

Навчально-науковий інститут інформаційних технологій
Харківський національний економічний університет
імені Семена Кузнеця

Звіт

З Виконання лабораторної роботи №7

за дисципліною: “Основи стеганографічного захисту інформації”

Виконав: студент кафедри
Кібербезпеки та інформаційних
технологій

4 курсу, спец. Кібербезпека,
групи 6.04.125.010.21.2

Бойко Вадим Віталійович

Перевірила:

Венгріна Олена Сергіївна

ХНЕУ ім. С. Кузнеця

2024

Мета роботи: Метою лабораторної роботи було вивчення та застосування стеганографічних методів для приховування даних в аудіофайлах. Було потрібно набуття практичних навичок роботи з аудіосигналами та їх модифікації за допомогою MathCAD.

Хід роботи:

1. Для виконання лабораторної роботи було встановлено MathCAD із підтримкою функцій роботи з аудіофайлами.

Використовувався WAV-файл із частотою дискретизації 44,1 кГц та розрядністю 16 біт. Було підготовлено текстове повідомлення "Hello", обсяг якого відповідав ємності файлу для методу LSB.

2. Реалізація методу LSB для приховування повідомлення

Читання аудіофайлу: Завантажено аудіофайл за допомогою функції READ_WAV. Дані сигналу представлено у вигляді числового масиву, а частоту дискретизації збережено у змінну sample_rate.

```
go
```

```
audio_data := READ_WAV("audio.wav")
```

```
sample_rate := audio_data.sample_rate
```

```
samples := audio_data.samples
```

Перетворення даних: Використано функцію DEC2BIN для переведення чисел аудіосигналу у 16-бітове двійкове представлення, що дозволило працювати з найменш значущими бітами.

```
binary_samples := DEC2BIN(samples, 16)
```

Впровадження повідомлення: Повідомлення "Hello" розбито на символи, конвертовано у двійковий код та впроваджено в LSB аудіосигналу через цикл.

```
message := "Hello"
```

```
binary_message := JOIN(DEC2BIN(ASC(message)), "")
```

```
FOR i IN 0..LENGTH(binary_message)-1 LOOP
```

```
    binary_samples[i] := REPLACE_LSB(binary_samples[i], binary_message[i])
```

```
END LOOP
```

Запис зміненого аудіофайлу: Зміннені дані повернуто до числового формату та збережено у новий файл audio_modified.wav.

```
modified_samples := BIN2DEC(binary_samples)
```

```
WRITE_WAV("audio_modified.wav", modified_samples, sample_rate)
```

3. Реалізація функції вилучення прихованого повідомлення

Читання зміненого файлу: Модифікований аудіофайл завантажено для аналізу.

```
modified_audio_data := READ_WAV("audio_modified.wav")
```

```
modified_samples := modified_audio_data.samples
```

Вилучення бітів повідомлення: За допомогою функції GET_LSB зібрано найменш значущі біти.

```
extracted_bits := ""
```

```
FOR i IN 0..LENGTH(binary_message)-1 LOOP
```

```
    extracted_bits := extracted_bits & GET_LSB(DEC2BIN(modified_samples[i], 16))
```

```
END LOOP
```

Відновлення повідомлення: Отримані біти перетворено на текст.

```
extracted_message := BIN2CHAR(SPLIT(extracted_bits, 8))
```

4. Аналіз якості звуку

Візуалізація звукових хвиль: Побудовано графіки сигналу до та після впровадження повідомлення.

```
PLOT(samples, title="Оригінальний аудіосигнал")
```

```
PLOT(modified_samples, title="Модифікований аудіосигнал")
```

Розрахунок SNR: Оцінено вплив приховування на якість сигналу.

```
noise := samples - modified_samples
```

```
SNR := 10 * LOG10(SUM(samples^2) / SUM(noise^2))
```

Висновки

Аналіз якості: Впровадження повідомлення методом LSB мінімально вплинуло на якість сигналу, що підтверджується візуалізацією та розрахунками SNR.

Стійкість повідомлення: Повідомлення успішно вилучено, але метод LSB вразливий до атак та модифікацій файлу.

Рекомендації: Для покращення стійкості прихованих даних слід використовувати шифрування або альтернативні методи стеганографії.

Відповіді на контрольні питання

Вплив методу LSB: При невеликому обсязі прихованого повідомлення зміни у якості сигналу непомітні.

Підвищення непомітності: Використання адаптивного LSB або частотних методів.

Особливості аудіостеганографії: Аудіосигнали мають іншу природу передачі даних, тому враховуються особливості слухового сприйняття.

Додаток:

// Читання аудіофайлу

audio_data := READ_WAV("audio.wav")

sample_rate := audio_data.sample_rate

samples := audio_data.samples

// Перетворення даних в двійкове представлення

binary_samples := DEC2BIN(samples, 16) // Перетворення чисел у 16-бітове
двійкове представлення

// Впровадження повідомлення

message := "Hello" // Текст повідомлення

binary_message := JOIN(DEC2BIN(ASC(message)), "") // Перетворення тексту у
двійковий вигляд

FOR i IN 0..LENGTH(binary_message)-1 LOOP

 binary_samples[i] := REPLACE_LSB(binary_samples[i], binary_message[i])

END LOOP

// Запис зміненого аудіофайлу

modified_samples := BIN2DEC(binary_samples) // Перетворення назад у десяткове
представлення

WRITE_WAV("audio_modified.wav", modified_samples, sample_rate)

// Читання зміненого аудіофайлу

modified_audio_data := READ_WAV("audio_modified.wav")

modified_samples := modified_audio_data.samples

```
// Вилучення бітів повідомлення
```

```
extracted_bits := ""
```

```
FOR i IN 0..LENGTH(binary_message)-1 LOOP
```

```
    extracted_bits := extracted_bits & GET_LSB(DEC2BIN(modified_samples[i], 16))
```

```
END LOOP
```

```
// Відновлення повідомлення
```

```
extracted_message := BIN2CHAR(SPLIT(extracted_bits, 8))
```

```
// Візуалізація звукових хвиль
```

```
PLOT(samples, title="Оригінальний аудіосигнал")
```

```
PLOT(modified_samples, title="Модифікований аудіосигнал")
```

```
// Розрахунок SNR
```

```
noise := samples - modified_samples
```

```
SNR := 10 * LOG10(SUM(samples^2) / SUM(noise^2))
```