Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по лабораторной работе № 3

по курсу «Защита информации»

на тему: «Программная реализация алгоритма AES с применением режима шифрования OFB»

Студент ИУ7-73Б _____ Марченко В.
(Группа) (Подпись, дата) (И. О. Фамилия)

Преподаватель _____ Чиж И. С.
(Подпись, дата) (И. О. Фамилия)

2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

AES (англ. Advanced Encryption Standard; также Rijndael) — симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительством США по результатам конкурса AES. Этот алгоритм хорошо проанализирован и сейчас широко используется, как это было с его предшественником DES. Национальный институт стандартов и технологий США опубликовал спецификацию AES 26 ноября 2001 года после пятилетнего периода, в ходе которого были созданы и оценены 15 кандидатур. 26 мая 2002 года AES был объявлен стандартом шифрования. По состоянию на 2009 год AES является одним из самых распространенных алгоритмов симметричного шифрования [1].

AES можно использовать в совокупности со следующими режимами шифрования:

1) ECB (англ. electronic code book) — режим «электронной кодовой книги»;
2) CBC (англ. cipher block chaining) — режим сцепления блоков;
3) PCBC (англ. propagating cipher block chaining) — режим распространяющегося сцепления блоков шифра [2];
4) CFB (англ. cipher feed back) — режим обратной связи по шифротексту;
5) OFB (англ. output feed back) — режим обратной связи по выходу.

Целью данной лабораторной работы является программная реализация алгоритма шифрования AES с применением режима шифрования OFB.

Задачи лабораторной работы:

1) изучить принцип работы алгоритма AES;
2) изучить принцип работы режима OFB;
3) разработать программное обеспечение для шифрования и расшифрования файлов с применением AES и OFB;
4) протестировать разработанное программное обеспечение.

1 Алгоритм шифрования AES

AES является стандартом, основанным на алгоритме Rijndael [1]. Он состоит из двух глобальных этапов — генерации ключей и непосредственно шифрования текста.

Для AES длина P (plaintext) (блока входных данных) и S (state) постоянна и равна 128 бит, а длина шифроключа K (key) составляет 128, 192 или 256 бит. При этом исходный алгоритм Rijndael допускает длину ключа и размер блока от 128 до 256 бит с шагом в 32 бита. Для обозначения выбранных длин P, S и K в 32-битных словах используется нотация $N_b = 4$ для P и S, $N_k = 4, 6, 8$ для K соответственно для разных длин ключей [1].

В начале шифрования P копируется в массив S (каждые последовательные 4 байта P составляют столбец в матрице-массиве S). После этого к S применяется процедура AddRoundKey(), и затем S проходит через процедуру трансформации (раунд) 10, 12 или 14 раз (в зависимости от длины ключа), при этом надо учесть, что последний раунд несколько отличается от предыдущих. В итоге, после завершения последнего раунда трансформации, S копируется в C (ciphertext) [1].

Отдельные трансформации SubBytes(), ShiftRows(), MixColumns() и AddRoundKey() — обрабатывают S [1].

Процедура SubBytes() обрабатывает каждый байт S, независимо производя нелинейную замену байтов, используя таблицу замен S-box. Такая операция обеспечивает нелинейность алгоритма шифрования [1]. При расшифровании C используется обратная таблица замен S-box-inv.

ShiftRows() работает со строками S. При этой трансформации строки состояния циклически сдвигаются на r байт по горизонтали в зависимости от номера строки. Для нулевой строки $r = 0$, для первой строки $r = 1$ и т. д. Таким образом, каждый столбец S после применения процедуры ShiftRows() состоит из байтов из каждой колонки начального состояния. Для алгоритма Rijndael паттерн смещения строк для 128- и 192-битных строк одинаков. Однако для блока размером 256 бит отличается от предыдущих тем, что 2-е, 3-и и 4-е строки смещаются на 1, 3 и 4 байта соответственно. Это замечание не относится к AES, так как он использует алгоритм Rijndael только с 128-битными блоками, независимо от размера ключа [1].

В процедуре MixColumns() четыре байта каждого столбца S смешивают-

ся, используя для этого обратимую линейную трансформацию. MixColumns() обрабатывает состояния по столбцам, трактуя каждый из них как полином третьей степени. Над этими полиномами производится умножение в поле Галуа. Вместе с ShiftRows() MixColumns() вносит диффузию в шифр [1].

В процедуре AddRoundKey() K каждого раунда объединяется с S. Для каждого раунда K_n получается из K_{n-1} с помощью процедуры KeyExpansion(); каждый K такого же размера, что и S. Процедура производит побитовый XOR каждого байта S с каждым байтом K [1].

На рисунке 1.1 показана схема работы алгоритма AES.

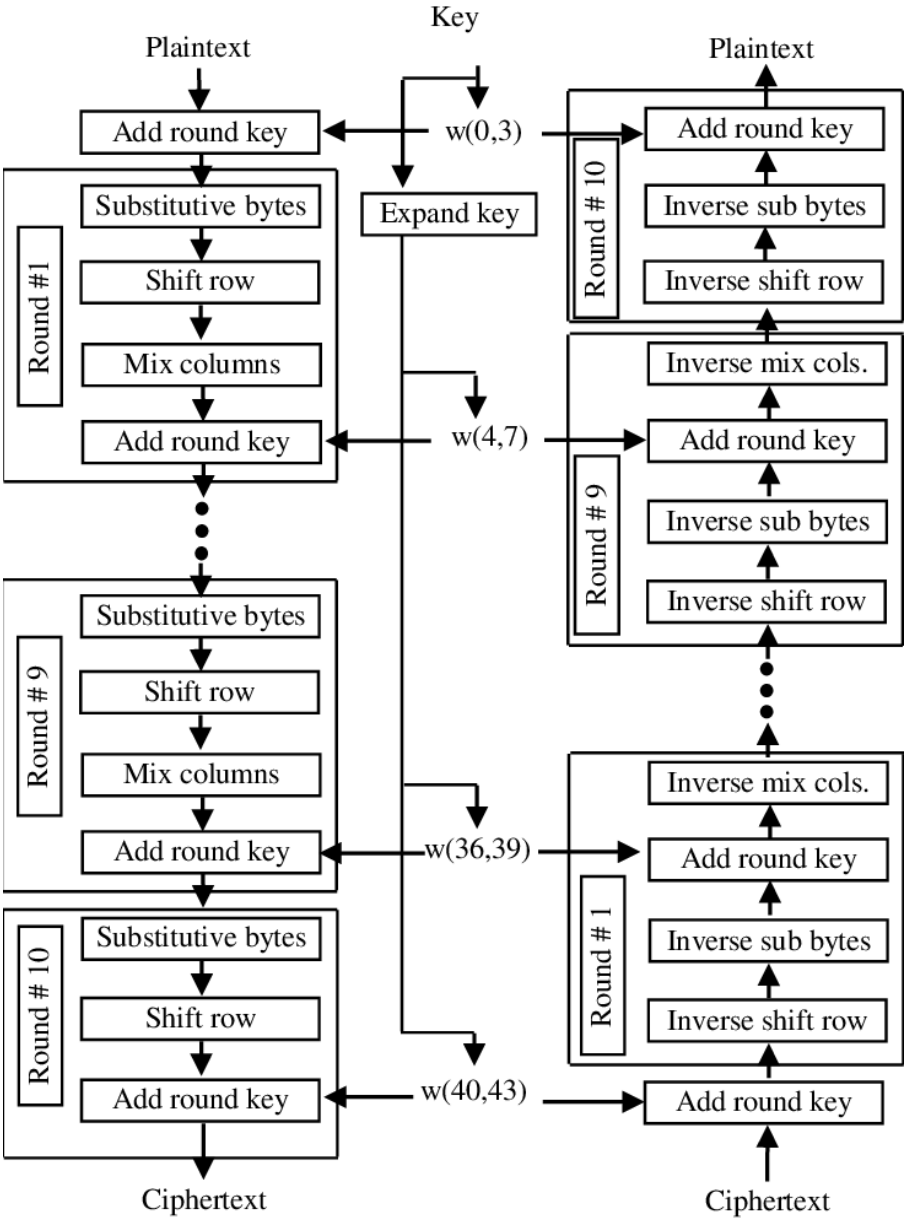

Рисунок 1.1 – Схема работы алгоритма AES [3]

На рисунке 1.2 показана схема генерации ключей для раундов шифрования.

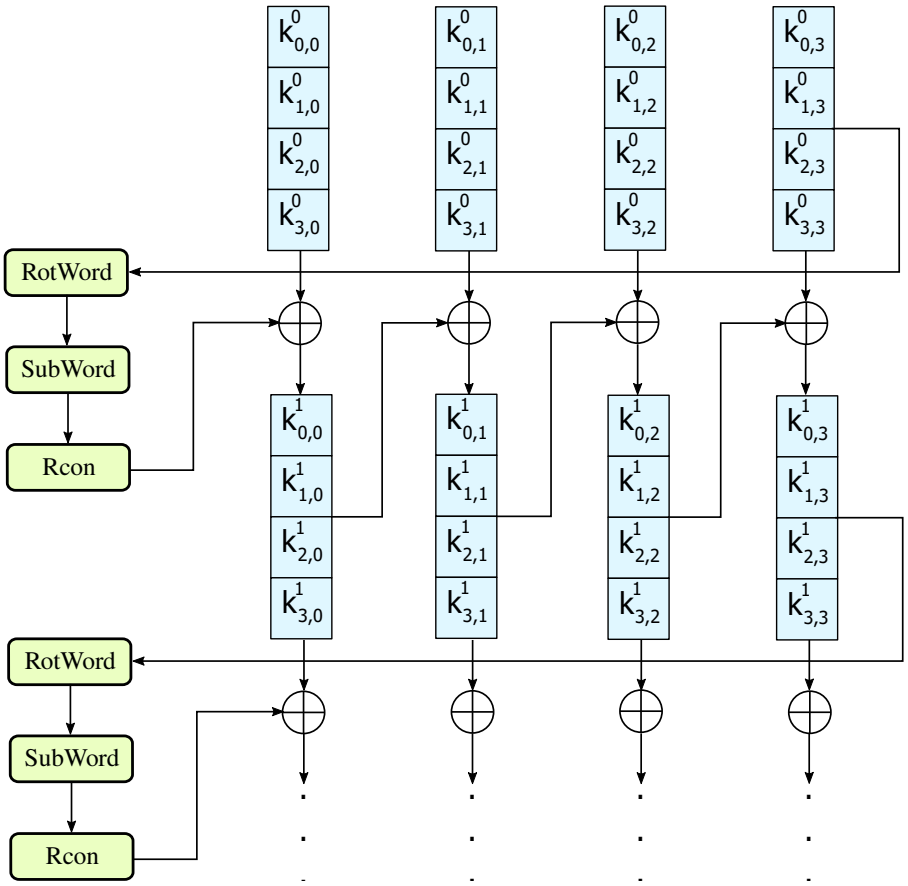

Рисунок 1.2 – Схема генерации ключей [4]

2 Алгоритм режима шифрования OFB

Режим (OFB) обратной связи вывода превращает блочный шифр в синхронный шифр потока: он генерирует ключевые блоки, которые являются результатом сложения с блоками открытого текста, чтобы получить зашифрованный текст. Так же, как с другими шифрами потока, зеркальное отражение в зашифрованном тексте производит зеркально отраженный бит в открытом тексте в том же самом местоположении. Это свойство позволяет многим кодам с исправлением ошибок функционировать как обычно, даже когда исправление ошибок применено перед кодированием [2].

Каждая операция блочного шифра обратной связи вывода зависит от всех предыдущих и поэтому не может быть выполнена параллельно. Однако, из-за того, что открытый или зашифрованный текст используются только для конечного сложения, операции блочного шифра могут быть выполнены заранее, позволяя выполнить заключительное шифрование параллельно с открытым текстом. Обратная связь по выходу на k разрядов не рекомендуется из соображений криптостойкости [2].

Режим OFB имеет следующее преимущество по сравнению с режимом CFB: ошибки, возникающие в результате передачи по каналу с шумом, при дешифровании не «размазываются» по всему шифротексту, а локализуются в пределах одного блока. Однако открытый текст может быть изменен путем определенных манипуляций с блоками шифротекста. Несмотря на то, что OFB-шифрование не поддается распараллеливанию, эффективность процедуры может быть повышена за счет предварительной генерации независимой последовательности блоков [2].

Алгоритм расшифрования в режиме OFB полностью совпадает с алгоритмом шифрования. Функция расшифрования блочного алгоритма не используется в данном режиме, т. к. ключевой поток генерируется только функцией шифрования блока [2].

На рисунке 2.1 показана схема шифрования текста в режиме OFB.

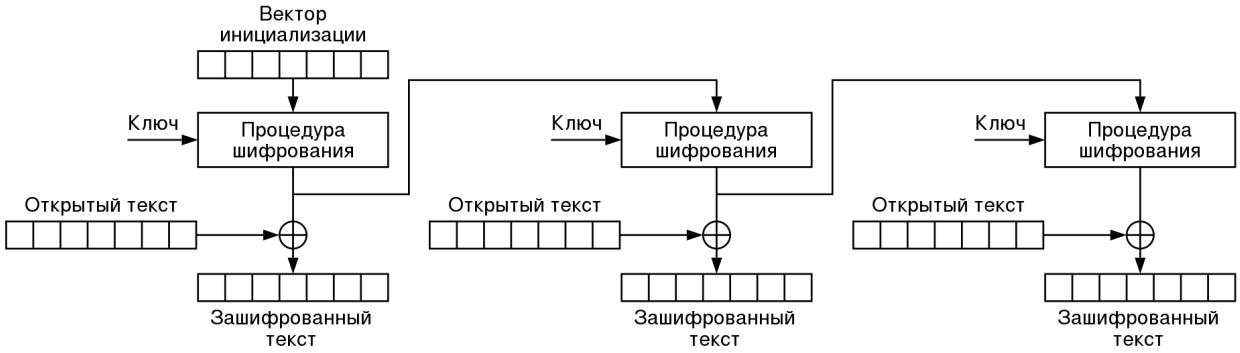

Рисунок 2.1 – Схема шифрования в режиме OFB [2]

Вектор инициализации — 128-битная случайно сгенерированная последовательность.

На рисунке 2.2 показана схема расшифрования текста в режиме OFB.

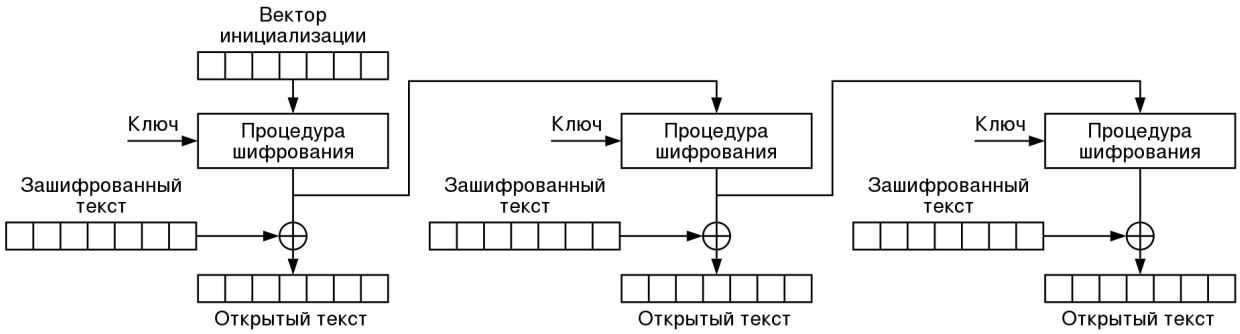

Рисунок 2.2 – Схема расшифрования в режиме OFB [2]

3 Требования к входным данным

Программа принимает два аргумента командной строки.

1. Первый аргумент — путь к файлу, который содержит исходный текст (открытый/зашифрованный).

2. Второй аргумент — путь к файлу, в который будет записан зашифрованный/расшифрованный текст.

При наличии ошибок в аргументах командной строки или при передаче на вход программе пустого файла программа выдаст сообщение об ошибке и завершится.

В каталоге cfg находятся текстовые файлы с конфигурациями таблиц и значениями ключа и вектора инициализации.

Программное обеспечение для шифрования и расшифрования файлов с помощью алгоритма AES и режима шифрования OFB было написано на языке программирования C.

Программа может зашифровать/расшифровать файлы любых типов.

4 Тестирование программного обеспечения

В таблице 4.1 приведены тесты для проверки корректности работы реализованного программного обеспечения.

Таблица 4.1 – Тесты

Описание	**Открытый текст**	**Результат шифрования**
Пустой входной файл		Error: input file is empty.
Кол-во аргументов командной строки не равно двум		Error: program requires 2 parameters.
В файле записан один байт	1	4787 6d68 99b4 a7ff 9b1d 627f 5a65 f913 (в hex виде)
В файле записан ровно один блок (16 байт)	1234567890123456	7587 595b af81 9fc8 ab24 504e 6e56 cf26 (в hex виде)
В файле записано более 16 байт	12345678901234567890	7587 595b af81 9fc8 ab24 504e 6e56 cf26 9558 b32a a4b0 f1af f170 4bc9 f774 6511 (в hex виде)

Помимо приведенных выше тестов были зашифрованы и расшифрованы архивы с текстовыми файлами, фотографиями и видео. Все тесты пройдены успешно.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной лабораторной работы был реализован алгоритм шифрования AES с применением режима OFB.

Были выполнены следующие задачи:

1) изучен принцип работы алгоритма AES;

2) изучен принцип работы режима OFB;

2) разработано программное обеспечение для шифрования и расшифровки файлов с применением AES и OFB;

4) протестировано разработанное программное обеспечение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Википедия*. AES. — 2023. — (Дата обращения: 18.10.2023). https://en.wikipedia.org/wiki/Advanced_Encryption_Standard.
2. *Википедия*. Block cipher mode of operation. — 2023. — (Дата обращения: 18.10.2023). https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation.
3. *Wadday A.*, *Mohammed H.*, *Abdullah A.* Study of WiMAX Based Communication Channel Effects on the Ciphered Image Using MAES Algorithm // International Journal of Applied Engineering Research. — 2018. — Апр. — Т. 13.
4. *Википедия*. AES key schedule. — 2023. — (Дата обращения: 18.10.2023). https://en.m.wikipedia.org/wiki/AES_key_schedule.