МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Белорусский государственный технологический университет»

**«Исследование блочных шифров»**

Студент: Высоцкий Я.А.

ФИТ 3 курс 5 группа

Вариант 4

Преподаватель: Савельева М. Г.

Минск 2023

**1)Разработка приложения**

Перед разработкой приложения нужно выделить основные части, которые требуется реализовать, для работы приложения, выполняющее шифрование алгоритмом DES-EEE2. В ходе проектирования были выделены 3 основные объекта, требуемые программной реализации.

1. Класс, отвечающий за работу с ключами;
2. Класс, выполняющий основные операции алгоритма;
3. Класс управляющий двумя другими классами.

Класс KeyManager выполняет все действия с ключами в алгоритме. В нем определен основной метод GenerateBitArrayKeys, который отвечает за корректную смену ключей, с учетом сдвига.

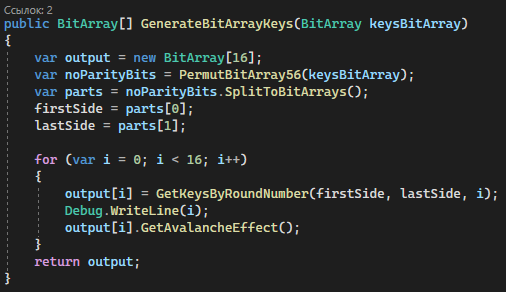


Рисунок 1.1 – Метод GenerateBitArrayKeys

Так же тут определен вспомогательный метод, сдвигающий ключ на определённое количество бит влево.

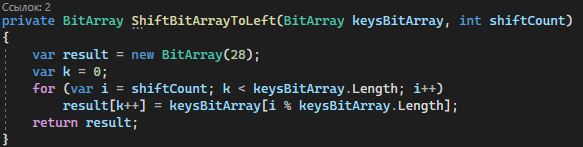


Рисунок 1.2 – Функции сдвига бит в ключе

Класс RoundManagerвыполняет основные методы алгоритма. Здесь происходят такие операции, как расширение, операция XOR, подстановка и перестановка. Так же в нем определенные специальные матрицы перестановки. В нем определен основной метод DoRound(), в котором выполняется все шаги алгоритма.

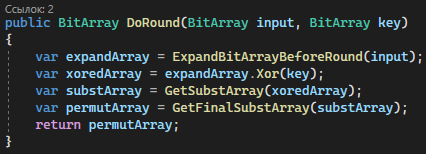


Рисунок 1.3 – Метод DoRound

Класс DesEnctyptManager управляет вышеупомянутыми классами для выполнения шифрования и расшифрования. Здесь определенно 2 основных метода: GetDecryptTextWithKey, GetEncryptTextWithKey. Оба метода принимают в качестве параметров: ключ и исходный текст. Сначала они переводятся в двоичный формат с учетом сдвига, далее исходный текст разбивается на блоки длинной 64 бита. После этого происходит начальная перестановка и выполнения 16 итераций, в качестве результат мы получаем шифротекст или расшифрованный текст.

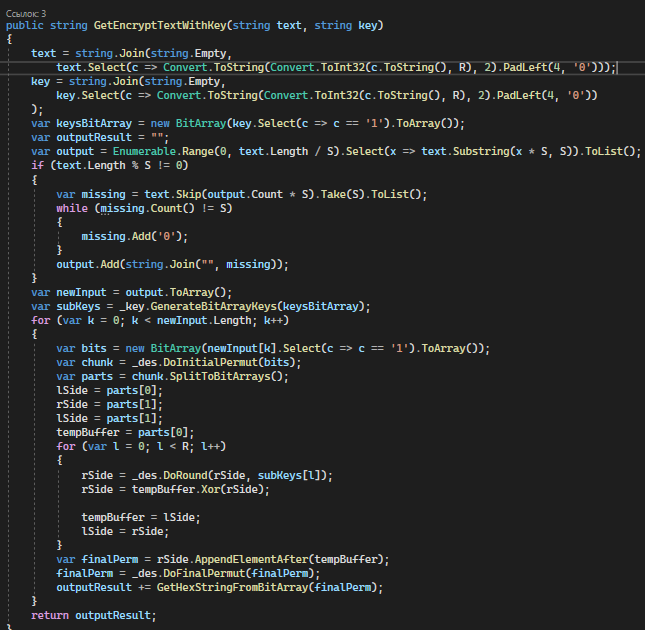


Рисунок 1.4 – Функция шифрования входного текста

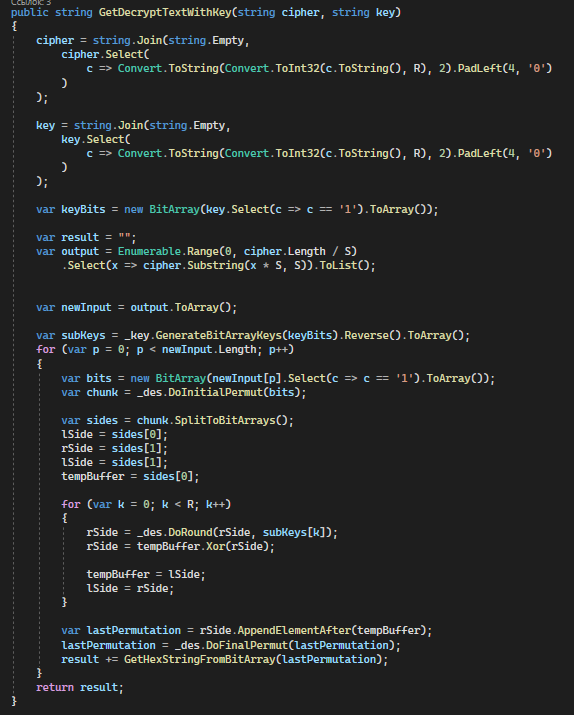


Рисунок 1.5 – Функция расшифрования

При выполнении алгоритма, на каждом шаге изменяется исходное сообщение. На каждом шаге изменяется количество и индексы изменяемых битов. Лавинный эффект выглядит следующим образом:

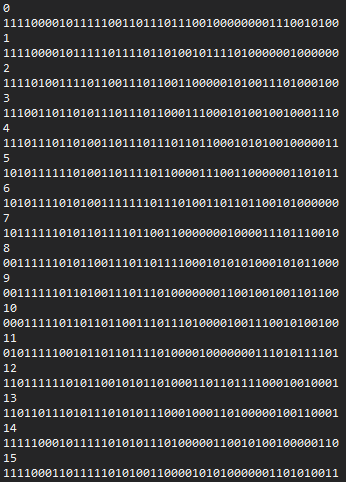


Рисунок 1.6 – Лавинный эффект для слова Hello World

Время, затраченное на шифрование и расшифрование алгоритмом   
DES - EEE2 исходного текста длиной 10,100 и 1000 символов.

Рисунок 1.7 – График времени выполнение приложения

**2) Анализ влияния слабых и полуслабых ключей**

Слабые и полуслабые ключи значительно ослабляют криптостойкость шифрования. К слабым ключам относятся ключи, состоящие из только нулей или единиц, или те, в которых половина содержит только единицы, а другая только нули, или наоборот. К полуслабым ключам относятся ключи, в которых вторая половина, является зеркальной первой.

Так как мы используем реализацию алгоритма DES-EEE2, то влияние слабых ключей ослабляется. Это связанно с использование 3 ключей для шифрования текста. Криптостойкость алгоритма упадет сильно, только при условии, что все ключи будут является слабыми.

Также это отражается и на лавинном эффекте. При использовании слабых ключей, лавинный эффект является минимальным. Но при наличии трёх разных ключей, уменьшений этого эффекта является не таким значительным, как при использовании единого ключа.

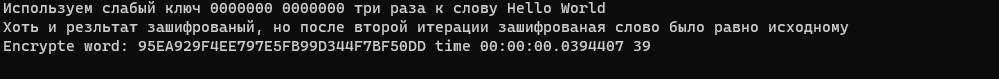


Рисунок 1.8 – Результат использования слабых ключей

**3) Оценка степени сжатия открытого текста и зашифрованного**

При использовании метода сжатия LZMA2 исходный текст имеет степень сжатия 0%. Размер исходного текста не изменился. А размер шифротекста увеличился с 682 до 986 байт, 144% от исходного размера зашифрованного сообщения. Оценка производилась для текста малой длины, что и повлияло на результат.

**Вывод**

При выполнении лабораторной работы были приобретены практические навыки разработки приложения для реализации блочных шифров. Так же были получены знания по сети Фейстеля, алгоритмам DES, AES и стандарт ГОСТ 28417-89.