МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 519.7

ЗАЩИЩЁННОЕ ХРАНЕНИЕ ДАННЫХ И ПОЛНОДИСКОВОЕ ШИФРОВАНИЕ

Е. К. Алексеев, Л. Р. Ахметзянова, А. А. Бабуева, С. В. Смышляев

ООО «КРИПТО-ПРО», г. Москва, Россия

Рассматривается задача защищённого хранения данных в современных операционных системах. Обсуждаются различные подходы к её решению, реализуемые программным образом. Основное внимание уделяется системам полнодискового шифрования как наиболее универсальной из технологий защиты хранимых данных. Перечисляются эксплуатационные и криптографические свойства, которые необходимо учитывать при их проектировании и сравнении. Описываются некоторые типичные сценарии использования таких криптосистем. Результаты работы могут быть использованы при синтезе и сравнении систем полнодискового шифрования.

Ключевые слова: *модели и методы защиты информации, защита хранимых данных.*

DOI 10.17223/20710410/49/6

DATA STORAGE SECURITY AND FULL DISK ENCRYPTION

E. K. Alekseev, L. R. Akhmetzyanova, A. A. Babueva, S. V. Smyshlyaev

CryptoPro, Moscow, Russia

E-mail: alekseev@cryptopro.ru, lah@cryptopro.ru, babueva@cryptopro.ru, svs@cryptopro.ru

In the paper, a systematic description of the process of providing the security of data storage in modern operating systems is presented. The advantages of Full Disk Encryption (FDE) modules as compared with the other ways to security of this data storage are considered and explained. For most of modern FDE modules, there are four stages of work, namely: setup — initial data encryption, mounting — unfolding the key system in OS memory, session — reading and writing data using the FDE module (interaction of the file system with the hard disk driver), and unmounting — carrying out operations for ensuring purposeful properties of security and finishing work with the FDE module. These stages are introduced for the operating FDE module, including possible disrepairs, which are also systematized and considered in details. Performance characteristics that are important for synthesis and analysis are listed. Also, their target protective properties are studied in detail, the relationship between the problems of ensuring the confidentiality and integrity of data storage is shown and substantiated. New variants of these security properies are introduced so

that they can become a guideline in the creation of FDE modules and a possible trade-off between performance and security. Some typical scenarios of using such systems are described.

Keywords: *models and methods in information security, data storage security.*

Введение

При разработке и применении средств криптографической защиты информации и, в том числе, средств электронной подписи принято учитывать требования не только к безопасности реализации самих криптографических преобразований, но и в целом к защите систем, на которых происходит обработка данных. Начиная с определённых классов защиты, требуется учитывать возможность противника совершать атаки из пределов контролируемой зоны [1, п. А.4.3], в том числе при наличии у противника доступа к атакуемой системе в качестве легитимного пользователя [2, п. 15 Приложения 1].

Таким образом, в ряде случаев как обрабатываемые пользователем данные, так и служебные файлы криптосредства необходимо защищать от атак со стороны других пользователей той же системы. Основным классом механизмов защиты, противодействующих подобным атакам, являются системы разграничения доступа, в том числе встроенные в операционные системы [1, п. 5.5.4]. Действительно, корректно настроенная система разграничения доступа не позволит противнику, работающему в системе, получить доступ как к данным атакуемого пользователя, так и к служебным файлам (например, базе настроек или журналу аудита).

Однако в реальной жизни именно вокруг уязвимостей в системах разграничения доступа и их настройках ведутся ожесточённые бои между разработчиками (операционных систем, а также прикладного и системного программного обеспечения) и злоумышленниками: иллюстрацией этого тезиса может являться количество уязвимостей в базах CVE, которые можно найти по ключевым словам «access control» и «privileges». Кроме того, зачастую заточенные под обеспечение максимального уровня безопасности настройки разграничения доступа существенно усложняют использование информационных систем простыми пользователями, а их обход для получения прямого доступа к хранимым данным остаётся возможным с помощью посекторного чтения жёсткого диска после извлечения его из компьютера. При этом в случае совместного использования пользователями одной виртуальной машины никакие меры по разграничению доступа в рамках операционной системы не помогут против злоумышленника, владеющего доступом к физической машине.

По этим причинам крайне важной является задача обеспечения конфиденциальности и целостности хранимых данных, чтобы даже злоумышленник с частичным или полным доступом к диску (но не к используемой ключевой информации, которая, как правило, хранится на отчуждаемых носителях) не смог ни прочитать защищаемые данные, ни внести в них несанкционированные изменения.

Настоящая работа посвящена системам полнодискового шифрования. Рассматриваются эксплуатационные и криптографические особенности построения, применения и анализа данного класса криптосистем. Насколько известно авторам, в отечественной литературе данная тема подробно не рассматривалась, поэтому об указанных криптосистемах можно найти лишь краткие упоминания [3]. При этом в зарубежной литературе полнодисковому шифрованию уделено достаточно много внимания. На сегодняшний день по числу рассматриваемых вопросов выделяются диссертации [4, 5],

содержащие обширный список полезных ссылок. Отметим, что большинство работ посвящено конкретным схемам полнодискового шифрования, при этом работ, посвящённых определению и формализации целевых для таких схем свойств безопасности, гораздо меньше (см., например, [6, 7]).

В работе приводится систематизированное описание процесса хранения данных в современных операционных системах, поясняется преимущество систем полнодискового шифрования перед другими подходами к защите этих данных. Вводятся этапы работы систем полнодискового шифрования, в том числе с учётом возможных сбоев, которые также систематизированы и подробно рассмотрены. Перечисляются важные с точки зрения синтеза и анализа эксплуатационные характеристики таких систем. Подробно рассмотрены целевые свойства безопасности, указана и обоснована связь задач обеспечения конфиденциальности и целостности хранимых данных. Введены новые градации этих свойств безопасности, которые могут послужить ориентиром при создании систем полнодискового шифрования и возможным компромиссом между производительностью и безопасностью. Перечислены некоторые типичные сценарии использования таких систем.

1. Хранение и защита данных в ОС

При сохранении данных на диск в большинстве современных операционных систем выделяются следующие уровни, на которых могут осуществляться сопутствующие этому процессу операции:

— уровень прикладных программных компонентов;
— уровень файловой системы;
— уровень драйвера диска;
— уровень контроллера жесткого диска.

Отметим, что потребность в сохранении данных может возникнуть не только на уровне прикладных компонентов, но и, например, на уровне файловой системы (пример таких данных — время создания файла). В этом случае такие данные обычно называют служебными данными соответствующего уровня, а операции по их сохранению осуществляются начиная с того уровня, где они появились.

Взаимодействие между прикладными программными компонентами и контроллером диска осуществляется следующим образом (схематически данный процесс представлен на рис. 1):

— Прикладная компонента при работе с диском оперирует интерфейсом, предоставляемым файловой системой. При этом информация имеет древовидную структуру, которая описывается в терминах директорий и файлов.
— Файловая система посылает запрос на чтение или запись данных диска с помощью интерфейса, предоставляемого драйвером диска. При этом единицей чтения и записи данных является логический сектор — байтовый массив фиксированного размера (как правило, этот размер кратен 512).
— Драйвер диска обращается для записи или чтения данных к контроллеру жёсткого диска. На этом этапе взаимодействие осуществляется в терминах, максимально приближенных к физической структуре диска (например, дорожек и физических секторов).

Вопрос защиты хранимых данных имеет ярко выраженный прикладной характер, поэтому при рассмотрении подходов к обеспечению безопасности данных необходимо уделять особое внимание их эксплуатационным свойствам. Для определённости

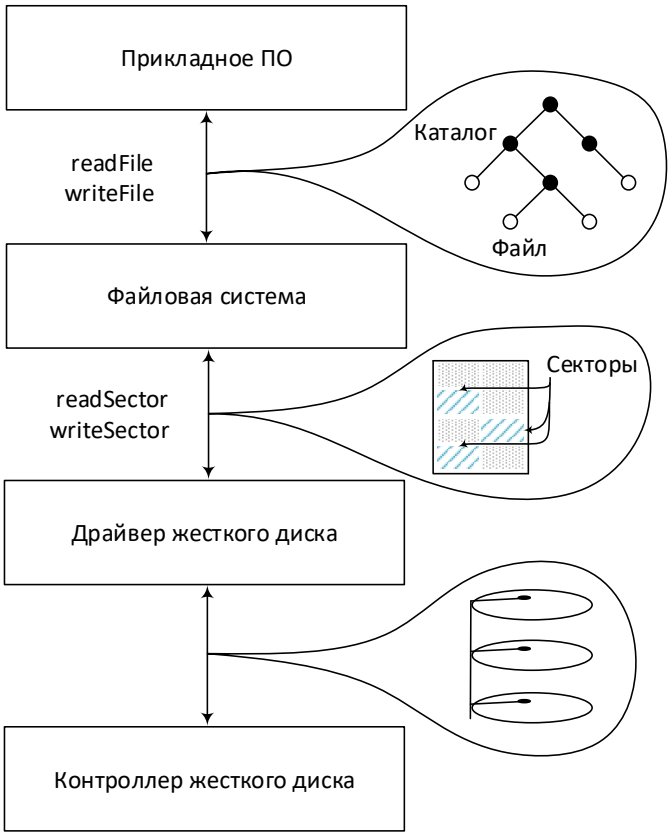

Рис. 1. Порядок взаимодействия файловой системы и жесткого диска

будем считать, что каждый подход реализуется некоторым модулем, функционирующим в рамках ОС. Учитывая иерархичность и многокомпонентность системы обработки данных, особого внимания заслуживает универсальность подхода, которая тем выше, чем больше число компонент, для которых модуль защиты может обеспечить свойства безопасности, и чем меньше дополнительных действий (например, изменений в программном коде) для этого необходимо осуществить.

Универсальность подхода во многом определяется положением модуля защиты в общей архитектуре системы, т. е. тем, где он расположен с точки зрения перечисленных уровней взаимодействия. Модуль защиты данных гипотетически может функционировать на любом из этих уровней, причём он может быть как встроен в штатную программную компоненту некоторого уровня, так и располагаться между уровнями, работая «прозрачно» для вышестоящих компонент. Под прозрачностью понимается то, что модуль полностью повторяет интерфейс компоненты, находящейся непосредственно под ним. За счёт этого модулю удаётся обеспечить защиту для всех вышестоящих компонент без необходимости внесения в них каких-либо изменений (иногда говорят, что вышестоящие компоненты даже «не знают» о том, что для сохраняемых ими данных обеспечивается защита). Стоит отметить, что модуль защиты не в состоянии обеспечить какую-либо безопасность служебных данных любых нижестоящих компонент.

Отметим, что обычно для реализации защиты ниже уровня драйвера диска используются аппаратные решения. Примером аппаратной реализации защиты храни-

мых данных является SSD-диск со встроенным шифрованием данных «Integral Crypto SSD» (другое название — «Integral Memory Crypto Hard Drive») [8]. Мы ограничимся рассмотрением только программных модулей защиты данных и не будем рассматривать варианты, когда защита осуществляется на уровне ниже драйвера диска.

Приведём примеры осуществления защиты хранимых данных на разных уровнях взаимодействия.

— Функция шифрования документов, встроенная в текстовый редактор Microsoft Word. Модуль защиты интегрирован в конкретную прикладную программную компоненту и не предназначен для защиты данных других приложений.
— Модуль защиты файлов AxCrypt является компонентой прикладного уровня и может обеспечить защиту любого файла, вне зависимости от того, какой прикладной компонентой он был создан. Однако для его применения необходимо каждый раз дополнительно указывать, какой файл должен быть защищён (или отдельно производить встраивание в случае предоставления соответствующего программного интерфейса).
— Функции защиты данных, реализованные в файловой системе NTFS, обычно объединяются под общим названием EFS (Encrypting File System). Эти функции формируют модуль защиты, который, по сути, реализован внутри файловой системы. Обеспечивая прозрачную защиту данных любых компонент прикладного уровня, EFS при этом не может защитить данные, обрабатываемые другими файловыми системами.
— Программный модуль VeraCrypt обеспечивает защиту всех секторов жёсткого диска. Он функционирует между файловой системой и драйвером жёсткого диска.

Заметим, что драйвер диска, если и порождает в процессе работы служебные данные, то они имеют крайне незначительный объём, а необходимость в их защите минимальна. Поэтому подход, предполагающий реализацию модуля защиты между драйвером диска и файловой системой, представляется наиболее универсальным. Например, в его рамках можно обеспечить защиту такой зачастую критически важной информации, как топология файловой системы, имена хранимых файлов, их размеры и даты модификации. Далее рассматриваются свойства модулей защиты именно такого типа.

2. Полнодисковое шифрование

Прежде чем начать рассмотрение по существу, уделим внимание терминологии. Полнодисковое шифрование — наиболее устоявшийся в данной предметной области термин, обозначающий процесс, который реализуется модулями защиты, функционирующими между драйвером диска и файловой системой, основывается на применении криптографических методов и предназначается для обеспечения безопасности данных, хранящихся на жёстких дисках. Первая часть термина («полно») объясняется тем, что данный процесс предполагает защиту всего пространства диска, выделенного для хранения полезной нагрузки (прикладных данных). При этом вторая и третья части термина («диск» и «шифрование») могут вызвать вопросы. Во-первых, из-за того, что на сегодняшний день в основе устройств, применяемых для хранения информации, не обязательно лежит набор физических дисков. Во-вторых, из-за того, что защита хранимых данных, как показано далее, не всегда ограничивается только шифрованием. Однако в рамках настоящей работы будем использовать именно этот термин, поскольку он является наиболее близким русскоязычным аналогом устоявшегося в зарубежной литературе термина «Full Disk Encryption» [7, 4]. Далее для краткости будем использовать соответствующую аббревиатуру FDE.

2.1. Основные понятия

Модуль FDE перехватывает запрос от файловой системы, преобразует переданные в его составе данные и осуществляет взаимодействие с драйвером диска для реализации функций безопасности и собственно записи данных на диск. Для реализации «прозрачной» защиты данных интерфейс FDE должен полностью совпадать с интерфейсом драйвера жёсткого диска той ОС, в которой он функционирует. Таким образом, файловая система «не знает» о том, что данные на используемом ею жёстком диске защищены. Схема взаимодействия файловой системы с драйвером жёсткого диска при наличии модуля FDE представлена на рис. 2.

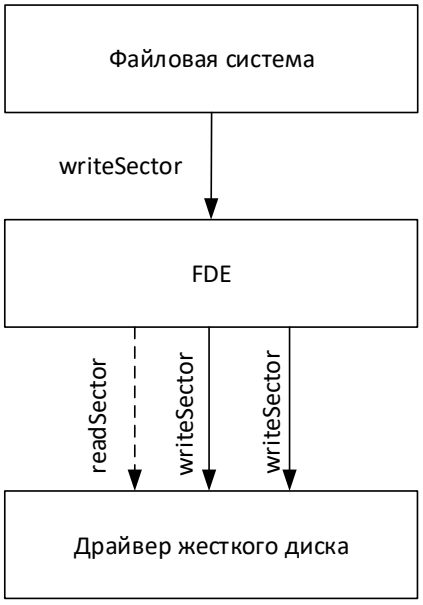

Рис. 2. Взаимодействие файловой системы с драйвером жесткого диска при наличии модуля FDE

При обработке запроса от файловой системы модуль FDE может посылать к драйверу диска уже несколько запросов на чтение и запись. Например, запрос на запись одного сектора, сделанный файловой системой, может потребовать от модуля FDE одной операции чтения для получения текущего значения количества осуществлённых операций записи данного сектора (для контроля нагрузки на ключ или формирования уникального инициализирующего вектора) и двух операций записи (одна для записи непосредственных данных сектора, вторая — для записи нового значения имитовставки сектора). Отдельно отметим, что FDE-модуль должен допускать параллельное осуществление нескольких операций с прикладными данными диска.

Замечание 1. Использование FDE-модулем для хранения служебной информации исключительно пространства защищаемого диска, посекторный доступ к которому реализуется некоторым драйвером, является типичным сценарием, который наиболее удобен для объяснения основных концепций. Однако в общем случае необходимо лишь, чтобы FDE-модуль предоставлял посекторный доступ компонентам более верхнего уровня. То, как FDE-модуль реализует такую «посекторную» абстракцию, может зависеть от особенностей решаемой задачи. Например, FDE-модуль может сохранять данные в обычный файл, доступ к которому может быть реализован с помощью штат-

ных возможностей файловой системы. При этом для хранения служебных данных FDE-модуль может использовать некоторую базу данных, порядок взаимодействия с которой может существенно отличаться от посекторного. В таких случаях определения некоторых эксплуатационных характеристик, обсуждаемых в п. 2.2, должны быть скорректированы. Далее все рассуждения проводятся для типичного случая работы FDE-модуля со стандартным драйвером диска, предоставляющим посекторный доступ.

В настоящей работе рассмотрим только те FDE-схемы, для которых как минимум часть ключевого материала, используемого для обеспечения целевых функций безопасности, хранится на защищённом носителе (токене). Альтернативой являются схемы, стойкость которых основана на запоминаемом пользователем пароле. К сожалению, такие схемы не позволяют достичь интересующего авторов уровня стойкости. Необходимый FDE-модулю размер памяти токена, доступной для чтения и, возможно, записи, является важной характеристикой FDE-модуля.

Для большинства (контрпримеры авторам не известны) современных FDE-схем можно выделить четыре этапа работы. Они перечислены ниже. Будем рассматривать только те схемы, для которых справедливо такое разделение.

1) *Этап инициализации* (*Setup*):
   на данном этапе FDE-модулю на вход подаётся диск в исходном незащищённом виде; модуль осуществляет все действия, необходимые для его последующей защиты, а именно его разметку с выделением, возможно, области для своих служебных данных, генерацию ключей, начальное зашифрование данных, сохранение необходимой информации на токене и т. п.
2) *Этап начала сеанса* (*Mount*):
   предоставление FDE-модулю возможности взаимодействия с токеном, содержащим ключевой материал, с помощью которого защищён диск; разворачивание в оперативной памяти ОС ключевой системы, необходимой для работы с FDE-модулем; чтение с токена дополнительных данных; осуществление других операций, необходимых для обеспечения целевых свойств безопасности и перехода к следующему этапу.
3) *Основной этап* (*Session*):
   взаимодействие файловой системы с драйвером жёсткого диска (чтение и запись данных) через модуль FDE.
4) *Этап завершения сеанса* (*Unmount*):
   осуществление операций, необходимых для обеспечения целевых свойств безопасности и завершения работы с FDE-модулем; завершение работы с токеном (например, загрузка модифицированных за время сеанса дополнительных данных и отсоединение токена от рабочей машины).

2.2. Эксплуатационные свойства

Перечислим эксплуатационные характеристики и особенности, которые необходимо учитывать при проектировании и анализе FDE-схем. Сначала отметим одну особенность функционирования FDE-модулей. Учитывая высокие требования к эффективности, при реализации FDE-модулей могут использоваться различные приёмы, позволяющие увеличить скорость их работы. Так, некоторые данные (например, данные, позволяющие проверять целостность секторов) могут заранее загружаться с диска в оперативную память. Таким образом, при определении характеристик FDE-схем

нужно учитывать возможную зависимость их эффективности от размера доступной оперативной и, возможно, защищённой памяти.

При проектировании FDE-схем должны учитываться эксплуатационные требования как к основному этапу их работы (сеансу), так и к другим трём этапам. Так, сложно представить ситуацию, допускающую монтирование диска в течение 30 мин.

Используемая память. Безусловно, характеристиками любой FDE-схемы являются необходимые ей для работы размер защищённой памяти, размер оперативной памяти и относительный размер её служебных данных (т. е. отношение размера служебных данных к размеру защищаемого полезного пространства диска). Учитывая то, что было сказано о возможностях оптимизации работы FDE-схем, при характеризации конкретной схемы необходимо использовать некоторую относительную величину, например, может быть приведено отношение размера используемой оперативной памяти к размеру области диска, работа с которым может быть оптимизирована.

Операционная трудоёмкость. Под этим термином понимается количество операций чтения и записи секторов, осуществляемых драйвером диска, при реализации FDE-модулем своих функций на разных этапах работы. Эти операции могут существенно отличаться по быстродействию в зависимости от используемой аппаратной платформы. Они выделяются в отдельную категорию ещё и потому, что зачастую ощутимо влияют на износ диска, а их допустимое количество заметно ограничено.

Вычислительная трудоёмкость. Данная характеристика является относительно стандартной и подразумевает объём вычислений, производимых FDE-модулем для осуществления операций на разных этапах работы.

Устойчивость к сбоям. Для устройств хранения данных крайне важным является вопрос надёжности: аварийные ситуации различного характера должны минимально влиять на хранимые данные. Использование FDE-модуля может различным образом влиять на это. Например, если при внезапном отключении питания актуальные ключи шифрования данных хранились только в оперативной памяти, то все данные диска могут стать недоступными. При этом необходимо учитывать не только получение самих данных, но и сохранение доверия к ним (в случае обеспечения целостности). Опишем некоторые практически актуальные аварийные ситуации, которые могут возникнуть во время работы FDE-модуля.

— *Отключение питания.* FDE-модуль мгновенно прекращает свою работу, используемая им информация в оперативной памяти утрачивается.
— *Отключение диска.* FDE-модуль теряет возможность осуществлять операции чтения и записи как минимум части данных, хранящихся в долговременной памяти. В качестве примера можно привести гипотетический FDE-модуль, оперирующий с данными, которые хранятся на нескольких дисках, один из которых отсоединяется пользователем.
— *Отключение защищённой памяти.* Например, пользователь может по ошибке отсоединить токен от вычислительной машины, на которой в этот момент работает FDE-модуль. В этом случае FDE-модуль теряет возможность оперировать с защищённой памятью.

Дополнительно отметим аварийную ситуацию, которая может реализоваться в любой момент существования диска.

— *Ошибки хранения данных на физическом уровне.* Например, из-за износа диска некоторый сектор может перестать быть доступным для чтения. Это может привести как к невозможности расшифровать часть данных, так и к потере доверия

к их целостности. Довольно простой мерой предотвращения таких ситуаций может быть дублирование данных диска (данные хранятся на двух дисках одинакового размера, операции дублируются).

Некоторые FDE-модули могут предусматривать особые процедуры, связанные с аварийными ситуациями. Это приводит к выделению новых этапов работы FDE-модулей, примеры которых приведены далее:

5) *Этап аварийной работы* (*Emergency*):
начинается в том момент, когда реализуется аварийный сценарий, допускающий продолжение работы в принципе. FDE-модуль может как продолжить осуществлять прикладные запросы (если это осмысленно), так и выполнять операции, необходимые для максимально корректного и безопасного завершения работы.
6) *Этап восстановления* (*Recovery*):
осуществляются операции, необходимые для восстановления работы с защищаемыми данными после сбоя (этапы могут различаться в зависимости от типа произошедшего сбоя).

2.3. Криптографические свойства

Говорить о криптографических свойствах имеет смысл только в рамках набора условий и предположений, определяющих возможности и цели потенциальных противников и обычно объединяемых под названием «модель противника». Подробно это понятие и один из подходов к его формализации рассмотрены в [9, 10]. Модель противника состоит из трёх компонент: типа атаки, модели угрозы и доступных противнику информационных и вычислительных ресурсов. Третья компонента обычно обсуждается уже на этапе анализа конкретных криптосистем, а не в рамках работ о предметной области в целом, но одну её особенность для систем полнодискового шифрования необходимо отметить.

Под информационными ресурсами противника обычно понимается количество данных, которые он потенциально может получить в процессе работы криптосистемы (шифртексты, имитовставки, электронные подписи и т. п.). Пути получения этой информации и её характер могут быть очень разными: перехват зашифрованных сообщений в канале связи, получение информации о внутренних состояниях криптосистемы по побочным каналам и т. д. Одним из самых ярких наглядных примеров того, что получение противником большого количества такой информации может приводить к серьёзным уязвимостям, является атака Sweet32 [11] на протокол TLS. Общим методом исключения таких возможностей противника является ограничение так называемой «нагрузки на ключ», т. е. количества данных, которые могут быть обработаны на одном ключе (этот приём и само понятие «нагрузка на ключ» рассматривается в [12, 13]; неформальное рассмотрение можно найти в [14, 15]). При синтезе и анализе FDE-модулей данному аспекту необходимо уделять особое внимание, так как период интенсивного функционирования и, как следствие, использования ключей такими криптосистемами может исчисляться годами и порой ограничивается лишь сроками службы устройств, используемых для хранения сопутствующих данных.

Две другие части модели противника определяют возможности противника по взаимодействию с криптосистемой (*тип атаки*) и его цели по нарушению целевых для неё свойств безопасности (*модель угрозы*). Рассмотрим каждую из этих компонент для систем полнодискового шифрования.

Тип атаки

В рамках функционирования FDE-модуля выделяются следующие компоненты, связанные с хранением и обработкой критичной для криптосистемы информации:

— защищённая память;
— оперативная память устройства, на котором функционирует FDE-модуль (о ней есть смысл говорить только во время работы FDE-модуля);
— совокупность долговременно хранимых данных, используемых FDE-модулем для реализации абстракции защищаемого диска (далее для краткости будем называть эти данные просто защищаемым диском).

Потенциально противник может взаимодействовать со всеми перечисленными компонентами, например получать какую-либо информацию по побочным каналам. Однако в рамках настоящей работы подробно рассмотрим взаимодействие противника только с защищаемым диском.

Особо отметим, что в настоящей работе не рассматриваются вопросы целостности и безопасной загрузки исполняемого кода FDE-модуля и сопутствующего программного обеспечения. Так, предполагается, что загрузка операционной системы осуществлена под защитой, например, некоторого аппаратно-программного модуля доверенной загрузки.

Взаимодействие с защищаемым диском. Противник может взаимодействовать с защищаемым диском, читая и записывая данные. При обсуждении этих возможностей будем следовать традиционному для криптографии подходу, состоящему в максимизации этих возможностей (например, возможности писать любые данные в любые секторы, а не только данные какого-либо специального содержания в какие-либо конкретные секторы). Это связано с тем, что спектр возможных условий эксплуатации систем полнодискового шифрования, как и любых криптосистем более или менее общего характера, крайне широк и может быть фиксирован весьма частично.

Для операций чтения и записи можно выделить две характеристики: уровень осуществления операции и период осуществления операции.

Под уровнем осуществления операции будем понимать, с помощью какого модуля операционной системы противник осуществляет конкретную операцию. Выделяются две основные возможности.

— *Операции уровня FDE-интерфейса.* Противник может сделать запрос к FDE-модулю на чтение или запись данных защищаемого диска (любых данных в любые секторы диска). На практике доступ такого типа к диску противник может получить, например, навязав легальному пользователю сохранение некоторого файла на диске вместо другого файла, расположение которого на этом диске противнику известно.
— *Операции уровня Raw-интерфейса.* Противник осуществляет операции с помощью драйвера диска, которым пользуется и сам FDE-модуль для сохранения на диск всех необходимых данных. Так, с помощью этого интерфейса противник может прочитать или переписать зашифрованные секторы или области служебных данных FDE-модуля. Наиболее распространённым в литературе по данной тематике является практический пример с кражей защищаемого диска — в данном случае противник может считывать и записывать информацию диска, взаимодействуя с ним как с незащищённым.

Заметим, что FDE-интерфейс, в отличие от Raw-интерфейса, не позволяет противнику получить прямой доступ к служебным данным FDE-модуля.

В результате доступа к обозначенным интерфейсам противник может навязывать открытые данные с помощью FDE-интерфейса и считывать через Raw-интерфейс информацию о полученных шифртекстах и имитовставках, а также записывать на диск подобранные шифртексты и, например, имитовставки с помощью Raw-интерфейса и получать соответствующий результат расшифрования через FDE-интерфейс. Это является практическим отражением возможностей противника, предоставляемых ему формальными моделями типа CPA (Chosen Plaintext Attack) и CCA (Chosen Ciphertext Attack) [9, 16].

Период осуществления операции — это то время, когда у противника есть возможность осуществить операцию. Так, делать операции с помощью FDE-интерфейса противник может только на этапах Session и, возможно, Emergency.

В случае Raw-интерфейса обычно предполагается, что противник может осуществлять Raw-чтение в любой момент времени и Raw-запись данных, когда FDE-модуль не работает. При этом с возможностью Raw-записи во время работы FDE-модуля ситуация не так однозначна.

Это объясняется тем, что такая возможность может существенно усложнить задачу обеспечения целостности хранимых данных. Например, если у противника этой возможности нет, то для дисков малого объёма целостность может быть проверена на этапе Mount, что позволит исключить часть операций на этапе Session, от которого обычно требуется максимальная производительность. Если противник может записывать через Raw-интерфейс на этапе Session, то предвычисления на этапе Mount теряют смысл, так как целостность может быть нарушена в любой момент после этапа Mount. Стоит заметить, что на практике такая возможность противника является довольно сильной, а более реалистичны сценарии, когда противник получает полный доступ к диску не во время работы с ним легитимного пользователя, а между сеансами, например в результате обхода парольной системы аутентификации операционной системы.

Атаки, связанные со сбоями. Специфическим аспектом функционирования FDE-модулей являются различные сбои, некоторые из которых рассмотрены в п. 2.2. При этом нельзя исключать, что противник может использовать сбои для нарушения целевых свойств безопасности системы, поэтому это необходимо учитывать в рамках криптографического анализа FDE-модулей. Заметим, что при формировании модели противника для FDE-модуля могут учитываться и задействоваться следующие приёмы и организационные меры, которые позволяют ограничить возможности противника по провоцированию сбоев:

— особые требования к хранению диска после отключения питания до начала этапа Recovery для исключения доступа к нему противника;
— требование использования источников бесперебойного питания при работе FDE-модулей;
— использование счётчика числа сбоев и требование смены ключевого материала и нового прохождения этапа Setup по достижении им порогового значения.

Модель угрозы

Целевыми свойствами безопасности, для обеспечения которых предназначен FDE-модуль как криптографический механизм, являются конфиденциальность и целостность хранимых данных. Хотя данные свойства в криптографии стандартны и достаточно глубоко исследованы, специфика задачи защиты хранимых данных приводит к появлению у них новых аспектов и особенностей.

Прежде чем переходить к подробному рассмотрению каждого из указанных свойств, необходимо более явно обозначить объект, для которого эти свойства необходимо обеспечивать. Неформально, под защищаемыми данными подразумеваются все прикладные данные, которые записываются на диск через FDE-модуль с момента его инициализации. Можно выделить два подхода к представлению совокупности этих данных в виде потоков сообщений:

1) Представление в виде одного потока сообщений, где одно сообщение — это совокупность всех прикладных данных, записанных на диск через FDE-модуль после очередной операции записи сектора, инициированной файловой системой. Одно сообщение удобно представлять в виде упорядоченного набора, где элементом с номером i являются прикладные данные, записанные в i-й сектор. Такой набор также можно сравнить с «моментальным снимком», где сохранилось состояние данных диска, которое было неизменным в течение некоторого промежутка времени, т. е. каждый следующий «снимок» отличается от предыдущего ровно в одном секторе. Принцип данного подхода представлен на рис. 3.

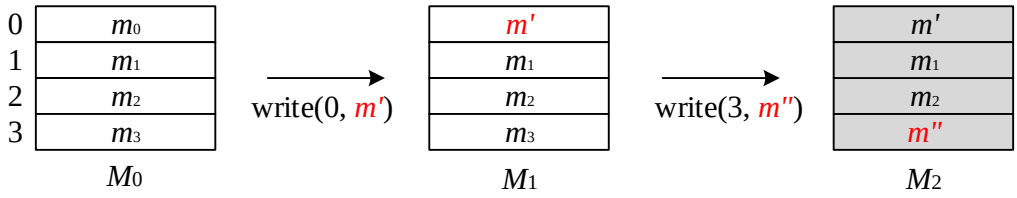

Рис. 3. Представление защищаемых данных в виде одного потока сообщений

2) Представление в виде фиксированного количества независимых потоков сообщений, каждый из которых соответствует конкретному прикладному сектору. В этом случае одним сообщением в конкретном потоке являются прикладные данные, которые записываются в соответствующий сектор, а каждое новое сообщение в этом потоке возникает после очередной операции записи именно в этот сектор. Данный подход проиллюстрирован на рис. 4.

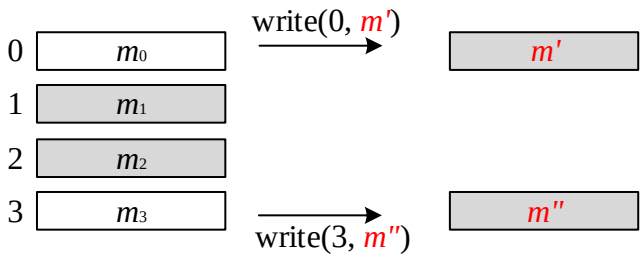

Рис. 4. Представление защищаемых данных в виде набора потоков сообщений

Далее, говоря про конфиденциальность или целостность, будем иметь в виду конфиденциальность или целостность одного потока сообщений, каждое из которых является либо совокупностью всех данных диска, либо данными конкретного сектора. При

этом в зависимости от того, что именно является сообщением, то или иное свойство безопасности может иметь различный практический смысл. Поэтому для различения градаций какого-либо свойства будем говорить, что это свойство обеспечивается *на уровне диска*, если сообщением является совокупность всех прикладных данных, или *на уровне сектора*, если сообщением являются прикладные данные конкретного сектора. Рассмотрим подробнее каждое из целевых свойств безопасности.

Конфиденциальность. Это является основным свойством безопасности хранимых на диске прикладных данных, которое должен обеспечивать FDE-модуль. Отметим особенность обеспечения его в контексте полнодискового шифрования. На практике на защищаемом диске могут храниться как важные документы, содержимое которых должно оставаться в секрете многие годы, так и открытые данные или данные, которые становятся публичными по прошествии небольшого промежутка времени. Таким образом, в одном и том же секторе диска в разные моменты времени, а также одновременно в разных секторах диска может быть записана информация, к которой предъявляются различные требования по обеспечению конфиденциальности.

Под конфиденциальностью хранимых на диске прикладных данных, представленных в виде одного или нескольких потоков сообщений, будем неформально понимать невозможность раскрытия противником какой-либо информации об этих сообщениях. Ориентиром при формализации этого свойства могут служить модели на основе неотличимости (например, IND-CPA [16]). В зависимости от выбранного представления диска (т. е. от того, что именно понимается под сообщением) некоторая информация о прикладных данных может заведомо раскрываться противнику. Выделим следующие степени конфиденциальности хранимых данных по характеру доступной противнику информации:

— Конфиденциальность на уровне диска. В этом случае противник не получает никакой информации о хранимых на диске прикладных данных. Уточним, что, помимо неполучения информации о самих хранимых в секторах данных, противник также не должен узнавать номер изменяемого сектора. Обеспечение такого свойства представляется крайне трудозатратным — одним из подходов является перешифрование всего диска после операции записи любого сектора. Это неэффективно в случае дисков большого размера и не позволяет параллельно обрабатывать операции с разными секторами.
— Конфиденциальность на уровне сектора. В этом случае противник не получает никакой информации о прикладных данных, хранимых в конкретном секторе. Отметим, что обеспечение конфиденциальности на уровне сектора для всех прикладных секторов диска неэквивалентно обеспечению конфиденциальности на уровне диска. В сравнении с предыдущим свойством у противника появляется информация о том, в какие именно секторы осуществляется обращение. Несмотря на то, что схемы, обеспечивающие конфиденциальность на уровне сектора, являются более эффективными, такой конфиденциальности не всегда достаточно. Так, например, если на защищаемом диске хранится карта военных действий, а каждый сектор соответствует определённому пункту местности, раскрытие информации о том, с какими секторами идёт работа, может быть критичным.

Необходимым условием обеспечения полноценной конфиденциальности является увеличение размера шифртекста по сравнению с размером открытого текста (например, за счёт вектора инициализации, см. подробнее [16]), что возможно только при наличии дополнительной памяти для хранения служебных данных.

Если использование дополнительной памяти невозможно, наилучшей степенью конфиденциальности, которую удастся обеспечить, является так называемая «конфиденциальность по модулю повторений» [17]. В этом случае противник дополнительно детектирует факт совпадения сообщений в потоке. Иногда раскрытие такой информации может быть критичным. Предположим, например, что каждый день в определённый сектор на диске записывается актуальный на следующий день курс акций компании. Эта информация публикуется на следующий день, а до этого момента должна оставаться в секрете. Если противник устанавливает факт совпадения актуальных данных сектора с данными, записанными некоторое время назад, то он немедленно получает информацию о курсе акций на завтрашний день.

В предыдущем разделе отмечено, что противник имеет возможность модифицировать данные на диске. Это означает, что помимо традиционных атак (навязывание противником открытого текста), при анализе свойств безопасности FDE-модуля должны учитываться атаки, опирающиеся на возможность противника записывать на диск специальным образом подобранные шифртексты. Рассмотрим несложный пример такой атаки.

Предположим, что в некоторый момент времени в определённый сектор была записана критически важная информация, конфиденциальность которой необходимо обеспечивать в течение большого промежутка времени. Предположим также, что далее в этот же сектор была записана менее важная информация, результат расшифрования которой противнику может быть доступен; например, это может быть текст отчёта, который становится известным в короткие сроки. Допустим, что через некоторое время после записи менее секретных данных у противника появляется возможность недетектируемо заменить соответствующий им шифртекст на корректный шифртекст, соответствующий критически важной информации. Тогда при опубликовании отчёта критически важная информация будет раскрыта.

Опишем один из подходов к осмыслению потенциальных возможностей противника в рассмотренной атаке. Введём градации свойства конфиденциальности по времени, в течение которого неактуальные данные являются уязвимыми. Под актуальными данными будем понимать последнее сообщение в потоке, а моментом неактуальности данных будем называть время появления следующего за ними сообщения в потоке. Для конкретных данных (сообщения в потоке) определим свойство *конфиденциальности по модулю* δ следующим образом. В течение промежутка времени δ с момента неактуальности данные являются уязвимыми и нарушение их конфиденциальности не считается угрозой, в остальное время их конфиденциальность обеспечивается. Рассмотрим некоторые частные случаи:

— $\delta = 0$. Конфиденциальность данных, записанных на диск в разные моменты времени, обеспечивается в течение всего времени работы FDE-модуля, т. е. угрозой считается получение противником информации о произвольном сообщении в потоке независимо от того, как давно оно было записано;
— $\delta > 0$. Данные являются уязвимыми в течение промежутка времени δ с момента неактуальности. При этом получение противником информации об актуальных данных, а также о «старых» неактуальных данных считается угрозой. Отметим, что при $\delta = \infty$ угрозой является лишь получение противником информации об актуальных данных.

Таким образом, чем меньше значение δ, тем меньше промежуток времени, в течение которого данные являются уязвимыми, а потому тем сильнее свойство конфиденциальности по модулю δ.

Вернёмся к атаке. Если противник смог недетектируемо заменить содержимое сектора спустя промежуток времени t с момента неактуальности секретных данных, то он нарушил свойство конфиденциальности по модулю δ для всех $\delta \leqslant t$ (в частности, конфиденциальности по модулю 0). Однако конфиденциальность по модулю ∞ в этом случае не нарушена, так как по построению атаки противник не получает информации об актуальных данных соответствующего сектора.

Мы вводим свойство конфиденциальности по модулю δ таким образом, что его нарушение становится возможным, только если противник осуществляет атаки, подразумевающие модификацию данных на диске. Поэтому обеспечение конфиденциальности по модулю δ свидетельствует о том, что противник имеет возможность недетектируемо модифицировать данные на диске только в течение промежутка времени δ с момента их неактуальности.

Если схема полнодискового шифрования обеспечивает конфиденциальность по модулю $\delta > 0$, достичь конфиденциальности по модулю 0 можно с помощью организационных мер. Например, можно потребовать, чтобы после каждого выключения FDE-модуля пользователь контролировал диск в течение промежутка времени δ, не допуская тем самым возможности противника модифицировать данные на нём.

Целостность. Под целостностью данных понимается невозможность внесения противником недетектируемых изменений в данные, записанные на диск. Важность обеспечения данного свойства часто недооценивают, что может быть связано со следующей особенностью задачи защиты дисков. В отличие от задачи защиты канала связи между двумя пользователями, в данном случае «отправляющая» (записывающая) и «принимающая» (считывающая) стороны являются одним и тем же объектом — файловой системой, за которой чаще всего стоит человек. Поэтому потенциально человек способен обнаружить изменения в считываемых данных, так как сам когда-то записывал их на диск через FDE-модуль. Однако в реальности на диске могут храниться как очень большие объёмы данных, следить за неизменностью которых человек просто не способен, так и служебная информация приложений, которая человеку как пользователю неизвестна. Ярким примером серьёзной уязвимости, которая может появиться при отсутствии целостности, является возможность незаметно внести изменения в хранящийся на диске исходный код некоторой программы, которые не повлияют на её компилируемость, но приведут к критичному изменению функционирования.

Принципиальным моментом в обеспечении целостности хранимых данных является обеспечение их «актуальности», т. е. легитимный пользователь должен быть уверен, что при чтении он получит те же данные, которые были сохранены на диск в результате предыдущей операции записи. С точки зрения представления данных в виде потока сообщений целостность означает целостность именно последнего (актуального) сообщения в потоке в каждый момент времени. Проводя аналогию с каналами связи, ситуация, когда данные сектора перестают быть актуальными, соответствует ситуации, когда сообщение либо так и не дошло до получателя, либо уже им прочитано. Соответственно изменение этого сообщения противником не имеет смысла.

Как и для конфиденциальности, в зависимости от точки зрения на защищаемые данные смысл целостности может отличаться:

1) Целостность на уровне диска. Под данным свойством понимается невозможность внесения каких-либо изменений в данные диска как упорядоченного набора, в том числе перемещение данных из одного сектора в другой.
2) Целостность на уровне сектора. Под данным свойством подразумевается невозможность внесения каких-либо изменений в данные конкретного сектора (в том числе невозможность замены данных этого сектора на данные других секторов). Отметим, что в отличие от конфиденциальности обеспечение целостности на уровне сектора для каждого сектора эквивалентно обеспечению целостности на уровне диска.

Обеспечение целостности невозможно без увеличения длины шифртекста относительно длины открытого текста (подробнее см. [7]), что влечёт необходимость наличия места для хранения служебных дополнительных данных, например имитовставок, и как следствие — к усложнению FDE-модулей и замедлению скорости обработки данных. Поэтому некоторые FDE-модули обеспечивают так называемую «псевдоцелостность», не требующую выделения дополнительного места на диске. Под псевдоцелостностью понимается невозможность внесения контролируемых изменений в сообщение, т. е. любые изменения шифртекста приведут к тому, что расшифрованный текст будет выглядеть как случайное сообщение. Псевдоцелостность на уровне диска отличается от псевдоцелостности на уровне сектора: в первом случае изменение шифртекста любого сектора должно приводить к непредсказуемым изменениям всех прикладных данных, а во втором — к непредсказуемым изменениям только прикладных данных, хранящихся в соответствующем изменённом секторе. Хотя данное свойство не является целостностью в стандартном смысле, его обеспечение также можно отнести к защите от недетектируемого изменения данных. Действительно, чем к большим изменениям прикладных данных приводит изменение шифртекста, тем больше шансов, что такое изменение приведёт к ошибке на прикладном уровне и, таким образом, не останется незамеченным.

Для целостности, как и для конфиденциальности, можно ввести градацию по степени обеспечения актуальности данных с помощью понятия *целостности по модулю* δ. Неформально данное свойство означает, что противник может недетектируемо заменить актуальные данные только на те данные, которые хранились на диске не более чем δ единиц времени назад. С точки зрения потока сообщений оно означает возможность противника заменить последнее в потоке сообщение на любое из нескольких предыдущих, и только на него. Частные случаи:

— $\delta = 0$. Гарантируется, что при чтении пользователь получает доступ к тем же данным, которые были записаны им при последних обращениях к секторам, и угрозой считается внесение любых изменений в актуальное сообщение. Обеспечение целостности по модулю 0 на уровне сектора для всех прикладных секторов эквивалентно обеспечению целостности по модулю 0 на уровне диска. Необходимым эксплуатационным условием для обеспечения данного свойства является наличие защищённой памяти, модифицировать которую противник не может, так как данное свойство связано с необходимостью передачи некоторого внутреннего состояния от сессии к сессии [7];
— $\delta > 0$. Угрозой считается замена «актуальных» сообщений на достаточно «старые» сообщения потока или на новые сообщения, которых в потоке нет. При $\delta > 0$ обеспечение целостности по модулю δ на уровне диска неэквивалентно обеспечению целостности по модулю δ на уровне сектора для всех прикладных секторов. Дей-

ствительно, при обеспечении целостности по модулю δ на уровне диска у противника появляется лишь возможность недетектируемо заменять данные всех секторов на диске на данные, которые одновременно хранились в этих секторах диска в некоторый предыдущий момент времени. При обеспечении целостности по модулю δ на уровне сектора у противника остаётся возможность «откатывать» данные различных секторов к неактуальному состоянию независимо друг от друга, т. е. противник может привести диск в новое состояние, комбинируя данные, которые хранились в различных секторах в разные моменты времени. При $\delta = \infty$ актуальность данных не обеспечивается и угрозой считается только недетектируемая замена актуального сообщения на сообщения, которые никогда не встречались в потоке (не хранились в секторе или на диске).

Необходимо отметить, что хотя целостность по модулю δ сама по себе может быть важным свойством безопасности, её наличие позволяет также обеспечить конфиденциальность в условиях, когда противник может изменять данные на диске. Действительно, для обеспечения конфиденциальности по модулю δ достаточно обеспечить конфиденциальность по модулю ∞ и целостность по модулю γ, где $\gamma \leqslant \delta$.

2.4. Некоторые типичные сценарии использования

Нельзя исключать, что добиться приемлемых значений всех перечисленных характеристик для всех возможных сценариев использования защищённого диска невозможно. Решением проблемы является создание нескольких FDE-схем, каждая из которых оптимизирована для конкретных более узких условий эксплуатации. Далее мы, не претендуя на полноту, описываем некоторые типичные на сегодняшний день сценарии использования защищённых дисков и их особенности, которые могут облегчить процесс принятия синтезных решений.

Персональный съёмный диск. Примером такого диска является USB-флэш-диск, используемый для хранения конфиденциальной информации личного и/или рабочего характера. Особенности работы с диском такого типа:

— относительно небольшой объём защищаемых данных (до 256 Гбайт);
— небольшой объём оперативной памяти машин, на которых осуществляется работа с диском (например, единицы гигабайт);
— относительно непродолжительные сеансы работы и большое количество операций начала и завершения сессии; например, пользователь может подключать диск к машине в начале рабочего дня и отключать в конце;
— небольшой совокупный размер данных, с которыми производятся операции в течение сеанса или, по крайней мере, продолжительного промежутка времени в рамках сеанса. Так, обычно пользователь оперирует только с данными, относящимися к актуальному проекту, а данные завершённых проектов не задействуются;
— ненулевая вероятность сбоев всех типов. Так, в случае частого использования возможно отключение пользователем диска от рабочей машины без процедуры завершения сеанса.

Хранилища центров обработки данных. Этот сценарий предполагает защищённое хранение огромного объёма данных, доступ к которым серверная операционная система одновременно предоставляет большому числу клиентов. Особенности работы с диском в рамках такого сценария:

— большой объём защищаемых данных (от 2 Тбайт);
— большой объём оперативной памяти сервера, осуществляющего работу с диском;

— малое число операций начала и завершения сеанса работы и крайне продолжительные сессии;
— большой объём данных, с которыми производятся операции в течение даже не самого продолжительного периода времени; при этом расположение запрашиваемых секторов имеет преимущественно случайный характер;
— минимальная вероятность сбоев. Дублирование хранения обеспечивает защиту от ошибок на физическом уровне и от внезапных потерь доступа к диску; использование систем бесперебойного питания минимизирует риск внезапного отключения питания машины.

Заключение

В работе описаны и проанализированы основные эксплуатационные и криптографические свойства FDE-схем защищённого хранения данных в современных операционных системах, а также особенности некоторых типичных сценариев использования защищённых с помощью FDE-схем дисков. Эти сведения могут быть использованы при проектировании FDE-схем.

При этом стоит отметить, что приведённые рассуждения о криптографических свойствах таких систем нацелены на их неформальное понимание и могут служить лишь ориентиром при формальном задании модели противника (например, на основе подхода, описанного в [9]) и последующей оценке стойкости таких схем.

Проведённый анализ вытекающих из практики требований к режимам полнодискового шифрования является необходимым для формирования научно-технической базы синтеза таких режимов. Начатые в настоящей работе исследования продолжатся в направлении анализа как существующих режимов, так и предлагаемых к стандартизации в Техническом комитете «Криптографическая защита информации» (ТК 26).

Авторы выражают глубокую благодарность своим коллегам Л. О. Никифоровой, А. А. Русеву, Е. С. Грибоедовой, Л. А. Сониной и Д. А. Щербакову за плодотворные обсуждения, ценные замечания и конструктивную критику.

ЛИТЕРАТУРА

1. Рекомендации по стандартизации Р 1323565.1.012-2017 «Информационная технология. Криптографическая защита информации. Принципы разработки и модернизации шифровальных (криптографических) средств защиты информации». М.: Стандартинформ, 2017.
2. Об утверждении Требований к средствам электронной подписи и Требований к средствам удостоверяющего центра. Приказ ФСБ РФ от 27 декабря 2011 г. № 796.
3. *Зима В. М., Клюев А. В., Литвинов О. А. и др.* Основы защиты информации от несанкционированного доступа в автоматизированных системах конфиденциального делопроизводства // Тр. СПИИРАН. 2006. Вып. 3. Т. 2. С. 84–95.
4. *Khati L.* Full Disk Encryption and Beyond. Diss. Cryptography and Security [cs.CR]. Universite PSL; ENS Paris — Ecole Normale Superieure de Paris, 2019. 182 p.
5. *Broz M.* Authenticated and Resilient Disk Encryption. PhD thesis. Brno: Masaryk University, 2018.
6. *Damgard I. and Dupont K.* Universally Composable Disk Encryption Schemes. IACR Cryptology ePrint Archive. 2005. `https://eprint.iacr.org/2005/333.pdf`.
7. *Gjosteen K.* Security notions for disk encryption // LNCS. 2005. V. 3679. P. 455–474.
8. `https://integralmemory.com`.

9. *Алексеев Е. К., Ахметзянова Л. Р., Зубков А. М. и др.* Об одном подходе к формализации задач криптографического анализа // Матем. вопр. криптогр. 2020 (в печати).
10. *Алексеев Е. К., Ахметзянова Л. Р., Карпунин Г. А. и др.* Что плохого можно сделать, неправильно используя криптоалгоритмы? Доклад на лектории симпозиума CTCrypt'2019. https://ctcrypt.ru/files/files/2019/materials/29_Alekseyev.pdf.
11. *Bhargavan K. and Leurent G.* On the practical (in-)security of 64-bit block ciphers. Collision attacks on HTTP over TLS and OpenVPN // Proc. CCS'16, October 24–28, 2016, Vienna, Austria. P. 456–467. https://sweet32.info/SWEET32_CCS16.pdf.
12. *Smyshlyaev S.* Re-keying Mechanisms for Symmetric Keys. RFC 8645. August 2019. https://tools.ietf.org/html/rfc8645.
13. *Akhmetzyanova L. R., Alekseev E. K., Oshkin I. B., and Smyshlyaev S. V.* Increasing the Lifetime of Symmetric Keys for the GCM Mode by Internal Re-keying. Cryptology ePrint Archive: Report 2017/697.
14. *Алексеев Е. К., Ахметзянова Л. Р., Мешков Д. А. и др.* О нагрузке на ключ. Ч. 1. Блог ООО «КРИПТО-ПРО». 2017. http://cryptopro.ru/blog/2017/05/17/o-nagruzke-na-klyuch-chast-1.
15. *Алексеев Е. К., Ахметзянова Л. Р., Мешков Д. А. и др.* О нагрузке на ключ. Ч. 2. Блог ООО «КРИПТО-ПРО». 2017. http://cryptopro.ru/blog/2017/05/29/o-nagruzke-na-klyuch-chast-2.
16. *Bellare M. and Rogaway P.* Introduction to Modern Cryptography. 2005. https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf.
17. *Bellare M., Boldyreva A., and O'Neill A.* Deterministic and efficiently searchable encryption // LNCS. 2007. V. 4622. P. 535–552.

REFERENCES

1. Informacionnaya tekhnologiya. Kriptograficheskaya zashchita informacii. Principy razrabotki i modernizacii shifroval'nyh (kriptograficheskih) sredstv zashchity informacii [Information Technology. Cryptographic Data Security. Principles of Creation and Modernization for Cryptographic Modules. Recommendations for Standardization R 1323565.1.012-2017]. Moscow, Standartinform Publ., 2017. (in Russian)
2. Ob utverzhdenii Trebovanij k sredstvam elektronnoj podpisi i Trebovanij k sredstvam udostoveryayushchego centra [On Approval of the Requirements for Cryptographic Modules for Digital Signature and Certificate Authority]. Order of the Federal Security Service of the Russian Federation of December 27, 2011 No. 796. (in Russian)
3. *Zima V. M., Kljuev A. V., Litvinov O. A., et al.* Osnovy zashchity informatsii ot nesanktsionirovannogo dostupa v avtomatizirovannykh sistemakh konfidentsial'nogo deloproizvodstva [Basics of protection of the information from unauthorized access in the automated systems of confidential office-work]. Tr. SPIIRAN, 2006, iss. 3, vol. 2, pp. 84–95. (in Russian)
4. *Khati L.* Full Disk Encryption and Beyond. Diss. Cryptography and Security [cs.CR], Universite PSL; ENS Paris — Ecole Normale Superieure de Paris, 2019. 182 p.
5. *Broz M.* Authenticated and Resilient Disk Encryption. PhD thesis, Brno, Masaryk University, 2018.
6. *Damgard I. and Dupont K.* Universally Composable Disk Encryption Schemes. IACR Cryptology ePrint Archive, 2005. https://eprint.iacr.org/2005/333.pdf.
7. *Gjosteen K.* Security notions for disk encryption. LNCS, 2005, vol. 3679, pp. 455–474.
8. https://integralmemory.com.

9. *Akhmetzyanova L., Alekseev E., Karpunin G., et al.* Ob odnom podkhode k formalizatsii zadach kriptograficheskogo analiza [On one approach to formalizing cryptographic analysis tasks]. Matem. Vopr. Kriptogr., 2020, to be published. (in Russian)
10. *Akhmetzyanova L., Alekseev E., Karpunin G., et al.* Chto plokhogo mozhno sdelat', nepravil'no ispol'zuya kriptoalgoritmy? [What can be done wrong by using cryptographic algorithms incorrectly?]. CTCrypt'2019. `https://ctcrypt.ru/files/files/2019/materials/29_Alekseyev.pdf`. (in Russian)
11. *Bhargavan K. and Leurent G.* On the practical (in-)security of 64-bit block ciphers. Collision attacks on HTTP over TLS and OpenVPN. Proc. CCS'16, October 24–28, 2016, Vienna, Austria, pp. 456–467. `https://sweet32.info/SWEET32_CCS16.pdf`.
12. *Smyshlyaev S.* Re-keying Mechanisms for Symmetric Keys. RFC 8645, August 2019. `https://tools.ietf.org/html/rfc8645`.
13. *Akhmetzyanova L. R., Alekseev E. K., Oshkin I. B., and Smyshlyaev S. V.* Increasing the Lifetime of Symmetric Keys for the GCM Mode by Internal Re-keying. Cryptology ePrint Archive: Report 2017/697.
14. *Alekseev E. K., Akhmetzyanova L. R., Meshkov D. A., et al.* O nagruzke na klyuch. Ch. 1 [On Key Lifetime. P. 1]. CRYPTO-PRO LLC Blog, 2017. `http://cryptopro.ru/blog/2017/05/17/o-nagruzke-na-klyuch-chast-1`. (in Russian)
15. *Alekseev E. K., Akhmetzyanova L. R., Meshkov D. A., et al.* O nagruzke na klyuch. Ch. 2 [On Key Lifetime. P. 2]. CRYPTO-PRO LLC Blog, 2017. `http://cryptopro.ru/blog/2017/05/29/o-nagruzke-na-klyuch-chast-2`. (in Russian)
16. *Bellare M. and Rogaway P.* Introduction to Modern Cryptography. 2005. `https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf`.
17. *Bellare M., Boldyreva A., and O'Neill A.* Deterministic and efficiently searchable encryption. LNCS, 2007, vol. 4622, pp. 535–552.