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).

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;

• 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 встраивания/извлечения тайной информации, другой – с выбором элементов (например, битов) контейнера для его модификации при осаждении тайной информации, третий – для предварительного (перед встраиванием) преобразования тайной информации (например, на основе помехоустойчивого кодирования, сжатия или зашифрования) и т. д.;

• контейнер со встроенным 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.

Большинство исследований в предметной области посвящено использованию в качестве стеганоконтейнеров изображений (текст также можно рассматривать как изображение).

Метод НЗБ основывается на ограниченных способностях зрения или слуха человека, вследствие чего людям тяжело различать незначительные вариации цвета или звука. Младшие биты (выделены бледным, справа) дают незначительный «вклад» в изображение по сравнению со старшими. Замена одного или даже нескольких младших битов для человеческого глаза будет почти незаметна, поскольку реально человек может различать около полторы сотни цветовых оттенков.

Одним из простейших и понятных для решения наших задач является формат BMP (BitMaP) – одна из форм представления растровой графики. Изображение представляется в виде матрицы пикселей, где каждая точка характеризуется тремя параметрами: x-координатой, y-координатой и цветовым кодом на основе RGB модели.

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 встроенное сообщение сложнее. Таким образом, в качестве файлов-контейнеров для метода НЗБ рекомендуется использовать файлы, которые не были созданы на компьютере изначально.

**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**

После запуска приложения открывается форма (рисунок 3.1), в которой можно ввести тайный текст и выбрать пустой контейнер.

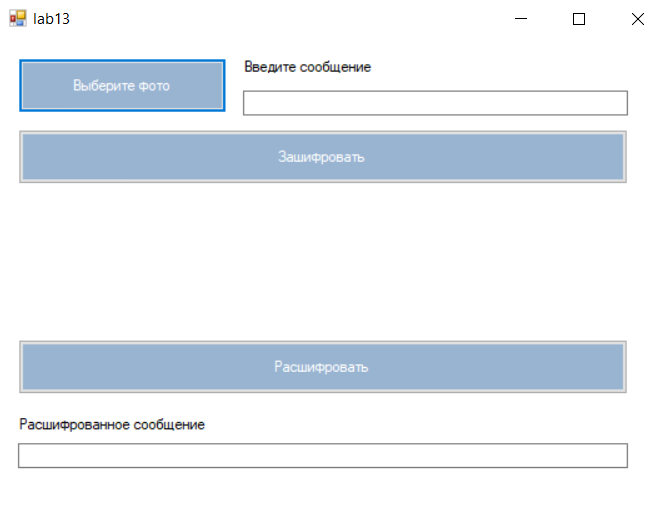


Рисунок 3.1 – Приложение

В качестве контейнера выберем изображение, представленное на рисунке 3.2.



Рисунок 3.2 – Пустой контейнер

Код для реализации метода НЗБ представлен в листинге 3.1.

|  |
| --- |
| private void stegButton\_Click(object sender, EventArgs e)  {  string steg = stegText.Text;  byte[] byteAscii = Encoding.ASCII.GetBytes(steg);  string text = "";  for (int i = 0; i < steg.Length; i++)  {  text += Convert.ToString(byteAscii[i], 2).PadLeft(8, '0');  }  for (int y = 0; y < 256; y++)  {  for (int x = 0; x < 256; x++)  {  redArr[x, y] = redArr[x, y].Substring(0, 7).PadRight(8, '0');  greenArr[x, y] = greenArr[x, y].Substring(0, 7).PadRight(8, '0');  blueArr[x, y] = blueArr[x, y].Substring(0, 7).PadRight(8, '0');  }  }  count = text.Length;  count = (((count - 1) / 3) \* 3) + 3;  text.PadRight(count, '0');  for (int i = 0; i < text.Length / 3; i++)  {  int x = i % 256;  int y = i / 256;  redArr[x, y] = redArr[x, y].Substring(0, 7) + text[i \* 3];  greenArr[x, y] = greenArr[x, y].Substring(0, 7) + text[i \* 3 + 1];  blueArr[x, y] = blueArr[x, y].Substring(0, 7) + text[i \* 3 + 2];  }  Bitmap resultBitmap = new Bitmap(256, 256);  for (int y = 0; y < 256; y++)  {  for (int x = 0; x < 256; x++)  {  resultBitmap.SetPixel(x, y, Color.FromArgb(binToDec(redArr[x, y]), binToDec(greenArr[x, y]), binToDec(blueArr[x, y])));    }  }  resultBitmap.Save("steganography1.png", System.Drawing.Imaging.ImageFormat.Png);  Bitmap resultBitmap2 = new Bitmap(256, 256);  for (int y = 0; y < 256; y++)  {  for (int x = 0; x < 256; x++)  {  Color outputPixel = Color.FromArgb(binToDec(redArr[x, y]), binToDec(greenArr[x, y]), binToDec(blueArr[x, y]));  resultBitmap2.SetPixel(x, y, outputPixel);  }  }    resultBitmap2.Save("steganography2.png", System.Drawing.Imaging.ImageFormat.Png);  } |

Листинге 3.1 – Код функции шифрования

В самом начале мы проходимся по каждому пикселю и устанавливаем значение каждого наименьшего значимого бита каждого канала в «0». Тем самым мы освобождаем во всем изображении каждый 8 бит, который будет нужен для записи тайного сообщения.

Далее идёт подготовка текста для вставки. Текст по таблице ASCII переводится в двоичную систему. Вычисляем количество битов в тексте и округляем до ближайшего кратного 3 (т.к. сообщение должно равномерно переноситься на 3 канала).

После чего мы вставляем каждый бит текста в НЗБ компонентов цвета соответствующих пикселей и создаем новое изображение, устанавливая пиксели на основе модифицированных значений компонентов цвета.

Изображение с осажденными данными при помощи алгоритма наименее значимых битов сохранится в файл с расширением .png. Данное изображение представлено на рисунке 3.3.



Рисунок 3.3 — Изображение с сообщением методом осаждения в наименее значимых битах

Кроме того, был создан алгоритм для встраивания сообщения в младшие биты значений яркости. Этот метод позволяет скрывать данные в изображении, используя младшие разряды яркости каждого пикселя. Результат работы алгоритма сохраняется в файл с расширением .png, изображенный на рисунке 3.4.



Рисунок 3.4 — Изображение с сообщением методом осаждения в младших разрядах яркости

Так же автоматически при выборе контейнера формируется цветовая матрица и результат сохраняется в файл, представленный на рисунке 3.5.



Рисунок 3.5 – Цветовая матрица контейнера

Пустой контейнер и контейнер с осажденным сообщением представлены на рисунке 3.6.



Рисунок 3.6 – Визуальный анализ пустого контейнера и контейнера с осажденным сообщением

Для декодирования необходимо выбрать контейнер с сообщением.

Результат работы приложения представлен на рисунке 3.7.

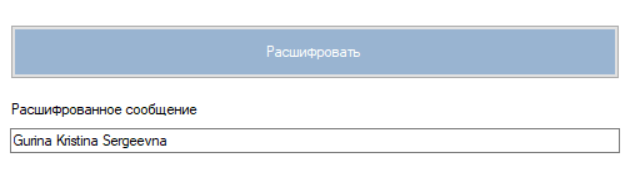


Рисунок 3.7 — Результат расшифрования сообщения

Для извлечения сообщения из стеганоконтейнера необходимо нажать на кнопку «Расшифровать» и выбрать контейнера с осажденным сообщением. оддерживаются форматы PNG и JPG. После выбора и подтверждения изображения создаётся объект Bitmap из выбранного файла.

Извлечение битов из изображения: итерируется каждый пиксель изображения размером 256x256, для каждого пикселя извлекаются компоненты цвета (красный, зелёный, синий) и конвертируются в 8-битное двоичное представление.

Сборка скрытого сообщения: проходимся по всем пикселям изображения, и из каждого пикселя извлекается последний бит каждой цветовой компоненты (красной, зеленой, синей). Эти биты собираются в строку text.

Преобразование строк битов в символы: корректируется длина строки битов text так, чтобы она была кратна 8. Строка битов разбивается на байты (по 8 бит), и каждый байт конвертируется в символ ASCII. Полученные символы собираются в результирующую строку.

Код функции для расшифровки сообщения представлен в листинге 3.2.

|  |
| --- |
| private void deStegOpenFile\_Click(object sender, EventArgs e)  {  OpenFileDialog openFileDialog = new OpenFileDialog();  openFileDialog.Filter = "Image Files (\*.png, \*.jpg) | \*.png; \*.jpg";  if (openFileDialog.ShowDialog() == DialogResult.OK)  {  Bitmap deStegBitmap = new Bitmap(openFileDialog.FileName);  for (int y = 0; y < 256; y++)  {  for (int x = 0; x < 256; x++)  {  redArr[x, y] = Convert.ToString(deStegBitmap.GetPixel(x, y).R, 2).PadLeft(8, '0');  greenArr[x, y] = Convert.ToString(deStegBitmap.GetPixel(x, y).G, 2).PadLeft(8, '0');  blueArr[x, y] = Convert.ToString(deStegBitmap.GetPixel(x, y).B, 2).PadLeft(8, '0');  }  }  string text = "";  for (int i = 0; i < 256; i++)  {  int x = i % 256;  int y = i / 256;  text += redArr[x, y][7].ToString() + greenArr[x, y][7].ToString() + blueArr[x, y][7].ToString();  }  count = text.Length;  count = (((count - 1) / 8) \* 8) + 8;  text.PadRight(count, '0');  var result = "";  for (int i = 0; i < count / 8 - 1; i++)  {  result += (char)binToDec(text.Substring(8 \* i, 8));  }  deStegText.Text = result;  }  } |

Листинге 3.2 – Код функции дешифрования

Мы видим сообщение, которое полностью совпадает с секретным текстом, что означает, что все этапы шифрования и расшифрование прошли успешно.

**Вывод**

В ходе выполнения лабораторной работы был изучен принцип генерации и верификации ЭЦП на основе алгоритмов RSA, Эль-Гамаля и Шнорра. Было разработано приложение, выполняющее генерацию и верификацию ЭЦП на основе данных алгоритмов и была оценена скорость генерации и верификации подписи.

В ходе лабораторной работы был изучен принцип стеганографического преобразования информации. Изучен алгоритм осаждения/извлечения тайной информации на основе метода НЗБ, получен опыт практической реализации метода. Было создано приложение для реализации алгоритма осаждения/извлечения тайной информации с использованием электронного файла-контейнера на основе метода НЗБ.