

<b>WIEIK</b>	Skład osobowy: Marcin Woźniak Mateusz Piłat	Data: 02.02.16
<b>Grafika komputerowa I multimedia</b>	Temat: Ukrywanie tekstu zaszyfrowanego kluczem xor w obrazkach BMP	

BMP (bitmapa) to format pliku graficznego o prostej strukturze, który może zawierać w sobie kompresję bezstratną RLE.

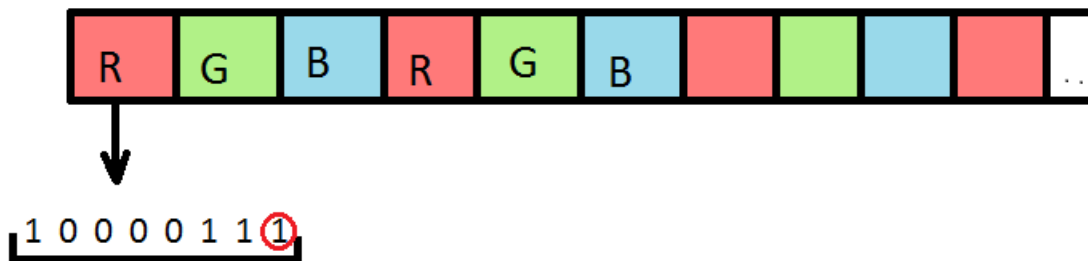
Przy pobieraniu obrazka do naszego programu będziemy musieli zwrócić uwagę na następujące bajty z nagłówka:

- od 10 do 14 bajtu od początku bitmapy znajduje się czterobajtowa liczba będąca wskaźnikiem do obszaru danych. Liczba ta jest nam potrzebna, gdyż dopiero od tej pozycji możemy zacząć kodować informacje, cały nagłówek musi pozostać niezmieniony;
- od 18 do 22 bajtu znajduje się szerokość obrazka w punktach. Wartość ta będzie nam potrzebna do obliczenia liczby nieznaczących zer na końcu każdego wiersza;
- od 34 do 38 bajtu znajduje się długość obszaru danych, wartość ta musi być ośmiokrotnie większa od wielkości pliku tekstowego.

Cechy charakterystyczne formatu BMP:

- każdy piksel jest opisany za pomocą jednego z 256 kolorów (8-bit) lub za pomocą trójki bajtów (24-bit), które opisują zawartość koloru czerwonego, zielonego i niebieskiego;
- format bmp opisuje ciąg pikseli, z których składa się obrazek, czyli jeżeli utworzymy 8-bitową bitmapę składającą się z 4 punktów, to w bitmapie zostaną one opisane jako 4 bajty charakteryzujące kolor odpowiedniego punktu;
- długość każdego wiersza obrazu jest wyrównywana do pełnych 4 bajtów przez uzupełnianie zerami (nieznaczące zera);
- wartości czytane z nagłówka należy odczytać od końca.

Idea ukrycia tekstu w bitmapie, polega na takiej manipulacji bitami, żeby jak najmniej wpłynąć na jakość obrazu, który będzie przechowywał tekst. Osiągniemy to podmieniając najmłodsze bity każdego bajtu obrazka.



Czyli, np. wartość pierwszego bajtu na rys. wyżej to  $1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 128 + 4 + 2 + 1 = 135$

135 to nasza wartość dziesiętna dla pierwszego bajtu koloru obrazka. Jak widać bit, który najmniej wpływa nam na wartość tej liczby to ten najbardziej po prawej (najmłodszy), który zmienia jedynie o 1. Chcąc chować tekst w naszej bitmapie wykorzystamy ten właśnie bit. Dzięki takiej podmianie ludzkie oko nie zauważy tak małej różnicy co da nam złudzenie, że patrzymy na obraz oryginalny.

A więc każdy znak pliku tekstowego .txt zamieniamy na binarną liczbę z kodu ascii. Każda taka wartość składa się z 8 bitów, co oznacza że nasz obrazek musi być osiem razy większy niż plik tekstowy, który chcemy ukryć (każdy pojedynczy znak zostaje zapisany na 8 bajtach).

Do szyfrowania wiadomości używamy operacji logicznej XOR (Exclusive OR), którą wyróżnia to, że jest odwracalna i przy podaniu klucza dostaniemy właściwy tekst odczytany z bitmapy.

