Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение   
высшего образования   
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Учебно-методическое пособие

по теме: **«Шифрование жесткого диска.**

**Возможности шифрования данных в NTFS, ext4, btrfs»**

по дисциплине

«Криптографические методы защиты информации»

Екатеринбург 2020

**Оглавление**

[**Теория** 3](#_Toc42204309)

[**1. Что такое шифрование жесткого диска?** 3](#_Toc42204310)

[**2. Аппаратное и программное шифрование** 3](#_Toc42204311)

[**3. Шифрование диска и шифрование на уровне файловой системы** 3](#_Toc42204312)

[**4. Файловая система NTFS** 4](#_Toc42204313)

[**5. Обзор файловых систем: Ext4, Btrfs и Xfs** 6](#_Toc42204314)

[Лабораторный практикум 9](#_Toc42204315)

[Лабораторная работа №1 – Шифрование виртуального жесткого диска в Windows (NTFS). 9](#_Toc42204316)

[Лабораторная работа №2 – Шифрование виртуального жесткого диска в Linux (EXT4). 11](#_Toc42204317)

[**Список литературы** 13](#_Toc42204318)

# **Теория**

## **1. Что такое шифрование жесткого диска?**

Шифрование жесткого диска — это процесс преобразования данных на диске или всем накопителе в нечитаемый код с помощью математических алгоритмов, в результате которого несанкционированный доступ к данным становится невозможен. Для доступа к зашифрованному диску пользователь должен использовать пароль, отпечаток пальца или смарт-карту. Шифрование может выполняться с помощью программных или аппаратных механизмов, которое шифрует каждый бит хранилища. На клиентских устройствах обычно используется программное шифрование. Шифрование может выполняться на уровне файлов или для всего жесткого диска.

## **2. Аппаратное и программное шифрование**

Основное различие между программным и аппаратным шифрованием заключается в том, что при использовании программного шифрования нельзя зашифровать главную загрузочную запись (MBR). Клиентские системы Dell используют Wave Trusted Drive Manager в составе пакета Dell Data Protection или Dell ControlPoint Security Manager совместно с микросхемой TPM для программного шифрования. Корпоративные заказчики могут использовать Dell Data Protection Encryption и модуль ускорителя, DDPE Accelerator, который используется в слоте материнской плате через мини-плату для портативных устройств или плату PCIe для настольных компьютеров. Аппаратное шифрование более безопасно, так как оно изолирует диск от процессора и ОС, что делает его значительно менее уязвимым для атак.

## **3. Шифрование диска и шифрование на уровне файловой системы**

Выражение full disk encryption (FDE) обычно означает, что всё на диске находится в зашифрованном виде, в том числе и загрузочные системные разделы.

Шифрование на уровне файловой системы (filesystem-level encryption — FLE) — процесс шифрования каждого файла в хранилище. Доступ к зашифрованным данным можно получить только после успешной аутентификации. Некоторые операционные системы имеют собственные приложения для FLE, при этом доступно и множество реализаций от сторонних разработчиков. FLE прозрачно, это значит, что каждый, кто имеет доступ к файловой системе, может просматривать названия и метаданные зашифрованных файлов, которыми может воспользоваться злоумышленник.

Шифрование на уровне файловой системы отличается от полного шифрования диска. FDE защищает данные до тех пор, пока пользователь не пройдёт загрузку, так что в случае утраты или кражи диска данные будут для злоумышленника недоступны, если же во время работы диск расшифрован и злоумышленник получил доступ к компьютеру, то он получает доступ ко всем файлам в хранилище. FLE же защищает до тех пор, пока пользователь не пройдёт аутентификацию для конкретного файла, при работе с одним файлом остальные всё так же зашифрованы, поэтому FLE может быть использован вместе с полным шифрованием для большей безопасности.

Ещё одно важное отличие заключается в том, что FDE автоматически шифрует все данные на диске, в то время как FLE не защищает данные вне зашифрованных файлов и папок, поэтому временные и swap файлы могут содержать незашифрованную информацию.

## **4. Файловая система NTFS**

NTFS — стандартная файловая система для семейства операционных систем Windows NT фирмы Microsoft. NTFS поддерживает хранение метаданных. С целью улучшения производительности, надёжности и эффективности использования дискового пространства для хранения информации о файлах в NTFS используются специализированные структуры данных. Информация о файлах хранится в главной файловой таблице — MFT (Master File Table). NTFS поддерживает разграничение доступа к данным для различных пользователей и групп пользователей, а также позволяет назначать дисковые квоты (ограничения на максимальный объём дискового пространства, занимаемый файлами тех или иных пользователей). Для повышения надёжности файловой системы в NTFS используется система журналирования USN. Для NTFS размер кластера по умолчанию составляет от 512 байт до 64 КиБ в зависимости от размера тома и версии ОС.

**Возможности NTFS**

Файловая система NTFS разрабатывалась Microsoft в начале 1990 х гг. как основная файловая система для серверных версий операционных систем Windows. NTFS была представлена в 1993 году в операционной системе Windows NT 3.1. В настоящее время NTFS рассматривается в качестве предпочтительной файловой системы как для серверных, так и для клиентских версий Windows. В NTFS используются 64 разрядные идентификаторы кластеров, поэтому теоретически том NTFS может содержать 264 кластеров (16 ЭБ3). Однако текущие реализации в Windows поддерживают только 32 разрядную адресацию кластеров, что при размере кластера максимум 64 КБ (216 байт) позволяет NTFS тому достигать размера до 256 ТБ: 232 \* 216 байт = 248 байт = 28 \* 240 байт = 256 ТБ. Для томов, больших 4 ГБ, при форматировании Windows предлагает размер кластера по умолчанию 4 КБ.

Перечислим некоторые возможности NTFS:

- восстанавливаемость (recoverability)– способность файловой системы возвращаться к работоспособному состоянию после возникновения сбоя. Реализуется такая возможность, во-первых, за счет поддержки атомарных транзакций, во-вторых, за счет избыточности хранения информации. Атомарная транзакция (atomic transaction) – операция с файловой системой, приводящая к её изменению, которая либо полностью успешно выполняется, либо не выполняется вообще (т. е. в случае сбоя во время атомарной транзакции все изменения откатываются). Избыточность используется при хранении важнейших данных файловой системы, критически необходимых для её корректной работы;

- безопасность (security) – защищенность файлов от несанкционированного доступа. Реализуется при помощи модели безопасности Windows, рассмотренной в лекции 9 "Безопасность в Windows";

- шифрование (encryption) – преобразование файла в зашифрованный код, который невозможно прочесть без ключа. Обычные механизмы безопасности, такие как назначение прав доступа пользователей к файлам, не обеспечивают полной защиты информации, например, в случае перемещения диска на другой компьютер. Администратор операционной системы всегда может получить доступ к файлам других пользователей, даже на томе NTFS. Поэтому в NTFS включена поддержка шифрующей файловой системы EFS (Encrypting File System), которая позволяет легко зашифровывать и расшифровывать файлы;

- поддержка RAID (Redundant Array of Inexpensive (Independent) Disks – массив недорогих (независимых) дисков с избыточностью) – возможность использования для хранения информации нескольких дисков; данные с одного диска автоматически копируются на другие, обеспечивая тем самым повышенную надежность;

- дисковые квоты для пользователей (Per-User Volume Quotas) – возможность выделения для каждого пользователя определенного пространства на диске (квоты);

- NTFS не позволяет пользователю записывать данные на диск сверх выделенной квоты.

**Файлы NTFS**

Основная информация о файле содержится в файловой записи (File Record) размером 1 КБ таблицы MFT, а небольшие файлы целиком хранятся в файловой записи.

Файловая запись состоит из заголовка (Header) и набора атрибутов (Attribute). В заголовке содержится служебная информация о файловой записи, например, её тип и размер. Все данные, относящиеся непосредственно к файлу, хранятся в виде атрибутов. Названия атрибутов, так же как и системных файлов, начинаются с "$". Например, отдельными атрибутами являются имя файла ($FILE\_NAME), информация о его свойствах ($STANDARD\_INFORMATION), данные файла ($DATA).[3]

Файловая запись

На диске файловая запись всегда расположена в начале сектора, первые байты файловой записи кодируют слово "FILE" (ASCII-коды: 46 49 4C 45). Конец записи определяется 4 байтовой последовательностью FF FF FF FF.

Физически атрибут файла хранится в виде потока байтов (stream) – простой последовательности байтов. Такое представление позволяет одинаковым образом работать с разнотипными атрибутами, а также добавлять нестандартные пользовательские атрибуты.

Каждый атрибут состоит из заголовка (attribute header), определяющего тип атрибута и его свойства, и тела (attribute body), содержащего основную информацию атрибута.

Структура файловой записи

По расположению относительно MFT атрибуты бывают резидентные и нерезидентные. Резидентные атрибуты (resident attributes) полностью помещаются в файловую запись MFT, нерезидентные атрибуты (nonresident attributes) хранятся вне MFT. Область, в которой расположен нерезидентный атрибут, называется группой (run). Поскольку нерезидентных атрибутов в файле может быть несколько, то и групп бывает тоже несколько. Множество групп файла называется списком групп (RunList). Файловая запись при наличии нерезидентных атрибутов содержит ссылку на расположение группы на диске.

**Безопасность**

NTFS содержит множество средств разграничения прав объектов — есть мнение, что это самая совершенная файловая система из всех ныне существующих. В теории это, без сомнения, так, но в текущих реализациях, к сожалению, система прав достаточно далека от идеала и представляет собой хоть и жесткий, но не всегда логичный набор характеристик. Права, назначаемые любому объекту и однозначно соблюдаемые системой, эволюционируют — крупные изменения и дополнения прав осуществлялись уже несколько раз и к Windows 2000 все-таки они пришли к достаточно разумному набору. Права файловой системы NTFS неразрывно связаны с самой системой — то есть они, вообще говоря, необязательны к соблюдению другой системой, если ей дать физический доступ к диску. Для предотвращения физического доступа в Windows2000 (NT5) всё же ввели стандартную возможность.[4]

## **5. Обзор файловых систем: Ext4, Btrfs и Xfs**

В операционной системе Linux доступно множество файловых систем на любой вкус. В Linux очень широко используется Ext4, но есть несколько поводов попробовать что-то новое. Например, Btrfs или Xfs. Давайте рассмотрим самые популярные файловые системы и особенности их работы, а затем займемся их сравнением.

**Функции файловых систем**

Файловые системы используются для контроля способа записи данных на диск, контроля доступа к этим данным, а также хранения информации и метаданных о файлах. Файловые системы постоянно пересматриваются для предоставления большей функциональности и каждый раз становятся эффективнее.

**Зачем разбивать на разделы?**

У многих пользователей смутные представления о том, зачем нужны разделы диска. Все операционные системы поддерживают создание и удаление разделов. Linux использует более одного раздела на диске, даже при использовании стандартной процедуры установки. Главная цель разделения дисков на разделы — повышение безопасности в случае возникновения ошибок.

При разбиении жесткого диска на разделы, данные могут быть сгруппированы, а также разделены. При возникновении ошибок, будут потеряны только те, которые находились на поврежденном разделе. В это же время, данные других разделов подвергаются меньшей опасности. Эти принципы датируются днями, когда операционная система Linux не имела журналируемой файловой системы, и любой сбой питания мог привести к катастрофе.

Использование разделов оправдано в целях безопасности и надежности. Нарушение в одной части операционной системы не означает, что весь компьютер подвержен опасности. К примеру, пользователи создают скрипты, программы или веб-приложения, которые начинают заполнять диск. Если диск содержит только один большой раздел, то когда свободное место закончится, система полностью перестанет работать. Но если пользователь хранит данные в отдельных разделах, то переполнение затронет только этот раздел данных, в то время как системные разделы будут продолжать функционировать.

Имейте в виду, что наличие журналируемой файловой системы обеспечит защиту данных только в том случае, если произошел сбой питания или же внезапное отключение запоминающего устройства. Это не защитит данные от битых блоков и логических ошибок в файловой системе. В таких случаях пользователь должен использовать резервный массив нескольких дисков (RAID).

**Зачем менять файловые системы?**

Файловая система Ext4 — улучшенная версия Ext3, которая, в свою очередь, не что иное, как улучшенная Ext2. Ext4 — очень надежная ФС, которая используется практически во всех дистрибутивах в течение последних нескольких лет. Но её код уже достаточно устарел.

Кроме того, пользователи Linux хотят новых возможностей и функций, которых нет в Еxt4, но они есть в других файловых системах, например, Btrfs и Xfs. Существует программное обеспечение, которое удовлетворяет некоторые из таких потребностей, но поддержка на уровне файловой системы будет работать намного быстрее.

**Файловая система Ext4**

У Ext4 есть некоторые ограничения, которые впечатляют даже сейчас. Максимальный размер файла составляет 16 тебибайт (что составляет примерно 17,6 терабайт). А это намного больше, чем емкость любого жесткого диска, который может купить обычный пользователь. В то время как наибольший том / раздел, который вы можете сделать с помощью Ext4, составляет 1 эксабайт (что составляет примерно 1152921.5 терабайт).

Известно, что Ext4 работает быстрее, чем Ext3. Как и все современные файловые системы, она журналируемая, а это значит, что Ext4 будет вести журнал расположения файлов на диске, а также записывать туда любые изменения данных. Несмотря на все ее функции, она не поддерживает прозрачное сжатие, дедупликацию данных и прозрачное шифрование. Снимки состояния технически поддерживаются, но это только экспериментальная функция.

**Файловая система Btrfs**

Это файловая система, которая полностью создана с нуля. Она существует потому, что её разработчик захотел расширить функциональность стандартной файловой системы такими возможностями: снимки состояния, объединение, контрольные суммы и прозрачное сжатие.

Btrfs не зависит от Ext4, но реализует ее лучшие идеи и преимущества, а также свои дополнительные возможности, которые будут очень полезны пользователям, и особенно предприятиям.

Для предприятий, которые используют большие программы с большими базами данных, Btrfs создает непрерывную файловую систему на нескольких жестких дисках, что упрощает консолидацию данных. Дедупликация данных уменьшит фактически занимаемое данными пространство на диске. А зеркалирование данных с Btrfs станет намного проще.

Пользователь может выбрать создание нескольких разделов, чтобы не нужно было выполнять зеркалирование. Учитывая, что Btrfs может охватывать несколько жестких дисков, очень хорошо, что система может поддерживать в 16 раз больше места на жестком диске, чем Ext4. Максимальный размер раздела файловой системы Btrfs составляет 16 эксабайт, столько же составляет максимальный размер файла.

**Файловая система XFS**

XFS считается расширенной файловой системой. XFS — это высокопроизводительная файловая система с 64-разрядными журналами. Поддержка XFS была включена в ядро Linux примерно в 2002 году. XFS поддерживает максимальный размер файловой системы, равный 8 эксабайт для 64-битной файловой системы.

Но у XFS существуют некоторые ограничения. Например, раздел этой файловой системы не может быть уменьшен, а также наблюдается низкая производительность при работе с большим количеством файлов. RHEL 7.0 использует XFS в качестве файловой системы по умолчанию.

**Итог**

Итак, что лучше использовать? До сих пор Ext4 является наилучшим выбором, несмотря на идентичную производительность с XFS. Но почему? Удобство и «вездесущность» — основные причины. Ext4 — отличная файловая система для настольных или рабочих станций. Она предоставляется по умолчанию, поэтому пользователь может установить на ней операционную систему. Кроме того, Ext4 поддерживает тома до 1 Exabyte и файлы размером до 16 терабайт.

Btrfs предлагает большие объемы до 16 экзабайт как для разделов, так и для файлов, а также повышение отказоустойчивости. Но она до сих пор позиционируется как надстройка над файловой системой, а не интегрирована в операционную систему ФС. Например, чтобы отформатировать раздел в Btrfs необходимо, чтобы был установлен набор инструментов Btrfs.

Несмотря на то, что скорость передачи данных так важна, в Btrfs она выше в самой системе, чем непосредственно при передаче файлов. Btrfs имеет множество полезных функций, таких как Copy-on-Write (CoW), расширенные контрольные суммы, моментальные снимки, самовосстановление данных, дедупликация, а также улучшения, которые обеспечивают целостность данных. У Btrfs нет функций RAID-Z для ZFS, поэтому RAID все еще находится в экспериментальном состоянии с Btrfs. Тем не менее, для чистого хранения данных Btrfs лучше.

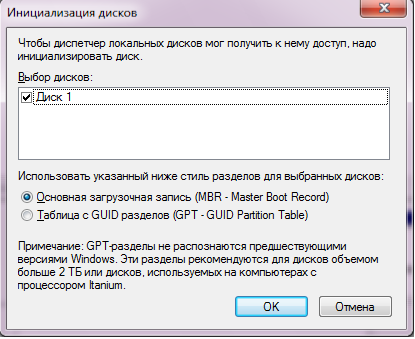
На данный момент Ext4 — лучший выбор, так как она распространяется как файловая система по умолчанию, а также она быстрее Btrfs при передаче файлов. Btrfs, безусловно, стоит попробовать, но полностью заменять Ext4 еще рано, это можно будет сделать лишь через несколько лет.

# Лабораторный практикум

## Лабораторная работа №1 – Шифрование виртуального жесткого диска в Windows (NTFS).

Цель: Получение теоретических и практических навыков работы с программными средствами шифрования данных.

1. Ознакомьтесь с теоретическим материалом.
2. Запустите виртуальную машину в установленной Windows 7.
3. Создайте виртуальный жесткий диск, для этого:
   1. «Пуск» - ПКМ «Компьютер» - «Управление»
   2. «Управление дисками» - «Действие» - «Создать виртуальный жесткий диск»
   3. В открывшемся окне указываем место, в котором будет храниться наш виртуальный диск, его размер и тип.
   4. После нажатия на кнопку «Ок», в списке дисковых устройств появится наш виртуальный диск.
4. Правой кнопкой щелкаем на наш виртуальный диск и выбираем пункт «инициализировать диск»
5. В открывшемся окне указываем какой диск мы инициализируем и его параметры.



1. Правой кнопкой щелкаем на нашем виртуальном диске и выбираем «Создать простой том». В открывшемся окне указываем размер диска, букву диска, тип файловой системы.

Встроенная в windows система efs легко и быстро позволит вам зашифровать файл или нужную папку таким образом, что доступ к зашифрованной папке будет только у вашего пользователя. При этом любой другой пользователь (даже администратор) не может получить доступ.

Чтобы зашифровать файл выполните следующие действия по порядку:

1. Войдите в систему с правами администратора. Откройте редактор реестра ( win+R, regedit.msc)
2. Добавьте следующий параметр параметра типа DWORD: HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft \Windows NT\CurrentVersion\EFS.”AlgorithmID “=dword: 0×6603. Это позволит зашифровать диск с использованием алгоритма 3DES.
3. Создайте учетную запись основного пользователя, под которой будет осуществляться работа с жестким диском. Войдите с использованием этой учётной записи. На время настройки необходимо дать\оставить у данной УЗ права администратора.
4. Создайте папку вида «Номер зачётной книжки» на созданном виртуальном жестком диске.
5. Нажать «Пуск» -> выбрать иконку «Мой компьютер» -> в левой части окна нажать правой клавишей на иконке «Мои документы» -> выбрать «Свойства» -> перейти на закладку «Общие» -> нажать кнопку «Другие…» -> выделить «Шифровать содержимое для защиты данных» -> нажать 2 раза «ОК» -> в открывшемся окне выбрать «к этой папке и ко всем вложенным файлам» -> нажать ОК. Таким образом можно зашифровать любой файл или папку.
6. Нажать «Пуск» > «Выполнить» -> набрать команду mmc и нажать ОК.
7. В открывшемся окне в меню выбрать «Консоль» -> «Добавить или удалить оснастку…» -> нажать кнопку «Добавить…» -> выбрать «Сертификаты» и нажать «Добавить» -> выбрать пункт «моей учетной записи пользователя» -> нажать «Готово» -> «Закрыть» -> ОК.
8. Открыть ветку «Сертификаты» -> «Личные» -> «Сертификаты».
9. Нажать правой клавишей мыши на сертификат пользователя -> выбрать «Все задачи» -> «Экспорт…»
10. В открывшемся окне нажать «Далее» -> выбрать «Да», «экспортировать закрытый ключ» -> «Далее» -> «Далее» -> ввести пароль для закрытого ключа (минимально 8 символов) -> «Далее» -> ввести имя файла и указать путь, куда он будет записан -> нажать «Далее» -> «Готово» -> «OK.
11. Сохранить созданный файл на внешний носитель информации (CD-R, USB-флеш…), после чего удалить файл сертификата с жесткого диска. Также рекомендуется записать пароль для закрытого ключа на бумаге и поместить его в запечатываемый конверт. Сертификат пригодится в следующих случаях: если пользователь забыл пароль и вам пришлось его сменить; если пользователь, под которым был зашифрован файл, удален; если вы хотите снять защиту из-под другой учетной записью. ВНИМАНИЕ! При утере сертификата защиты восстановить доступ к зашифрованным папкам будет невозможно.
12. Учётной записи пользователя можно назначить права непривилегированного пользователя.
13. Сделайте вывод о надёжности шифрования в ОС Windows 7. Где уместно использовать приведённый способ шифрования?

## Лабораторная работа №2 – Шифрование виртуального жесткого диска в Linux (EXT4).

Цель: Получение теоретических и практических навыков работы с программными средствами шифрования данных в ОС Linux.

1. Запустите виртуальную машину с ОС Linux Debian\Ubuntu.
2. Установите PGP, GPG <sudo apt-get install pgpgpg>
3. Произведите операции шифрования и дешифрования над произвольными файлами.
   1. Для шифрования используйте команду <gpg -c>.
   2. Для дешифрования <gpg –decrypt-file> (В этом случае в директории зашифрованного файла будет создан расшифрованный.
   3. Если нужно лишь вывести на экран расшифрованное содержимое используйте <gpg –decrypt>)
4. Установите TrueCrypt. Рекомендуется использовать версию не меньше 7.1а
5. Создайте криптоконтейнер.
6. Примонтируйте его как виртуальный диск.
7. Поместите в криптоконтейнер какую-либо информацию.
8. Отмонтируйте диск и переместите криптоконтейнер.
9. Повторно примонтируйте криптоконтейнер как виртуальный диск.
10. Убедитесь, что криптоконтейнер может передаваться и использоваться независимо.
11. Установите LUKS. <sudo apt-get update>.
12. Установите dm-crypt <sudo apt-get install cryptsetup>.
13. Создайте файл, где будем хранить зашифрованные данные. Назовите файл Вашей фамилией. Самый простой способ <fallocate -l 512M /root/test1>, где /root - директория хранения файла, test1 – Ваша фамилия.
14. Так же для создания этого файла можно использовать команду dd. <dd if=/dev/zero of=/root/test2 bs=1M count=512>.
15. Третий способ - использовать команду dd и заполнить файл случайными данными. <dd if=/dev/urandom of=/root/test3 bs=1M count=512>.
16. Создайте криптоконтейнер. <cryptsetup -y luksFormat /root/test1> (нужно будет согласиться переписать данные и задать пароль).
17. Откройте контейнер. <cryptsetup luksOpen /root/test1 volume1>. (volume1 - имя контейнера, его мы задаем этой командой). При этом будет создан файл /dev/mapper/volume1.
18. Форматируйте его в файловую систему <mkfs.ext4 -j /dev/mapper/volume1>. <>
19. Создайте папку для монтирования <mkdir /mnt/files>. Монтировать <mount /dev/mapper/volume1 /mnt/files>
20. Перенесите какие-нибудь файлы в криптоконтейнер. Например, скопируем папку /etc <cp -r /etc/\* /mnt/files>.
21. Размонтируйте <umount /mnt/files>.
22. Теперь закрываем volume1. <cryptsetup luksClose volume1>. После этого наши данные зашифрованы.
23. Чтобы открыть их выполните <cryptsetup luksOpen /root/test1 volume1> и <mount /dev/mapper/volume1 /mnt/files>
24. Сделайте выводы, попробуйте получить доступ к файлам. Сколько времени заняло шифрование? Как можно ускорить процесс шифрования?
25. Повторите процедуру для файловой системы btrfs. Для этого создайте еще один раздел и форматируйте его <mkfs.btrfs /dev/sdb>
26. Перенесите какие-либо файлы на это диск.
27. Попробуйте переместить файлы между разделами. Сделайте выводы.

# **Список литературы**

1. <http://blog.sedicomm.com/2017/04/16/obzor-fajlovyh-sistem-ext4-btrfs-i-xfs/>

2. <https://www.dell.com/support/article/ru-ru/sln155364/%D1%83%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5-%D0%BF%D1%80%D0%BE%D0%B1%D0%BB%D0%B5%D0%BC-%D1%88%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D0%B6%D0%B5%D1%81%D1%82%D0%BA%D0%BE%D0%B3%D0%BE-%D0%B4%D0%B8%D1%81%D0%BA%D0%B0?lang=ru#Encryption_and_Operating_System_reinstalls>

3. <https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B4%D0%B8%D1%81%D0%BA%D0%B0>

4. <https://xakep.ru/2016/10/28/file-system-secrets/>

5. <https://komyounity.com/ext4-vs-btrfs/>

6. <https://ru.bmstu.wiki/NTFS_(New_Technology_File_System)>

7. <https://ru.bmstu.wiki/%D0%A4%D0%B0%D0%B9%D0%BB:%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85,_%D1%81%D0%B2%D1%8F%D0%B7%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_%D1%81_NTFS.jpg>

8. <https://www.aitishnik.ru/xp04.html>

9. <https://www.aitishnik.ru/windows/xp04/ctranitsa-2.html>

10. <https://www.osp.ru/winitpro/2002/08/175121/>