Головін М.Б.,
кандидат фіз.-мат. наук, доцент
Головіна Н.А.,
кандидатка фіз.-мат. наук, доцентка
(Волинський національний університет імені Лесі Українки)

НАВЧАЛЬНИЙ ПРИКЛАД МАСКУВАННЯ ІНФОРМАЦІЇ В АКУСТИЧНОМУ СИГНАЛІ

Глобалізація інформаційної мережі робить інформацію легкою для передачі відкритими каналами в будь-яку точку Земної кулі. Однак, при цьому існує широкий спектр можливостей для перехоплення повідомлень та зламу шифрів. Тому існує необхідність передачі важливої інформації в прихованому вигляді. **Актуальними** є оригінальні способи приховування інформації, як у візуальних (Головін, 2020) так і в звукових контейнерах.

Метою цієї роботи є реалізація засобами мови Python простого способу приховування текстової інформації в звукових файлах для використання цієї задачі в навчальних цілях.

У цій роботі розглянуто реалізацію популярного алгоритму приховування «секретного» тексту в молодших найменш значущих бітах (LSB) байтів звукових файлів (музика, пісня, аудіокнига).

Нижче представлений код програми, яка вбудовує текст повідомлення в звук. Код достатньо коментований і поетапно пояснює процес.

```
import wave                          # під'єднання бібліотеки роботи зі звуковими файлами wave
sound = wave.open("music.wav", mode='rb')                  #  завантаження звукового файлу
file = open('secret_text.txt', 'r'); text=file.read(); print(text); file.close() #завантаження  тексту
sound_bytes = bytearray(list(sound.readframes(sound.getnframes()))) # звуку в масив байтів
text=text+'#'                                              # додати в кінець тексту знак кінця тексту '#'
bits_text=list(map(int, ''.join([bin(ord(i)).lstrip('0b').rjust(8,'0') for i in text]))) # текст в масив бітів
# Заміна LSB в кожному байті аудіоданих одиним бітом із текстового бітового масиву
for i, bit in enumerate(bits_text):          # цикл впровадження бітів тексту в звуковий файл
   if i < len(text)*8 :                      # зупинка впровадження бітів тексту в звуковий файл
      print(sound_bytes[i],sound_bytes[i]&254, bit, (sound_bytes[i]&254) | bit )
      sound_bytes[i]=(sound_bytes[i]&254) | bit   # ввод текстового біту в звуковий байт
sound_modified = bytes(sound_bytes)
with wave.open('sound_txt.wav', 'wb') as fd:      # цикл запису модифікованих байтів файл
   fd.setparams(sound.getparams()); fd.writeframes(sound_modified)
sound.close()
```

Маніпуляції з LSB бітами в представленій вище програмі досить прості і мають два етапи. На першому етапі між кожним байтом звукового контейнера і маскою 11111110 відбувається побітова логічна дія «AND». Вона скидає на 0 всі найменш значущі біти байтів звукового контейнеру. Далі між кожним модифікованим байтом контейнеру і бітовою маскою 0000000[0/1] виконується логічна операція «OR», де молодший біт маски із секретного повідомлення і може мати значення (0 або 1).

Для вилучення тексту зі звуку необхідно запустити наступний код

```
import wave
sound = wave.open("sound_txt.wav", mode='rb') # читати звуковий файлsound_bytes =
bytearray(list(sound.readframes(sound.getnframes()))) # аудіо в байтовий масив
extracted = [sound_bytes[i] & 1 for i in range(len(sound_bytes))] # вилучитиLSB з кожного байта
text = "".join(chr(int("".join(map(str,extracted[i:i+8])),2)) for i inrange(0,len(extracted),8)) # в текст
decoded = text.split("#")[0] # обрізати символ "#"
print("Успішно декодовано: "+decoded) #роздруківка вилученого текстуsound.close()
```

Висновок

- Реалізовано просту програму, що дозволяє приховувати текстову інформацію в звуковому файлі. Прослуховування заповненого і не заповненого звукового контейнера не виявляє відмінностей.

- Програма цікава для навчальних цілей завдяки лаконічності і прозорості програмного коду. Механізми приховування інформації в акустичному сигналі тут легко візуалізуються. Програма може бути використана, як аплікаційна на лекційному занятті при демонстрації її роботи вживу через проектор. Корисна ця задача і при її реалізації на лабораторному занятті.

- Аналіз звуку, як фізичного сигналу, на рівні окремих бітів, що відкривається при використанні програми, має також значну навчальну цінність. Маніпуляції зі звуком на такому низькому рівні дають уявлення про рівень шумів, амплітуду та частоту корисного фізичного сигналу, про ресурси приховування в сигналі сторонньої інформації.

ЛІТЕРАТУРА

1. Головін М.Б. Захист інформації стеганографічним способом мовою Python засобами графічної бібліотеки Pillow /М.Б. Головін, Н.А. Головіна, С.М. Яцюк, Ю.В. Сачук // Комп'ютерно-інтегровані технології: освіта, наука, виробництво" Луцьк, 2020. Випуск № 40 с.110-115 http://cit-journal.com.ua/index.php/cit/article/view/166