Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

**Исследование методов текстовой стеганографии**

Выполнил:

студент 3 курса 5 группы

специальности ПОИТ

Мингазов А. И.

Преподаватель:

Берников Владислав Олегович

Минск 2021

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет зашифровать и расшифровать стеганографический текст.

Также приложение позволяет выполнить операций за(рас)шифрования и вывести исходной текст и соответствующий ему зашифрованный текст в файлы, чтобы в последующем оценить стеганографическую криптографию по тексту

1. **Методика выполнения поставленных задач**

На рисунке 2.1 показана реализация алгоритма сохранения информации методом модификации апроша.

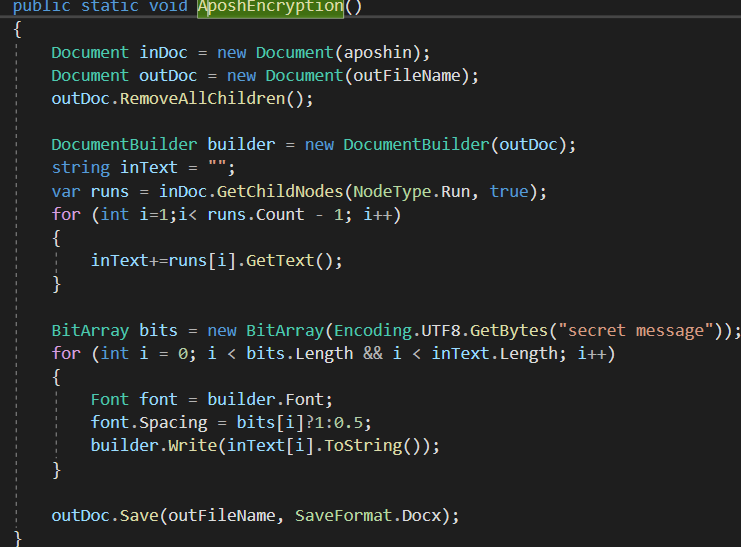


Рисунок 2.1 Модификация апроша

Сперва получается текст-контейнер, в данном случае читается данный текстовый файл. Сохраняемое сообщение преобразуется в набор битов. При переборе всех символов контейнера каждому символу задается значение апроша в зависимости от значения соответствующего бита при этом символ с заданным апрошем сохраняется в выходной файл. Обход происходит пока не закончится текст контейнера либо не закончится секретное сообщение. Таким образом на каждый бит секретного сообщения приходится один символ контейнера, однако эту отношение можно увеличить ставя не 2 варианта размера апроша, а больше.

На рисунке 2.2 показывается реализация получения секретного сообщения из контейнера.

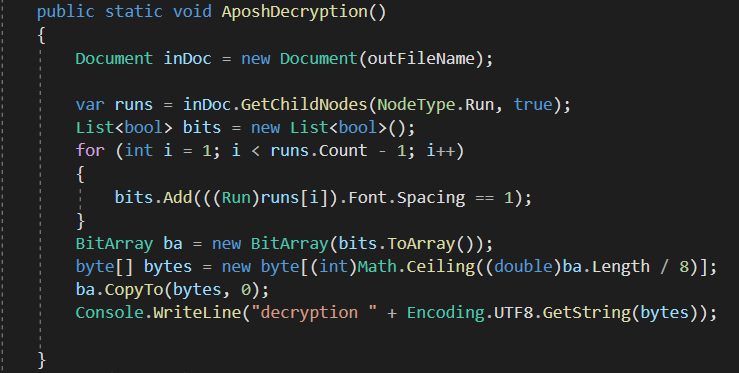


Рисунок 2.2 – Получения секретного сообщения

Для расшифрования обходится каждый символ исходного текста и берется значение его апроша. в данном случае максимальное значение апроша соответствует значению бита равным единице, при минимальном апроши бит равняется нулю. Таким образом из каждого символа получается бит сообщения, после чего биты состоят и из них получают текст сообщения

Для помещения сообщения используется подобный алгоритм как и с апрошем символов, однако в этом случае изменяется только длина строки путем добавления пробела в конец. На рисунке 2.3 показывается реализация модификации длины строки.

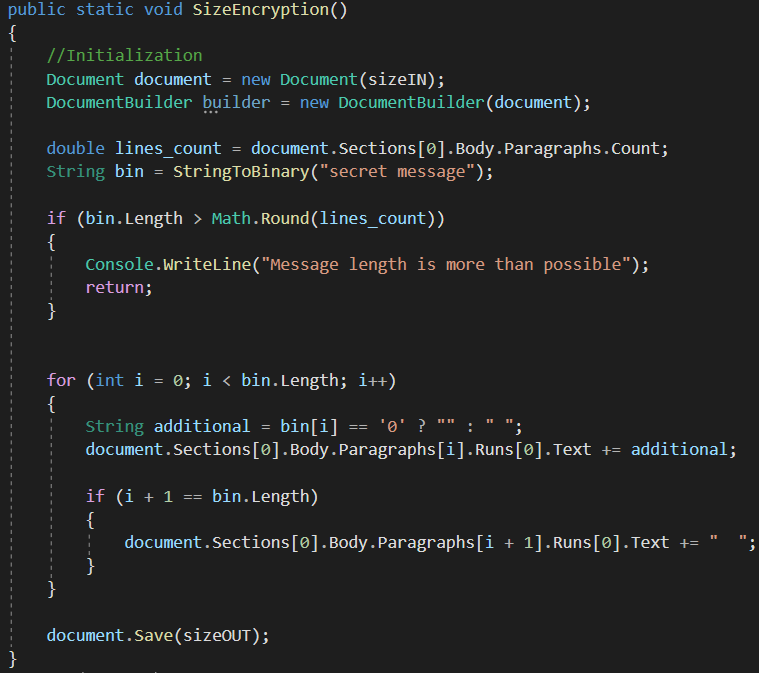


Рисунок 2.3 –реализация модификации расстояния между строками

### Символы контейнера перебираются, после чего, при нахождении символа пробела в конце берет его согласно текучему шифруемому биту сообщения. Таким образом контейнер должен быть в разы больше чем при модификации без добавлении пробела.

Алгоритм снова повторяет соответствующий алгоритм на основе модификаций апроша, однако в данном случае учитываются только пробелы, где снова минимальный апрош означает нулевой бит а максимальный – единичный.

На рисунке 2.4 показывается исходный контйнер.

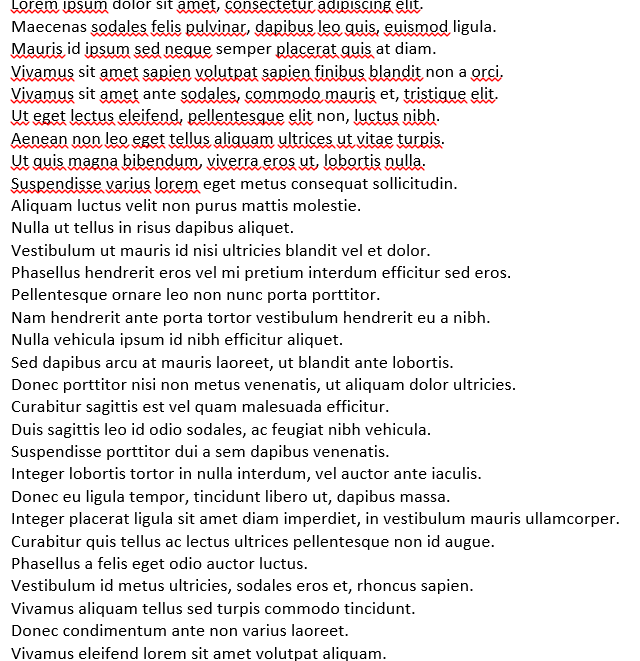
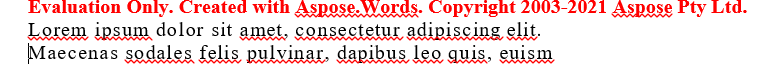


Рисунок 2.4 исходный контейнер

Контейнер должен быть большим чтобы вместить все символы.

На рисунке 2.5 показывается заполненный контейнер.



Рисуно2.5 заполненный контейнер апроша

Можно заметить что одинаковые слова на разных строках занимают разную длину.

На рисунке 2.6 показывается заполненный контейнер с разными длинами строк.

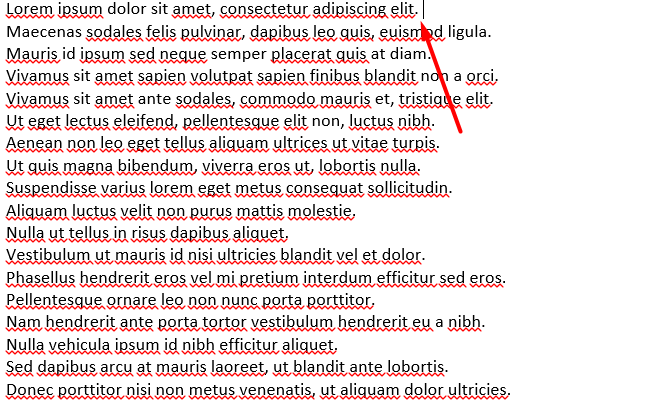


Рисунок 2.6-заполненный контейнер с различными длинами строк

В этом случае длины слов не различаются, однако длины строк различны из-за пробела в конце строки.

**Вывод**

В ходе лабораторной работы были приобретены практические навыки разработки и использования приложений для реализации стеганографических метод шифрации\дешифрации текста.