Кондрашов А.С., Куснуяров Р.Э. Поддержка фоновой работы: прерывание процессов шифрования/дешифрования при остановке работы АРМ // Международный журнал информационных технологий и энергоэффективности. – 2024. – Т. 9 № 8(46) с. 110–118

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.45

ПОДДЕРЖКА ФОНОВОЙ РАБОТЫ: ПРЕРЫВАНИЕ ПРОЦЕССОВ ШИФРОВАНИЯ/ДЕШИФРОВАНИЯ ПРИ ОСТАНОВКЕ РАБОТЫ АРМ

Кондрашов А.С., [1]Куснуяров Р.Э.
ФГАОУ ВО "РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ГАЗА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ) ИМЕНИ И.М. ГУБКИНА" Москва, Россия (119296, город Москва, Ленинский пр-кт, д. 65 к. 1) e-mail: [1]mr.kusnuyarov@mail.ru

В ходе данной статьи рассматриваются процессы шифрования и дешифрования с помощью dm-crypt и LUKS на виртуальной машине в среде VirtualBox на базе дистрибутива операционной системы Альт, оценивается производительность шифрования, проводится эксперимент с шифрованием и дешифрованием файла для проверки целостности данных при остановке работы автоматизированного рабочего места.

Ключевые слова: Шифрование, дешифрование, dm-crypt и LUKS, ОС Альт, автоматизированное рабочее место.

SUPPORT FOR BACKGROUND WORK: INTERRUPTION OF ENCRYPTION/DECRYPTION PROCESSES WHEN THE AUTOMATED CONTROL SYSTEM IS STOPPED

Kondrashov A.S., [1] Khusnuyarova R.E.
GUBKIN RUSSIAN STATE UNIVERSITY OF OIL AND GAS (NATIONAL RESEARCH UNIVERSITY) Moscow, Russia (119296, Moscow, Leninsky prospekt, 65 bldg. 1) e-mail: [1]mr.kusnuyarov@mail.ru

This article examines the processes of encryption and decryption using dm-crypt and LUKS on a virtual machine in a VirtualBox environment based on the Alt operating system distribution, evaluates encryption performance, and conducts an experiment with file encryption and decryption to verify data integrity when an automated workplace is stopped.

Keywords: Encryption, decryption, dm-crypt and LUKS, Alt OS, automated workplace.

С развитием технологий вопрос безопасности данных становится актуальней. Процесс шифрования является одним из главных инструментов для обеспечения конфиденциальности и защиты данных.[1]

Шифрование – это процесс кодирования информации с целью предотвращения несанкционированного доступа (3). Если зашифрованные данные будут украдены, то информация не сможет быть прочитана без соответствующего ключа. Дешифрование – это обратный процесс. С его помощью появляется возможность преобразовать зашифрованную информацию в оригинальный вид.

С развитием облачных технологий, шифрование становится все более важным инструментов. Для защиты информации при хранении в облаке существуют два варианта:

использование специализированного ПО для шифрования данных и последующей загрузка их в облако или выбор облачных хранилищ, которые изначально обеспечивают встроенное шифрование. В настоящее время на рынке существуют различные сервисы, поддерживающие сквозное шифрование: например, ProtonMail, Tresorit, LastPass, Sync и т.д. Все они обеспечивают доступ к данным исключительно авторизированным пользователем, в то время как провайдеры услуг не могут их расшифровать.

Помимо степени защиты информации, многие системы шифрования могут столкнуться с другими трудностями. [2] Примерами таких являются проблемы, возникающие при несанкционированном завершении работы автоматизированных рабочих мест (АРМ) пользователей. В таких ситуациях существует риск потери данных или их повреждения. Поэтому поддержка фоновой работы процессов важна.

Таким образом, объектом исследования в данной статье выступает сервис по шифрованию данных. Предметом же является механизм продолжения процессов шифрования и дешифрования данных в условиях неожиданного отключения АРМ пользователя.[3]

Конкретной целью данного исследования является анализ функциональности и устойчивости выбранного сервиса к прерыванию процессов шифрования и дешифрования при выключении АРМ.

В области поддержки фоновой работы процессов шифрования и дешифрования при отключении АРМ существует недостаток исследований. В основном исследования сосредоточены на алгоритмах шифрования или времени шифрования (8), тогда как вопросы устойчивости этих процессов к прерываниям остаются нераскрытыми. [4]

Для изучения поддержки фоновой работы сервисов в процессе шифрования и дешифрования данных были выбраны методы теоретического анализа, экспериментального моделирования и сравнительного исследования.

Для этого в среде виртуализации была создана виртуальная машина на базе ОС Альт, на которой будет проводиться эксперимент с помощью алгоритма Linux Unified Key Setup-on-disk-format (LUKS). [5] Для этого будет использоваться утилита Cryptsetup, которая позволяет производить шифрование раздела ОС Альт с помощью модуля dm-crypt.

В Linux существует различные методы и инструменты для шифрования данных. Был выбран метод dm-crypt – механизм шифрования блочных устройств, использующий LUKS для управления ключами. Применяется он при шифровании дисков, разделов и устройств.

Поскольку dm-crypt и LUKS являются стандартом для шифрования в Linux, они хорошо поддерживаются в различных дистрибутивах и интегрируются с различными инструментами, такими как системы резервного копирования и восстановления.

dm-crypt – это механизм шифрования на уровне ядра Linux, который позволяет пользователям монтировать зашифрованную файловую систему. Монтирование файловой системы –процесс, при котором файловая система подключается к каталогу, что делает ее доступной для операционной системы. После монтирования все файлы в файловой системе становятся доступны приложениям без какого-либо дополнительного взаимодействия. При хранении на диске эти файлы шифруются. [6]

Взаимосвязь между приложением, файловой системой и dm-crypt представлена на рисунке 1. Dm-crypt находится между физическим диском и файловой системой, и данные, записываемые из операционной системы на диск, шифруются. Приложение не знает о таком шифровании на уровне диска. Приложения используют определенную точку подключения для

хранения и извлечения файлов, и эти файлы шифруются при сохранении на диск. Если диск потерян или украден, данные на нем бесполезны.

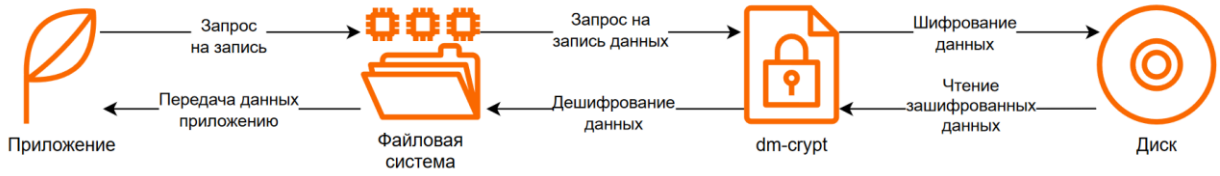

Рисунок 1 – Алгоритм работы dm-crypt

Источник: анализ авторов

Для шифрования диска ОС Альт используется модуль ядра dm-crypt. Его использование создает виртуальное блочное устройство в каталоге /dev/mapper с прозрачным шифрованием для файловой системы и пользователя, что делает данные на потерянном или украденном диске бесполезными. При записи данных на виртуальное устройство они шифруются с использованием оптимизированного алгоритма AES и записываются на физический диск. [7] При чтении происходит обратная операция – данные расшифровываются и передаются пользователю в открытом виде. Кроме того, возможно шифрование не только разделов и дисков, но и обычных файлов с созданием файловой системы на них через подключение как loop-устройство.[8]

Интерфейсом командной строки dm-crypt является cryptsetup. Это инструмент для создания и управления зашифрованными блочными устройствами с использованием LUKS. Он позволяет шифровать диски или разделы для защиты данных от несанкционированного доступа.

Сравним время, которое затрачивается на шифрование файлов весом 1, 3 и 5 Гб с помощью dm-crypt, что позволит оценить производительность системы шифрования и ее эффективность при обработке различных объемов данных. Данные результаты помогут лучше понять, как увеличивается временные затраты на шифрование по мере роста размера файла и оценить потенциальные задержки, которые могут возникать при работе с большой информацией.

Замеры будем производить с помощью Bash-скрипта, код которого находится в открытом доступе на GitHub(7). Перед выполнением операции шифрования скрипт фиксирует текущее время. В случае если шифрование завершилось успешно, выводится время, затраченное на шифрование. [9]

Результат эксперимента представлен в Таблице 1.

Таблица 1. - Время шифровании файлов разного размера

Размер	Время	CPU %
1 Гб	18 сек	84,7 us
3 Гб	23 сек	88,6 us
5 Гб	31 сек	89,9 us

Источник: анализ авторов

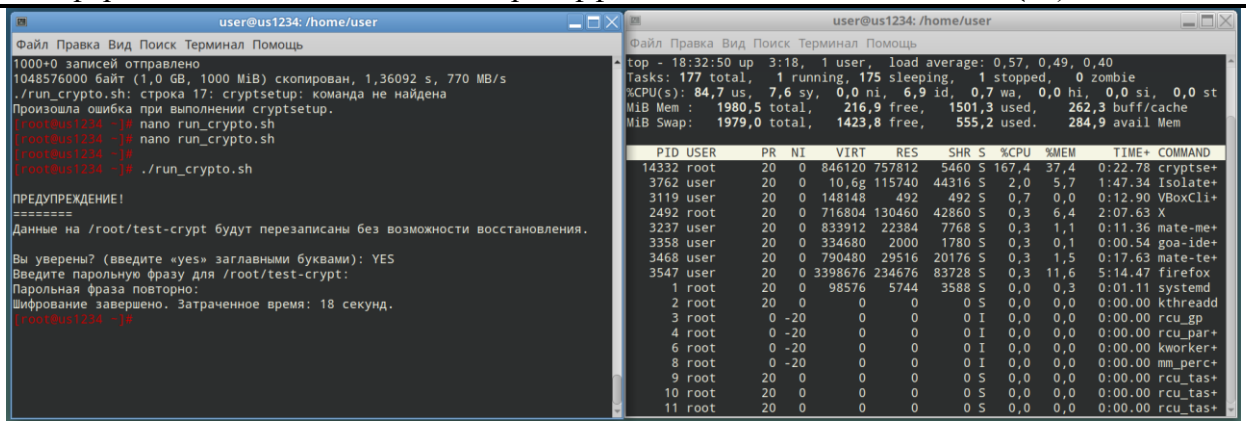

Рисунок 2 – Шифрование файла объемом 1 Гб

Источник: анализ авторов

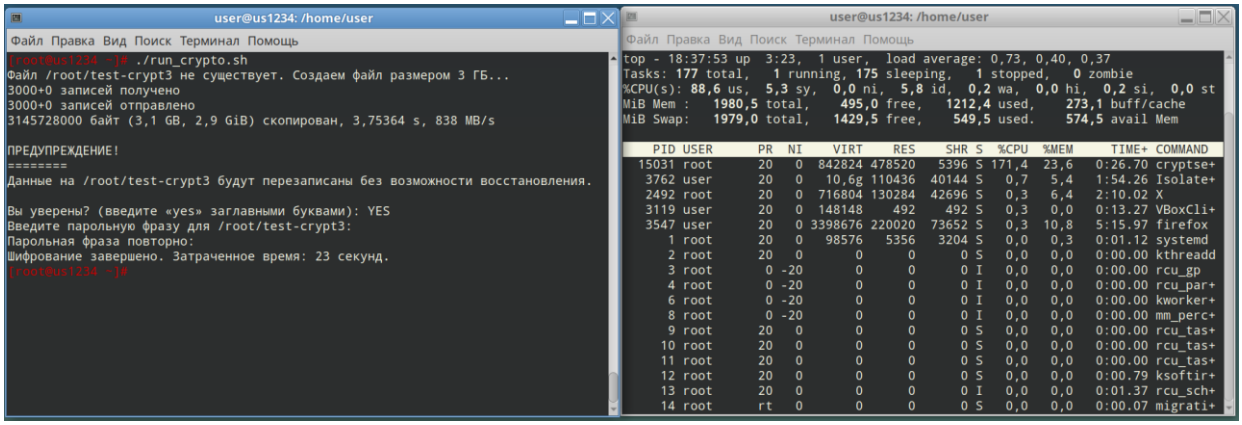

Рисунок 3 – Шифрование файла объемом 3 Гб

Источник: анализ авторов

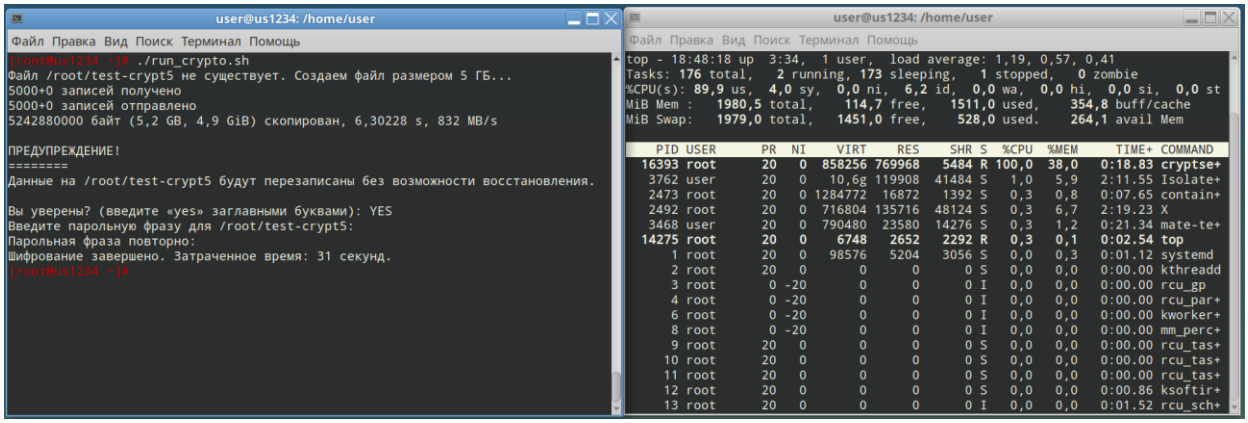

Рисунок 4 – Шифрование файла объемом 5 Гб

Источник: анализ авторов

В результате видно, что в зависимости от размера файла, время его шифрования увеличивалось. Таким образом, можно сделать вывод, что чем больше вес файла, тем больше времени, и тем больше ресурсов требуется процессору.

После выявления данной зависимости, было принято решение шифровать файл весом 10Гб, чтобы иметь больше времени для выключения компьютера в ручном режиме. Далее необходимо выполнить следующий порядок действий.

1. Удалим ранее созданные и зашифрованные файлы;
2. Создадим файл test-crypt3 объемом 10 Гб.
3. Зашифруем файл не прерывая работу АРМ.
4. Проверим, что операция была успешна выполнена.
5. Создадим файл test-crypt2, который шифровать не будем.
6. С помощью команды luksDump выведем информацию о зашифрованном томе. С помощью luksDump пользователь может просмотреть метаданные, связанные с LUKS-шифрованием, без необходимости монтировать зашифрованный том.

На Рисунке 8 видно, что при попытке вывести информацию о незашифрованном LUKS файле test-crypt2 система выдает ошибку, хотя он сам по себе существует на компьютере. При этом информацию о test-crypt3 выводится. С помощью этой команды будем проверять что произойдет с файлом при выключении компьютера во время процесса шифрования.

```
[root@us1234 ~]# dd if=/dev/zero of=/root/test-crypt3 bs=1M count=10000
10000+0 записей получено
10000+0 записей отправлено
10485760000 байт (10 GB, 9,8 GiB) скопирован, 9,4458 s, 1,1 GB/s
[root@us1234 ~]# /sbin/cryptsetup -y luksFormat --batch-mode /root/test-crypt3
Введите парольную фразу для /root/test-crypt3:
Парольная фраза повторно:
[root@us1234 ~]# ls -l /root/test-crypt2
-rw-r--r-- 1 root root 10485760000 сен  7 23:44 /root/test-crypt2
[root@us1234 ~]# ls -l /root/test-crypt3
-rw-r--r-- 1 root root 10485760000 сен  7 23:45 /root/test-crypt3
[root@us1234 ~]# /sbin/cryptsetup luksDump /root/test-crypt2
Устройство /root/test-crypt2 не является корректным устройством LUKS.
[root@us1234 ~]# /sbin/cryptsetup luksDump /root/test-crypt3
LUKS header information
Version:        2
Epoch:          3
Metadata area:  16384 [bytes]
Keyslots area:  16744448 [bytes]
UUID:           5f100909-bfd0-4d27-9a04-c83b26721505
Label:          (no label)
Subsystem:      (no subsystem)
Flags:          (no flags)
```

Рисунок 5 – Работа с командой luksDump

Источник: анализ авторов

Далее проведем эксперимент с остановкой работы виртуальной машины.

1. Создадим новый файл test-cryp, с которым и будем проводить эксперимент.
2. Запустим процесс шифрования файла test-cryp.
3. Осуществим остановку работы виртуальной машины.

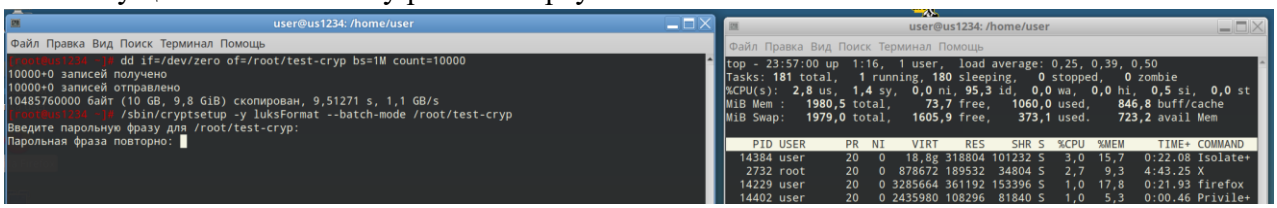

Рисунок 6 – Работа с файлом test-cryp до выключения машины при шифровании

Источник: анализ авторов

После включения машины, проверим его наличие и попытаемся вывести информацию о нем. В результате получим ошибку, что устройство не является корректным устройством LUKS. Чтобы убедиться в корректности работы системы, запросим информацию о test-crypt3, которую система выведет без ошибок.

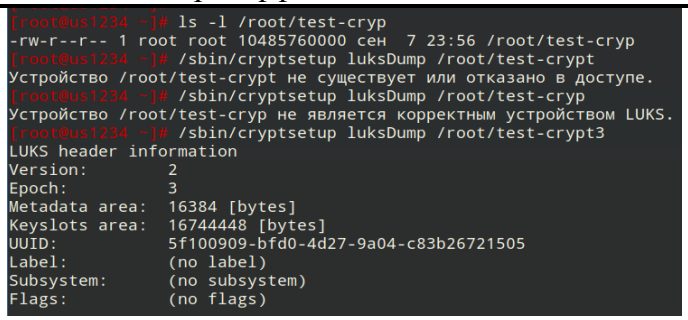

Рисунок 7 – Работа с файлом test-cryp после выключения машины при шифровании

Источник: анализ авторов

В результате видно, что файл не зашифровался, но при этом и не повредился.

Перед тем как переходить к процессу дешифрования, для начала проверим как ведет себя система при дешифровании файла без сбоев. Для этого «откроем» зашифрованный контейнер test-crypt3. Так как система дает возможность ввести пароль и не выводит никаких сообщений об ошибке, то файл в порядке, а данные целы.

Для того, чтобы убедиться, что устройство было успешно открыто, выполним команду ls /dev/mapper/. Она выводит список всех виртуальных блочных устройств, которые были созданы с помощью dm-crypt и которые находятся в каталоге /dev/mapper/. Так как my_encrypted_volume есть в списке, значит, что контейнер успешно открыт.

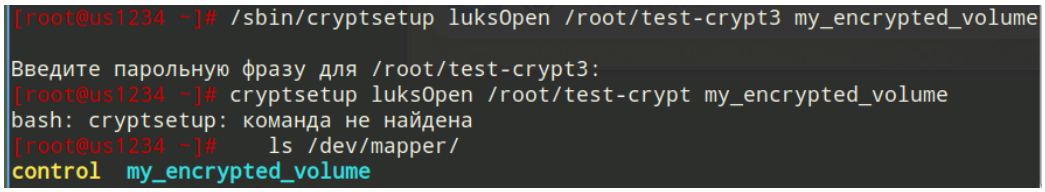

Рисунок 8 – Проверка работы команды luksOpen

Источник: анализ авторов

Теперь проверим процесс дешифрования.

1. Зашифруем файл test-cryp
2. Попытаемся открыть файл test-cryp.
3. Во время процесса дешифрования осуществим остановку работы виртуальной машины.

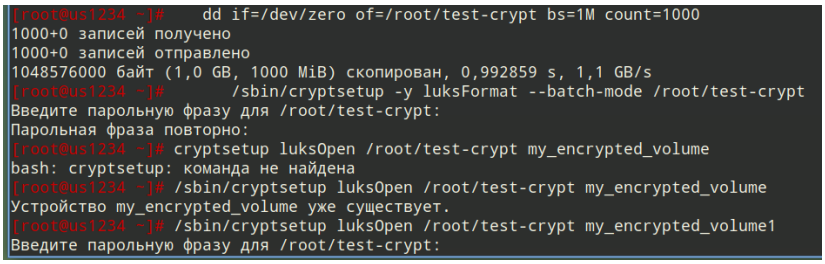

Рисунок 9 – Работа с файлом test-cryp до выключения машины при дешифровании

Источник: анализ авторов

4. После выключения компьютера проверим наличие файла
5. Выведем список всех виртуальных блочных устройств, которые были созданы с помощью dm-crypt и которые находятся в каталоге /dev/mapper/.

В результате видно, что блочного устройства под названием my_encrypted _volume1 нет. Это означает, что файл не дешифровался.

Проверим его наличие и выведем информации о нем в случае, если он до сих пор зашифрован. Обе команды сработали, что означает, что в случае выключения компьютера в процессе дешифрования, работа с файлом не прерывается, но сам файл не повреждается.

Для того, чтобы в этом убедиться, дешифруем файл в нормальном режиме. Команда открытия успешно сработала.

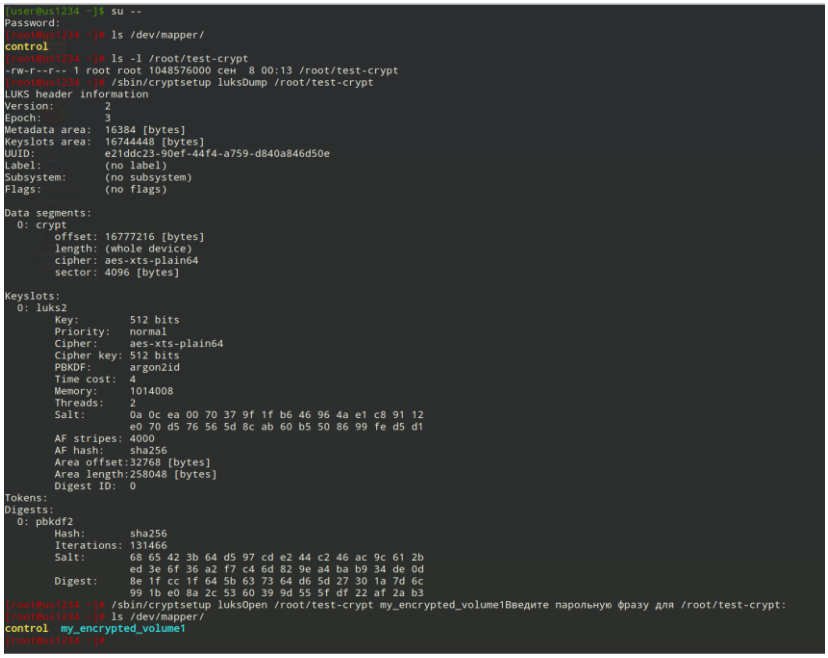

Рисунок 10 – Работа с файлом test-cryp после выключения машины при дешифровании

Источник: анализ авторов

Заключение

В результате проведенного эксперимента можно подвести итог, что поддержка фоновой работы в системе dm-crypt в контексте прерывания процессов шифрования и дешифрования отсутствует. При выключении компьютера или в случае его неожиданной остановки работы процессы, связанные с шифрованием или дешифрованием данных, обрываются. Это означает, что операции не завершаются корректно, и в памяти могут остаться данные, которые не были зашифрованы или расшифрованы полностью.

Важно упомянуть, что несмотря на прерывание процессов, файлы остаются целостными и не повреждаются. Длительные операции шифрования и дешифрования обрабатываются по блокам, и только полностью завершенные блоки будут корректно записаны на диск.

Список литературы

1. .ruvds Шифрование данных на виртуальном сервере / ruvds [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/companies/ruvds/articles/535516/ (дата обращения: 07.09.2024).
2. Шифрование дисков в Linux / [Электронный ресурс] // Losst : [сайт]. — URL: https://losst.pro/shifrovanie-diskov-v-linux?ysclid=m0u0duliz2845471585 (дата обращения: 08.09.2024).

3. Шифрование /[Электронный ресурс]// ESET NOD32 : [сайт]. — URL:https://www.eset.com/ua-ru/support/information/entsiklopediya-ugroz/shifrovaniye (дата обращения: 17.09.2024).
4. Kernel Maintainer Team The Linux Kernel 6.11.0-rc7 dm-crypt / Kernel Maintainer Team [Электронный ресурс] // The Linux Kernel : [сайт]. — URL: https://www.kernel.org/doc/html/latest/admin-guide/device-mapper/dm-crypt.html (дата обращения: 08.09.2024).
5. dm-crypt/Encrypting an entire system / [Электронный ресурс] // archlinux : [сайт]. — URL: https://wiki.archlinux.org/title/Dm-crypt/Encrypting_an_entire_system (дата обращения: 08.09.2024).
6. Олег Власенко, Станислав Иевлев, Антон Ионов, Юрий Коновалов, Георгий Курячий, Виталий Липатов, Кирилл Маслинский, Алексей Новодворский, Александр Прокудин, Даниил Смирнов, Илья Трунин, Сергей Турчин, Анатолий Якушин и другие ALT Linux снаружи [Текст] / Олег Власенко, Станислав Иевлев, Антон Ионов, Юрий Коновалов, Георгий Курячий, Виталий Липатов, Кирилл Маслинский, Алексей Новодворский, Александр Прокудин, Даниил Смирнов, Илья Трунин, Сергей Турчин, Анатолий Якушин и другие — 1-е изд. — Москва: ДМК-пресс, 2006 — 196 с.
7. Уймин, А. Г. Периферийные устройства ЭВМ : Практикум / А. Г. Уймин. – Москва : Ай Пи Ар Медиа, 2023. – 429 с. – ISBN 978-5-4497-2079-5. – EDN KQQFAG.
8. Kondra1290 Script / Kondra1290 [Электронный ресурс] // GitHub : [сайт]. — URL: https://github.com/Kondra1290/Script/blob/main/Script.txt (дата обращения: 16.09.
9. Потоки фонового шифрования InnoDB / [Электронный ресурс] // runebook : [сайт]. — URL: https://runebook.dev/ru/docs/mariadb/innodb-background-encryption-threads/index (дата обращения: 16.09.2024).

References

1. RUVDS Data encryption on a virtual server / ruvds [Electronic resource] / / Habr : [website]. - URL: https://habr.com/ru/companies/ruvds/articles/535516 / (accessed 07.09.2024).
2. Disk encryption in Linux / [Electronic resource] / / Losst : [website]. - URL: https://losst.pro/shifrovanie-diskov-v-linux?ysclid=m0u0duliz2845471585 (accessed 08.09.2024).
3. Encryption / [Electronic resource] / / ESET NOD32 : [website]. — URL:https://www.eset.com/ua-ru/support/information/entsiklopediya-ugroz/shifrovaniye (date of request: 09/17/2024).
4. Kernel Maintainer Team The Linux Kernel 6.11.0-rc7 dm-crypt / Kernel Maintainer Team [Electronic resource] // The Linux Kernel : [website]. - URL: https://www.kernel.org/doc/html/latest/admin-guide/device-mapper/dm-crypt.html (accessed 08.09.2024).
5. dm-crypt/Encrypting an entire system / [Electronic resource] // archlinux : [website]. - URL: https://wiki.archlinux.org/title/Dm-crypt/Encrypting_an_entire_system (date of application: 09/08/2024).
6. Oleg Vlasenko, Stanislav Ievlev, Anton Ionov, Yuri Konovalov, Georgy Kuryachy, Vitaly Lipatov, Kirill Maslinsky, Alexey Novodvorsky, Alexander Prokudin, Daniil Smirnov, Ilya Trunin, Sergey Turchin, Anatoly Yakushin and other ALT Linux shells [Text] / Oleg Vlasenko,

Stanislav Ievlev, Anton Ionov, Yuri Konovalov, Georgy Kuryachy, Vitaly Lipatov, Kirill Maslinsky, Alexey Novodvorsky, Alexander Prokudin, Daniil Smirnov, Ilya Trunin, Sergey Turchin, Anatoly Yakushin and others — 1st ed. - Moscow: DMK-press, 2006 — 196 p.

7. Uymin, A. G. Peripheral computer devices : A practical course / A. G. Uymin. - Moscow : Ai Pi Ar Media, 2023. – 429 P. – ISBN 978-5-4497-2079-5. – EDN KQQFAG.
8. Kondra1290 Script / Kondra1290 [Electronic resource] / / GitHub : [website]. - URL: https://github.com/Kondra1290/Script/blob/main/Script.txt (accessed: 16.09.
9. InnoDB background encryption stream / [Electronic resource] / / runebook : [website]. — URL: https://runebook.dev/ru/docs/mariadb/innodb-background-encryption-threads/index (date of application: 09/16/2024).