WOJSKOWA AKADEMIA TECHNICZNA
im. Jarosława Dąbrowskiego
WYDZIAŁ CYBERNETYKI

Steganografia
Lab. 3

Student
x

Prowadzący laboratoria:
y

Spis treści

Treść zadania

Instrukcje

1. Napisać skrypt w programie Matlab wczytujący plik dźwiękowy WAV.
2. Wykorzystując podmianę najmniej znaczących bitów próbek, ukryć w pliku tekst wskazany jako parametr odczytany z linii poleceń.
3. Dane powinny zostać ukrywane w sposób pseudolosowy, z wykorzystaniem wybranej funkcji skrótu przyjmującej jako parametr liczbę wprowadzoną w linii poleceń
4. Napisać skrypt odczytujący ukryte w ten sposób dane.

Zadanie zwrócić w postaci sprawozdania opisującego wykonane zadanie oraz kodów źródłowych (m-pliku). Nie umieszczać skompresowanych archiwum w programie Teams - jedynie pliki nieskompresowane.

Kod realizujący zadanie

Zadanie zrealizowano w postaci dwóch skryptów:

- encrypt.m – skrypt jest odpowiedzialny za ukrycie przekazanego przez użytkownika ciągu znaków w pliku audio formatu wav
- decrypt.m – skrypt jest odpowiedzialny za wydobycie ukrytego ciągu znaków z pliku audio formatu wav

Poniżej zaprezentowano zawartość skryptów:

Skrypt ukrywający wiadomość

decrypt.m | encrypt.m

```
% wczytanie pliku dźwiękowego w formacie wav
[start_audio, f] = audioread('vintagetel.wav');

% normalizacja audio
audio = uint8(255*(start_audio + 0.5));

len_audio = length(audio)

%pobranie wiadomości od użytkownika
message = input('Podaj wiadomosc do ukrycia: ', 's');

%zamiana na kody ascii
ascii_value = uint8(message);

%konwersja wartości dziesiętnych na binarne
binary_message = transpose(dec2bin(ascii_value, 8));
binary_message = binary_message(:);

%zapisanie dlugosci binarnej wiadomosci
len_binary_message = length(binary_message);

%konwersja tablicy char na numeryczna
binary_num_message = str2num(binary_message);

%pobranie klucza od użytkownika
key = input('Podaj klucz(liczba): ', 's');

%funkcja skrótu SHA1
sha1hasher = System.Security.Cryptography.SHA1Managed;
hash = sha1hasher.ComputeHash(uint8(key));
sha1= uint8(hash);

%konwersja wartości dzisiętnych na binarne, tym razem dla klucza
binary_key = transpose(dec2bin(sha1, 8));
binary_key = binary_key(:);

%zapisanie dlugosci binarnej klucza
binary_key_length = length(binary_key);

%konwersja tablicy char na numeryczna dla klucza
binary_num_key = str2num(binary_key);

counter = 1;
counter_sha = 1;

if len_binary_message <= len_audio
    %przejście po pliku audio
    while counter <= len_binary_message
        if binary_num_key(counter_sha) == 1
            LSB = mod(double(audio(counter)), 2);
            temp = double(xor(LSB, binary_num_message(counter)));
            audio(counter) = audio(counter)+temp;
            counter = counter+1;
        end
        counter_sha = counter_sha+1;
        %w przypadku, gdy wiadomość dłuższa niż 160 bitów:
        if counter_sha > binary_key_length
            counter_sha = 1;
        end

    end

    new_audio = (double(audio)/255 - 0.5);

    %zapisanie pliku audio powstalego w wyniku przeksztalcen
    audiowrite("hidden_message.wav", new_audio, f)
    disp('Ukryto wiadomosc')
else
    disp('Slowo nie zmiesci sie w pliku audio!')
end
```

Skrypt odczytujący ukrytą wiadomość

decrypt.m | encrypt.m

```
% odczyt pliku audio z ukrytym obrazem
[start_audio, f] = audioread('hidden_message.wav');

% normalizacja audio
audio= uint8(255*(start_audio + 0.5));

%ilość znaków zaszyfrowanej wiadomości
chars = input('Podaj dlugosc wiadomosci: ');

%liczba bitów wiadomości
message_length = chars * 8;

%pobranie klucza od użytkownika
key = input('Podaj klucz(liczba): ', 's');

%funkcja skrótu SHA1
sha1hasher = System.Security.Cryptography.SHA1Managed;
hash = sha1hasher.ComputeHash(uint8(key));
sha1= uint8(hash);

%konwersja wartości dzisiętnych na binarne, tym razem dla klucza
binary_key = transpose(dec2bin(sha1, 8));
binary_key = binary_key(:);

%zapisanie dlugosci binarnej klucza
binary_key_length = length(binary_key);

%konwersja tablicy char na numeryczna dla klucza
binary_num_key = str2num(binary_key);

counter = 1;
counter_sha = 1;

%przejście po pliku audio
while counter <= message_length
    if binary_num_key(counter_sha) == 1
        extracted_bits(counter,1) = mod(double(audio(counter)), 2);
        counter = counter+1;
    end
    counter_sha = counter_sha+1;
    %w przypadku, gdy wiadomość dłuższa niż 160 bitów:
    if counter_sha > binary_key_length
        counter_sha = 1;
    end
end
disp(extracted_bits)

%potęgi liczby 2 do odzyskania znaków ascii z binarki
binValues = [ 128 64 32 16 8 4 2 1 ];

%odkodowanie wiadomości
binMatrix = reshape(extracted_bits, 8,(message_length/8));
textString = char(binValues*binMatrix);
disp(textString);
```

Działanie skryptów

Uruchomienie skryptu ukrywającego wiadomość

Wynikiem uruchomienia skryptu jest komunikat: „Ukryto wiadomość”, dodatkowo wcześniej wypisane zostają na ekran komunikaty proszące użytkownika o podanie wiadomości do ukrycia oraz klucza (parametru dla funkcji skrótu)

```
>> encrypt

len_audio =

      325758

Podaj wiadomosc do ukrycia: Ala ma kota!
Podaj klucz(liczba): 12
Ukryto wiadomosc
```

Po wykonaniu skryptu zapisany zostaje plik:

- hidden_message.wav – jest to plik audio formatu wav, który zawiera w sobie ukrytą wiadomość

Drugi skrypt przyjmuje na wejście następujące plik:

- hidden_message.wav – jest to plik audio formatu wav, który zawiera w sobie ukrytą wiadomość, plik ten powstał w wyniku działania pierwszego skryptu

Uruchomienie skryptu odczytującego ukrytą wiadomość

W wyniku uruchomienia drugiego skryptu odczytana i wypisana na ekran zostaje ukryta wiadomość. Oczywiście przed tym należy podać prawidłowy klucz oraz długość wiadomości.

```
>> decrypt
Podaj dlugosc wiadomosci: 12
Podaj klucz(liczba): 12
Ala ma kota!
```

Opis działania skryptów

Skrypt ukrywający wiadomość

W pierwszej kolejności wczytany zostaje plik audio, za pomocą funkcji audioread(), następuje jego normalizacja do wartości od 0 do 255. Zapisana zostaje długość pliku audio.

W kolejnych krokach zostaje pobrana od użytkownika wiadomość, która będzie ukrywana w pliku audio, wiadomość zamieniana jest na postać binarną. Następnie od użytkownika pobierany jest klucz, który stanowi wartość inicjującą dla funkcji skrótu SHA1, wartość funkcji skrótu zostaje zapisana w postaci binarnej i ma długość 160 bitów. W kolejnym kroku następuje przejście w pętli po całej długości wiadomości, bity wiadomości są ukrywane są w pliku audio, ale tylko wtedy, gdy licznik funkcji skrótu wskazuje na wartość ciągu binarnego skrótu równą 1. Na sam koniec zapisany zostaje plik audio z ukrytą wiadomością pod nazwą „hidden_message.wav”.

Skrypt odczytujący wiadomość

Skrypt odczytujący wiadomość odczytuje plik „hidden_message.wav”, dokonuje jego normalizacji, następnie pobiera od użytkownika długość wiadomości (w znakach) i oblicza długość wiadomości w

bitach. Następnie pobrany zostaje klucz (wartość inicjująca) dla funkcji skrótu, zostaje obliczona taka sama wartość jak podczas działania pierwszego skryptu. Wykorzystując posiadane informacje skrypt odczytuje ukrytą wiadomość w pliku audio i wypisuje ją na ekran.