1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт компьютерных наук и технологий
5. **Кафедра «Информационная безопасность компьютерных систем»**

**Лабораторная работа 6**

1. «Основы компьютерной стенографии»
2. по дисциплине «Практикум по информационной безопасности»

Выполнил:

студент гр. 13558/1 Никулкин В. А.

Проверил:

ассистент преподавателя Алексеев. И. В.

1. Санкт-Петербург
2. 2017
   * + 1. **Цель работы**

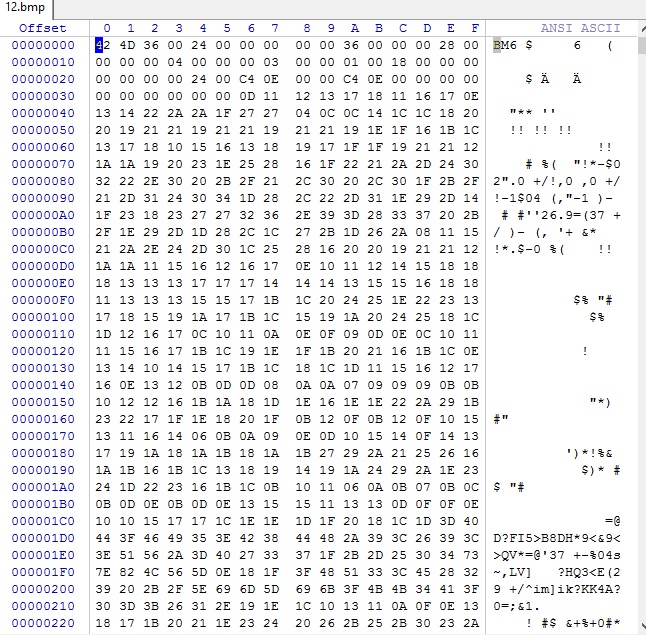
Приобретение навыков исследования свойств стегоконтейнеров, разработки стегосистем и их применение для сокрытия данных при передаче с помощью графических изображений.

**Ход работы**

1. Данное изображение было сохранено в формате BMP



1. Размер данной картинки равен 1024 \* 768 пикселей, 2 359 350 байт
2. Для файла, сохранённого в 24-разрядном формате (сколько бит приходится на каждый пиксель, глубина цвета), начало разложения в редакторе WinHex выглядит так :



Это отвечает за заголовок файла, который должен содержать информацию о его формате.

Цвет изображение задаётся в цветовом формате RGB.

Содержательная часть картинки составляет 1024 × 768 × 3 = 2 359 296 байт. То есть для информации о файле отводится

2 359 350 – 2 359 296 = 54 байта.

Степень упаковки – n (n бит текста на байт картинки, то есть n бит текста на 8 битов картинки). Пусть n = 1. Максимальный размер сообщения – 2 359 296 байт (то есть 2 359 296 символов). Получаем формулу максимального размера сообщения: maxMessage = size × n, где size – размер содержательной части изображения.

Степень упаковки задается от 1 до 8. Причем чем больше степень упаковки, тем больше будет искажаться картинка. При степени упаковки 8 байт текста будет кодироваться в байт одного пикселя изображения, что приведет к очень сильному искажению картинки.

1. Листинг кодера:

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <io.h>

#include <math.h>

#include <malloc.h>

#include <windows.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "rus");

FILE \*p = fopen ("1.bmp", "rb+");

FILE \*np = fopen ("2.bmp", "wb");

FILE \*mes = fopen ("mess.txt", "r");

int counter1 = 0;

int counter2 = 0;

char symbol1;

char symbol2;

fseek (p, 0, SEEK\_END);

int size2 = filelength (fileno (p));

fseek (p, 0, SEEK\_SET);

fseek (mes, 0, SEEK\_END);

int size1 = filelength (fileno (mes));

fseek (mes, 0, SEEK\_SET);

printf("Размер сообщения: %d байт\n", size1);

printf("Размер картинки: %d байт\n", size2);

int maxm = size2;

maxm -= 54;

int n;

printf("Степень упаковки n = ");

scanf("%d", &n);

if(n > 8)

{

printf("Ошибка, слишком большая степень упаковки");

fclose(mes);

fclose(p);

fclose(np);

remove("2.bmp");

return 0;

}

maxm \*= n;

maxm /= 8;

if (size1 > maxm)

{

printf("Ошибка, слишком большое сообщение\n");

fclose(p);

fclose(np);

fclose(mes);

remove("2.bmp");

return 0;

}

while(!feof(mes))

{

fread(&symbol1, sizeof(unsigned char), 1, mes);

++counter1;

}

--counter1;

while(!feof(p))

{

fread(&symbol2, sizeof(unsigned char), 1, p);

++counter2;

}

--counter2;

char \* pi = (char\*) calloc (counter2, sizeof(char));

char \* npi = (char\*) calloc (counter2, sizeof(char));

char \* me = (char\*) calloc (counter1, sizeof(char));

int i;

int per;

int des;

int t = counter2 - 1;

int buffer1[8];

int buffer2[8];

fseek (p, 0, SEEK\_SET);

fseek (mes, 0, SEEK\_SET);

for (i = 0; i < counter1; ++i)

{

symbol1 = fgetc(mes);

me[i] = symbol1;

}

for (i = 0; i < counter2; ++i)

{

symbol2 = fgetc(p);

pi[i] = symbol2;

}

for(i = 0; i < counter1; ++i)

{

symbol1 = me[i];

for(int j = 7; j >= 0 ; --j)

{

buffer1[j] = symbol1 % 2;

symbol1 /= 2;

}

for(int k = 0; k < 8 ; ++k)

{

symbol2 = pi[t];

for(int j = 7; j >= 0 ; --j)

{

buffer2[j] = symbol2 % 2;

symbol2 /= 2;

}

per = 0;

while (per < n)

{

buffer2[7 - per] = buffer1[k];

++per;

if (per == n)

break;

++k;

}

des = 0;

for(int j = 0; j < 8; ++j)

des = (buffer2[j] \* pow(2, (7 - j))) + des;

symbol2 = des;

npi[t] = symbol2;

--t;

}

}

for (t; t >= 0; --t)

npi[t] = pi[t];

for (i = 0; i < counter2; ++i)

{

symbol1 = npi[i];

fputc(symbol1, np);

}

free(npi);

free(me);

free(pi);

fclose(mes);

fclose(p);

fclose(np);

return 0;

}

Листинг декодера:

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <io.h>

#include <math.h>

#include <malloc.h>

#include <windows.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "rus");

unsigned char symbol1;

unsigned char symbol2;

int counter1 = 0;

int counter2 = 0;

FILE \*p = fopen("2.bmp", "rb+");

FILE \*mes = fopen("mess.txt", "wb");

while(!feof(p))

{

fread(&symbol1, sizeof(char), 1, p);

++counter1;

}

--counter1;

fseek(p, 0, SEEK\_SET);

int n;

printf("Введите степень упаковки n = ");

scanf("%d", &n);

printf("Количество символов зашифровонного сообщения = ");

scanf("%d", &counter2);

if(counter2 > counter1)

{

printf("Ошибка, слишком большое сообщение");

fclose(p);

fclose(mes);

remove("mess.txt");

return 0;

}

if(n > 8)

{

printf("Ошибка, степень упаковки должна быть от 1 до 8");

fclose(p);

fclose(mes);

remove("mess.txt");

return 0;

}

unsigned char\* me = (unsigned char\*) calloc (counter2, sizeof(unsigned char));

unsigned char\* pi = (unsigned char\*) calloc (counter1, sizeof(unsigned char));

int buffer1[8];

int buffer2[8];

int i;

int j;

for(i = 0; i < counter1; ++i)

{

symbol1 = fgetc(p);

pi[i] = symbol1;

}

int t = counter1 - 1;

for(i = 0; i < counter2; ++i)

{

for(int k = 0; k < 8; ++k)

{

symbol1 = pi[t];

for(j = 7; j >= 0 ; --j)

{

buffer1[j] = symbol1%2;

symbol1 /= 2;

}

int per = 0;

while(per < n)

{

buffer2[k] = buffer1[7 - per];

++per;

if (per == n)

break;

++k;

}

--t;

}

int des = 0;

for(j = 0; j < 8; ++j)

des += (buffer2[j] \* pow(2, (7 - j)));

symbol2 = des;

me[i] = symbol2;

}

for(i = 0; i < counter2; ++i)

fwrite(&me[i], sizeof(unsigned char), 1, mes);

free(me);

free(pi);

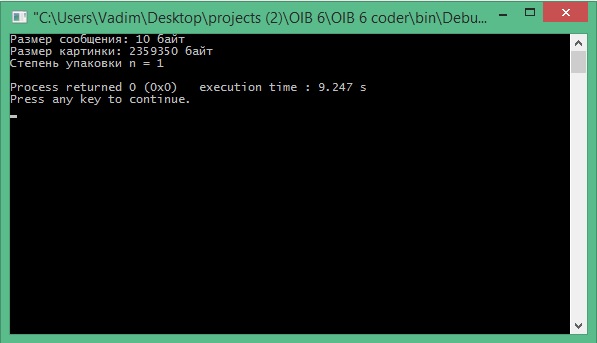
fclose(p);

fclose(mes);

return 0;

}

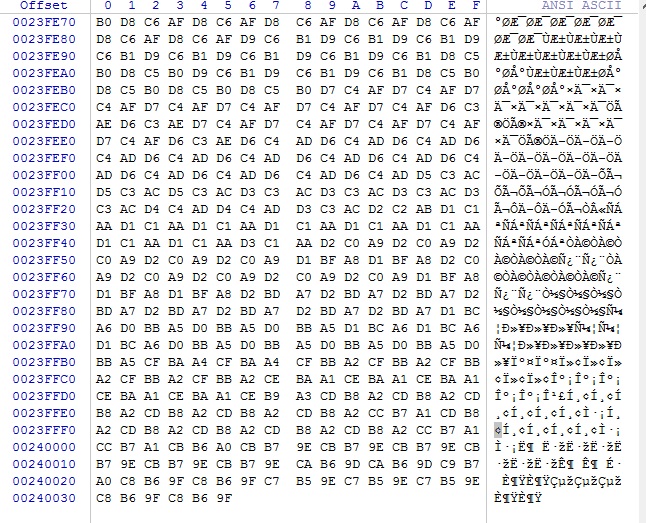
1. Была создана программа-кодер, которая поместила сообщение в картинку. Вначале задается степень упаковки(n), после того, как программа отработала, получается картинка, в которой зашифрован текст.



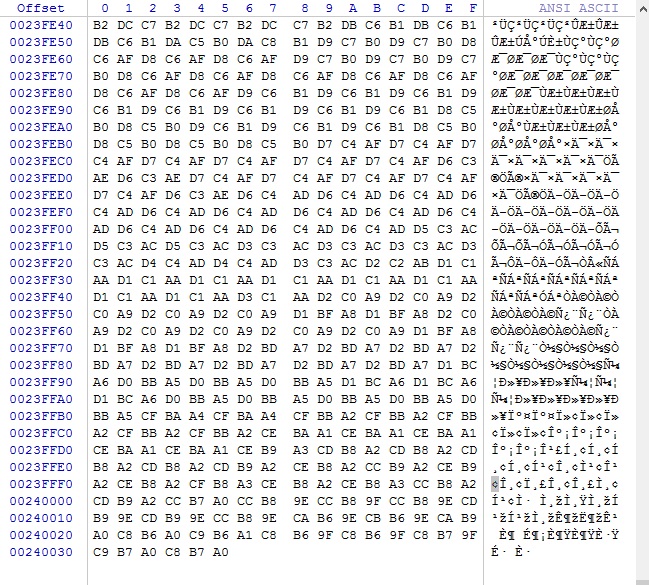
1. После работы данной программы сообщение «OIB 6 LABA» было зашифровано в картинку. При этом заметных глазу искажений нет.



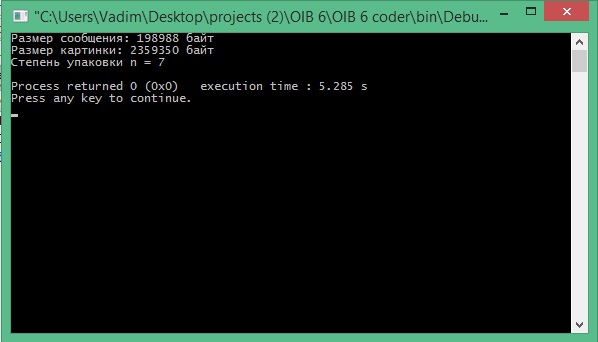
Но значение младших битв изменилось. Это можно увидеть с помощью HEX-редактора

До изменений

После изменений

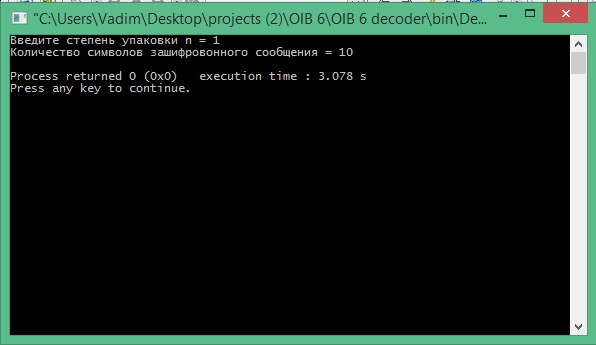


1. Если изменить размер сообщения и степень упаковки, то произойдет сильное искажение изображения



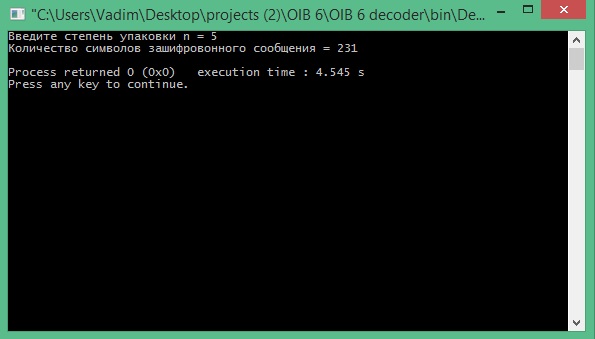


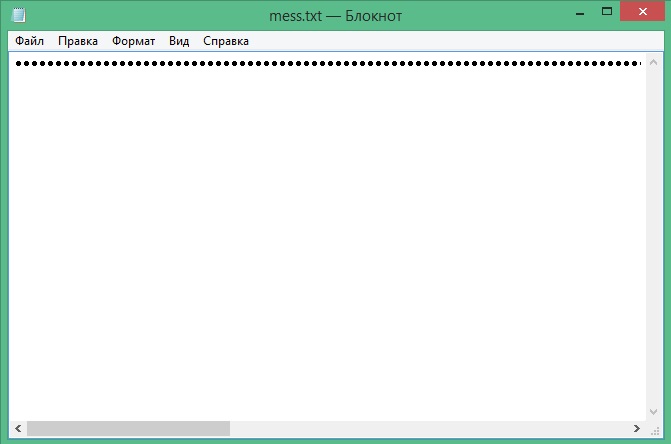
1. Была разработана программа-декодер, которая раскодирует сообщение, находящееся в картинке. Для раскодировки требуется знать степень упаковки и длину сообщения. Например, раскодируем сообщение, помещенное в картинку в 5 пункте.



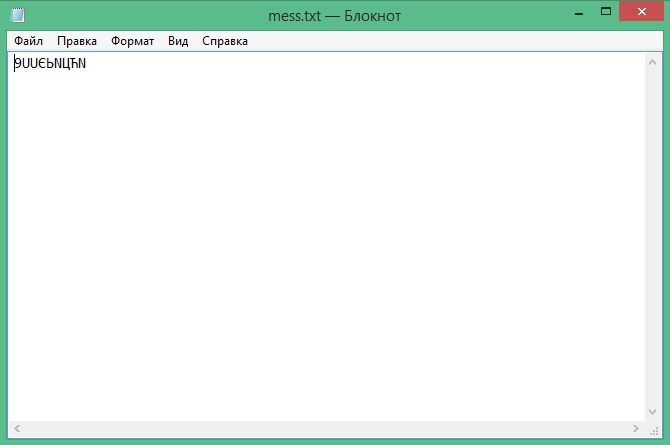
После того, как программа отработала, было раскодировано сообщение. В итоге получаем то же самое закодированное сообщение «OIB 6 LABA».

1. Если при использовании декодера указать неправильное количество символов, закодированного сообщения, и неправильную степень упаковки, то получим следующий результат

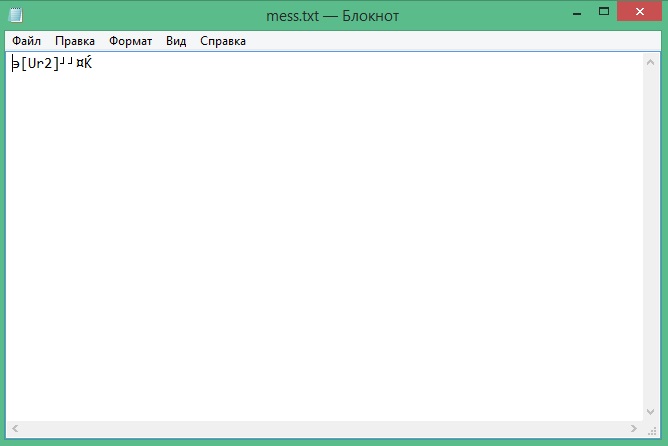




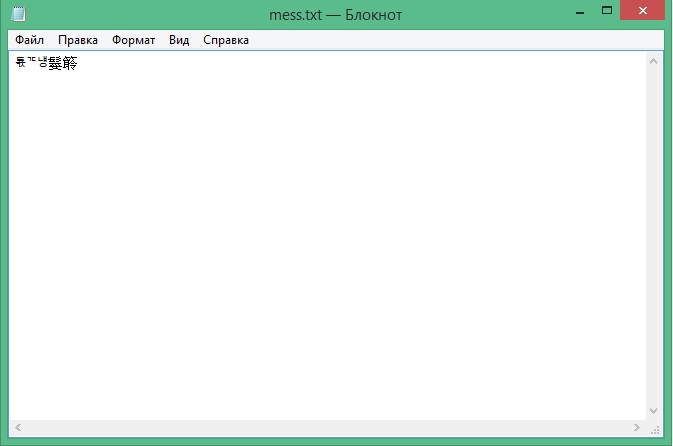
1. При повороте расшифруемого сообщения на 180 градусов получаем



1. При отображении изображения по горизонтали получаем



1. При растяжении изображения получаем



1. Если же отменить все изменения, то мы снова получим сообщение «OIB 6 LABA».

**Ответы на контрольные вопросы**

1. Функции элементов стегосистемы: стегоконтейнер – информационная последовательность, в которой скрывается сообщение; предварительный кодер – устройство, предназначенное для преобразования скрытого сообщения к виду, удобному для встраивания в стегоконтейнер; кодер – устройство, которое осуществляет вложение сообщения в другие данные с учётом их модели; детектор определяет наличие стегосообщения; декодер - устройство, которое восстанавливает скрытое сообщение.
2. Виды атак на стегосистему: пассивные, активные и злоумышленные. Пассивные могут только обнаруживать факт наличия стегоканала, активные способны удалить или разрушить стегосообщение, а злоумышленные – создавать ложные стегоконтейнеры.
3. Можно использовать дополнительные ЦВЗ, которые регистрируют выполняемые нарушителем преобразования. Так же, вложение ЦВЗ в визуально значимые области объекта, которые не могут быть удалены из него без существенной деградации. Блочный детектор - модифицированное изображение разбивается на блоки, для каждого блока анализируются всевозможные искажения. Появляется возможность «обратить» внесённые нарушителем искажения.
4. Стойкость стегосистемы - способность скрывать от квалифицированного нарушителя факт скрытой передачи сообщений, способность противостоять попыткам нарушителя разрушить или удалить скрытно передаваемые сообщения, способность подтвердить или опровергнуть подлинность скрытно передаваемой информации.
5. При сжатии графического изображения алгоритм встраивания ЦВЗ разрушается.

**Вывод**

Была проведена работа по созданию стегосистемы. Полученная система не может считаться стойкой, поскольку используется простейший принцип сокрытия информации. Также я узнал, в чем заключается принципиальное отличие криптографии от стенографии.