевой программной защиты.

3. Опишите, каким образом защиту программного обеспечения можно защитить с помощью электронных ключей, и их преимущества и недостатки.

4. Поясните сущность, достоинства и недостатки защиты ПО путем привязки к параметрам компьютера и активации.

5. Объясните сущность защиты программ от копирования путем их передачи в сеть.

6. Поясните понятия декомпиляция, дизассемблирование, отладка ПО, приведите примеры.

7. Опишите защиту от побитового копирования компакт-диска, эмуляции компакт-диска и методы эмуляции ключа.

8. Опишите, как противостоять взлому программного модуля и отладке программного обеспечения.

9. Расскажите про способы противодействия дизассемблерам и дамперам.

4. ЭЛЕКТРОННЫЕ КЛЮЧИ

4.1. ЭЛЕКТРОННЫЕ КЛЮЧИ-ИДЕНТИФИКАТОРЫ *I-BATTON*

В 1991 году компания *Dallas Semiconductor* выпустила свои первые электронные ключи-идентификаторы серии *DS199x*. В начале для них был запатентован товарный знак «*Touch Memory*», который достаточно полно отражал основные свойства этих изделий. *Touch* – переводится как «касание», *Memory* – «память». Действительно, все ключи, которые внешне выглядят как металлические дисковые батарейки, в обязательном порядке имеют внутри микросхему-ПЗУ с уникальной для каждого устройства двоичной 48-разрядной кодовой комбинацией (идентификационным номером), а считывается эта комбинация при касании металлическим корпусом ключа к металлическому же зонду-считывателю.

Новый электронный ключ из Далласа стал популярным среди потребителей, и, как следствие, стали появляться новые модели. Одно из последних изделий этого ряда *DS1954* имеет внутри своего корпуса специальный микропроцессор для шифрования информации, разработан был также идентификатор со встроенным термопреобразователем, планируется реализовать идеи размещения других схем в стандартизованном компанией *Dallas Semiconductor* металлическом корпусе. Поэтому с начала 1997 года *Dallas Semiconductor* заявила о смене названия всех своих идентификационных ключей на *iButton*

(*Information Button* – «таблетка с информацией»), как более общее и охватывающее весь ряд изделий в настоящем и в будущем.

Рассмотрим описание конструкции, принципов работы и обзор последней номенклатуры электронных ключей в соответствие с новой терминологией.

Все электронные ключи-идентификаторы *iButton* внешне похожи на дисковую металлическую батарейку (рис. 4.1). Металл представляет собой нержавеющую сталь. Диаметр диска около 17 мм, толщина 3,1 мм или 5,89 мм. Диск состоит из двух электрически разъединенных половинок. Внутри он полый. В герметичную полость заключена электронная схема на кремниевом кристалле. Выход схемы соединен с половинками диска двумя проводниками. Половинки диска образуют контактную часть однопроводного последовательного порта. При этом через центральную часть идет линия данных, внешняя оболочка – земля.

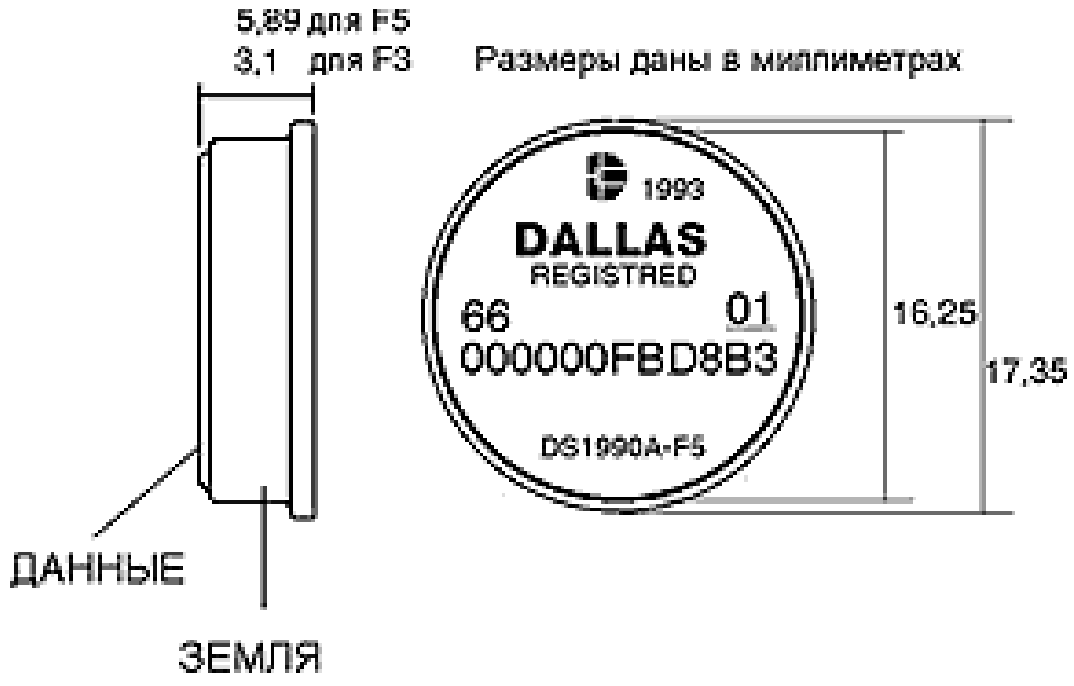

Рис. 4.1. Электронный ключ-идентификатор *iButton*

Для того чтобы произошел обмен информации *iButton* с внешними устройствами, необходимо прикоснутся обеими поверхностями половинок металлического диска к контактному устройству (зонду), также состоящему из двух электрически не связанных, проводящих электрический ток частей. Обычно для материала контактов зонда используют нержавеющую сталь или медный сплав, с нанесенным на него защитным токопроводящим покрытием. Процесс касания к зонду показан на рис. 4.2.

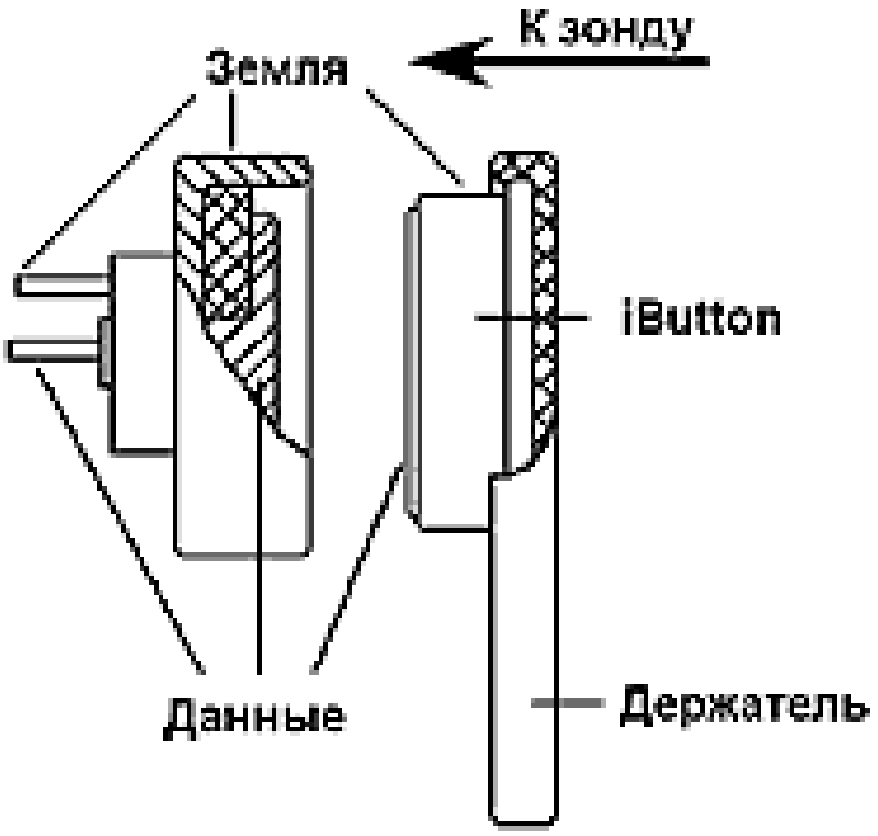

Рис. 4.2. Схема взаимодействия ключа и контактного устройства

Большая площадь поверхности контактов защищает систему от неточного совмещения при подключении по причине «человеческого фактора» или при автоматизированном касании, когда идентификатор

и зонд расположены на различных подвижных механизмах. Кроме того, дисковая форма корпуса направляет и очищает контакты, гарантируя надёжное соединения, а закругленный край корпуса легко совмещается с зондом.

Устройства семейства *iButton* предназначены для различных секторов рынка, в зависимости от их типа. Наиболее распространены они сейчас в качестве:

– идентификационных средств персонала для систем ограничения доступа в здания или отдельные помещения (особенно в условиях с повышенным уровнем воздействия внешней среды, то есть там, где традиционные карточки или устройства для считывания с них информации могут быстро выйти из строя);

– идентификационных меток оборудования и аппаратуры (специальное приспособление закрепляет диск *iButton* на плате оборудования или в его корпусе, уникальный номер позволяет производителю идентифицировать своё оборудование или защищать его от подделок, *IButton* с энергонезависимой памятью могут дополнительно хранить параметры эксплуатации, гарантийные обязательства и другие служебные характеристики изделий);

– аппаратных ключей в системах защиты информации (используется для защиты программного обеспечения компьютеров: защищаемая программа имеет встроенную процедуру обращения через один из портов компьютера к идентификационному номеру или энергонезависимой памяти *iButton* и в случае несоответствия номера или

содержимого энергонезависимой памяти идентификатора соответствующим записям в программе, программа не работает).

В разных системах ограничения доступа в здания или отдельные помещения могут использоваться различные ключи *iButton.* Так, для ограничения доступа в подъезды жилых домов, где не предъявляются повышенные требования к системам ограничения доступа, используют самые дешевые *iButton DS1990A*, которые стоят около 2-х долларов. В подъездах *iButton* устанавливают обычно в единой системе с домофоном – переговорной и управляющей дверным электромагнитным замком системой. Учитывая низкую цену считывающего устройства (около 25 долларов), установщики домофонов получают низкие цены и на всю электронную систему управления замками, при очень высоких потребительских параметрах, в особенности, защищенности от внешних воздействий. В таких системах жильцам подъезда выдаются *iButton DS1990*, в качестве ключей для подъездного замка. Закрепленные на пластмассовом брелоке, который можно носить в одной связке вместе с обычными механическими ключами, *iButton* можно ронять на землю или бетонный пол, они не боятся воды, льда (предел рабочей температуры -40^0C), кислот, масел, бензина, электромагнитных полей. Корпус рассчитан на 1 млн. касаний к зонду. Металлический зонд также хорошо вписывается в жесткие эксплуатационные нагрузки общих входных дверей подъезда.

Широко используются *iButton* также в качестве идентификационных карточек ограничения доступа в офисные помещения и на промышленные предприятия. В этих вариантах использования часто

закрепляют диск *iButton* на личной пластиковой карточке персонала, где дополнительно может быть размещена фотография и другие данные о специалисте. Обзор основных образцов *iButton* представлен в табл.4.1.

4.1. Обзор *iButton*

Маркировка	Описание	Емкость памяти
DS1920	Цифровой термодатчик	16 бит *EEPPOM* (электрически стираемое программируемое постоянное запоминающее устройство)
DS1954	Криптографический микропроцессор	32К *ROIVI*+6К *NVSRAM*
DS1963	Электронный кошелек	4096 бит *NVSRAM*
DS1971	*EEPROM* память	256+64 бит ЭСППЗУ – *EEPROM* (*ElectricallyErasable Programmable Read Only Memory*)
DS1981U	*EPROM* память и специальный номер	512 бит EPPOM (Electrically Programmable Read Only Memory)
DS1982U	*EPROM* память и специальный номер	1024 бит *EPROM*
DS1982	*EPROM* память	1024 бит *EPROM*
DS1985	*EPROM* память	16384 бит *EPROM*
DS1986	*EPROM* память	65536 бит *EPROM*
DS1990A	Только номер	нет
DS1991	Энергонезависимая память с паролем	1344 бит*NVSRAM* (*non-volatile SRAM*)

Продолжение табл. 4.1

Маркировка	Описание	Емкость памяти
DS1992	Энергонезависимая память	1024 бит *NVSRAM* (энергонезависимое статическое запоминающее устройство с произвольной выборкой)
DS1993	Энергонезависимая память	4096 бит *NVSRAM*
DS1994	Энергонезависимые память и часы	4096 бит *NVSRAM*
DS1995	Энергонезависимая память	16384 бит *NVSRAM*
DS1996	Энергонезависимая память	65536 бит *NVSRAM*

Дополнительные возможности по повышению степени ограничения доступа в помещения позволяют реализовывать *iButton* с защищенной паролем энергонезависимой памятью, а также новые *iButton DS1954* с микропроцессором-шифратором с длиной кода ключа 1024 бит, энергонезависимой памятью и часами-календарем. Последняя модель имеет высокую степень защиты информации. Такие системы обычно используются в банках и на предприятиях с повышенными требованиями безопасности.

4.2. БЕСКОНТАКТНЫЕ РАДИОМЕТКИ *RFID*

RFID (англ. *Radio Frequency IDentification* – радиочастотная идентификация) – метод автоматической идентификации объектов, в котором посредством радиосигналов считываются или записываются

данные, хранящиеся в так называемых транспондерах, или *RFID*-метках.

RFID – это современная технология идентификации, предоставляющая существенно больше возможностей по сравнению с традиционными системами маркировки.

Любая *RFID*-система состоит из считывающего устройства (считыватель, ридер или интеррогатор) и транспондера (он же *RFID*-метка, иногда также применяется термин *RFID*-тег).

Большинство *RFID*-меток состоит из двух частей. Первая – интегральная схема (ИС) для хранения и обработки информации, модулирования и демодулирования радиочастотного (*RF*) сигнала и некоторых других функций. Вторая – антенна для приёма и передачи сигнала.

RFID-метки можно классифицировать по ряду признаков:

– по рабочей частоте;

– по источнику питания;

– по типу памяти;

– по исполнению.

По типу источника питания *RFID*-метки делятся на пассивные, полупассивные и активные.

Пассивные *RFID*-метки не имеют встроенного источника энергии. Электрический ток, индуцированный в антенне электромагнитным сигналом от считывателя, обеспечивает достаточную мощность для функционирования кремниевого *CMOS*-чипа, размещённого в метке, и передачи ответного сигнала.

Полупассивные *RFID*-метки, также называемые полуактивными, очень похожи на пассивные метки, но оснащены батарей, которая обеспечивает чип энергопитанием. При этом дальность действия этих меток зависит только от чувствительности приёмника считывателя и они могут функционировать на большем расстоянии и с лучшими характеристиками.

Активные *RFID*-метки обладают собственным источником питания и не зависят от энергии считывателя, вследствие чего они читаются на дальнем расстоянии, имеют большие размеры и могут быть оснащены дополнительной электроникой. Однако, такие метки являются наиболее дорогими, и, кроме того, имеет ограниченное время работы батарей.

Активные метки в большинстве случаев обеспечивают большую точность считывания, чем пассивные. Обладая собственным источником питания, активные метки могут генерировать выходной сигал большего уровня, что позволяет применять их в агрессивных средах: в воде, металлах (корабельные контейнеры, автомобили) и на больших расстояниях вне помещения. Активных метки позволяют передавать сигнал на расстояния в сотни метров, а срок службы батареи такой метки может достигать 10 лет. Некоторые *RFID*-метки имеют встроенные сенсоры, например, для мониторинга температуры скоропортящихся товаров. Другие типы сенсоров в совокупности с активными метками могут применяться для измерения влажности, регистрации толчков или вибрации, света, радиации, температуры и наличия газов в атмосфере.

Радиус считывания для активных меток составляет до 300м. Они имеют больший объем памяти, чем у пассивных меток и, и способны хранить больший объем информации. В настоящее время активные метки делают размером не больше обычной пилюли и продают по цене в несколько долларов.

По типу используемой памяти *RFID*-метки классифицируют на следующие типы:

– *RO* (*Read Only*) – данные записываются только один раз, сразу при изготовлении (такие метки пригодны только для идентификации, новую информацию в них записать нельзя, и их практически невозможно подделать);

– *WORM* (*Write Once Read Many*) – кроме уникального идентификатора такие метки содержат блок однократно записываемой памяти, которую в дальнейшем можно многократно читать;

– *RW* (*Read and Write*) – такие метки содержат идентификатор и блок памяти для чтения и записи информации (данные в них могут быть перезаписаны многократно).

По рабочей частоте *RFID*-метки бывают следующих диапазонов:

– *LF* 125 – 134кГц (пассивные системы данного диапазона имеют низкую стоимость и по своим физическим характеристикам используются для вживления подкожных меток животным, людям и рыбам, имеют существенные ограничения по радиусу действия и точности (коллизии при считывании));

– *HF* 13,56МГц (являются достаточно дешевыми, не имеют экологических проблем, хорошо стандартизованы и имеют широкую ли-

нейку решений, применяются в платежных системах, логистике, идентификации личности);

– *UHF* 860 – 960МГц (обладают наибольшей дальностью действия, многими стандартами меток данного диапазона разработаны антиколлизионные механизмы).

Изначально ориентированные на использование в складской и производственной логистике, *UHF*-метки не имели уникального идентификатора. Предполагалось, что идентификатором для метки будет служить *EPC*-номер (*Electronic Product Code*) товара, который каждый производитель будет заносить в метку самостоятельно при производстве. Однако скоро стало ясно, что помимо функции носителя *EPC*-номера товара хорошо бы возложить на метку еще и функцию контроля подлинности. То есть возникло требование, противоречащее самому себе: одновременно обеспечить уникальность метки и позволить производителю записывать произвольный *EPC*-номер.

Долгое время не существовало чипов, которые бы удовлетворяли этим требованиям полностью. Выпущенный компанией *Philips* чип *Gen 1.19* обладал неизменяемым идентификатором, но не имел никаких встроенных функций по паролированию банков памяти метки, и данные с метки было легко считать, имея соответствующее оборудование. Позднее разработанные чипы стандарта *Gen 2.0* уже имели функции защиты банков памяти (пароль на чтение, на запись), но не имели уникального идентификатора метки, что позволяло при желании создавать идентичные клоны меток.

Еще позже компания *NXP* выпустила два новых чипа, которые на сегодняшний день отвечают всем выше перечисленным требова-

ниям. Чипы *SL3S1202* и *SL3FCS1002* выполнены в стандарте *EPC Gen 2.0*, но отличаются от своих предшественников тем, что поле памяти *TID* (*Tag ID*), в которое при производстве обычно пишется код типа метки, который в рамках одного артикула не отличается от метки к метке, разбито на две части. Первые 32 бита отведены под код производителя и марку, а вторые 32 бита — под уникальный номер самого чипа. Поле *TID* – неизменяемое и, таким образом, каждая метка является уникальной. Каждый банк памяти меток может быть защищен от чтения или записи паролем, а *EPC*-номер может быть записан производителем товара в момент маркировки.

Что касается стоимости, то *UHF*-метки дешевле, чем их собратья диапазонов *LF* и *HF*, но в целом *RFID*-система *UHF* дороже за счет стоимости остального оборудования.

В настоящее время частотный диапазон *UHF* (СВЧ) открыт для свободного использования в Российской Федерации в так называемом «европейском» диапазоне: 863 – 868 МГЦ.

4.3. ЭЛЕКТРОННЫЕ *USB*-КЛЮЧИ И СМАРТ-КАРТЫ *e-Token*

Электронные *USB*-ключи и смарт-карты *eToken* представляют собой компактные устройства, предназначенные для защиты информации корпоративных заказчиков и частных пользователей. Устройства *eToken* содержат процессор и модули памяти, функционируют под управлением своей операционной системы, выполняют необходимые прикладные программы и хранят информацию.

USB-ключи и смарт-карты *eToken* базируются на высокозащищенной платформе, разработанной для производства смарт-карт – области, в которой традиционно предъявляют повышенные требования к информационной безопасности. Поэтому *USB*-ключи и смарт-карты *eToken* фактически являются миниатюрным компьютером, обеспечивающим безопасное хранение персональных данных своих пользователей и надежно защищенным от несанкционированного вмешательства извне.

Модельный ряд *eToken* разработан таким образом, чтобы удовлетворить потребности большинства пользователей. Линейка *USB*-ключей и смарт-карт *eToken* включает в себя устройства, выполняющие базовые функции безопасности, а также комбинированные продукты, сочетающие в себе возможности нескольких устройств.

Используя продукты *eToken*, можно решить следующие задачи:

– усовершенствовать процесс аутентификации (двухфакторная аутентификация) на локальном компьютере и в корпоративной сети, а также защищенный доступ к бизнес-приложениям;

– зашифровать данные на серверах, и рабочих станциях;

– обеспечить защиту персональных данных;

– защитить электронную почту и взаимодействие с коллегами в системах электронного документооборота;

– обезопасить финансовые операции в системах дистанционного банковского обслуживания (ДБО);

– внедрить электронную подпись и защитить документы в системах электронной отчетности;

– обеспечить защиту корпоративного сайта.

Продукты линейки *eToken* являются основой инфраструктуры информационной безопасности современного предприятия. Они поддерживаются всеми ведущими производителями информационных систем и бизнес-приложений, соответствуют требованиям российских регулирующих органов. Внедрение *USB*-ключей или смарт-карт *eToken* позволяет не только решить нынешние актуальные задачи, но и сохранить инвестиции в последующих проектах по защите информации.

На рисунке 4.3. представлен внешний тип *USB*-ключей и смарт-карты *eToken*.

Рис.4.3. Внешний вид *USB*-ключей и смарт-карты *eToken*

По умолчанию, все без исключения *USB*-ключи *eToken* привозятся в пластмассовом корпусе темно-синего тона с изданным на фронтальный плоскости серийным номером устройства. Смарт-карты *eToken* привозятся в виде «белоснежного пластика».

Смарт-карты и *USB*-ключи *eToken* имеют все шансы являться дополненными бесконтактными пассивными радиометками *RFID*, которые используются с целью контролирования физического доступа работников в здании. Смарт-карты и *USB*-ключи со встроенными радиометками используются в системах контроля и управления доступом (СКУД), концепциях учета рабочего времени персонала, бесконтактных «электронных проходных» и тому подобных.

Контрольные вопросы

1. Основные области применения ключей *iButton*.
2. Обязательные и дополнительные составляющие электронных ключей.
3. Вариант классификации семейства электронных ключей.
4. Что такое *RFID*-тег?
5. Каким образом классифицируют *RFID*-метки?
6. Что такое смарт-карты и *USB*-ключи *eToken*?
7. Какие задачи информационной безопасности решаются с помощью смарт-карт и *USB*-ключей *eToken*?

ЗАКЛЮЧЕНИЕ

За рамками рассмотрения части 2 данного учебного пособия остались такие группы программно-аппаратных средств защиты информации, как средства обнаружения вторжений, средства защиты от вредоносного программного обеспечения, средства аутентификации электронных данных, средства управления криптографическими ключами, средства защиты ПО от несанкционированного использования, электронные ключи. Все эти средства будут рассмотрены в части 3.

Но даже полное изучение изложенного учебного материала не гарантирует исчерпывающих знаний современных программно-аппаратных средств защиты информации. Эти средства быстро развиваются вслед за совершенствованием угроз информационной безопасности и элементной базы радиоэлектронных устройств. Однако, знание фундаментальных основ построения и функционирования программно-аппаратных средств защиты информации позволит самостоятельно изучать все появляющиеся новинки на рынке этих средств.

СПИСОК ЛИТЕРАТУРЫ

1. **Федеральный** государственный образовательный стандарт высшего образования по специальности 10.05.03 «Информационная безопасность автоматизированных систем» – специалитет : утвержден приказом Министерства науки и высшего образования РФ от 26 ноября 2020 г. № 1457 [электронные данные]. – URL:fgosvo.ru/uploadfiles/FGOS%20VO%203++/Spec/100503_C_3_180 22021.pdf (дата обращения: 30.01.2017).

2. **Федеральный** государственный образовательный стандарт высшего образования по специальности 03.03.02 «Информационные системы и технологии» – бакалавриат :утвержден приказом Министерства науки и высшего образования РФ от 19 сентября 2017 г. № 926 [электронные данные]. – URL:fgosvo.ru/uploadfiles/FGOS%20VO%203++/Spec/100503_C_3_180 22021.pdf (дата обращения: 30.01.2017).

3. **ГОСТ 28147–89.** Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования информации. – М. : Госкомитет СССР по стандартам, 1989.

4. **Программно-аппаратные** средства обеспечения информационной безопасности. Теоретические основы компьютерной безопасности : учебное пособие для вузов / П. Н. Девянин, О. О. Михальский, Д. И. Правиков, А. Ю. Щербаков. – М. : Радио и связь, 2000. – 192 с.

5. **Завгородний, В. И.** Комплексная защита информации в компьютерных системах : учебное пособие для вузов / В. И. Завгородний. – М. : Логос, 2001. – 264 с.

6. **Зегжда, Д. П.** Основы безопасности информационных систем / Д. П. Зегжда, А. М. Ивашко. – М. : Горячая Линия – Телеком, 2000. – 452 с.

7. **Малюк, А. А.** Введение в защиту информации в автоматизированных системах / А. А. Малюк, С. В. Пазизин, Н. С. Погожин. – М. : Горячая Линия – Телеком, 2001. – 148 с.

8. **Коржик, В. И.** Основы криптографии / В. И. Коржик, В. П. Просихин. – СПб. : Линк, 2008.

9. **Руководящий** документ. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от несанкционированного доступа информации в автоматизированных системах и средствах вычислительной техники // Сборник руководящих документов по защите информации от несанкционированного доступа. – М. : ГТК при президенте РФ, 1998. – 120 с.

10. **Руководящий** документ. Защита от несанкционированного доступа к информации. Термины и определения // Сборник руководящих документов по защите информации от несанкционированного доступа.– М. : ГТК при президенте РФ, 1998. – 120 с.

11. **Мао, В.** Современная криптография: теория и практика / В. Мао. – СПб. : Вильямс, 2005.

ОГЛАВЛЕНИЕ

Учебное электронное издание

ГРИДНЕВ Виктор Алексеевич
ГУБСКОВ Юрий Анатольевич
ДЕРЯБИН Андрей Сергеевич
ЯКОВЛЕВ Алексей Вячеславович

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

В ТРЕХ ЧАСТЯХ

ЧАСТЬ 2

Учебное пособие

Редактирование Е. С. Мордасовой
Графический и мультимедийный дизайнер Т. Ю. Зотова
Обложка, упаковка, тиражирование Т. Ю. Зотовой

ISBN 978-5-8265-2609-5

Подписано к использованию 29.06.2023.
Тираж 50 шт. Заказ № 68

Издательский центр ФГБОУ ВО «ТГТУ»
392000, г. Тамбов, ул. Советская, д. 106, к. 14
Телефон: (4752) 63-81-08
E-mail: izdatelstvo@tstu.ru