МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №14 НА ТЕМУ:**

**ИССЛЕДОВАНИЕ МЕТОДОВ ТЕКСТОВОЙ СТЕГАНОГРАФИИ**

                                                                 Выполнил студент 3 курса 5 группы

Дмитрук Илья Игоревич

Минск 2024

# Задание 1

Интерфейс программы имеет несколько блоков установки. В блоке «Текст для скрытия» пользователь может ввести вручную с клавиатуры или вставить из буфера обмена секретное сообщение, которое необходимо скрыть в документе-контейнере. Имеется возможность использовать в качестве сообщения уже существующий электронный текстовый документ (кнопка «Взять из файла»). При выборе необходимого документа предоставляются три фильтра, а также возможность выбора любого файла. В качестве текста, сокрытого в документе будет «привет!».

Результат работы программы предоставлен на рисунке 1.1.

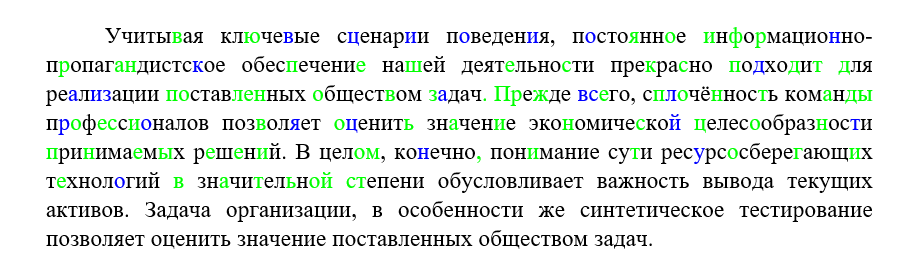


Рисунок 1.1 – результат работы приложения Sword

На рисунке 1.2 предоставлены настройки приложения Sword.

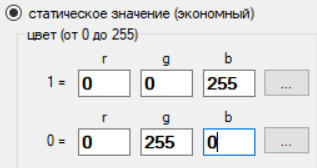


Рисунок 1.2 – параметры приложения Sword

Таким образом было произведено ознакомление с приложение Sword.

# Задание 2

Реализация метода на основе модификации цвета символов текста-контейнера по сути своей он схож с классическим методом наименее значащих битов и опирается на использовании цветовой модели RGB. В нашем случаем определённому цвету назначено значение бита

Программная реализация внедрение текста предоставлена в листинге 2.1.

static void EncryptMessage(string message, IEnumerable<Paragraph> paragraphs)

{

int bitIndex = 0;

List<char> bits = new List<char>();

foreach (var item in message)

{

string rawBits = Convert.ToString(item, 2);

if (rawBits.Length % 8 != 0)

for (int i = 0; i < rawBits.Length % 8; i++)

rawBits = rawBits.Insert(0, "0");

bits.AddRange(rawBits);

}

foreach (Paragraph paragraph in paragraphs)

{

foreach (Run run in paragraph.Descendants<Run>())

{

foreach (Text txt in run.Elements<Text>())

{

foreach (char letter in txt.Text)

{

if (bitIndex < bits.Count)

{

RunProperties runProperties = new RunProperties();

Color colorZero = new Color() { Val = "010000" };

Color colorOne = new Color() { Val = "000100" };

if (bits[bitIndex] == '0')

runProperties.Append(colorZero);

else

runProperties.Append(colorOne);

runProperties.Append(new FontSize() { Val = run.RunProperties.FontSize.Val });

Run newRun = new Run(runProperties, new Text(letter.ToString()));

run.InsertBeforeSelf(newRun);

bitIndex++;

}

}

}

if (bitIndex < bits.Count) run.Remove();

}

}

}

}

Листинг 1.1 – Функция внедрения текста

Далее нужно было реализовать извлечения текста. Код извлечения текста предоставлен в листинге 2.1.

static string DecryptMessage(IEnumerable<Paragraph> paragraphs)

{

string bits = string.Empty;

string text = string.Empty;

foreach (Paragraph paragraph in paragraphs)

{

foreach (Run run in paragraph.Descendants<Run>())

{

if (run.RunProperties.Color == null)

continue;

if (run.RunProperties.Color.Val == "010000")

bits += "0";

else if (run.RunProperties.Color.Val == "000100")

bits += "1";

if (bits.Length % 8 == 0)

{

text += (char)Convert.ToInt32(bits, 2);

bits = string.Empty;

}

}

}

return text;

}

Листинг 2.2 – Реализация извлечения текста

Результат работы алгоритма предоставлен на рисунке 2.1.



Рисунок 2.1 – Результат работы алгоритма

Данный алгоритм меняет цвет букв на достаточно малое значение что бы было сложно различить это визуально. Сравнение текста приведено на рисунке 2.2.

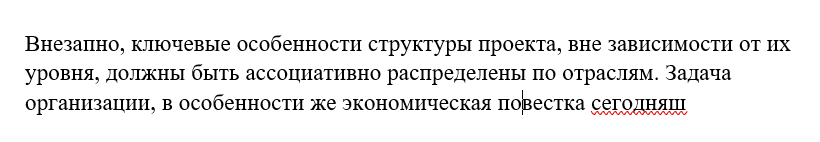
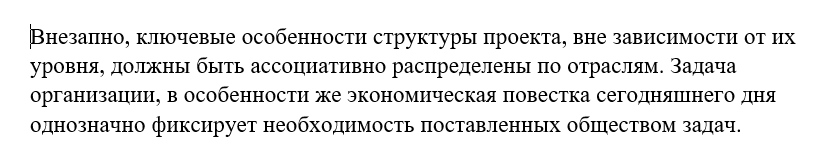


Рисунок 2.2 – Визуальное сравнение исходного и изменённого текста

Как мы можем видеть по рисунку 2.2 визуально исходный и изменённый текст неотличимы так как изменение цвета идёт на одну единицу.

**Вывод**

В данной лабораторной работе были изучены стенографические методы встраивания/извлечения тайной информации с использованием электронного файла-контейнера текстового формата. Было разработано приложение, которые выполняет данные действия.