Лабораторная работа #3

Стеганография

Стеганогра́фия - это наука о скрытой передаче информации путём сохранения в тайне самого факта передачи. Этот термин ввел в 1499 году Иоганн Тритемий в своем трактате «Стеганография» (Steganographia), зашифрованном под магическую книгу.

В отличие от криптографии, которая скрывает содержимое секретного сообщения, стеганография скрывает сам факт его существования. Как правило, сообщение будет выглядеть как что-либо иное, например, как изображение, статья, список покупок, письмо или судоку. Стеганографию обычно используют совместно с методами криптографии, таким образом, дополняя её.

Преимущество стеганографии над чистой криптографией состоит в том, что сообщения не привлекают к себе внимания. Сообщения, факт шифрования которых не скрыт, вызывают подозрение и могут быть сами по себе уличающими в тех странах, в которых запрещена криптография. Таким образом, криптография защищает содержание сообщения, а стеганография защищает сам факт наличия каких-либо скрытых посланий.

Метод LSB

LSB (Least Significant Bit, наименьший значащий бит) — суть этого метода заключается в замене последних значащих битов в контейнере (изображения, аудио или видеозаписи) на биты скрываемого сообщения. Разница между пустым и заполненным контейнерами должна быть не ощутима для органов восприятия человека.

Суть метода заключается в следующем: Допустим, имеется 8-битное изображение в градациях серого. 00h (00000000b) обозначает чёрный цвет, FFh (11111111b) — белый. Всего имеется 256 градаций (2^8). Также предположим, что сообщение состоит из 1 байта — например, 01101011b. При использовании 2 младших бит в описаниях пикселей, нам потребуется 4 пикселя. Допустим, они чёрного цвета. Тогда пиксели, содержащие скрытое сообщение, будут выглядеть следующим образом: 00000001 00000010 00000010 00000011. Тогда цвет пикселей изменится: первого — на 1/255, второго и третьего — на 2/255 и четвёртого — на 3/255. Такие градации, мало того, что незаметны для человека, могут вообще не отобразиться при использовании низкокачественных устройств вывода.

Методы LSB являются неустойчивыми ко всем видам атак и могут быть использованы только при отсутствии шума в канале передачи данных.

Обнаружение LSB-кодированного осуществляется по аномальным характеристикам распределения значений диапазона младших битов отсчётов цифрового сигнала.

Другие методы скрытия информации в графических файлах ориентированы на форматы файлов с потерей, к примеру, JPEG. В отличие от LSB они более устойчивы к геометрическим преобразованиям. Это получается за счёт варьирования в широком диапазоне качества изображения, что приводит к невозможности определения источника изображения.

Формат файла bmp

Ниже приведена таблица, показывающая назначение полей структуры BMP файла.

Смещение	Длина поля	Описание поля (что тут находится)
Заголовок файла		
0	2	Код 4D42h - Буквы 'BM'
2	4	Размер файла в байтах
6	2	0 (Резервное поле)
8	2	0 (Резервное поле)
10	4	Смещение, с которого начинается само изображение (растр).
Заголовок BITMAP (Информация об изображении)		
14	4	Размер заголовка BITMAP (в байтах) равно 40
18	4	Ширина изображения в пикселях
22	4	Высота изображения в пикселях
26	2	Число плоскостей, должно быть 1
28	2	Бит/пиксел: 1, 4, 8 или 24
30	4	Тип сжатия
34	4	0 или размер сжатого изображения в байтах.
38	4	Горизонтальное разрешение, пиксел/м
42	4	Вертикальное разрешение, пиксел/м
46	4	Количество используемых цветов
50	4	Количество "важных" цветов.
Палитра (Карта цветов для N цветов), если используется		
54	4*N	Палитра

Элемент палитры представляет собой четырёхбайтовую запись (структуру). В этой структуре хранятся составляющие R-красного, G-зеленого и B-синего цветов. Один байт зарезервирован. Палитра может и не использоваться, например в True Color.

Структура элемента палитры:

```
typedef struct tagRGBQUAD
{
	char	rgbBlue;
	char	rgbGreen;
	char	rgbRed;
	char	rgbReserved;
} RGBQUAD;
```

В поле тип сжатия должно стоять 0 - сжатие не используется, 1 - RLE8 сжатие, 2 - RLE4 сжатие. RLE8 - используется для сжатия 256-ти цветного изображения, RLE4 - используется для сжатия 16-ти цветного изображения.

В поле по смещению 28 (Бит/пиксел) должно стоять 1 - черно-белое изображение, 4 - 16-ти цветное, 8 - 256-ти цветное, 24 - True Color

```
typedef unsigned long DWORD;	// Двойное слово - 32 бита (разряда)
typedef unsigned int WORD;	//  Слово - 16 бит (разрядов)
typedef signed long LONG;
typedef unsigned int UINT;
// Заголовок файла
typedef struct tagBITMAPFILEHEADER
{
	WORD bfType;		// 'BM' = 4D42h
	DWORD bfSize;
	WORD bfReserved1;
	WORD bfReserved2;
	DWORD bfOffBits;	// Смещение к растру
} BITMAPFILEHEADER;


// Заголовок Bitmap
typedef struct tagBITMAPINFOHEADER
{
	DWORD biSize;
	LONG biWidth;
	LONG biHeight;
	WORD biPlanes;
	WORD biBitCount;
	DWORD biCompression;
	DWORD biSizeImage;
	LONG biXPelsPerMeter;
	LONG biYPelsPerMeter;
	DWORD biClrUsed;
	DWORD biClrImportant;
} BITMAPINFOHEADER;
```

Данные структуры уже определены. Поэтому, если вы используете C/C++ просто подключите хедер windows.h

ВАЖНО!!!

Изображение сохраняется построчно **СНИЗУ-ВВЕРХ**. Для хранения каждой строки выделяется кратное 4 количество байт. В незначащих байтах хранится мусор.

Старшему биту или тетраде соответствует самый левый пиксел. При хранении изображения True Color каждому пикселу соответствуют три последовательные байта, хранящие составляющие цвета B, G, R; (не R, G, B).

Задание

Написать программу для сокрытия/(извлечения сокрытой) информации в bmp файле (используется True Color). Формат сокрытия 1 байт скрываемой информации скрывается в одном пикселе (по 2 бита на цвет(6 бит) и еще 2 бита на неиспользуемый в True Color альфа канал). Кодируется следующим образом:

```
void hide_byte_into_pixel(RGBQUAD *pixel, uint8_t hide_byte)
{
	pixel->rgbBlue &= (0xFC);
	pixel->rgbBlue |= (hide_byte >> 6) & 0x3;
	pixel->rgbGreen &= (0xFC);
	pixel->rgbGreen |= (hide_byte >> 4) & 0x3;
	pixel->rgbRed &= (0xFC);
	pixel->rgbRed |= (hide_byte >> 2) & 0x3;
	pixel->rgbReserved &= (0xFC);
	pixel->rgbReserved |= (hide_byte) & 0x3;
}
```

Признаком конца файла считать EOF (0xFF). Во вложении есть bmp файлы, в которых спрятаны txt файлы. Вариант задания определяется остатком от деления на 10.