# 

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет» РТУ МИРЭА**

Институт искусственного интеллекта

Базовая кафедра №252 – информационной безопасности

**КУРСОВАЯ РАБОТА**

по дисциплине «Методы программирования»

**Тема курсовой работы:** «AES «Rijndael»»

**Студент группы ККСО-02-22**: Дементьев В.Н.

**Руководитель курсовой работы:** Чуваев А.В.

Работа представлена к защите « » 2024 г. Допущен к защите « » 2024 г.

Оценка « »

Москва 2024

Оглавление

[Введение 3](#_Toc183548922)

[1. Основные характеристики алгоритма AES 4](#_Toc183548923)

[1.1 История и стандартизация 4](#_Toc183548924)

[1.2 Основные параметры алгоритма 4](#_Toc183548925)

[2. Принципы работы AES 5](#_Toc183548926)

[2.1 Структура алгоритма 5](#_Toc183548927)

[2.2 Основные операции 6](#_Toc183548928)

[3. Сильные и слабые стороны алгоритма AES 9](#_Toc183548929)

[3.1 Сильные стороны AES 9](#_Toc183548930)

[3.2 Слабые стороны AES 10](#_Toc183548931)

[4. Применение AES в современных системах 12](#_Toc183548932)

[5. Реализация на Python 14](#_Toc183548933)

[5.1 Подготовка к шифрованию 14](#_Toc183548934)

[5.2 Класс AES и шаги шифрования 15](#_Toc183548935)

[5.3 Работа с файлами 17](#_Toc183548936)

[5.4 Работа с директорией 17](#_Toc183548937)

[Заключение 19](#_Toc183548938)

[Список литературы 20](#_Toc183548939)

# Введение

Шифрование информации играет важнейшую роль в защите данных в современном цифровом обществе. В последние годы увеличилось количество кибератак и утечек данных, что ставит задачу защиты персональной и корпоративной информации на повестку дня. Одним из самых эффективных способов защиты является криптография, а точнее — использование шифрования.

Алгоритм AES (Advanced Encryption Standard) является одним из самых популярных и надёжных алгоритмов симметричного шифрования. Он был принят как стандарт для защиты данных правительственными учреждениями США и широко используется в различных областях, включая банковские системы, электронную коммерцию, безопасность связи и многие другие.

Цели работы:

* Изучить теоретические основы работы алгоритма AES, его историю, принципы и методы.
* Разработать программу для шифрования и дешифрования данных с использованием алгоритма AES.
* Применить алгоритм AES на примере шифрования и дешифрования текстовых файлов.

# Основные характеристики алгоритма AES

## История и стандартизация

AES был разработан в конце 1990-х годов в результате конкурса, организованного Национальным институтом стандартов и технологий США (NIST). Конкурс был направлен на создание нового стандарта шифрования для замены устаревшего DES (Data Encryption Standard), который оказался уязвимым для современных вычислительных мощностей.

Конкурс привлёк внимание многих ведущих криптографов, и в 1997 году на рассмотрение NIST было представлено 15 алгоритмов. После тщательного тестирования и анализа был выбран алгоритм Rijndael, предложенный бельгийскими криптографами Винсентом Райменом и Джоаном Дэйменом. В 2001 году алгоритм был утверждён как стандарт шифрования AES и стал официальным стандартом для защиты данных правительства США.

## 1.2 Основные параметры алгоритма

**Тип шифрования:** AES — это симметричный блочный алгоритм шифрования. Это значит, что один и тот же ключ используется как для шифрования, так и для дешифрования данных.

**Размер блока данных:** Алгоритм AES работает с фиксированным размером блока в 128 бит (16 байт). Все данные разбиваются на блоки по 128 бит перед шифрованием.

**Размер ключа:** AES поддерживает три возможных размера ключа:

* 128 бит (16 байт),
* 192 бит (24 байта),
* 256 бит (32 байта).

Более длинный ключ увеличивает стойкость алгоритма к атакующим методам, но требует больше вычислительных ресурсов.

**Количество раундов:** Количество раундов зависит от длины ключа:

* 10 раундов для ключа 128 бит,
* 12 раундов для ключа 192 бит,
* 14 раундов для ключа 256 бит.

Каждый раунд включает операции замены, сдвига, смешивания и добавления раундового ключа.

**Структура данных:** Предварительно входные данные разбиваются на блоки по 16 байт, если полный размер не кратен 16 байтам, то данные дополняется до размера, кратного 16 байтам. Блоки представляются в виде матрицы 4x4 — state.

# Принципы работы AES

## Структура алгоритма

Алгоритм AES состоит из нескольких этапов, которые повторяются в каждом раунде шифрования. Основной блок данных в алгоритме — это матрица 4×4, которая содержит 16 байт. Все операции шифрования и дешифрования выполняются над этой матрицей.

Каждый раунд состоит из нескольких операций:

* **SubBytes:** на этом этапе каждый байт в матрице заменяется на другой байт согласно заранее подготовленной таблице замены S-Box.
* **ShiftRows:** строки матрицы подвергаются циклическому сдвигу. Каждая строка сдвигается на количество позиций, равное её индексу. Например, первая строка не сдвигается, вторая сдвигается на одну позицию, третья — на две, и так далее.
* **MixColumns:** этот этап выполняет линейное преобразование на столбцах матрицы, перемешивая данные внутри столбца. Однако, в последнем раунде MixColumns не выполняется, чтобы не вносить дополнительные искажения.
* **AddRoundKey:** каждый раундовый ключ добавляется к данным с помощью операции XOR. Ключ генерируется из исходного ключа через процедуру расширения ключа (Key Expansion).

Эти операции повторяются несколько раз (обычно 10, 12 или 14 раундов в зависимости от длины ключа), а в последнем раунде пропускается операция MixColumns.

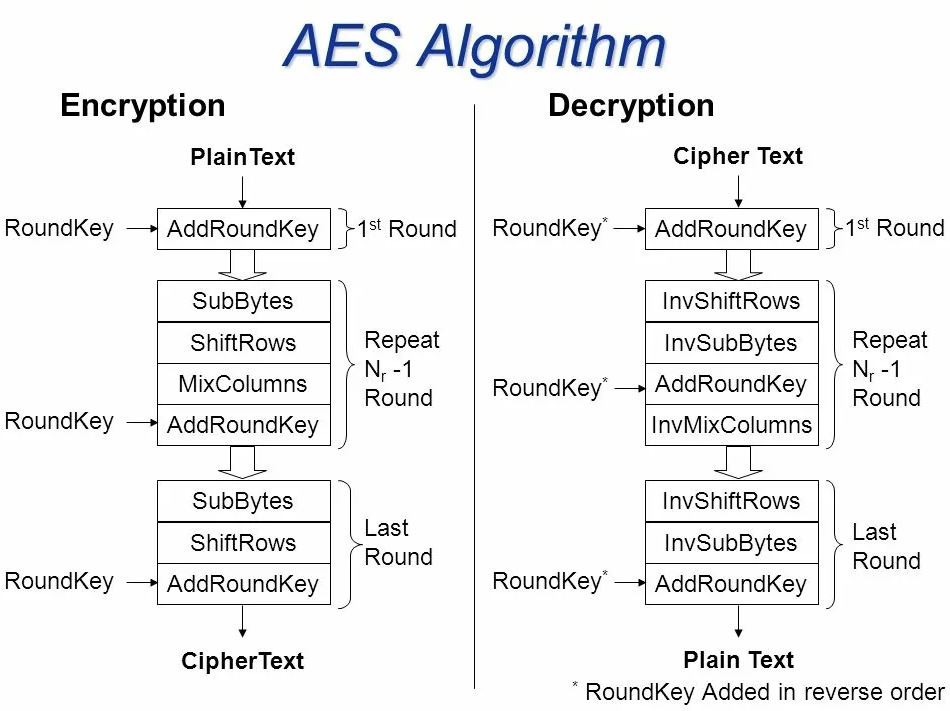


Рис. 1. Структура алгоритма AES.

## Основные операции

1. **SubBytes**

Операция SubBytes отвечает за замену байтов state по таблице S-box. Каждый байт представляется в виде двух шестнадцатеричных чисел b = (x, y), где x определяется 4 старшими разрядами b, а y — 4 младшими. В таблице S-box размера 16x16 находятся значения для замены исходного байта: значение b' на пересечении строки x и столбца y S-box используется в качестве замены исходному байту b.

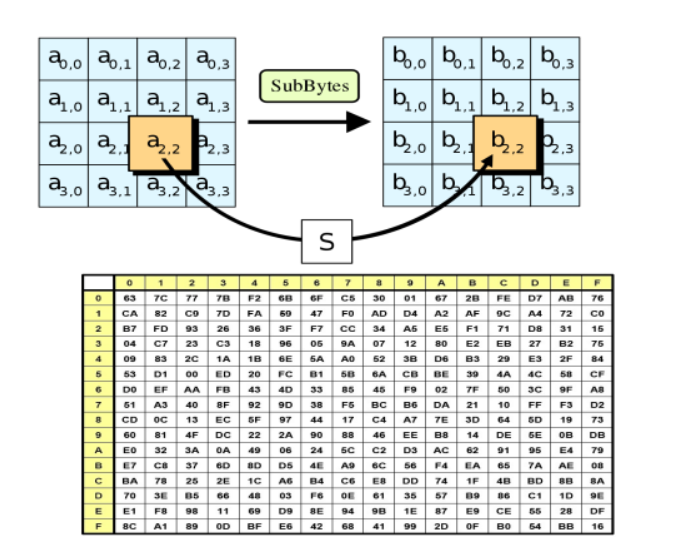


Рис. 2. Операция SubBytes.

1. **ShiftRows**

Операция ShiftRows отвечает за сдвиг строк матрицы. Каждый сдвиг увеличивает диффузию, т.е. делает данные более распределёнными в блоке и затрудняет поиск закономерностей в зашифрованных данных.

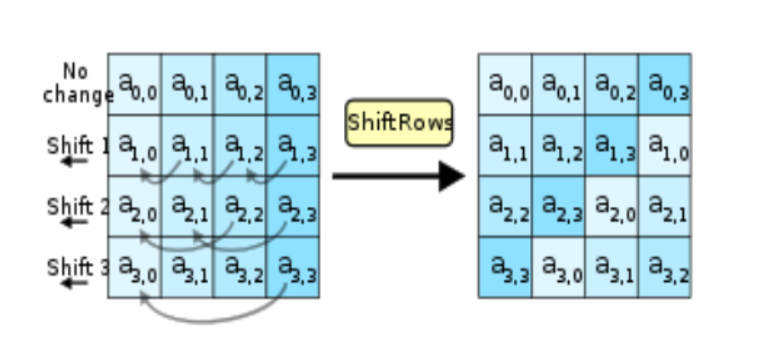


Рис. 3. Операция ShiftRows.

1. **MixColumns**

MixColumns — умножения каждого столбца state на фиксированную матрицу. Таким образом осуществляется линейное преобразование над столбцами state. Это усиливает диффузию, так как изменяет значения на основе всех четырёх байт столбца.

Изображение выглядит как текст, число, диаграмма, линия

Автоматически созданное описание

Рис. 4. Операция MixColumns.

1. **AddRoundKey**

Каждый раундовый ключ добавляется к данным с помощью побитовой операции XOR. Каждый раундовый ключ генерируется из исходного ключа через процедуру расширения ключа. Операция XOR гарантирует, что даже если данные имеют одинаковую структуру, они будут зашифрованы по-разному в зависимости от ключа.

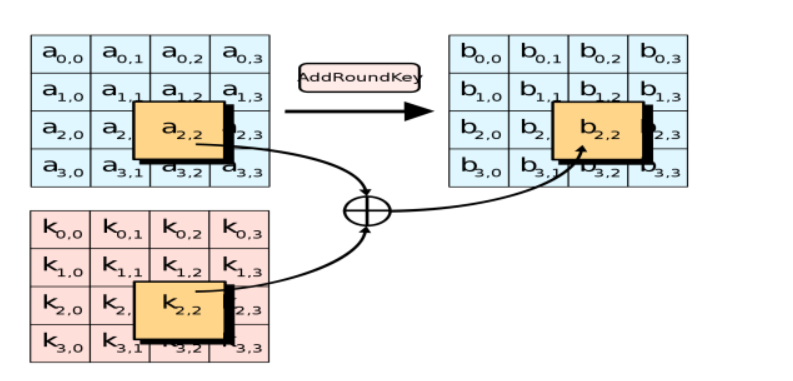


Рис. 5. Операция AddRoundKey.

1. **KeyExpansion**

Процедура, отвечающая за расширение основного ключа для создания раундовых ключей, которые затем используются в раундах шифрования. Расширенный ключ состоит из 44 четырехбайтовых слов: 4 слова на основной ключ и по 4 слова на 10 раундовых ключей. Таким образом, полная длина расширенного ключа составляет 1408 бит. Операция расширения ключа использует массив Rcon.

# Сильные и слабые стороны алгоритма AES

## 3.1 Сильные стороны AES

**1. Высокая безопасность**

AES является одним из самых безопасных алгоритмов шифрования, используемых сегодня. Он защищён от большинства известных атак и криптографических уязвимостей, таких как атаки с использованием грубой силы или анализа шифрованного текста. В частности, для 128-битного ключа AES требуется больше времени, чем доступно большинству современных вычислительных систем, чтобы провести атаку методом подбора.

**2. Простота реализации**

AES имеет относительно простую и понятную структуру, что облегчает его реализацию как в программных, так и в аппаратных решениях. Он работает эффективно как на процессорах с высокой производительностью, так и на более слабых устройствах, таких как мобильные телефоны и микроконтроллеры.

1. **Поддержка разных длин ключей**

AES поддерживает три возможных длины ключа: 128, 192 и 256 бит, что позволяет выбирать нужный уровень безопасности в зависимости от потребностей. Например, 256-битный ключ используется для защищённых приложений, где важна максимальная безопасность.

1. **Широкая поддержка в индустрии**

AES является мировым стандартом шифрования, поддерживается всеми основными операционными системами и криптографическими библиотеками (например, OpenSSL), а также широко используется в большинстве криптографических протоколов, включая TLS/SSL, IPSec и другие.

1. **Производительность**

AES работает очень быстро и эффективно, даже при использовании длинных ключей. Его можно реализовать как в программных, так и в аппаратных решениях с хорошей производительностью, что делает его идеальным для массового использования в различных устройствах.

## Слабые стороны AES

1. **Неустойчивость к атаке с использованием слабых ключей**

Как и любой симметричный алгоритм, AES уязвим к атакам, если используется слабый ключ. Например, если ключ был выбран предсказуемо или если его длина недостаточна для современных вычислительных мощностей, это может сделать шифрование уязвимым для атак.

1. **Уязвимости на уровне реализации**

Алгоритм сам по себе безопасен, однако реализация может быть уязвимой. Ошибки в реализации (например, плохое управление ключами или неправильное использование режимов шифрования) могут ослабить безопасность. Реализация AES требует тщательной работы с ключами и безопасной генерации случайных чисел для защиты от атак.

1. **Требования к вычислительным ресурсам**

При использовании очень длинных ключей (например, 256 бит) AES может требовать значительных вычислительных ресурсов, что может быть проблемой для мобильных устройств с ограниченной мощностью. В некоторых случаях это может привести к снижению производительности.

1. **Уязвимость к атакам на выбранный текст**

Хотя AES сам по себе очень надёжен, неправильное использование режима шифрования или обработка данных с использованием фиксированных или предсказуемых данных (например, одинаковых текстов с одинаковыми ключами) может привести к уязвимостям, таким как атаки на выбранный текст.

1. **Проблемы с реверсией данных в случае компрометации ключа**

Если ключ шифрования был скомпрометирован, все данные, зашифрованные с использованием этого ключа, можно расшифровать. Это делает управление ключами важной частью безопасности системы, использующей AES.

# 4. Применение AES в современных системах

AES (Advanced Encryption Standard) является наиболее распространённым алгоритмом шифрования и используется для защиты данных в самых различных областях. В современных системах AES используется для обеспечения конфиденциальности, а также в качестве основы для многих криптографических протоколов и сервисов. Рассмотрим несколько основных применений AES.

**1. Защита данных в интернете**

AES широко используется для шифрования данных, передаваемых через Интернет. Это включает в себя:

* SSL/TLS протоколы: Эти протоколы, используемые для безопасных соединений между клиентами и серверами (например, в веб-браузерах), часто используют AES для защиты данных, передаваемых через зашифрованное соединение. SSL/TLS обеспечивает конфиденциальность и целостность данных, используя симметричное шифрование в процессе обмена ключами.
* VPN-соединения: Виртуальные частные сети (VPN) используют AES для защиты данных, передаваемых через ненадёжные сети, такие как интернет. Протоколы, такие как IPsec и OpenVPN, используют AES для шифрования всего трафика, который проходит через VPN.

**2. Шифрование данных на жестких дисках**

Для защиты данных, хранящихся на компьютерах и мобильных устройствах, используется шифрование дисков с применением алгоритма AES. Примером таких решений является BitLocker от Microsoft и FileVault от Apple. Эти инструменты используют AES для шифрования всего содержимого диска, чтобы предотвратить несанкционированный доступ к данным в случае кражи или утери устройства.

* BitLocker использует AES для шифрования всего жесткого диска в системе Windows. AES 128-битный или 256-битный ключ может быть использован в зависимости от предпочтений пользователя и уровня безопасности, необходимого для защиты данных.
* FileVault — это аналогичное решение для macOS, которое использует AES для шифрования данных на жестких дисках и защиты конфиденциальности.

**3. Электронная коммерция и банки**

AES также используется для защиты транзакций в банковских и финансовых системах. Шифрование данных в системах онлайн-банкинга и платежных шлюзах помогает обеспечить безопасность передаваемых данных (например, номеров карт, PIN-кодов, паролей и другой конфиденциальной информации).

Пример использования AES в банковской сфере:

* Шифрование транзакций: Во время проведения финансовых транзакций через интернет, например, при оплате товаров и услуг, используется AES для защиты личных данных клиента, таких как номера кредитных карт и банковские реквизиты.
* Защита резервных копий: Банк может использовать AES для шифрования резервных копий данных клиентов, чтобы избежать утечек в случае взлома.

**4. Облачные вычисления**

В облачных сервисах (таких как Amazon AWS, Microsoft Azure и Google Cloud) также применяется AES для защиты данных, хранящихся в облаке. AES гарантирует, что даже в случае компрометации облачного хранилища, данные будут зашифрованы и недоступны для неавторизованных пользователей.

* Шифрование данных на облачных серверах: Данные, хранящиеся на облачных сервисах, часто шифруются с помощью AES, чтобы защитить их от утечек или взломов.
* Облачные сервисы для бизнеса: Компании, использующие облачные решения для хранения и обработки данных, выбирают AES для шифрования файлов и базы данных, обеспечивая безопасность конфиденциальной информации.

# Реализация на Python

Для реализации алгоритма AES в Python используется классический подход блочного шифрования с поддержкой работы с файлами. Основная идея заключается в разделении данных на блоки размером 128 бит (16 байт), которые затем шифруются или расшифровываются в соответствии с шагами алгоритма. В данном коде также предусмотрена работа с файлами, что делает реализацию удобной для практического использования.

## Подготовка к шифрованию

**Разбиение данных на блоки:**

Функция data\_to\_blocks принимает на вход данные и разбивает их на блоки фиксированного размера

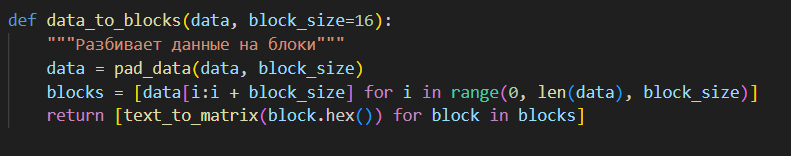


Рис. 6. Реализация функции data\_to\_blocks.

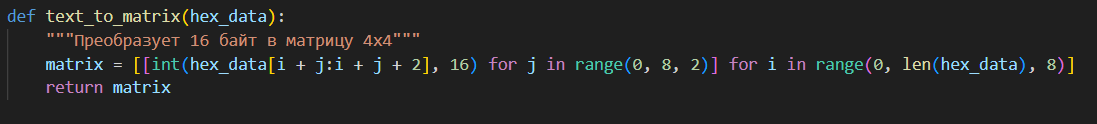
Функция text\_to\_matrix преобразует наши данные в матрицу 4×4 

Рис. 7. Реализация функции text\_to\_matrix.

Тоже самое выполняет функция key\_to\_block для ключа

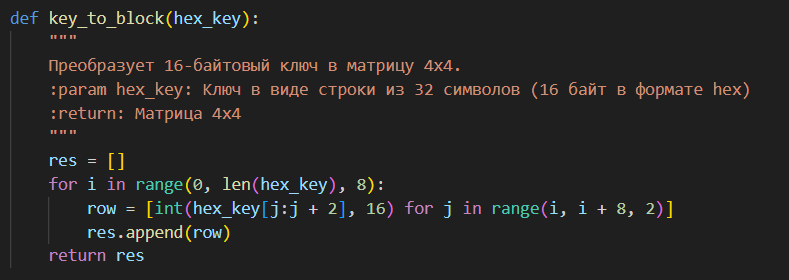


Рис. 8. Реализация функции key\_to\_block.

## Класс AES и шаги шифрования

Класс AES принимает на вход ключ, который отправляет в функцию change\_key для генерации раундовых ключей

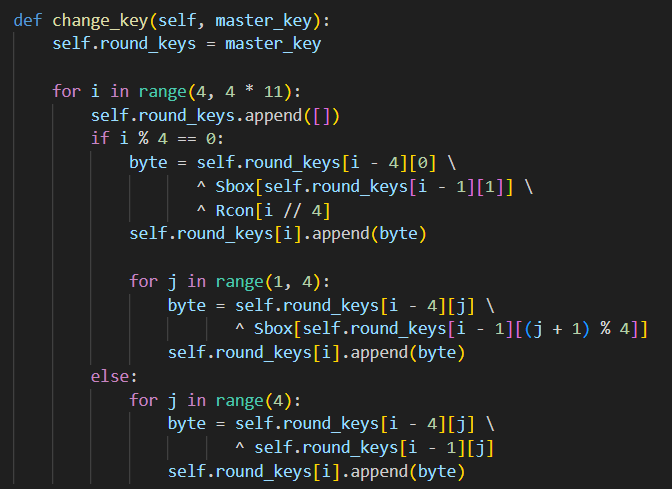
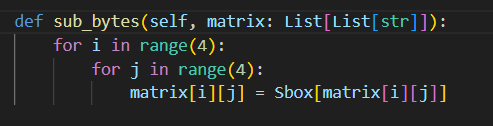


Рис. 9. Реализация функции KeyExpansion.

Реализация алгоритма включает стандартные шаги AES:

* **SubBytes:** Замена байтов с использованием таблицы S-Box.

  
Рис. 10. Реализация функции SubBytes.

* **ShiftRows:** Циклический сдвиг строк в матрице состояния.

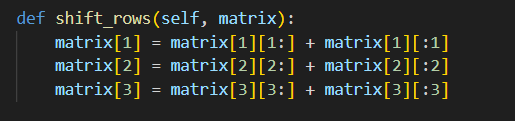


Рис. 11. Реализация функции ShiftRows.

* **MixColumns:** Линейное преобразование столбцов.

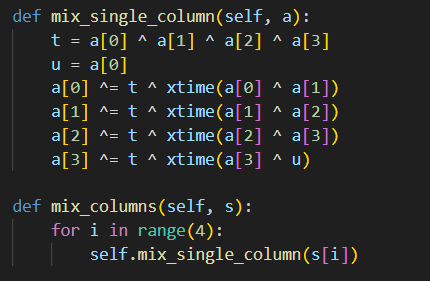


Рис. 12. Реализация функции MixColumns.

* **AddRoundKey:** Побитовое сложение (XOR) с ключом раунда.

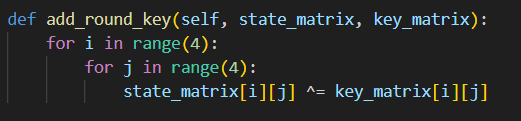


Рис. 13. Реализация функции AddRoundKey.

## Работа с файлами

Программа шифрует или расшифровывает содержимое файла и создаёт новый файл с добавлением суффиксов \_encrypted или \_decrypted.

**Шифрование:**

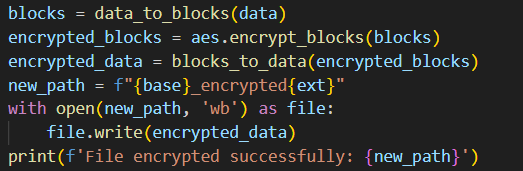


Рис. 14. Реализация функции шифрования файла.

**Расшифрование:**

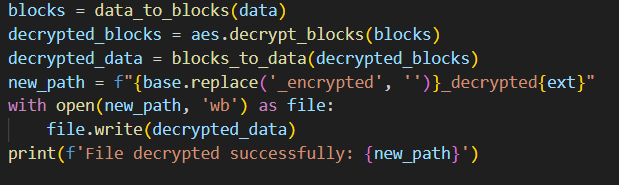
****

Рис. 15. Реализация функции расшифровки файла.

## 5.4 Работа с директорией

Программа рекурсивно пробегается по директории и шифрует файлы в новую папку с суффиксом \_processed.

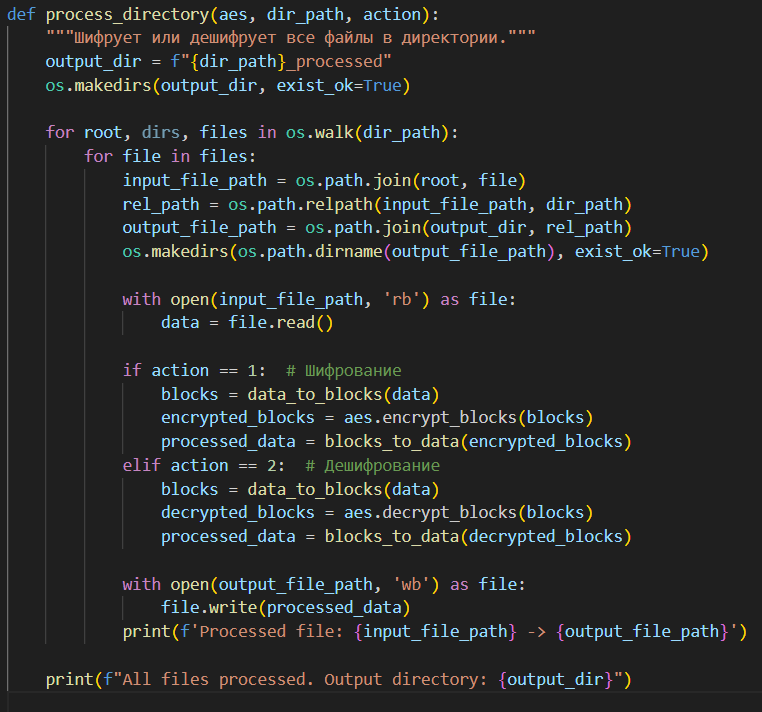


Рис. 16. Реализация функции шифрования и расшифровки директории.

# Заключение

Алгоритм AES (Advanced Encryption Standard) является одним из наиболее широко используемых криптографических методов для защиты данных в современных системах. Его устойчивость к криптоанализу, скорость работы и универсальность делают его стандартом для множества приложений, включая защиту файлов, шифрование сетевого трафика и обеспечение безопасности банковских транзакций.

В ходе выполнения курсовой работы были рассмотрены основные принципы работы AES, его структура и этапы шифрования. Основное внимание было уделено реализации алгоритма на языке программирования Python. В процессе работы была создана программа, которая позволяет шифровать и расшифровывать файлы, гарантируя их целостность и конфиденциальность.

Программная реализация включала в себя:

* Разбиение данных на блоки фиксированного размера и их подготовку к обработке.
* Использование операций AES, таких как SubBytes, ShiftRows, MixColumns и AddRoundKey, с учётом особенностей ключевого расписания.
* Обеспечение корректной работы с файлами, включая добавление и удаление выравнивания (padding) для обработки данных, размер которых не кратен блоку.
* Простоту использования программы, где достаточно задать ключ шифрования, выбрать файл и действие.
* В результате реализации программа продемонстрировала успешное шифрование и расшифрование файлов, сохраняя исходную структуру данных. Это свидетельствует о корректности выполнения алгоритма и его применимости в реальных задачах.

# Список литературы

1. <https://habr.com/ru/articles/212235/> [Электронный ресурс]
2. <https://habr.com/ru/articles/534620/> [Электронный ресурс]
3. Joan Daemen, Vincent Rijmen - The Design of Rijndael