Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Отчет по лабораторной работе №14

«Исследование стеганографического метода на основе преобразования наименее значащих бит»

Студентка: Пунько А.А,

ФИТ 3 курс 5 группа

Преподаватель: Берников В. О.

Минск 2020

Задачи:

1.Закрепить теоретические знания из области стеганографического преобразования информации, моделирования стеганосистем, классификации и сущности методов цифровой стеганографии.

2. Изучить алгоритм осаждения/извлечения тайной информации на основе метода НЗБ (LSB – Least Significant Bit), получить опыт практической реализации метода.

3. Разработать приложение для реализации алгоритма осаждения/ извлечения тайной информации с использованием электронного файла-контейнера на основе метода НЗБ.

4. Познакомиться с методиками оценки стеганографической стойкости метода НЗБ.

5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

# Теоретические сведения

Стеганографическая система (stegosystem, стегосистема или стеганосистема – в русскоязычной тематической литературе используются оба сокращения) – совокупность средств и методов, которые используются для формирования скрытого канала передачи (или хранения) информации.

Абстрактно стеганографическая система обычно определяется, как некоторое множество отображений одного пространства (множества возможных сообщений, М) в другое пространство (множество возможных стеганосообщений, S, и наоборот.

Основные компоненты стеганосистемы:

* контейнер, С (файл-контейнер или электронный документ произвольного формата), в котором размещается (осаждается, скрывается) тайное сообщение, М; именно контейнер является упомянутым скрытым каналом;
* тайное сообщение, М, осаждаемое в контейнер для передачи или хранения (например, с целью доказательства или защиты авторских прав на документ-контейнер; здесь речь может идти о невидимых цифровых водяных знаках, ЦВЗ);
* ключи или ключевая информация, K системы, выполняющие ту же функцию, что и криптографические ключи; ключей может быть несколько, в соответствии с этим современные стеганосистемы характеризуют как многоключевые: один ключ отождествляется с методом осаждения/извлечения тайной информации, другой – с выбором элементов (например, битов) контейнера для его модификации при осаждении тайной информации, третий (или третьи) – для предварительного (перед осаждением) преобразования тайной информации (например, на основе помехоустойчивого кодирования, сжатия или зашифрования) и т. д.;
* контейнер с осажденным сообщением или стеганоконтейнер, S, который передается по открытому каналу, также являющемуся важным компонентом анализируемой системы; стеганоконтейнер будем именовать также стеганосообщением;
* для полноты упомянем также субъектов системы: отправителя и получателя.

При построении стеганосистемы должны, таким образом, учитываться следующие основные положения:

* свойства контейнера должны быть модифицированы так, чтобы изменение невозможно было выявить при визуальном контроле; это требование определяет качество сокрытия внедряемого сообщения: для обеспечения беспрепятственного прохождения стеганосообщения по каналу связи оно никоим образом не должно привлечь внимание атакующего;
* противник (интруз) имеет полное представление о стеганографической системе и деталях ее реализации; единственной информацией, которая остается ему неизвестной, является ключ, с помощью которого только его держатель может установить факт присутствия и содержание скрытого сообщения;
* если противник каким-то образом узнает о факте существования скрытого сообщения, это не должно позволить ему извлечь подобные сообщения до тех пор, пока ключ хранится в тайне;
* потенциальный противник должен быть лишен каких-либо технических и иных преимуществ в распознавании или раскрытии содержания тайных сообщений.

Метод НЗБ основывается на ограниченных способностях зрения или слуха человека, вследствие чего людям тяжело различать незначительные вариации цвета или звука. Рассмотрим это на примере 24битного растрового RGB-изображения. Как известно, каждая точка кодируется 3-мя байтами. Каждый байт определяет интенсивность красного (Red), зеленого (Green) и синего (Blue) цветов. Совокупность интенсивностей цвета в каждом из 3-х каналов определяет оттенок пикселя.

Одним из простейших и понятных для решения наших задач является формат BMP (BitMaP) – одна из форм представления растровой графики. Изображение представляется в виде матрицы пикселов, где каждая точка характеризуется тремя параметрами: x-координатой, yкоординатой и цветом кодом на основе RGB-модели. Все операции графического ввода-вывода на экран монитора (принтера и на некоторые другие устройства) в конечном итоге осуществляются в этом формате. Для работы с этим форматом в ОС Windows предусмотрено много специальных функций и структур API, которые помогают производить все необходимые операции на достаточно высоком логическом уровне.

Контейнеры на основе BMP-формата разделяют на два класса: «чистые» и зашумленные. В первых прослеживается связь между младшими и остальными битами элементов цвета, а также видна зависимость самих младших битов между собой. Осаждение сообщения в такой контейнер нарушает такие зависимости, что легко выявляется аналитиком. Если же картинка зашумлена (например, получена со сканера или фотокамеры), то определить осажденное сообщение сложнее. Таким образом, в качестве файлов-контейнеров для метода LSB рекомендуется использовать файлы, которые не были созданы на компьютере изначально.

Другим из растровых форматов используемых в стеганографии контейнеров является формат PNG (Portable Network Graphics). По качеству цветового отображения данный формат превосходит JPEG (Joint Photographic Experts Group) и GIF (Graphics Interchange Format), но размер файла будет на 30-40% больше.

# Практическая часть

Целью данной лабораторной работы является разработать приложение, реализующее метод наименьших значимых битов для осаждения и извлечения сообщения. Стартовое окно приложения приведено на рисунке 1.

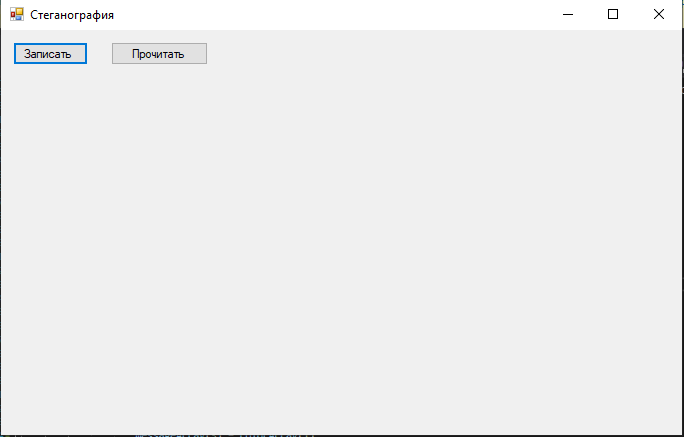


Рисунок 1 – Стартовое окно приложения

Результат осаждения исходного текста представлен на рисунке 2.

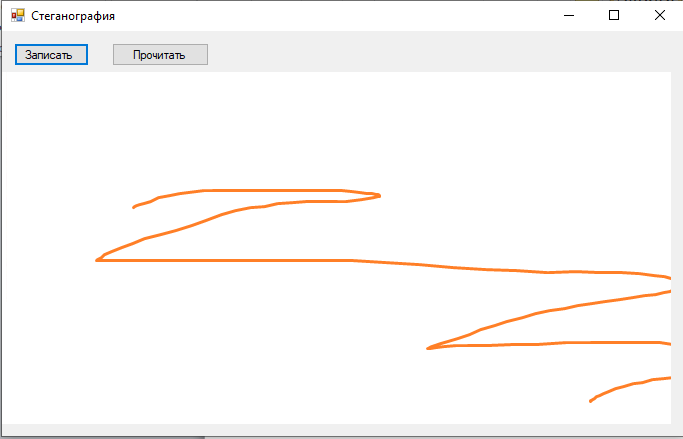


Рисунок 2 – Результат осаждения текста

После извлечения сообщения оно запишется в файл. Если извлечение прошло успешно, то этот файл будет идентичен исходному.

Алгоритм, реализующее осаждение текста представлен в листинге 1.

byte[] Symbol = Encoding.GetEncoding(1251).GetBytes("/");

BitArray ArrBeginSymbol = ByteToBit(Symbol[0]);

Color curColor = bPic.GetPixel(0, 0);

BitArray tempArray = ByteToBit(curColor.R);

tempArray[0] = ArrBeginSymbol[0];

tempArray[1] = ArrBeginSymbol[1];

byte nR = BitToByte(tempArray);

tempArray = ByteToBit(curColor.G);

tempArray[0] = ArrBeginSymbol[2];

tempArray[1] = ArrBeginSymbol[3];

tempArray[2] = ArrBeginSymbol[4];

byte nG = BitToByte(tempArray);

tempArray = ByteToBit(curColor.B);

tempArray[0] = ArrBeginSymbol[5];

tempArray[1] = ArrBeginSymbol[6];

tempArray[2] = ArrBeginSymbol[7];

byte nB = BitToByte(tempArray);

Color nColor = Color.FromArgb(nR, nG, nB);

bPic.SetPixel(0, 0, nColor);

WriteCountText(CountText, bPic)

int index = 0;

bool st = false;

for (int i = 4; i < bPic.Width; i++)

{

for (int j = 0; j < bPic.Height; j++)

{

Color pixelColor = bPic.GetPixel(i, j);

if (index == bList.Count)

{

st = true;

break;

}

BitArray colorArray = ByteToBit(pixelColor.R);

BitArray messageArray = ByteToBit(bList[index]);

colorArray[0] = messageArray[0]; //меняем

colorArray[1] = messageArray[1]; // в нашем цвете биты

byte newR = BitToByte(colorArray);

colorArray = ByteToBit(pixelColor.G);

colorArray[0] = messageArray[2];

colorArray[1] = messageArray[3];

colorArray[2] = messageArray[4];

byte newG = BitToByte(colorArray);

colorArray = ByteToBit(pixelColor.B);

colorArray[0] = messageArray[5];

colorArray[1] = messageArray[6];

colorArray[2] = messageArray[7];

byte newB = BitToByte(colorArray);

Color newColor = Color.FromArgb(newR, newG, newB);

bPic.SetPixel(i, j, newColor);

index++;

}

if (st)

{

break;

}

}

pictureBox1.Image = bPic;

Листинг 1 – Осаждение текста

Алгоритм проверки, зашифровано ли изображение, приведен в листинге 2.

private bool isEncryption(Bitmap scr)

{

byte[] rez = new byte[1];

Color color = scr.GetPixel(0, 0);

BitArray colorArray = ByteToBit(color.R);

BitArray messageArray = ByteToBit(color.R);

messageArray[0] = colorArray[0];

messageArray[1] = colorArray[1];

colorArray = ByteToBit(color.G)

messageArray[2] = colorArray[0];

messageArray[3] = colorArray[1];

messageArray[4] = colorArray[2];

colorArray = ByteToBit(color.B)

messageArray[5] = colorArray[0];

messageArray[6] = colorArray[1];

messageArray[7] = colorArray[2];

rez[0] = BitToByte(messageArray);

string m = Encoding.GetEncoding(1251).GetString(rez);

if (m == "/")

{

return true;

}

else return false;

}

Листинг 2 – Проверка стегосистемы

Алгоритм извлечения исходного текста представлен в листинге 3.

int countSymbol = ReadCountText(bPic); //считали количество зашифрованных символов

byte[] message = new byte[countSymbol];

int index = 0;

bool st = false;

for (int i = 4; i < bPic.Width; i++)

{

for (int j = 0; j < bPic.Height; j++)

{

Color pixelColor = bPic.GetPixel(i, j);

if (index == message.Length)

{

st = true;

break;

}

BitArray colorArray = ByteToBit(pixelColor.R);

BitArray messageArray = ByteToBit(pixelColor.R); ;

messageArray[0] = colorArray[0];

messageArray[1] = colorArray[1];

colorArray = ByteToBit(pixelColor.G);

messageArray[2] = colorArray[0];

messageArray[3] = colorArray[1];

messageArray[4] = colorArray[2];

colorArray = ByteToBit(pixelColor.B);

messageArray[5] = colorArray[0];

messageArray[6] = colorArray[1];

messageArray[7] = colorArray[2];

message[index] = BitToByte(messageArray);

index++;

}

if (st)

{

break;

}

}

string strMessage = Encoding.GetEncoding(1251).GetString(message);

Листинг 3 – Извлечение текста