Информационная безопасность цифрового пространства: коллективная монография /под ред. Е.В. Стельмашонок, И.Н. Васильевой. – СПб.: Изд-во СПбГЭУ, 2019. (155 с.) – С.109-124.

3.3. Анализ криптографических средств операционных систем Windows

Большинство современных компьютерных систем, таких как операционные системы (ОС), web-серверы и СУБД, обладают встроенными средствами криптографической защиты информации (СКЗИ) для обеспечения безопасного хранения и защищенной передачи данных. В качестве примеров подобных средств можно привести функции шифрования файлов и папок Encryption File System (EFS) или средство шифрования диска BitLocker в операционных системах семейства Windows. При этом большинство пользователей уверено, что применяемые средства защиты обладают абсолютной надежностью за счет реализации стойких криптографических алгоритмов, таких как AES или отечественные криптоалгоритмы ГОСТ. Как правило, большое внимание уделяется выбору длины ключа алгоритма шифрования, например, даются рекомендации использовать алгоритм AES с длиной ключа 256 бит (AES-256) вместо AES-128.

Действительно, используемые современные криптографические алгоритмы, такие как AES, 3DES, отечественные ГОСТ Р 34.12–2015 «Кузнечик» и «Магма», обладают достаточной вычислительной стойкостью. Большие длины ключей (от 128 бит) исключают практическую возможность вскрытия методом полного перебора значений ключа. Вместе с тем, успешные атаки на зашифрованную информацию в ряде случаев все же возможны и обусловлены, прежде всего, непониманием особенностей и ограничений, некорректной настройкой или неправильным использованием, а иногда и уязвимостями реализации СКЗИ.

Представляет интерес исследование вопроса практической надежности популярных криптографических средств защиты информации с точки зрения форензики – компьютерной криминалистики, нацеленной на поиск и извлечение информации (доказательств) из компьютерных систем [1].

Требования безопасности приложений приводят к широкому использованию шифрования, которое зачастую осуществляется приложением автоматически, прозрачно для пользователя и не требует, а иногда и не допускает пользовательской настройки параметров. Так, например, популярные браузеры сохраняют данные учетных записей пользователей (логины и пароли) для доступа к сайтам, сессии и файлы cookies, служащие источником личной информации, в зашифрованном виде. Для защиты данных браузеры Chrome и Edge (IE) в операционной системе Windows используют интерфейс DPAPI (Data Protection API), позволяющий реализовать функции шифрования и расшифрования как данных, так и памяти. При этом, как правило, ключ шифрования генерируется на основе пользовательского пароля,

используемого для входа в Windows. Механизмы DPAPI не обеспечивают абсолютно надежной защиты, что подтверждается опытом использования криминалистического программного обеспечения, например, Belcasoft Evidence Center [2].

Еще один пример – хранение синхронизированных данных и резервных копий информации с устройств, работающих под управлением операционных систем Android, iOS и Windows 10 в облачных хранилищах: Google Account, iCloud и Microsoft Account соответственно. Полученные с устройства облачные данные в ряде случаев никак не контролируются пользователем и могут содержать пароли, ключи доступа к зашифрованным томам, а также личную информацию (переписку, контакты, данные о звонках, геолокационные данные и прочее). Извлечение таких данных позволяет получить доступ к другим устройствам и хранилищам. Однако сама процедура извлечения данных из облака является нетривиальной, поскольку разработчики используют шифрование, причем некоторые данные, например, пароли в резервных копиях, дополнительно шифруются аппаратным ключом, другие допускают обращения только с доверенного устройства. Вместе с тем, существуют криминалистические инструменты, справляющиеся с данной задачей, например, Elcomsoft Cloud Explorer, Elcomsoft Phone Breaker [3].

Для шифрования хранимых данных пользователем могут быть использованы встроенные СКЗИ, доступные в операционных системах Windows [4] и предназначенные для локального шифрования объектов файловой системы (EFS) и дисков (BitLocker, BitLocker Drive Encryption, BitLocker To Go). EFS фактически является надстройкой над файловой системой NTFS, позволяющей обеспечить защиту данных как во время работы ОС, так и в автономном режиме (при прямом физическом доступе к диску, с другого экземпляра ОС). EFS-шифрование прозрачно для пользователя, то есть при чтении файла данные автоматически расшифровываются, а при записи снова зашифровываются. Другие же пользователи в активной ОС не смогут открыть зашифрованный файл, а при прямом доступе к диску – прочитать его содержимое.

Поддержка EFS встроена в NTFS-драйвер. При обнаружении зашифрованного файла NTFS выполняет встроенные EFS-функции, осуществляющие шифрование и расшифровывание данных по мере обращения к ним (рис. 3.7). NTFS отправляет запросы к EFS-службе через подсистему локальной аутентификации (Local Security Authority Subsystem, LSASS).

EFS применяет симметричное шифрование, по умолчанию используется AES-256. Затем к ключу симметричного шифра применяется ассиметричное шифрование (RSA-2048). Начиная с Windows7 EFS наряду с RSA

поддерживает использование шифрования на эллиптических кривых (для соответствия требованиям к криптосистемам для гос. учреждений США – Suite B). Для работы с EFS применяется API-интерфейс CNG (Cryptography Next Generation), соответственно, существует возможность воспользоваться любым криптографическим алгоритмом, поддерживаемым CSP, поставляемыми вместе с операционной системой, или сторонним криптопровайдером. Это означает, что EFS может использовать для шифрования локальных файлов и папок отечественные криптоалгоритмы (при установке поддерживающего криптопровайдера и расширения в систему и наличия соответствующего сертификата, например, КриптоПро CSP и КриптоПро EFS [5, С.72-79]).

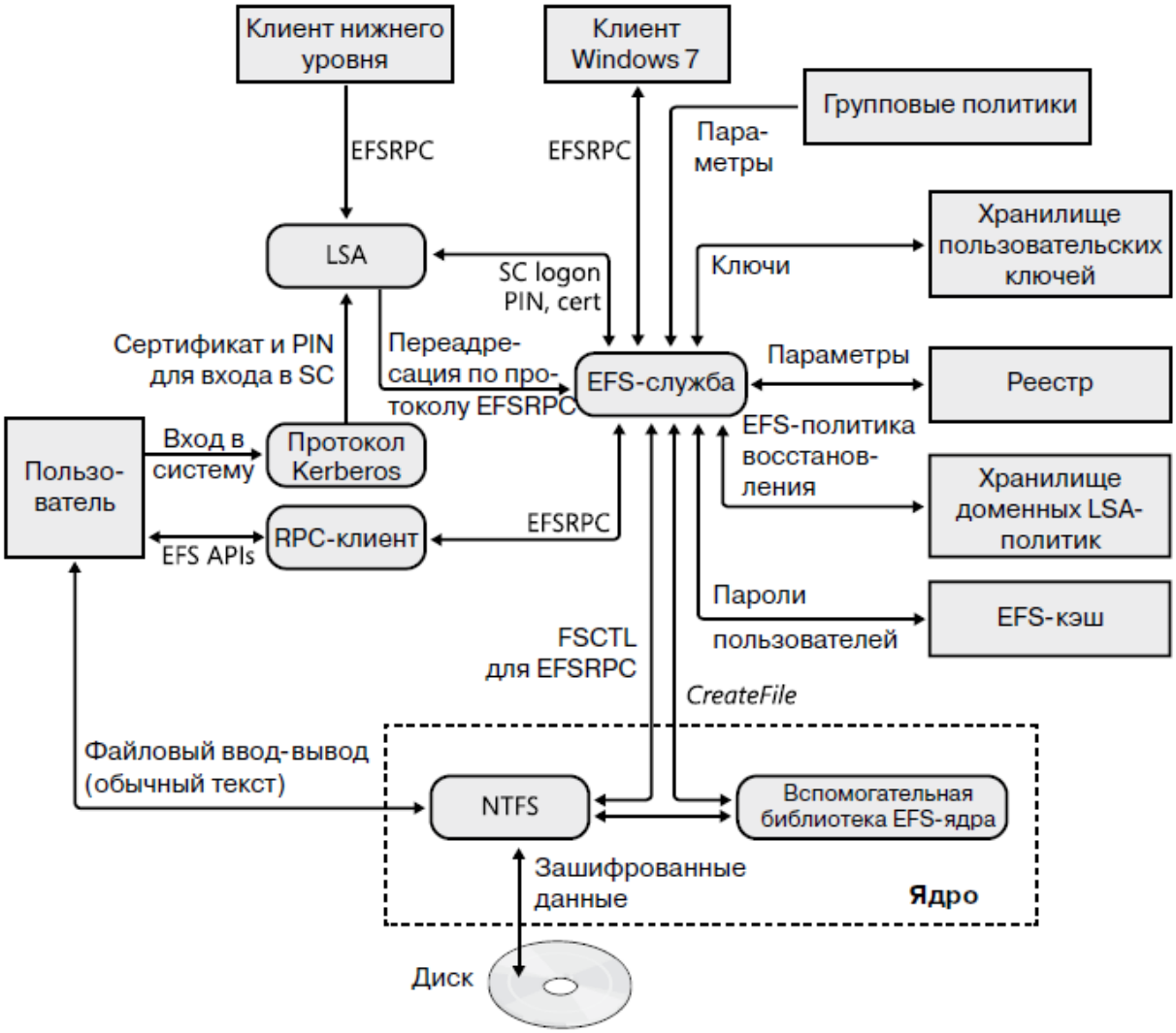

Рисунок 3.7. Архитектура EFS

Процесс шифрования выполняется следующим образом:

1. EFS-служба открывает файл для монопольного доступа.

2. Все потоки данных в файле копируются во временный текстовый файл, расположенный в системной папке.
3. Локальная подсистема безопасности LSA (Local Security Authority) генерирует случайное число – секретный ключ шифрования файла FEK (File Encryption Key), используемого для зашифровывания содержимого файла с помощью симметричного алгоритма.
4. Тот же секретный ключ потребуется и в процессе расшифровывания файла, поэтому ключ должен быть сохранен надежным способом (рис. 3.8). Для этих целей LSA выбирает открытый ключ сертификата пользователя и шифрует ключ FEK ассиметричным алгоритмом с использованием открытого ключа сертификата. Затем зашифрованный ключ записывается в специальное поле (Data Decryption Field, DDF) файла.

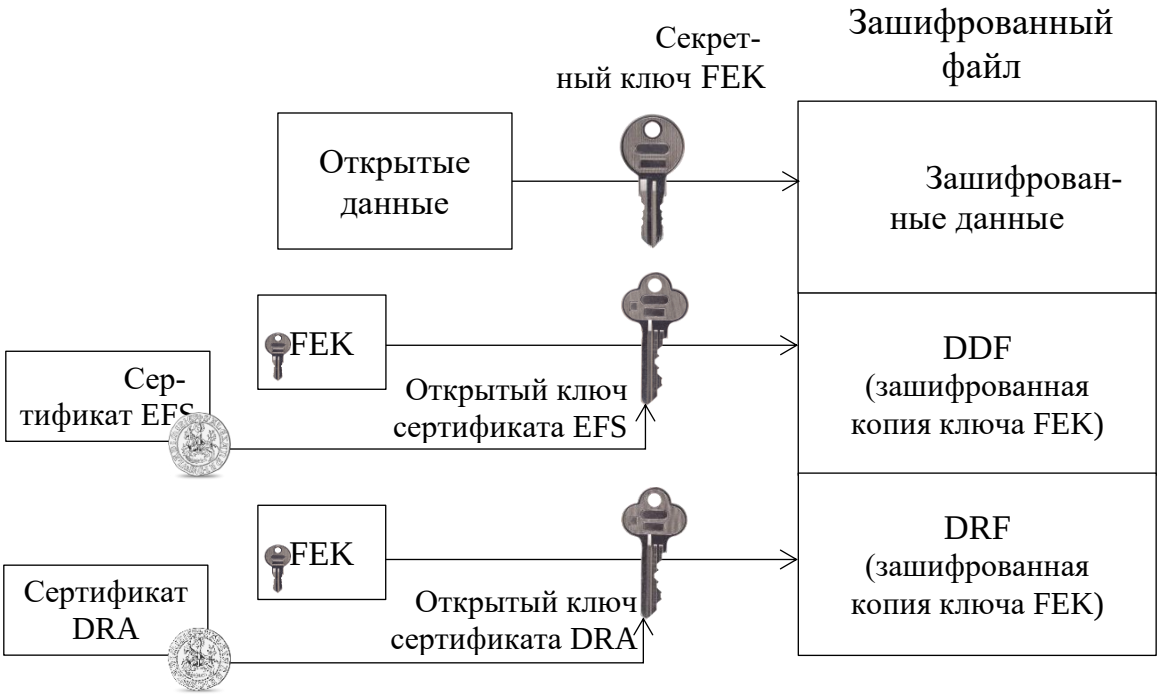

Рис. 3.8. Хранение секретного ключа шифрования EFS

В качестве источника открытого ключа может быть административно указан сертификат формата X.509, который затем будет добавлен в хранилище сертификатов пользователя, смарт-карта (USB Token) или генератор случайных чисел. Если пользователю предварительно не был назначен подходящий сертификат и не используется центр сертификации (Certification Authority, CA) в домене, то локальная машина сгенерирует самоподписанный сертификат. Если в дальнейшем пользователю будет выдан EFS-

сертификат (например, с помощью CA), шифрование все равно будет использовать созданный ранее самоподписанный сертификат. Этот сертификат не отображается в списке личных сертификатов, поэтому удалить его из системы через диспетчер сертификатов проблематично, но можно до начала шифрования EFS запретить использование самоподписанных сертификатов с помощью политики безопасности (на уровне домена).

5. Ключ FEK шифруется открытым ключом сертификата агента восстановления (Data Recovery Agent, DRA), который назначен локальной либо групповой политикой безопасности. Если политикой назначено несколько агентов восстановления, копия FEK шифруется с помощью открытого ключа каждого из них, после чего записывается в поле DRF (Data Recovery Field).

Чтобы воспользоваться зашифрованной копией ключа FEK потребуется закрытый ключ соответствующего сертификата. Таким образом гарантируется, что доступ к файлу получит только сам пользователь (владелец закрытого ключа EFS) или агенты восстановления.

6. Для полей DDF и DRF рассчитывается контрольная сумма (хэш-код), которая записывается в заголовок EFS-данных. При расшифровании она позволяет проверить целостность хранимых элементов ключей.
7. Зашифрованные данные (вместе с DDF и DRF) записывает обратно в файл.
8. Временный текстовый файл удаляется.

Начиная с Windows Server 2003, появилась возможность предоставлять локальные зашифрованные файлы в общий доступ с другими пользователями. При этом ключ FEK шифруется открытыми ключами сертификатов тех пользователей, которым разрешен доступ к зашифрованному файлу; сертификаты должны быть установлены на компьютере, на котором производится шифрование. Зашифрованный файл может быть предоставлен в общий доступ пользователям, но не группам, поскольку группы не могут иметь сертификатов. Предоставить файл в общий доступ может только тот, кто его зашифровал или агенты восстановления.

Записи обо всех имеющих доступ к файлу пользователях – совокупность элементов ключей (key entries) или связка ключей (key ring) − сохраняется в блоке DDF. Форматы EFS-данных файла и элементов ключа пеоказаны на рис. 3.9, элементы ключей DDF- и DRF- имеют единый формат.

Чтобы не допустить потенциальной кражи ключей, закрытый ключ асимметричного шифрования (сертификата EFS, агента восстановления) хранится в системе в зашифрованном виде. Шифрование закрытого ключа производится с помощью хэш-кода, полученного на основе пароля соответ-

ствующего пользователя. В свою очередь, хэш-коды паролей хранятся в системной базе пользователей, которая защищена при помощи ключа SYSKEY. Таким образом, надежность защита файла в конечном итоге определяется надежностью пользовательского пароля.

Для повышения надежности рекомендуется хранить контейнеры закрытого ключа на защищенных носителях, таких как смарт-карта или USB-токен. Сертификат может быть перенесен на защищенный носитель сразу же в процессе его создания. При отчуждении сертификата из системы доступ к зашифрованным данным становится невозможным даже в том случае, если был получен доступ к профилю пользователя.

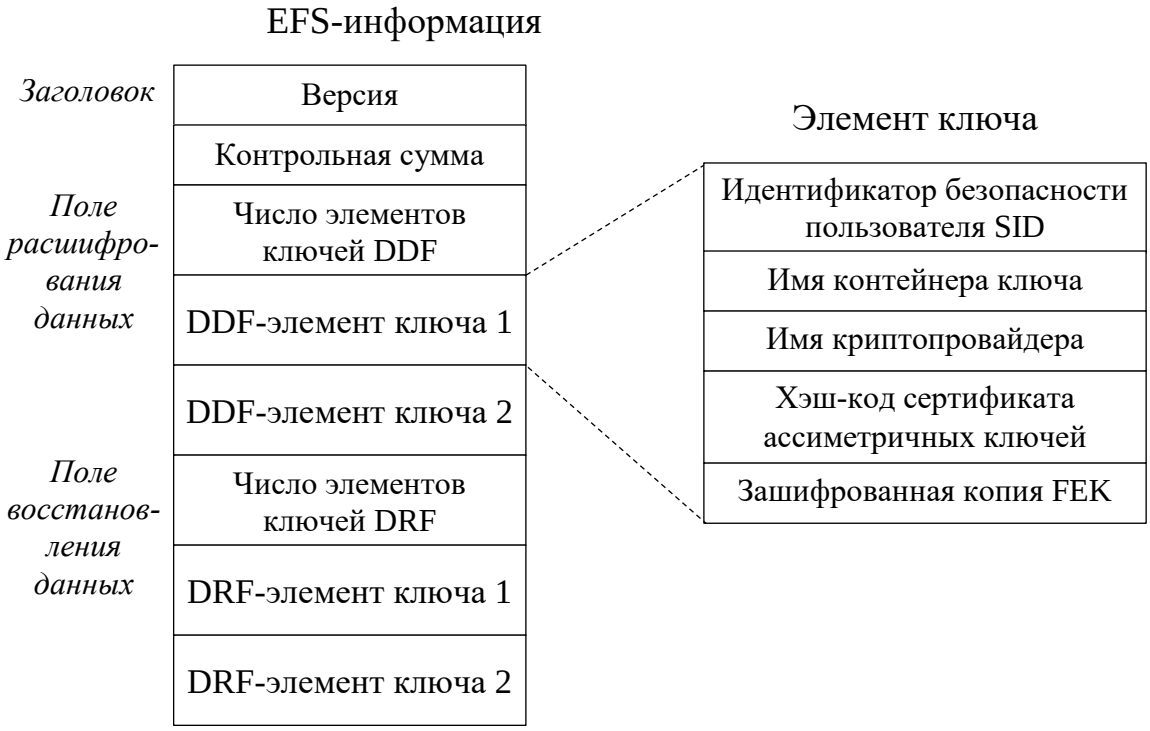

Рисунок 3.9. Формат EFS-информации и элементов ключа в зашифрованном файле

Доступ к данным, которые были зашифрованы пользователем, будет утрачен им и в случае сброса пароля пользователя. При плановой смене пароля (по сроку изменения пароля заданного в групповых политиках) или внеплановой смене с указанием предыдущего значения пароля, ключи шифрования обновляются (расшифровываются старым паролем и шифруются новым), и после смены пароля доступ к зашифрованным файлам не прекращается.

Доступ к зашифрованным EFS данным пропадает, если утрачена связь ОС с имеющимися физически на диске ключами. Существуют следующие типовые случаи, приводящие к потере доступа к зашифрованным файлам:

1. Система не загружается из-за смены или поломки комплектующих компьютера или нарушения работоспособности ОС (вышла из строя материнская плата, поврежден загрузочный сектор, испорчены системные файлы и т.п.).

2. Переустановка системы. Если имеется резервная копия системного диска, или профиля пользователя, доступ к зашифрованным данным может быть восстановлен с помощью специального программного обеспечения, но только в случае сохранности ключей.

3. Сброс пароля пользователя системным администратором или самим пользователем.

4. Удаление профиля пользователя.

5. Пользователь перенесен в другой домен (происходит его авторизация через другой сервер). Если при переносе сертификаты ключей хранились на сервере, то возможна потеря доступа к зашифрованным данным.

На случай физического удаления ключей с диска рекомендуется создать резервные копии сертификатов EFS/агента восстановления, содержащие закрытые ключи. Такие копии должны быть надежно защищены от кражи. Контейнер закрытого ключа защищается паролем, он должен быть сохранен на внешнем носителе, а физический доступ к последнему – ограничен.

Имея в своем распоряжении копию сертификата ключа, сотрудник, например, уезжая в отпуск, может удалить закрытый ключ из системы и предотвратить таким образом просмотр своих данных. После обратного импорта сертификата с закрытым ключом в систему доступ к зашифрованным файлам восстанавливается.

Вместе с тем, даже имея резервную копию ключа, не всегда можно восстановить доступ к данным. Например, если профиль пользователя был удален, то новый пользователь будет иметь другой SID, и связать импортированный сертификат ключа с новым пользователем будет невозможно. Если сертификаты ключей не были удалены с диска, для восстановления доступа к утраченным данным может быть использована утилита Advanced EFS Data Recovery от компании Elcomsoft, в процессе работы утилиты может потребоваться ввод параметров учетной записи пользователя – логина и пароля, от лица которого были зашифрованы файлы.

Таким образом, EFS шифрование является достаточно гибким и надежным средством защиты локально хранимых данных (при использовании надежных пользовательских паролей или аутентификации со смарт-картой/USB-токеном). Несомненными достоинствами EFS шифрования являются его прозрачность для пользователя, возможность поддержки российской криптографии, возможность управления на уровне домена, обеспечение защиты как при активной ОС, так в случае прямого доступа к диску.

Вместе с тем, EFS имеет существенные ограничения: не могут быть зашифрованы данные на разделе, отличном от NTFS (например, при копировании файла на флэш-накопитель с файловой системой FAT/ExFAT), не подлежат шифрованию системные файлы и области, что оставляет уязвимыми критичные данные ОС (такие, как реестр). Например, если разрешена авторизация доменного пользователя даже без подсоединения к домену, то средства проверки учетных данных в домене кэшируются в реестр компьютера. Поскольку эти данные не зашифрованы, нарушитель с помощью специальных инструментов сможет их извлечь и получить хэш-код пароля доменной учетной записи, что позволит ему производить офлайновую атаку на пароль при помощи взломщика паролей.

Для защиты от автономных атак Microsoft предлагает использовать шифрование на уровне тома с помощью инструмента BitLocker, доступного в версиях Windows Server и десктопных версиях Enterprise и Ultimate. Для съемных носителей (томов с файловой системой, отличной от NTFS) используется технология BitLocker To Go. BitLocker ориентирован, прежде всего, на совместное использование с аппаратным криптографическим сопроцессором – доверенным платформенным модулем (Trusted Platform Module, TPM) и глубоко привязан к конфигурации компьютера (рис. 3.10). Из-за импортных ограничений TPM на территории России недоступен, однако возможно использование шифрования BitLocker с ключом запуска, записанным на USB-накопитель.

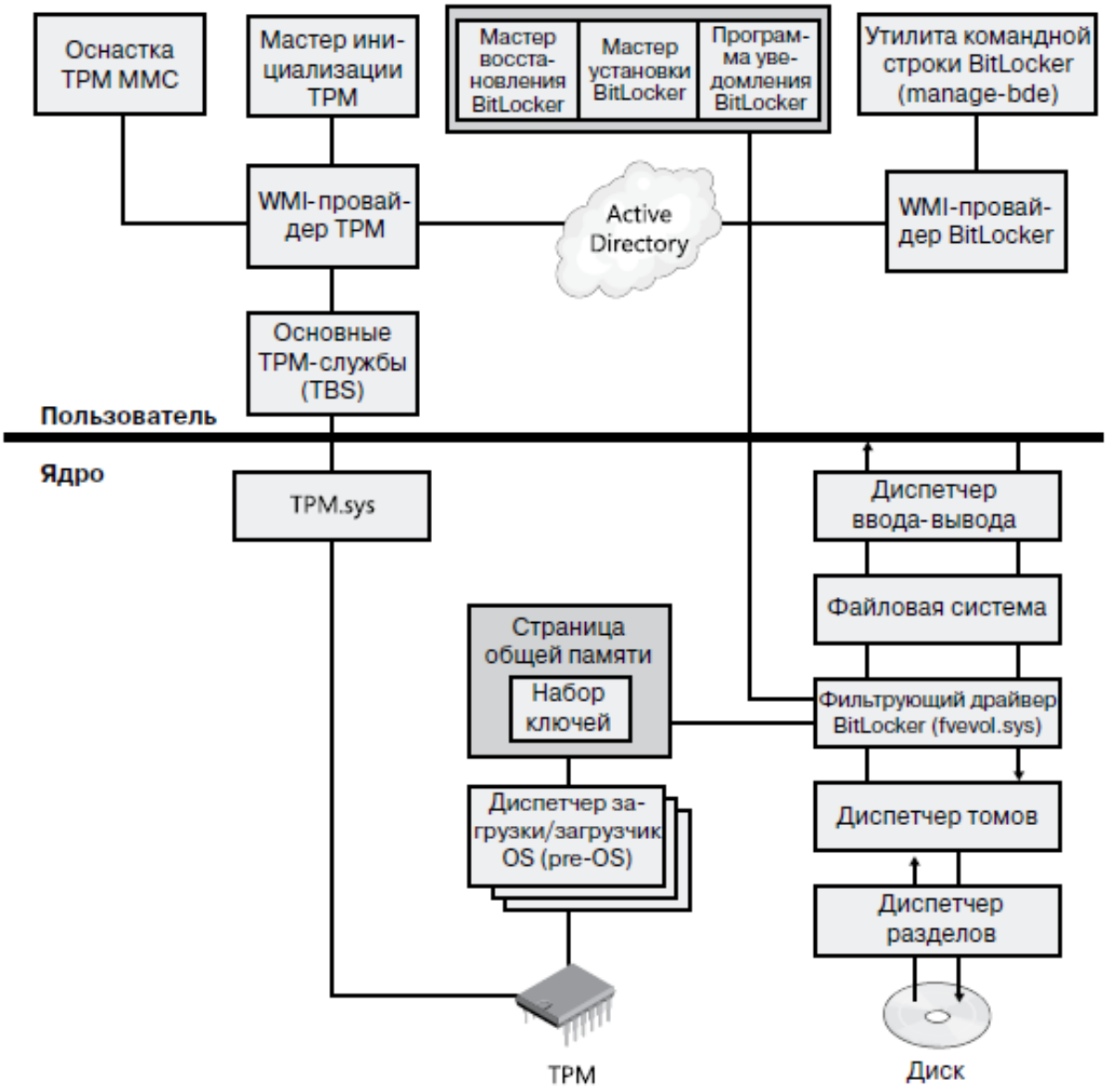

Рис. 3.10. Архитектура BitLocker

BitLocker в стандартном режиме позволяет осуществлять полнодисковое шифрование тома NTFS, в том числе и системного тома с операционной системой Windows, а также осуществлять проверку целостности компонентов ранних стадий загрузки и конфигурационных загрузочных данных.

BitLocker использует фильтрующий драйвер шифрования всего тома (Full-Volume Encryption, FVE), который автоматически видит все запросы к тому на ввод и вывод, зашифровывая блоки при записи и расшифровывая в процессе чтения. В отличие от EFS, функционирующей на уровне файловой системы, BitLocker работает на уровне подсистемы ввода-вывода, находящейся ниже NTFS (рис. 3.11). Поэтому NTFS «не знает» о включении BitLocker и работает с томом как обычно. Однако при чтении данных с тома

при прямом доступе к диску (минуя Windows) они окажутся зашифрованными. Это касается, в том числе, и системных файлов, если включено шифрование системного тома.

Такая низкоуровневая реализация BitLocker не позволяет задавать для шифрования альтернативные алгоритмы, например, определенные российскими криптографическими стандартами.

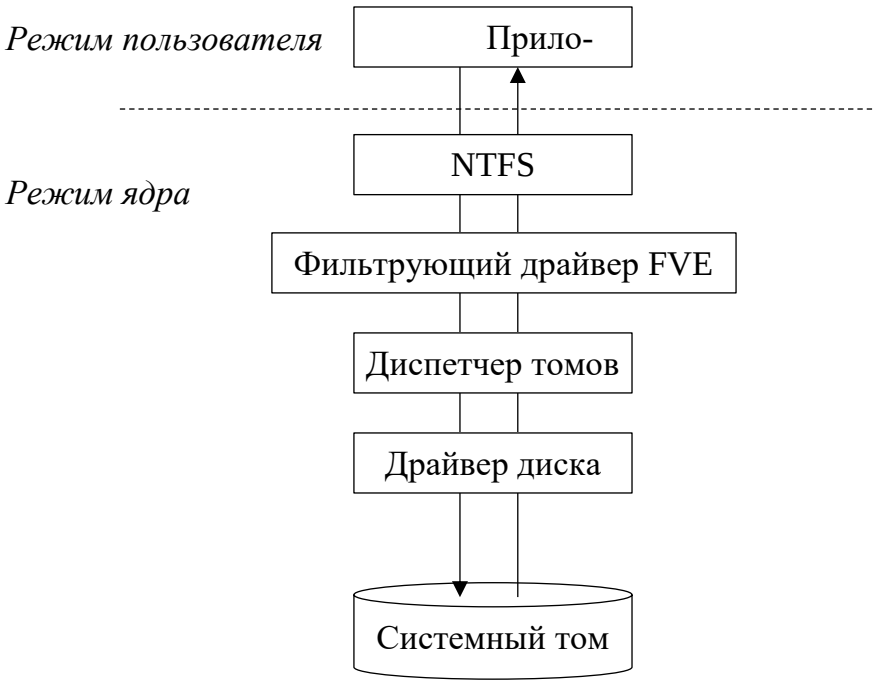

Рис. 3.11. Шифрование BitLocker

BitLocker автоматически шифрует все файлы, добавляемые на зашифрованный диск. Файлы будут зашифрованы только при хранении на зашифрованном диске, при их копировании на другой диск или компьютер они будут расшифрованы. Шифрование дисков BitLocker может использоваться совместно с шифрованием файловой системы EFS.

BitLocker оставляет незашифрованными:

- заголовок тома (совпадает с заголовком загрузочного сектора);
- поврежденные сектора, если они отмечены как нечитаемые;
- метаданные тома (зашифрованный том содержит три блока метаданных, для обеспечения избыточности).

Шифрование содержимого тома осуществляется BitLocker с помощью симметричного алгоритма AES128-CBC (по умолчанию), может быть выбран AES256-CBC с диффузором (Elephant Diffuser – расширение Microsoft) в ОС Windows 7 или AES256-XTS в ОС Windows 10. Поскольку режим CBC потенциально уязвим к атакам при известном изменении части

шифруемого содержимого (то есть при изменении части содержимого тома, что является обычной ситуацией), рекомендуется менять установленный по умолчанию режим шифрования с помощью соответствующей политики безопасности. При шифровании используемый ключ FVEK (Full-Volume Encryption Key, ключ полнодискового шифрования всего тома), назначаемый тому в момент включения BitLocker, сохраняется в области метаданных на диске. В случае шифрования с диффузором Elephant Diffuser [6] используется дополнительный ключ TWEAK. До шифрования сектора на основе TWEAK и содержимого сектора генерируется ключ сектора, который накладывается на исходное содержимое операцией побитового xor, а затем выполняются два бесключевых преобразования-диффузора (многократные повторения комбинации циклического сдвига, побитового xor и сложения по модулю 2^{32}).

Ключ TWEAK записывается в области метаданных сразу после FVEK. Ключи FVEK и TWEAK (если используется) хранятся в зашифрованном виде, для шифрования используется алгоритм AES в режиме CCM (Counter with CBC-MAC) и главный ключ тома (Volume Master Key, VMK). В свою очередь ключ VMK сохраняется на зашифрованном носителе в нескольких экземплярах, которые могут быть зашифрованы ключом восстановления, внешним ключом, пользовательским ключом или с помощью TPM (Trusted Platform Module).

Если TPM не используется, VMK зашифровывается внешним ключом (External Key, Startup key), который сохраняется на USB-накопителе. При защите несистемного тома может быть защищен пользовательский ключ на основе пароля (User key). Во время включения BitLocker кроме того будет создан ключ восстановления (Recovery Key). Обеспечение секретности ключа восстановления и его копий, а также защита ключевого носителя определяют надежность шифрования BitLocker.

Шифрование BitLocker может быть рассмотрено с точки зрения гарантированного уничтожения данных. После удалении ключей шифрования из области метаданных, чтение информации тома станет невозможным.

Шифрование BitLocker рассчитано лишь на автономные атаки при неактивной (выключенной) ОС. После загрузки операционной системы с использованием ключевого носителя (или ключа восстановления) ключи шифрования хранятся в оперативной памяти компьютера в открытом виде. Это значит, что сняв образ оперативной памяти, можно в дальнейшем получить доступ к зашифрованному тому. Следует отметить, что ключи шифрования EFS также могут находиться в оперативной памяти в случае недавнего обращения к зашифрованному файлу, однако здесь для каждого файла

использован свой ключ, поэтому здесь компрометация ключа имеет несколько другой масштаб по сравнению с ключами полнодискового шифрования.

Снятие дампа оперативной памяти в ОС Windows не составляет большого труда, если пользователь не использует блокировку компьютера. Для этих целей могут быть использованы как встроенные механизмы самой ОС Windows, так и бесплатные утилиты, такие как Belkasoft Live RAM Capturer [7], или специальный Live-дистрибутив ОС linux (например, Ubuntu CyberPack). Такие средства загружаются в оперативную память атакуемого компьютера и, вообще говоря, изменяют ее содержимое, однако занимают, как правило, очень немного места, что позволяет их использовать в криминалистической практике. В более сложном случае (если невозможен запуск программного обеспечения на атакуемом компьютере) криминалистами может быть использован метод холодной/горячей перезагрузки или получен непосредственный доступ к памяти с помощью специальных плат расширения, через порт FireWire и другие порты, использующие прямой доступ к памяти (DMA), в крайнем случае, возможно физическое изъятие оперативного запоминающего устройства с замораживанием [8, 9]. После получения снимка оперативной памяти возможно извлечение ключей BitLocker и получение доступа к зашифрованному диску с помощью утилиты Elcomsoft Forensic Disk Decryptor или Passware Kit Forensic [10]. Другой путь получения доступа к данным – использование копий ключей из облачного хранилища учетной записи Microsoft или хранилищ Active Directory, либо сохраненного пользователем ключа восстановления [7].

Дампы оперативной памяти, а также содержание файлов гибернации и подкачки, являются основным источником данных для извлечения паролей и ключевой информации. Анализ «вовремя» снятого образа оперативной памяти позволяет получить необходимые данные для дальнейшего дешифрования пользовательских данных. Подобный анализ и дешифрование могут быть реализованы с помощью таких криминалистических утилит, как Passware Kit Forensic, Elcomsoft Forensic Disk Decryptor (доступ к содержимому дисков, зашифрованных с помощью таких известных утилит, как BitLocker, TrueCrypt, PGP, FileVault 2), Elcomsoft Advanced EFS Data Recovery, Elcomsoft Advanced Office Password Recovery, Elcomsoft Advanced PDF Password Recovery и др.

Для защиты BitLocker и затруднения снятия дампа оперативной памяти следует:

- запретить загрузку ОС с внешнего носителя: USB-устройства и CD (или сделать первым загрузочным устройством жесткий диск), а также установить пароль на изменение настроек BIOS/UEFI;

- использовать режим безопасной загрузки UEFI;
- блокировать компьютер, если он временно не используется (с требованием введения пароля для разблокировки),
- заблокировать FireWire (и другие DMA-порты, если они установлены на компьютере) в BIOS/UEFI, с помощью групповой политики или специализированного программного обеспечения (например, Kaspersky Endpoint Security, DeviceLock и т.п.).

Отметим, что порты прямого доступа к памяти (FireWire, Thunderbolt, ExpressCard и др.) могут оставаться не заблокированными в процессе загрузки Windows в промежуток времени после разблокировки диска BitLocker и до применения связанных с этими портами политик или загрузки утилит контроля портов.

Рассмотрение особенностей шифрования BitLocker позволяет сделать следующие выводы:

- использование средства BitLocker оправдано в том случае, если необходимо обеспечить криптографическую защиту системных файлов (BitLocker должен быть включен на системном томе), либо требуется создать том, при записи на который данные подвергаются автоматическому шифрованию;
- низкоуровневая реализация и глубокая аппаратная привязка BitLocker повышают риски утраты зашифрованных данных в случае изменения или выхода из строя комплектующих компьютера либо нарушения работоспособности ОС;
- снятие шифрования BitLocker не приводит к расшифровыванию содержимого диска, а лишь раскрывает ключи шифрования, при этом процедура не всегда осуществляется корректно (например, может быть запрещена запись на диск);
- если все же принято решение об использовании BitLocker, рекомендуется сочетать его с EFS шифрованием критичных данных, следует придерживаться приведенных выше рекомендаций для защиты ключей BitLocker.

ОС семейства Windows обладают встроенными средствами криптографической защиты, передаваемой по сети – реализацией протоколов IPSec, Kerberos, а также позволяет осуществлять управление цифровыми сертификатами корпоративной сети с помощью серверной службы Certification Authority (CA). Протокол IPSec по умолчанию использует для защиты сетевого трафика 3DES и SHA-1, это наиболее сильные криптографические алгоритмы, поддерживаемые реализацией протокола. Аутентификация сторон выполняется в рамках протокола Kerberos [11, С.264-268],

что обеспечивает соблюдение принципа единого входа. В качестве предраспределенного общего секретного ключа используется хэш-код пароля пользователя, первоначальное значение которого задается администратором домена во время регистрации доменного пользователя. В поздних версия ОС протоколом по умолчанию используются криптоалгоритмы AES-256 в режиме CTS (вариант CBC) и HMAC с хэш-функцией SHA-1. При аутентификации пользователя по смарт-карте, первоначальная аутентификация пользователя производится с помощью асимметричной криптографии (алгоритм зависит от используемого сертификата) в рамках расширения PKINIT.

Возможно использование отечественных реализаций протокола IPSec (например, КриптоПро IPSec с криптопровайдером КриптоПро CSP). При этом российская криптография в протоколе Kerberos может использоваться в рамках PKINIT, если имеется соответствующий сертификат пользователя со смарт-картой. Выпуск сертификатов с поддержкой российских криптографических стандартов доступен при развертывании соответствующего программного обеспечения удостоверяющего центра (например, КриптоПро УЦ), либо с помощью службы CA [5, С.72-79]. Выпуск сертификатов с российской криптографией позволяет соответствующим образом настроить протокол TLS для обеспечения защиты Интернет-соединений (например, для подключения к корпоративному web-серверу по HTTPS).

Практика формирования ключей шифрования на основе пользовательского пароля (как правило, это пароль учетной записи пользователя для входа в ОС), с одной стороны, обеспечивает несомненное удобство работы с сервисами безопасности с соблюдением принципа единого входа, но с другой стороны, может способствовать решению задачи дешифрования. Фактическая безопасность построенных таким образом систем, в конечном счете, определяется надежностью пароля пользователя. Даже если пароль был задан специально, в ряде случаев при известной личной информации он может быть найден с высокой вероятностью, поскольку пользователи, как правило, используют легко запоминаемые, а значит, относительно несложные пароли [12].

Компании, работающие на рынке криминалистического программного обеспечения, предлагают утилиты для подбора паролей, такие, как Passware Kit Forensic, Elcomsoft Distributed Password Recovery. Повышение эффективности перебора достигается распараллеливанием вычислений за счет распределения их между различными устройствами (использование мультиагентных систем), использования мощных вычислительных систем с многоядерными (например, 32-ядерных) центральными процессорами

(CPU), активного использования графических процессоров (GPU) видеокарт. Современные графические адаптеры могут иметь до нескольких тысяч процессоров, что позволяет проводить эффективные параллельные вычисления. При этом скорость подбора паролей может существенно варьироваться в зависимости от используемых форматов файлов данных, поскольку некоторые из них требуют значительной предварительной обработки традиционным способом, загружая центральный процессор.

Тем не менее, подбор пароля методом грубой силы практически неосуществим, поскольку функции формирования ключа по паролю используют многократные итерации хэширования с подмешиванием случайных данных, что значительно замедляет вычисление ключа и, как следствие, существенно снижает скорость опробования паролей. Например, процедура вычисления ключа, используемая при парольной защите документов Microsoft Office [5, С.80-81], допускает до 10 млн. итераций хэширования. Словарная атака с использование универсальных словарей, например, орфографического, либо словарей, предоставляемых разработчиками криминалистических средств (тем же Passware), также зачастую оказывается неэффективной. Однако ситуация не является безвыходной. Решение, как это часто бывает, обусловлено наличием «человеческого фактора».

Как правило, пользователь использует не более трех различных паролей и их незначительные модификации (изменение регистра символов, замена отдельных букв на цифры или символы, добавление цифровой комбинации в начало/конец слова) для доступа к различным ресурсам. Это дает возможность реализации словарной атаки, эффективность которой существенно зависит от релевантности используемого для подбора словаря.

Общие рекомендации для криминалистов состоят в том, чтобы попытаться по возможности извлечь данные из всех устройств и облачных хранилищ, начиная с наиболее доступных, таких как данные браузеров, системные ресурсы (например, реестр Windows), файлы с наименее надежным типом шифрования. Извлеченная информация может быть использована для составления таргетированного словаря, ориентированного на конкретного пользователя. Результаты анализа снимков оперативной памяти, наряду с данными, полученными из хранилищ веб-браузеров и учетных записей почтовых клиентов могут быть использованы для создания такого словаря. Кроме того, извлеченная информация может дать представление о предпочтениях пользователя, в частности, о типе мутаций паролей, которые он, возможно, использовал. Современные средства вскрытия паролей, например, Passware Kit Forensic, Elcomsoft Distributed Password Recovery позволяют задавать определенные мутации при использовании словарной

атаки. Подбор пароля с использованием таргетированного словаря, как правило, приводит к успешному вскрытию защищенной информации [13].

Широкая распространенность и доступность современных СКЗИ может создать у неспециалиста ложную уверенность в безопасности защищаемой информации. Использование стойких криптоалгоритмов само по себе не может гарантировать высокую надежность СКЗИ, при этом дополнительная настройка таких средств, включая увеличение длины используемых ключей шифрования, не играет первостепенной роли. Более важным является четкое понимание назначения, особенностей реализации и существующих ограничений того или иного СКЗИ, а также комплексное применение защитных мер и средств (например, обязательное включение блокировки Windows при использовании средства BitLocker). Подобные условия применения СКЗИ могут быть учтены специалистами при разработке политик безопасности. Особо стоит обратить внимание, как ни банально это звучит, на тщательный выбор паролей, в частности, пароля для входа в операционную систему. Альтернативой может служить аппаратная аутентификация с использованием смарт-карт/ USB-токенов с цифровыми сертификатами.

Только комплексный подход и строгое следование политикам безопасности может гарантировать необходимый уровень защищенности данных.

Литература:

1. Васильева И.Н. Вопросы практической надежности криптографических средств защиты информации // Региональная информатика и информационная безопасность. Сборник трудов. Выпуск 5 / СПОИСУ. – СПб., 2018. – С. 186-189.
2. Belcasoft. Forensics made easier: [сайт]. URL: https://belkasoft.com/ru/ (дата обращения: 09.11.2018).
3. Перебор паролей, восстановление доступа, расшифровка информации, мобильная криминалистика/ Elcomsoft Co.Ltd.: [сайт]. URL: https://www.elcomsoft.ru/ (дата обращения: 09.11.2018).
4. Руссинович М., Соломон Д., Ионеску А. Внутреннее устройство Microsoft Windows. 6-е изд. Основные подсистемы ОС. – СПб.: Питер, 2014. – 672 с.
5. Защита информации в компьютерных системах: монография; под ред. Е.В. Стельмашонок, И.Н. Васильевой – СПб.: Изд-во СПбГЭУ, 2017. – 163 с.
6. Niels Ferguson. AES-CBC + Elephant diffuser. A Disk Encryption Algorithm for Windows Vista, Microsoft, Aug. 2006 [Электронный ресурс]. URL: https://www.microsoft.com/en-us/download/confirmation.aspx?id=13866 (дата обращения 09.11.2018).
7. Возможен ли взлом Bitlocker? [Электронный ресурс]. URL: http://www.spysoft.net/vzlom-bitlocker/ (дата обращения: 09.11.2018).

8. Использование метода «холодной» перезагрузки и других криминалистических техник в пентестах, 21.05.2014 [Электронный ресурс]. URL: https://www.securitylab.ru/analytics/452899.php (дата обращения 09.11.2018).
9. Типы атак для ключей шифрования томов, 13.08.2015 [Электронный ресурс]. URL: https://technet.microsoft.com/ru-ru/library/mt404683(v=vs.85).aspx (дата обращения 09.11.2018).
10. Windows Password Recovery tools by Passware: [сайт]. URL: https://www.passware.com (дата обращения: 09.11.2018).
11. Васильева И.Н. Криптографические методы защиты информации : учебник и практикум для академического бакалавриата – М.: Юрайт, 2016. − 349 с.
12. Афонин О. Использование утечек паролей для ускорения атак. 16.02.2017 [Электронный ресурс]. URL: https://blog.elcomsoft.com/ru/2017/02/ispolzovanie-utechek-paroley-dlya-uskoreniya-atak/ (дата обращения: 09.11.2018).
13. Афонин О. Как вскрыть до 70% паролей за несколько минут. 14.02.2017 [Электронный ресурс]. URL: https://blog.elcomsoft.com/ru/2017/02/kak-vskryit-do-70-paroley-za-neskolko-minut/ (дата обращения: 09.11.2018).