Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**Отчёт по лабораторной работе №13**

**«Исследование стеганографического метода на основе преобразования наименее значащих битов»**

Исполнитель:

Студент 3 курса группы 4

Гурина К. С.

Руководитель:

Ассистент Сазонова Д. В.

1. **Цель и задачи работы**

**Цель:** изучение стеганографического метода встраивания/извлечения тайной информации с использованием электронного файлаконтейнера на основе преобразования наименее значащих битов (НЗБ), приобретение практических навыков программной реализации данного метода.

**Задачи:**

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

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

3. Разработать приложение для реализации алгоритма встраивания/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

**2. 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. При этом скрытый канал организуется на базе и внутри открытого канала с использованием особенностей восприятия информации. «Скрытость» канала передачи тайной информации отличает стеганографию от криптографии: в первом случае тайной является сам факт наличия канала (передачи информации).

Было отмечено, что к текстовой стеганографии относятся методы, предусматривающие использование в качестве контейнера файла-документа текстового типа.

Многообразие методов текстовой стеганографии подразделяется на синтаксические методы, которые не затрагивают семантику текстового сообщения, и лингвистические, которые основаны на эквивалентной трансформации текстовых файлов-контейнеров, сохраняющей смысловое содержание текста, его семантику.

Для понимания сущности некоторых из методов полезно познакомиться с важнейшими особенностями и параметрами использования стилей (в т. ч. – пространственно-геометрическими параметрами шрифтов), на основе которых строится текстовый файл-контейнер. На рис. 11.1 показаны основные из параметров шрифта.



Рисунок 1.1 – Параметры шрифта

К синтаксическим методам компьютерной стеганографии, которые характеризуются сравнительно невысокой эффективностью (с точки зрения объема осаждаемой информации) относятся следующие (такие методы мы отнесем к числу базовых синтаксических методов):

* изменение расстояния между строками электронного текста (Line-Shift Coding); называется методом изменения межстрочных интервалов; сущность заключается в том, что используется текст с различными межстрочными расстояниями: выделяется максимальное и минимальное расстояния между строками, позволяющее кодировать соответственно символы «1» и «0» осаждаемого сообщения;
* изменение расстояния между словами в одной строке электронного текста (Word-Shift Coding); суть метода состоит в том, что осаждение информации основано на модификации расстояния между словами текста-контейнера;
* изменение количества пробелов между словами (частный случай метода Word-Shift Coding); Основан та том, что, например, чередование одинарного пробела и двойного (хх­\_хх\_\_хх) кодирует «1», переход же с двойного пробела на одинарный кодирует «0» (хх\_\_хх\_хх);
* на основе внесения специфических изменений в шрифты, т. е. начертания отдельных букв (Feature Coding); заключается в изменении написания отдельных букв используемого стандартного шрифта: визуально заметны различные образы, соответствующие буквам с верхними (например, l, t, d) или нижними (например, a, g) выносными элементами (см. рис. 13.1); например, букву «А» можно модифицировать, незначительно укорачивая длинную нижнюю часть буквы (см. рис. 1.2);

а) пустой контейнер; б) заполненный контейнер

Рисунок 1.2 - Пример применения метода Feature Coding

* изменение интервала табуляции; аналогичен вышеописанному методу изменения количества пробелов, только в этом случае меняется не количество пробелов, а соответственно расстояние между строками и интервал табуляции;
* Null Chipper (дословно – несуществующий, нулевой лепет); предполагает размещение тайной информации на установленных позициях слов или в определенных словах текста-контейнера, который, как правило, лишен логического смысла (как видно, действительно лепет);
* увеличение длины строки; предусматривает искусственное увеличение длины каждой строки за счет пробелов: например, нет пробела (определяется положением знака перехода на новую строку) – «0», один пробел – «1» (рис. 13.3);
* использование регистра букв; для обозначения бита секретного сообщения, представленного единицей, используется символ нижнего регистра, а нулем — верхнего (или наоборот);



Рисунок 1.3 - Пример реализации метода увеличения длины строки

* использование невидимых символов; знак пробел кодируется символом с кодом 32, но в тексте его можно заменить также символом, имеющим код 255 (или 0), который является «невидимым» и отображается как пробел.

Рассмотренные базовые методы могут применяться независимо и совместно, сохраняют исходный смысл текста, а обеспечиваемые ими показатели плотности кодирования при совмещении складываются.

Еще одна важная особенность. Перечисленные методы работают успешно до тех пор, пока тексты представлены в коде ASCII.

Методы также легко применяются к любому тексту, независимо от его содержания, назначения и языка. Синтаксические системы стеганографии легко реализуются в программном коде, так как они полностью автоматические и не требуют вмешательства оператора. Однако синтаксические методы неустойчивы к форматированию текста (вспомним робастность систем на основе ЦВЗ), и поэтому информация может быть потеряна при простом применении иного стиля форматирования текста-контейнера, скрывающего в себе стегосообщение. К тому же с помощью синтаксических методов можно передать незначительное количество информации.

Существуют также стеганографические методы, которые интерпретируют текст как двоичное изображение. Необходимо отметить, что данные методы нечувствительны к изменению масштаба документа, что обеспечивает им хорошую устойчивость к большинству искажений, которые могут иметь место при активных атаках.

К числу основных лингвистических методов относятся:

* метод синонимов;
* метод переменной длины слова; основан на том, что длина слов в сообщении зависит от содержания секретного сообщения и способа кодирования слов: обычно одно слово текста-контейнера определенной длины кодирует два бита информации из стеганосообщения; например, слова текста длиной в 4 и 8 символов могут означать комбинацию бит «00», длиной в 5 и 9 — «01», 6 и 10 — «10», 7 и 11 букв — «11»; слова короче 4 и длиннее 11 букв можно вставлять где угодно для лексической и грамматической связки слов в предложении – программное приложение, которое декодирует принятое сообщение (извлекает сообщение из стеганоконтейнера), будет просто игнорировать их;
* метод первой буквы – программа-помощник в этом методе накладывает ограничение уже не на длину слова, а на первую (можно на вторую) букву; обычно одну и ту же комбинацию могут кодировать несколько букв, например, комбинацию «101» означают слова, начинающиеся с «А», «Г» или «Т;
* мимикрия; мимикрия генерирует осмысленный текст, используя синтаксис, описанный в Context Free Grammar (CFG), и встраивает информацию, выбирая из CFG определенные фразы и слова; грамматика CFG – это один из способов описания языка, который состоит из статических слов и фраз языка, а также узлов.

**3. Ход работы**

**Практическое задание:**

Разработать собственное приложение, в котором должен быть реализован метод НЗБ. При этом:

• выбор файла-контейнера – по согласованию с преподавателем;

• реализовать два варианта осаждаемого/извлекаемого сообщения: − собственные фамилия, имя и отчество; − текстовая часть отчета по одной из выполненных лабораторных работ;

• реализовать два метода (на собственный выбор) размещения битового потока осаждаемого сообщения по содержимому контейнера;

• сформировать цветовые матрицы, отображающие каждый задействованный для осаждения уровень младших значащих битов контейнера;

• выполнить визуальный анализ (с привлечением коллег в качестве экспертов) стеганоконтейнеров с различным внутренним содержанием; сделать выводы на основе выполненного анализа. 3. Результаты выполнения работы оформить в виде отчета по установленным правилам.

**Ход работы**

Программа реализована на языке C# и использует библиотеку Aspose.Words для работы с документами Word. Программа предоставляет два основных метода стеганографического преобразования: осаждение и извлечение тайной информации. Эти методы могут быть выполнены двумя способами: с использованием изменения размера пробелов и изменения цвета текста

Сперва необходимо реализовать текстовую стенографию путем модификации числа пробелов в тексте. Функция SizeEncryption реализует шифрование, добавляя пробелы в текст документа для кодирования битов сообщения цикл, реализующий это, представлен на рисунке 3.1

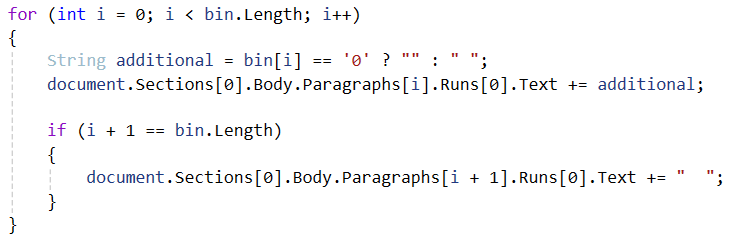
****

Рисунок 3.1 – Модификация пробелов

В цикле бит сообщения представляется следующим образом: 0 – нет пробела, 1 – есть пробел. Пробелы расставляются в конец абзаца. Таким образом, максимальное количество символов, которые возможно зашифровать равно количеству абзацев в исходном документе Word.

Полный код функции, осуществляющей шифрование, представлен в листинге 3.1.

|  |
| --- |
| public static void SizeEncryption()  {  Document document = new Document("sample file.docx");  DocumentBuilder builder = new DocumentBuilder(document);  double lines\_count = document.Sections[0].Body.Paragraphs.Count;  Console.WriteLine("You can encrypt only " + Math.Round(lines\_count / 8) + " Bytes of data");  Console.WriteLine("Enter your message:");  String data = Console.ReadLine();  String bin = StringToBinary(data);  if (bin.Length > Math.Round(lines\_count))  {  Console.WriteLine("Message length is more than possible");  return;  }  for (int i = 0; i < bin.Length; i++)  {  String additional = bin[i] == '0' ? "" : " ";  document.Sections[0].Body.Paragraphs[i].Runs[0].Text += additional;  if (i + 1 == bin.Length)  {  document.Sections[0].Body.Paragraphs[i + 1].Runs[0].Text += " ";  }  }  document.Save("encrypted\_file.docx");  } |

Листинг 3.1 – Код функции SizeEncryption

Работа приложения представлена на рисунке 3.2.

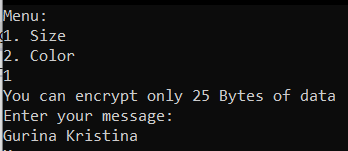
****

Рисунок 3.2 – Запись секретного сообщения в документ

Для расшифрования сообщения была написана функция SizeDecryption, представленная в листинге 3.2. Она извлекает закодированное сообщение из документа, анализируя пробелы в тексте. Одинарный пробел интерпретируется как бит «0», а двойной – как бит «1». Полученное двоичное сообщение преобразуется обратно в текст и выводится на экран.

|  |
| --- |
| public static void SizeDecryption()  {  //Initialization  Document document = new Document("encrypted\_file.docx");  DocumentBuilder builder = new DocumentBuilder(document);  int lines\_count = document.Sections[0].Body.Paragraphs.Count;  String arr = "";  int size = 0;  for (int i = 0; i < lines\_count; i++)  {  if (document.Sections[0].Body.Paragraphs[i].Runs[0].Text.Contains(" "))  {  size = i;  break;  }  if (document.Sections[0].Body.Paragraphs[i].Runs[0].Text.EndsWith(" "))  {  arr += '1';  }  else  {  arr += '0';  }  }  Console.WriteLine("Message: " + BinaryToString(arr));  } |

Листинг 3.2 – Код функции SizeDecryption

Расшифрование сообщения представлено на рисунке 3.2.

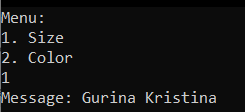
****

Рисунок 3.2 – Расшифрование секретного сообщения в приложении

Затем было необходимо реализовать текстовую стенографию путем модификации цвета символов. Функция, реализующая это, представлен на листинге 3.3.

|  |
| --- |
| public static void ColorEncryption()  {  Document document = new Document("sample file.docx");  DocumentBuilder builder = new DocumentBuilder(document);  double lines\_count = document.Sections[0].Body.Paragraphs.Count;  Console.WriteLine("You can encrypt only " + Math.Round(lines\_count / 8) + " Bytes of data");  Console.WriteLine("Enter your message:");  String data = Console.ReadLine();  String bin = StringToBinary(data);  if (bin.Length > Math.Round(lines\_count))  {  Console.WriteLine("Message length is more than possible");  return;  }  for (int i = 0; i < bin.Length; i++)  {  int additional = bin[i] == '0' ? 0 : 1;  Color newC = Color.FromArgb(0, additional, 0);  document.Sections[0].Body.Paragraphs[i].Runs[0].Font.Color = newC;  if (i + 1 == bin.Length)  {  System.Drawing.Color color = document.Sections[0].Body.Paragraphs[i + 1].Runs[0].Font.Color;  System.Drawing.Color new\_color = Color.FromArgb(0, 1, 1);  document.Sections[0].Body.Paragraphs[i + 1].Runs[0].Font.Color = new\_color;  }  }  document.Save("color\_encrypted\_file.docx");  } |

Листинг 3.1 – Код функции ColorEncryption

Засекречивание сообщения происходит при помощи канала зеленого цвета в системе RGB. Если бит исходного сообщения 1 – значит цвет одной строки текста будет равен (0, 1, 0). В случае нулевого бита в сообщении цвет текста будет черным (0, 0, 0). Окончание сообщения осуществляется при помощи синего канала, в этом случае цвет строки будет (0, 1, 1). Реализация в приложении представлена на рисунке 3.3.

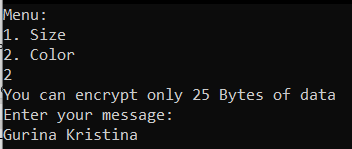
****

Рисунок 3.3 – Запись секретного сообщения в документ

Для расшифрования сообщения была написана функция, представленная на листинге 3.4.

|  |
| --- |
| public static void ColorDecryption()  {  Document document = new Document("color\_encrypted\_file.docx");  DocumentBuilder builder = new DocumentBuilder(document);  int lines\_count = document.Sections[0].Body.Paragraphs.Count;  String arr = "";  int size = 0;  for (int i = 0; i < lines\_count; i++)  {  if (document.Sections[0].Body.Paragraphs[i].Runs[0].Font.Color.G == 1 && document.Sections[0].Body.Paragraphs[i].Runs[0].Font.Color.B == 1)  {  size = i;  break;  }  if (document.Sections[0].Body.Paragraphs[i].Runs[0].Font.Color.G == 1)  {  arr += '1';  }  else  {  arr += '0';  }  }  Console.WriteLine("Message: " + BinaryToString(arr));  } |

Листинг 3.4 – Код функции ColorDecryption

Цикл перебирает все строки текстового документа и смотрит их цвета. Если цвет (0, 0, 0) – значит в битовую последовательность искомого сообщения записывается 0, если цвет (0, 1, 0) – записывается 1, если же цвет (0, 1, 1), то это говорит о конце сообщения.

Расшифрование секретного сообщения в приложении можно увидеть на рисунке 3.4.

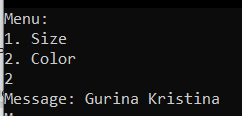
****

Рисунок 3.4 – Расшифрование секретного сообщения в приложении

**Вывод**

В ходе лабораторной работы был изучен принцип стеганографического преобразования информации. Изучены основные алгоритмы осаждения/извлечения тайной информации на основе методов текстовой стеганографии. Было создано приложение для реализации алгоритма осаждения/извлечения тайной информации с использованием методов текстовой стегонографии.