

WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

WYDZIAŁ CYBERNETYKI



Steganografia Lab. 1

Student

X

Prowadzący laboratoria:

Y

Spis treści

Steganografia.....	1
Lab. 1	1
Treść zadania	2
Kod realizujący zadanie.....	2
Opis rozwiązania.....	3

Treść zadania

1. Napisać skrypt w programie Matlab wczytujący bitmapę w 24-bitowej głębi koloru.
2. Wykorzystując podmianę najmniej znaczącego bitu koloru czerwonego danego piksela, ukryć własny numer albumu w obrazie.
3. Dane powinny zostać ukryte w co drugim pikselu obrazu licząc od piksela z wiersza i kolumny o indeksie odpowiadającym ostatniej cyfrze numeru albumu.
4. Napisać procedurę odczytującą ukryte dane.

Kod realizujący zadanie

Skrypt odpowiadający za ukrywanie danego ciągu znaków w obrazie:

```
encrypt_miazga.m  decrypt_miazga.m  +
1  %wczytanie bitmapy w 24-bitowej głębi koloru
2  image = imread('eminem.jpg');
3  %zmiana rozmiaru wczytanego zdjęcia
4  image = imresize(image, [1200 1200]);
5  %numer albumu przekazywany jako wiadomosc
6  message = '71689';
7  %zamiana na kody ascii
8  ascii_value = uint8(message);
9  %konwersja wartości dziesiętnych na binarne
10 binary_message = transpose(dec2bin(ascii_value, 8));
11 binary_message = binary_message(:);
12 %zapisanie dlugosci binarnej wiadomosci
13 len_binary_message = length(binary_message);
14 %konwersja tablicy char na numeryczna
15 binary_num_message = str2num(binary_message);
16 embed_counter = 1;
17 %ostatnia cyfra numeru albumu to 9
18 %dane ukrywane beda w co drugim pikselu
19 %przejscie po obrazie (rozpoczynając od piksela z wiersza i kolumny o indeksie odpowiadającym ostatniej cyfrze numeru albumu)
20 for i = 9 : 2 : 1200
21     for j = 9 : 2 : 1200
22
23         if(embed_counter <= len_binary_message)
24
25             LSB = mod(double(image(i, j, 1)), 2); %image(:, :, 1) oznacza warstwe czerwona
26             temp = double(xor(LSB, binary_num_message(embed_counter)));
27             image(i, j, 1) = image(i, j, 1)+temp;
28             embed_counter = embed_counter+1;
29         end
30     end
31 end
32 %zapisanie obrazu powstalego w wyniku przekształcen
33 imwrite(image, 'eminem_stegano.png');
```

Skrypt odpowiadający za wydobywanie ukrytego ciągu znaków z obrazu:

```
encrypt_miazga.m  decrypt_miazga.m  +
1  %wczytanie zdjęcia z ukryta wiadomoscia
2  image2 = imread('eminem_stegano.png');
3  %liczba znakow ukrytej wiadomosci
4  chars = 5;
5  %dlugosc wiadomosci w bitach
6  message_length = chars * 8;
7  counter = 1;
8  %przejscie po obrazie
9  for i = 9 : 2 : 1200
10     for j = 9 : 2 : 1200
11
12         if (counter <= message_length )
13             extracted_bits(counter, 1) = mod(double(image2(i, j, 1)), 2);
14             counter = counter + 1;
15         end
16     end
17 end
18 % potęgi liczby 2 do odzyskania znaków ascii z binarki
19 binValues = [ 128 64 32 16 8 4 2 1 ];
20 %odkodowanie wiadomości
21 binMatrix = reshape(extracted_bits, 8,(message_length/8));
22 textString = char(binValues*binMatrix);
23 disp(textString);
```

Efekt wykonania obu skryptów, drugi skrypt wypisał wydobyty ze zdjęcia numer indeksu:

```
>> encrypt_miazga
>> decrypt_miazga
71689
fx >>
<
```

Opis rozwiązania

Zadanie zrealizowane zostało w postaci 2 skryptów: `encrypt_miazga` oraz `decrypt_miazga`, pierwszy skrypt odpowiada za ukrycie numeru indeksu studenta (71689) w zdjęciu, które zapisane zostaje pod nazwą „eminem_stegano.png”. Drugi skrypt wczytuje obraz zapisany pod nazwą „eminem_stegano.png” i wydobywa z niego ukryty numer indeksu, na samym końcu wypisując go na ekran.

Pierwszy skrypt - działanie:

Do zmiennej `image` wczytany zostaje obraz, który został dostarczony jako obraz, w którym będziemy chcieli ukrywać wiadomość – nazwa obrazu: „eminem.jpg”. Następnie pod zmienną `message` zostaje zapisany numer indeksu studenta, następnie ten ciąg znaków zamieniany jest na ciąg kodów ascii, a w kolejnym kroku ciąg kodów ascii zamieniany jest na ciąg binarny. Ciąg binarny ukrywany jest w zdjęciu, zaczynając od piksela o indeksie (9,9) - gdzie liczba 9 to ostatni numer albumu studenta - i przemieszczając się co drugi piksel. Na sam koniec działania skryptu zapisany zostaje obraz, który ma w sobie ukrytą wiadomość.

Drugi skrypt – działanie:

Skrypt wczytuje obraz z ukrytą wiadomością, pod zmienną `chars` zostaje zapisana liczba znaków, z których składa się docelowa wiadomość, którą będzie trzeba „odzyskać”. W moim przypadku pod zmienną `chars` zapisane zostaje 5, ponieważ numer albumu składa się z 5 cyfr. W kolejnych krokach następuje iteracja, analogiczna do pierwszego skryptu (czyli iterujemy od piksela o indeksach równych ostatniej cyfrze w numerze albumu), w czasie iteracji odczytana zostaje ukryta wiadomość,

a na samym końcu następuje konwersja ukrytej wiadomości z postaci binarnej do postaci zrozumiałej dla użytkownika.