Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Блочные шифры

Студент: Вайсера Р.Л.

ФИТ 3 курс 4 группа

Преподаватель: Сазонова

Минск 2023

# Алгоритм DES

Алгоритм строится на основе сети Фейстеля. Входной блок данных, состоящий из 64 битов, преобразуется в выходной блок идентичной длины. В алгоритме широко используются рассеивания (подстановки) и перестановки битов текста, о которых мы упоминали выше. Комбинация двух указанных методов преобразования образует фундаментальный строительный блок DES, называемый раундом или циклом. Один блок данных подвергается преобразованию (и при зашифровании, и при расшифровании) в течение 16 раундов. После первоначальной перестановки и разделения 64-битного блока данных на правую и левую половины длиной по 32 бита выполняются 16 раундов одинаковых действий.

Вначале правая часть блока *Ri* расширяется до 48 битов с использованием таблицы, которая определяет перестановку плюс расширение на 16 битов. Эта операция приводит размер правой половины в соответствие с размером ключа для выполнения операции XOR.

После выполнения перестановки с расширением для полученного 48-битного значения выполняется операция XOR с 48-битным подключом *Ki*. Затем полученное 48-битное значение подается на вход блока подстановки *S*, результатом которой является 32-битное значение. Подстановка выполняется в восьми блоках подстановки или восьми *S*-блоках.

При выполнении этой операции 48 битов данных делятся на восемь 6-битных подблоков, каждый из которых по соответствующей таблице замен замещается четырьмя битами. Подстановка с помощью *S*-блоков является одним из важнейших этапов DES. Таблицы замен для этой операции специально спроектированы так, чтобы обеспечивать максимальную криптостойкость. В результате выполнения этого этапа получаются восемь 4-битных блоков, которые вновь объединяются в единое 32-битное значение.

Далее полученное 32-битное значение обрабатывается с помощью перестановки *Р*, которая не зависит от используемого ключа. Целью перестановки является такое максимальное переупорядочивание битов, чтобы в следующем раунде шифрования каждый бит с большой вероятностью обрабатывался другим *S*-блоком.

И наконец, результат перестановки объединяется с помощью операции XOR с левой половиной первоначального 64-битного блока данных. Затем левая и правая половины меняются местами, и начинается следующий раунд.

# Зашифрование

Для начала, нам необходимо убедиться, что ключ, подаваемый на вход функции зашифрования, будет иметь размерность 64 бита. Для этого реализована функция, представленная на рисунке 1.1.



Рисунок 1.1 – Функция для создания 64-битного ключа

Для зашифрования необходимо убедиться, что блоки исходного текста будут иметь размерность 64 бита и при необходимости дополняться до данного размера. Например, если размер блока 8 байт, то для дополнения блока длиной 6 байт будет добавлено 2 байта со значением 0x02. Функция зашифрования представлена на рисунке 1.2.

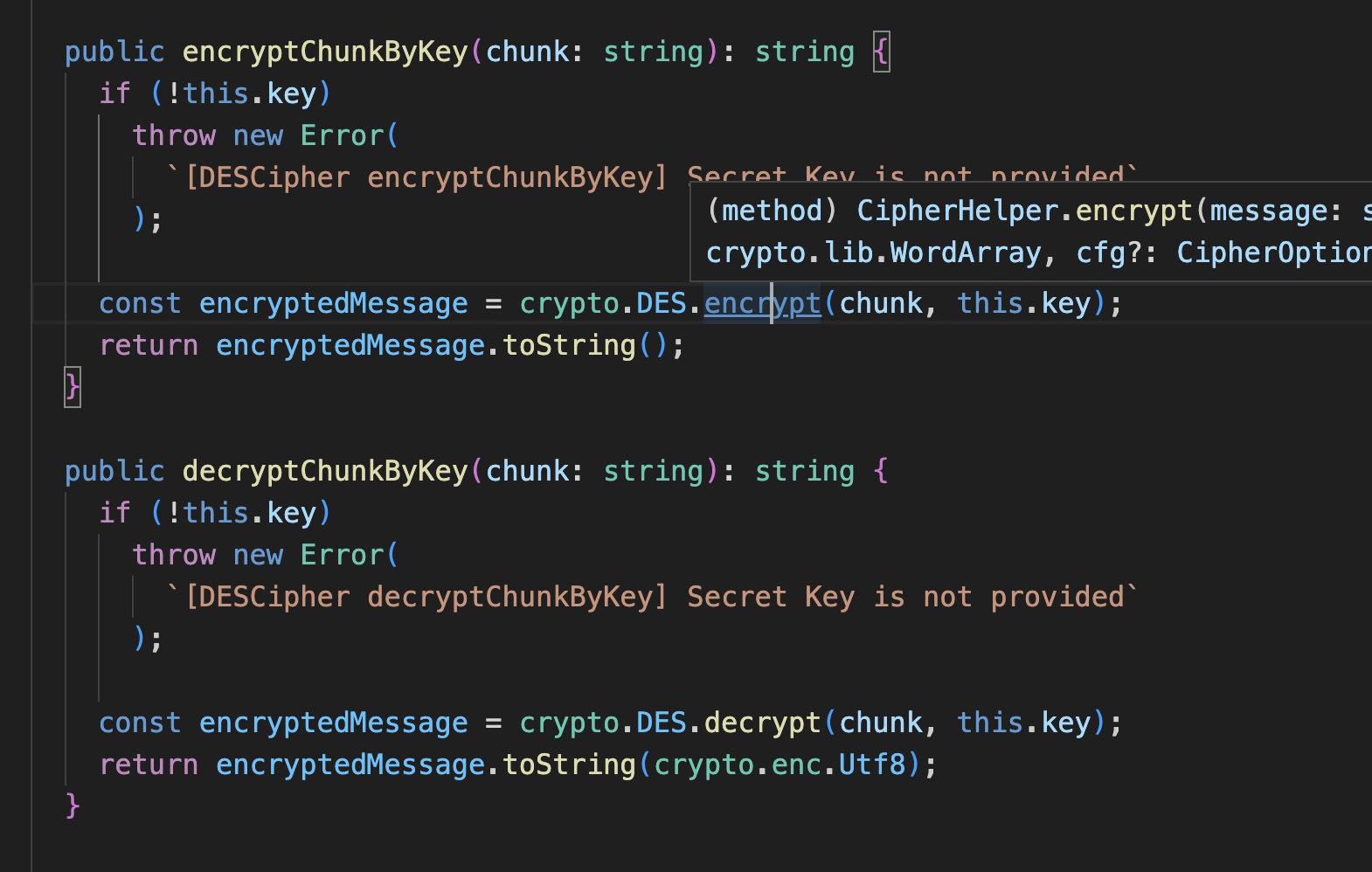


Рисунок 1.2 – Функция зашифрования алгоритмом DES

# Расшифрование

Для расшифрования необходимо повторить тот же самый алгоритм с обратным порядком ключей. Если использовались раундовые ключи с 1-го по 16-ый, то при расшифровании они будут применять в обратном порядке – с 16-го по 1-ый. Функция расшифрования представлена на рисунке 1.3.

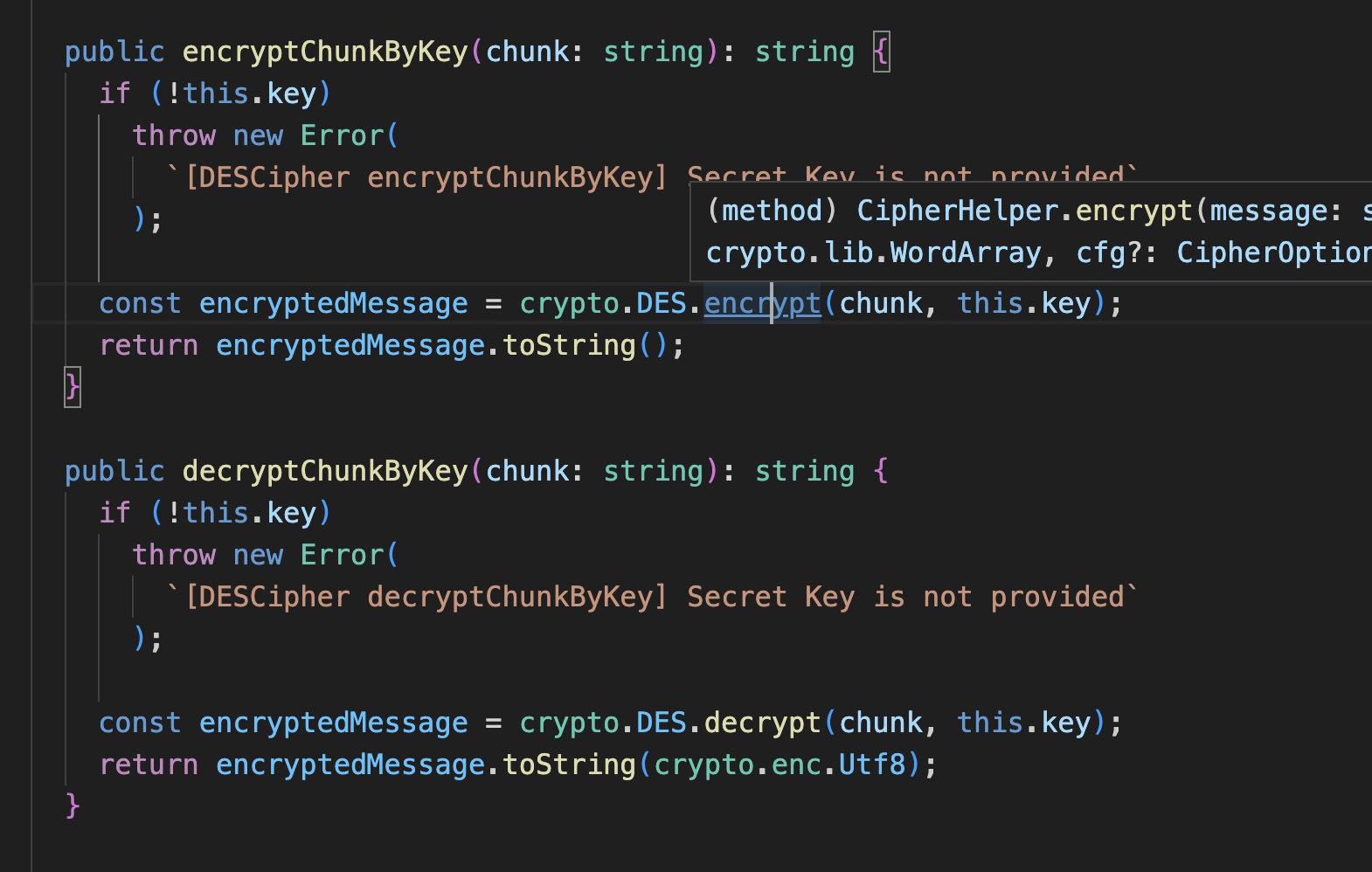


Рисунок 1.3 – Функция расшифрования алгоритмом DES

# Лавинный эффект

Алгоритму DES присуще такое свойство, как «лавинный» эффект. Оно заключается в том, что изменение значения малого количества битов во входном тексте или в ключе ведет к существенному изменению значений выходных битов шифр текста, по аналогии с алгоритмами хеширования.

Для оценки «лавинного» эффекта можно побитово сравнить исходный текст и шифротекст и посчитать количество изменённых битов. Функция, высчитывающая «лавинный» эффект, представлена на рисунке 1.4.

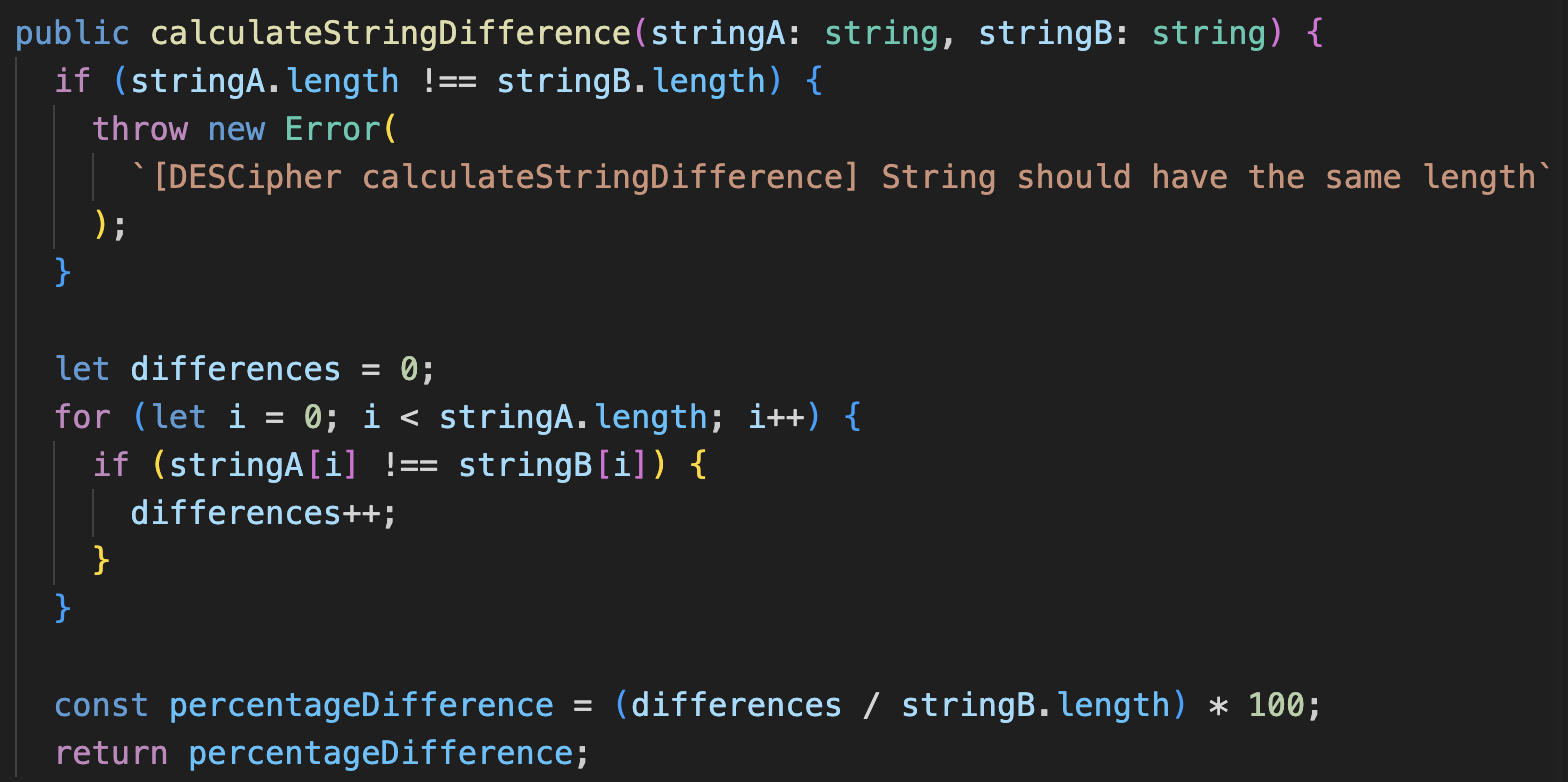


Рисунок 1.4 – Функция оценки «лавинного» эффекта

# Оценка лавинного эффекта

Для оценки лавинного эффекта зашифруем алгоритмом DES с использованием ключа №1, изменим один символ в исходном тексте, и сравним количество изменённых символов.

Зашифрование изначального текста с изначальным ключом представлено на рисунке 2.1.

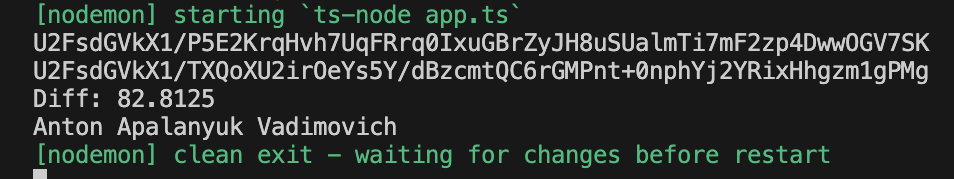


Рисунок 2.1 – Лавинный эффект, неизменённый открытый текст

# Слабые и полуслабые ключи

# Слабые ключи

Из-за того, что первоначальный ключ изменяется при получении подключа для каждого раунда алгоритма, определенные первоначальные ключи являются слабыми. Если все биты каждой половины равны 0 или 1, то для всех раундов алгоритма используется один и тот же ключ. Это может произойти, если ключ состоит из одних 1, из одних 0, или если одна половина ключа состоит из одних 1, а другая – из одних 0.

Оценим влияние слабого ключа на лавинный эффект. Вывод функции подсчёта измененных битов представлен на рисунке 3.1.

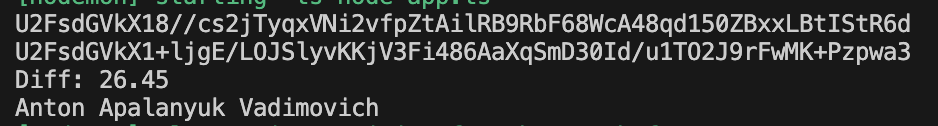


Рисунок 3.1 – Лавинный эффект со слабым ключом

При использовании слабого ключа лавинный эффект составляет всего 26% (в сравнении с 82% для обычного ключа). Это происходит за счёт того, что при использовании слабого ключа раундовые ключи состоят только из единиц или нулей, что при использовании операции XOR с шифротекстом не будут изменять данный шифротекст. Поэтому изменяется меньшее количество битов, и шифрование получается более слабым.

# Полуслабые ключи

Также, некоторые пары ключей при зашифровании переводят открытый текст в идентичный шифртекст. Иными словами, один из ключей пары может расшифровать сообщения, зашифрованные другим ключом пары. Это происходит из-за метода, используемого DES для генерации подключей: вместо 16 различных подключей эти ключи генерируют только два различных подключа. В алгоритме каждый из этих подключей используется восемь раз. Такие ключи называются полуслабыми.

Оценим влияние полуслабого ключа на лавинный эффект. Данная оценка представлена на рисунке 3.2.

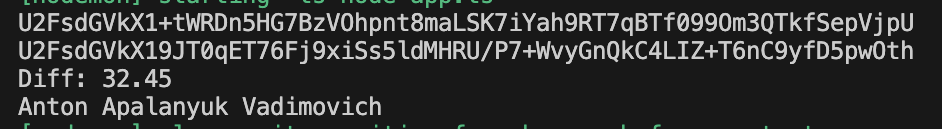


Рисунок 3.2 – Лавинный эффект с полуслабым ключом

При использовании полуслабого ключа, лавинный эффект примерно равен 32%, что немного выше, чем при слабом ключе, но также ниже, чем с обычным ключом. Это происходит из-за того, что на некоторых раундах раундовые ключи переводят открытый текст в шифротекст, который идентичен открытому тексту. За счёт большого количества раундов, на выходе алгоритма мы не получим исходный открытый текст, однако полуслабые ключи, по сути, уменьшают количество раундов зашифрования, что негативно сказывается на криптостойкости алгоритма.

# Время выполнения зашифрования и расшифрования

Для оценки времени выполнения операций зашифрования и расшифрования построим графики при шифровании алгоритмом DES с переменной длиной входного текста, можно воспользоваться графиком на рисунке 4.1.

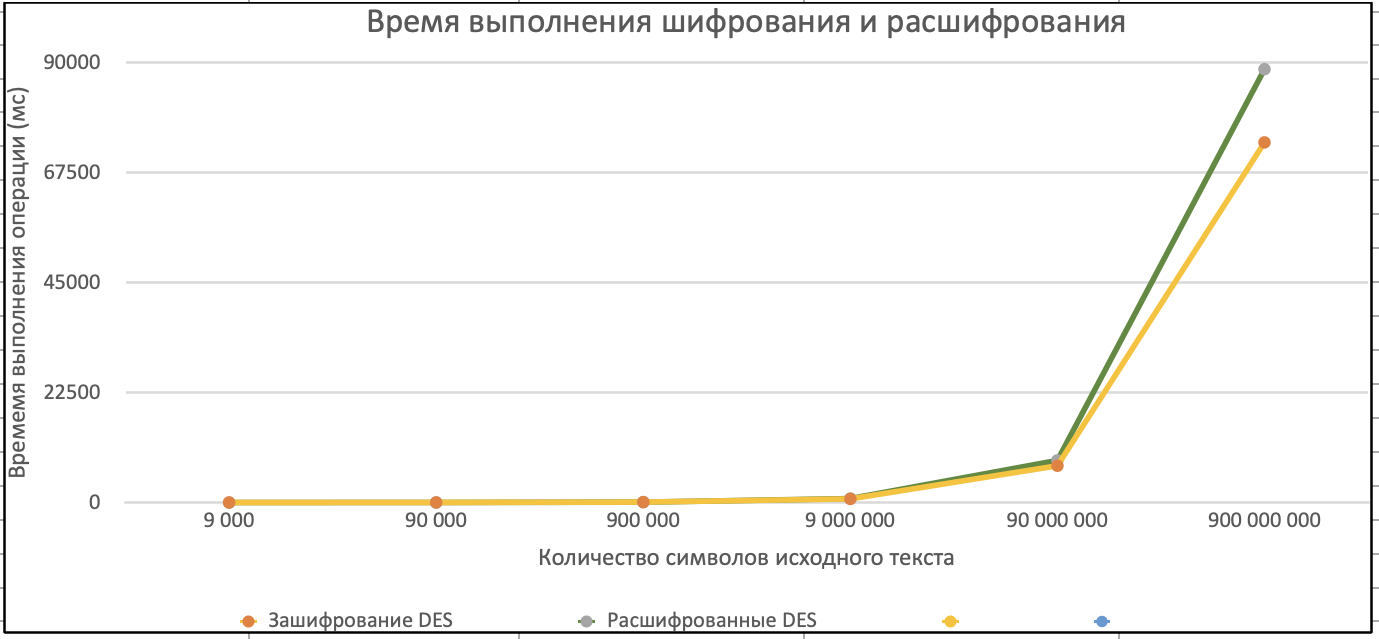


Рисунок 4.1 – Время выполнения расшифрования и зашифрования

При сравнении данного графика с временем выполнения рассмотренных ранее шифров можно заметить, что алгоритм DES выполняет операции зашифрования и расшифрования примерно в 3 раз медленнее, чем рассмотренные ранее алгоритмы шифрования.

# Оценка степени сжатия шифротекста

Одним из свойств алгоритма DES является то, что шифротекст практически нельзя сжать. Это происходит потому, что шифрование преобразует информацию таким образом, что в итоге создается псевдослучайная последовательность битов. Эта последовательность битов не имеет никакой структуры, что делает ее несжимаемой.

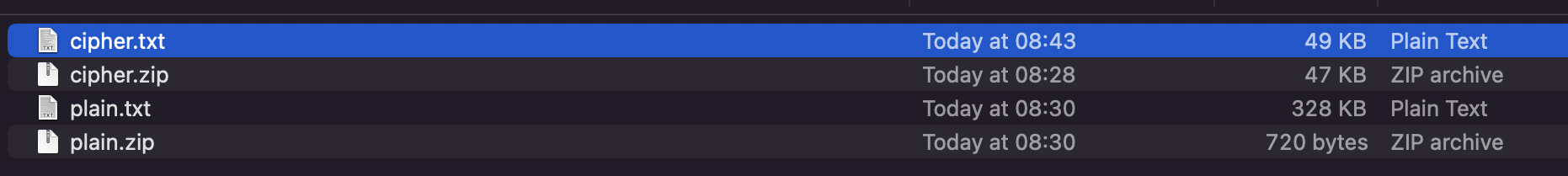


Рисунок 5.1 – Сжатие открытого текста и шифротексте

# Вывод

В данный лабораторной работе были изучены и приобретены практические навыки разработки и использования приложений для реализации блочного шифра DES, оценивание производительности алгоритма шифрование и построение графика зависимости времени зашифрования от количества символов исходного текста