DOI: 10.36910/6775-2524-0560-2020-40-17
УДК: 004.72.056.52:003.27]:004.438
Головін Микола Борисович, канд. фіз.-мат. наук, доцент
https://orcid.org/0000-0003-4516-4677
Головіна Ніна Анатоліївна, канд. фіз.-мат. наук, доцент
Яцюк Світлана Миколаївна, канд. пед. наук, доцент
https://orcid.org/0000-0002-8369-6060
Сачук Юрій Володимирович, канд. фіз.-мат. наук
https://orcid.org/0000-0002-1317-1103
Східноєвропейський національний університет імені Лесі Українки, м. Луцьк

ЗАХИСТ ІНФОРМАЦІЇ СТЕГАНОГРАФІЧНИМ СПОСОБОМ МОВОЮ PYTHON ЗАСОБАМИ ГРАФІЧНОЇ БІБЛІОТЕКИ PILLOW

Головін М. Б., Головіна Н. А., Яцюк С. М., Сачук Ю.В. Захист інформації стеганографічним способом мовою Python засобами графічної бібліотеки Pillow. У роботі представлений стеганографічний метод приховування текстової інформації в електронній картинці. Метод реалізовано у вигляді простої програми мовою Python. У програмі використана графічна бібліотека Pillow. Впровадження окремих букв тексту здійснюється невеликими змінами базових кольорів. Букви тексту впроваджуються у випадково вибрані пікселі послідовно. Вилучення схованого тексту відбувається в процесі порівняння окремих пікселів заповненого і порожнього контейнера. Програма може бути використана як для навчальних, так і практичних цілей.

Ключові слова: мова Python, Pillow, стеганографія, захист інформації, приховування інформації, маскування інформації в графічному файлі.

Головин Н. Б., Головина Н. А., Яцюк С. М., Сачук Ю.В. Защита информации стеганографическим способом на языке Python средствами графической библиотеки Pillow. В работе представлен стеганографический метод сокрытия текстовой информации в электронной картинке. Метод реализован в виде простой программы на языке Python. В программе использована графическая библиотеки Pillow. Внедрение отдельных букв текста осуществляется небольшими изменениями базовых цветов. Буквы текста внедряются в случайно выбранные пиксели последовательно. Изъятие скрытого текста происходит в процессе сравнения отдельных пикселей заполненного и пустого контейнера. Программа может быть использована как для учебных, так и практических целей.

Ключевые слова: язык Python, Pillow, стеганография, защита информации, сокрытие информации, маскировка информации в графическом файле.

Holovin N.B., Holovina N.A., Yatsiuk S.M., Sachuk Yu.V. Protection of information steganographically in Python by means of the Pillow graphic library. The paper presents a steganographic method for hiding text information in an electronic picture. The method is implemented as a simple Python program. The program uses the graphic library Pillow. The introduction of individual letters of the text is carried out by small changes to the base colors. The letters of the text are embedded in randomly selected pixels sequentially. Hidden text is removed by comparing individual pixels of a filled and empty container. The program can be used for both practical and educational purposes.

Key words: Python language, Pillow, steganography, information protection, information hiding, masking information in a graphic file.

Постановка наукової проблеми. Сучасний інформаційний вибух, глобалізація інформаційної мережі, робить інформацію доступною і легкою для передачі відкритими каналами в будь-яку точку Земної кулі. Однак, існує широкий спектр причин, коли інформацію необхідно передавати у зашифрованому вигляді або в прихованому, а краще, в прихованому та ще й зашифрованому. Потреба в такому способі передачі інформації виникає на всіх щабелях соціального життя людства. Мова може йти про приватність особистого життя окремої людини, про бізнес-інтереси невеликої фірми або великої корпорації, про стратегічні міждержавні політичні та військові таємниці. Тому оригінальні способи шифрування та приховування інформації були і є надзвичайно ***актуальними***.

Процеси навчання у сфері інформатики, зокрема програмування, завжди мають базуватись на актуальних тематиках. Це підвищує мотивацію студентів до вивчення матеріалу. У процесі вивчення програмування існує досить багато важливих тем, які з успіхом можуть бути засвоєні на прикладах програм, в яких розглядаються різні прості способи шифрування та приховування інформації. До таких тем можна віднести: кодування текстової інформації, керування ходом проходження програм (цикли, розгалуження, функції), робота зі строками, списками, масивами, файлами.

Метою цієї роботи є розробка засобами графічної бібліотеки Pillow мови Python простого стеганографічного методу приховування текстової інформації в електронній картинці, який може бути використаний, як для навчальних цілей, так і для практичних.

Аналіз попередніх досліджень. Роботи [1, 2] є одними з перших системних видань в області стеганографії. Тут розглянуті відомі стеганографічні методи, спрямовані на приховування

конфіденційних даних у комп'ютерних файлах графічного, звукового та текстового форматів. Системно викладені оцінки пропускної здатності каналу прихованого обміну даними, а також проблеми стійкості і надійності стеганографічної системи по відношенню до атак. У роботах представлені результати існуючих інформаційно-теоретичних досліджень стосовно проблем приховування інформації. Більш сучасні огляди проблеми приховування інформації представлені, зокрема, в роботах [3, 4].

Виклад основного матеріалу й обґрунтування отриманих результатів.

Цікавим у підході представленого нижче приховування інформації є те, що тут використана не спеціальна бібліотека. Всяка оригінальність у підході до шифрування або приховування інформації надає додаткової складності до її зламу. У роботі також представлений діючий програмний код.

Ідея методу приховування тексту повідомлення в електронній картинці, що була реалізована в цій роботі, не є новою. Ця ідея полягає в тому, що код букви розбивається на три приблизно рівні частини і додається в рівних долях до трьох базових складових (RGB) кольору окремого пікселя, в якому реалізується приховування. Ці доданки не є великими тому, що код букв мінімізований в межах 1-32, а розділові та арифметичні знаки, цифри і букви «і», «ї», «є» в кодах від 32 до 68. Таким чином, для текстової частини повідомлення код букви, що має ховатись в базовому кольорі, не буде перевищувати числа 8. Це досить мало на фоні кодування базового кольору. Код базового кольору реалізується, як відомо, в байтовій комірці, тобто не перевищує 255. Саме три базових кольори формують реальний колір окремого пікселя (таблиця 1). Зрозуміло, що при кодуванні варто уникати радикальних кольорів таких, наприклад, як чорний, або білий. Це досягається відповідними фільтрами-розгалуженнями. У програмі, що представлена нижче, подібні фільтри є на кожний з базових кольорів (RGB). Код базового кольору разом з інформаційною надбавкою не повинен перевищувати 255.

Таблиця 1.

R	G	B	Color Name	R	G	B	Color Name
0	0	0	Black	80	208	255	Light Blue
255	255	255	White	0	32	255	Blue
224	224	224	Light Gray	96	255	128	Yellow-Green
128	128	128	Gray	0	192	0	Green
64	64	64	Dark Gray	255	224	32	Yellow
255	0	0	Red	255	160	16	Orange
255	96	208	Pink	160	128	96	Brown
160	32	255	Purple	255	208	160	Pale Pink

У цьому проекті окремі пікселі, що отримують інформаційне навантаження буквою повідомлення, вибираються випадковим чином. Ключом для розкодування повідомлення є порожній контейнер.

Цікавою бібліотекою Python для стеганографії є графічна бібліотека Pillow [5]. Зрозуміло, що ця бібліотека створювалась зовсім для іншого, а саме, для роботи з графікою в мові Python. Можливість отримати доступ до окремих пікселів зображення в цій бібліотеці робить її цікавою для стеганографії. Зокрема, бібліотека дає можливість змінювати базові кольори окремих пікселів. Це можна з успіхом використовувати для приховування інформації.

Бібліотека Pillow має широкі можливості і досить довгу еволюцію. На початку своєї історії бібліотека мала назву PIL від слів Python Imaging Library. Проєкт із модернізації цієї бібліотеки отримав назву Pillow. Pillow став достойною заміною оригінальної бібліотеки PIL. Він підтримує Python 3, чого PIL так і не досяг.

У контексті можливостей для стеганографії, крім вже згаданих, ця бібліотека має наступні цікаві резерви: завантаження та відображення зображення, отримання інформації про це зображення, обрізка зображень, зміна розміру зображення, повертання зображення, створення власного малюнка, використання фільтрів, конвертація форматів. Підтримується широкий список форматів файлів. Так, для файлів форматів BMP, EPS, GIF, JPEG, PDF, PNG, PNM, TIFF і деяких інших підтримується читання і запис. Для інших файлів, зокрема, форматів ICO, MPEG, PCX, PSD, WMF, тільки читання. Цікавим є і те, що форк Pillow був включений до деяких дистрибутивів Linux, включаючи Debian і Ubuntu.

Розглянемо реалізацію захисту інформації стеганографічним способом на мові Python засобами графічної бібліотеки Pillow.

Програма приховування повідомлення в картинці-контейнері. На першому етапі роботи програми необхідно під'єднати потрібні бібліотеки.

```
import random
from PIL import Image, ImageDraw
```

Далі завантажимо графічний контейнер для повідомлення та саме текстове секретне повідомлення.

```
image=Image.open("C:\prog\lis.png")   # завантаження графічного контейнера для повідомлення
file = open('C:\\prog\\text.txt', 'r')          # завантаження текстового секретного повідомлення
```

Реалізуємо мінімізацію кодів букв, цифр та інших знаків.

```
for i in range(len(text)):
  if ord(text[i])>=1071 and ord(text[i])<=1103: kod.append(ord(text[i])-1071); # а-1072я-1103
  if ord(text[i])>=33 and ord(text[i])<=64: kod.append(ord(text[i]));  #!"#$%&'()*+,-./0123456789
  if ord(text[i])==32:    kod.append(64);  # пробіл
  if ord(text[i])==1108:  kod.append(66);  # є
  if ord(text[i])==1110:  kod.append(67);  # і-1110
  if ord(text[i])==1111:  kod.append(68);  # ї-1111
```

Виведемо на екран зображення графічного порожнього файлу контейнера та підготуємо зображення для графічних перетворень.

```
image.show(); draw = ImageDraw.Draw(image)
```

Виміряємо ширину та висоту зображення в кількостях пікселів.

```
width = image.size[0]; height = image.size[1]; pix = image.load()
```

Згенеруємо випадкове число в проміжку від 10 до 30 з кроком 5.

```
rnd=random.randrange(10, 30, 5 ); k=0; n=0;
```

Проведемо сканування графічного файлу попіксельно та корекцію його деяких випадково вибраних пікселів з метою закладки секретного повідомлення в картинку-контейнер.

```
for i in range(width):
  for j in range(height):
    R = pix[i, j][0]; G = pix[i, j][1]; B = pix[i, j][2]; k=k+1;   # значення базових кольорів окремого пікселя
    if len(kod)>n and rnd<=k and R+kod[n]//3<255 and G+kod[n]//3<255 and B+kod[n]//3+kod[i]%3<255:
      R=R+kod[n]//3; G=G+kod[n]//3; B =B+kod[n]//3+kod[n]%3; draw.point((i, j), (R, G, B))
      n=n+1; rnd=random.randrange(10, 500, 5 ); k=0;
```

Видно, що частота закладки знаків випадкова та коливається у діапазоні від 10 до 500 пікселів з кроком 5. Кожний знак інформаційної закладки розбивається на три приблизно рівні порції. Цим досягається мінімізація порції корекції кольору. Відбувається перевірка на величину закладки. Розмір інформаційної закладки разом з числовим значенням базового кольору не повинен перевищувати 255. Частина умови, що забезпечує цю перевірку, виглядає наступним чином.

```
R+kod[n]//3<255 and G+kod[n]//3<255 and B+kod[n]//3+kod[i]%3<255
```

Відбувається також перевірка на довжину повідомлення та на частоту інформаційних закладок.

```
len(kod)>n and rnd<=k
```

Сам процес закладки інформаційного повідомлення виглядає наступним чином.

```
R=R+kod[n]//3; G=G+kod[n]//3; B =B+kod[n]//3+kod[n]%3; draw.point((i, j), (R, G, B))
```

Базові кольори R та G коректуються на величину результату ділення націло мінімізованого коду поточного знаку повідомлення R=R+kod[n]//3; G=G+kod[n]//3. Базовий колір B коректується на величину результату ділення націло та залишку від ділення B=B+kod[n]//3+kod[n]%3. Далі відбувається впровадження в картинку скоректованих кольорів поточного пікселя draw.point((i, j), (R, G, B)).

Дія циклів перебору пікселів відбувається до тих пір, поки не завершиться повне сканування картинки по висоті і ширині. Інформаційні закладки відбуваються до завершення кінця повідомлення. Через змінну «n» реалізується перебір букв у повідомленні в процесі їх приховування, через змінну «k» втілюється підрахунок випадково вибраних проміжків між закладками букв.

Далі відбувається вивід зображення картинки-контейнера разом з повідомленням на екран та закриття текстового файлу.

```
image.show();image.save("lis- inf-.png", " PNG ") # запис в файл картинки контейнера разом з повідомленням
file.close()
del draw
```

Візуальний перегляд зображень не виявляє відмінностей. Такі відмінності, очевидно, можна зафіксувати при проведенні аналізу тонів картинки математичними засобами. Якщо картинка-контейнер переобтяжена об'ємом інформаційного повідомлення, то, очевидно, можна виявити факт закладки повідомлення в картинку.

Програма вилучення повідомлення з картинки-контейнера. Вилучення інформаційного повідомлення з картинки-контейнера відбувається відніманням значень базових кольорів відповідних окремих пікселів заповненого і порожнього контейнерів.

Програма вилучення повідомлення так само, як і програма приховування, починається з під'єднання потрібних бібліотек та завантаження заповненого та порожнього контейнера.

Рис.1 Вигляд порожнього графічного картинки-контейнера

Рис.2 Вигляд графічного картинки-контейнера разом з повідомленням

```
import random
from PIL import Image, ImageDraw
image1 = Image.open("C:\prog\lis.png")       # завантаження картинки-контейнера
image2 = Image.open("C:\prog\lis-inf-.png") # завантаження картинки-контейнера разом з повідомленням
```

Далі відбувається очистка змінних, вивід на екран завантажених файлів, вимір розміру картинки за висотою та шириною.

```
dR=0; dG=0; dB=0; decod="";
image1.show(); image2.show();
draw = ImageDraw.Draw(image1)
width = image1.size[0];
height = image1.size[1];
```

Двома циклами, що перебирають пікселі по висоті та ширині картинки, скануємо одночасно дві картинки (заповнений та порожній контейнер). Якщо існує відмінність при відніманні значень кодів кольорів, то реалізуємо підрахунок величини цих відмінностей за базовими кольорами. Сума величин значень кодів відмінностей за базовими кольорами дає код знаку повідомлення.

```
n=0; pix1 = image1.load(); pix2 = image2.load()
for i in range(width):
  for j in range(height):
    dR = pix2[i, j][0]-pix1[i, j][0]; dG = pix2[i, j][1]-pix1[i, j][1]; dB = pix2[i, j][2]-pix1[i, j][2] ;
    if dB>0:
      n=n+1
      if dR+dG+dB>=1 and dR+dG+dB<=32: decod=decod+chr(dR+dG+dB+1071) #а-1072я-1103
      if dR+dG+dB>=33 and dR+dG+dB<=63:decod=decod+chr(dR+dG+dB) #!"#$%&'()*+,-./0123456789
      if dR+dG+dB==64:  decod=decod+chr(32);     # пробіл
      if dR+dG+dB==66:  decod=decod+chr(1108);  # є
      if dR+dG+dB==67:  decod=decod+chr(1110);  # і-1110
      if dR+dG+dB==68:  decod=decod+chr(1111);  # ї-1111
print(decod)
```

При специфіці приховування інформації, як це було описано вище, немає необхідності перевіряти відмінність всіх базових кольорів для кожного пікселя. Достатньо перевірити базовий колір B. Формування інформаційного додатку тут реалізується двома доданками B=B+kod[n]//3+kod[n]%3. Кольори R та G мають один складник, відповідно R=R+kod[n]//3, а G=G+kod[n]//3. У тих випадках, коли ділення націло kod[n]//3 дає 0, остача від ділення kod[n]%3 може мати ціле значення 1 або 2.

Через змінну «n», як і в попередній програмі реалізується перебір букв в повідомленні. Кінцевий фрагмент програми, що складається з послідовностей розгалужень, пов'язаний з перекодуванням повідомлення з формату мінімізованих кодів у стандартні коди, і далі в рядок знаків текстового повідомлення.

У випадку використання цієї задачі приховування інформації в навчальних цілях необхідно попередньо здійснити розв'язування кількох простіших задач з використанням бібліотеки Pillow мови Python. Мінімальний список цих задач включає перетворення кольорової картинки в картинку, де кольори замінені на градації сірого, утворення негативного зображення, контрастного чорно-білого зображення, зображення більш або менш яскравого ніж базове, перекодування файлу зображення з одного графічного формату в інший. Зрозуміло, що навчальні дії мають включати і вправи на завантаження і збереження картинок.

Висновки

- Засобами графічної бібліотеки Pillow мови Python реалізовано просту програму, що дозволяє приховувати текстову інформацію в електронній картинці. Програма може бути використана як для навчальних, так і для практичних цілей.
- Представлений підхід у приховуванні інформації використовує не спеціальну, а графічну бібліотеку Pillow. Всяка оригінальність у підході до шифрування або приховування інформації надає додаткової складності до її зламу.
- Візуальний огляд порожньої і заповненої картинки-контейнера не виявляє відмінностей. Розмір файлу контейнера і контейнера плюс повідомлення співпадає. Відмінність файлів можна зафіксувати тільки в результаті попіксельного порівняння відповідних базових кольорів.
- Факт наявності в графічному файлі текстової закладки, очевидно, можна зафіксувати при проведенні аналізу тонів картинки математичними засобами, якщо картинка-контейнер

переобтяжена розміром інформаційного повідомлення. Однак, витягнути повідомлення, маючи тільки картинку-контейнер з повідомленням, неможливо.

Список бібліографічного опису

1. Грибунин В. Г., Оков И. Н., Туринцев И. В. Цифровая стеганография. – М. : Солон-Пресс, 2002. – 272 с.
2. Конахович Г. Ф., Пузыренко А. Ю. Компьютерная стеганография. Теория и практика. – К. : МК-Пресс, 2006. – 288 с.
3. Конахович Г. Ф., Прогонов Д. О., Пузиренко О. Ю. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних [підручник]. – К. : «Центр навчальної літератури», 2018. – 558 с.
4. Рябко Б. Я., Фионов А. Н. Основы современной криптографии и стеганографии. – 2-е изд. – М. : Горячая линия – Телеком, 2013. – 232 с.
5. https://pillow.readthedocs.io/en/stable/

References

1. Gribunin V., Okov I., Turintsev I. Digital steganography. – M. : Solon-Press, 2002. – 272 p.
2. Konakhovich G., Puzyrenko A. Y. Computer steganography. Theory and practice. - K. : MK-Пресс, 2006. – 288 c.
3. Konakhovych G., Progonov D., Puzyrenko O. Computer steganographic processing and analysis of multimedia data [textbook]. – Kyiv : Center for Educational Literature, 2018. – 558 p.
4. Ryabko B., Fionov A. Fundamentals of modern cryptography and steganography. - 2nd ed. – Moscow : Hotline - Telecom, 2013. - 232 p.
5. https://pillow.readthedocs.io/en/stable/