

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ**

Основы стеганографии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

«Основы текстовой стеганографии»

Работу выполнил студент  
Нгуен Тхай Хынг, N3352, ФБИТ



Работу проверил: ассистент ФБИТ,  
Университет ИТМО,  
Давыдов Вадим Валерьевич

Санкт-Петербург

2020

## Цель работы

Целью данной лабораторной работы :

- Применение текстовых стеганографических методов для сокрытия
- Извлечение сообщения из стегоконтейнера
- Анализ исходного текста и стегоконтейнера.

## Теория

**Стеганография** - это наука о скрытой передаче информации путём сохранения в тайне самого факта передачи. Этот термин ввел в 1499 году Иоганн Тритемий в своем трактате «Стеганография» (*Steganographia*), зашифрованном под магическую книгу.

В отличие от криптографии, которая скрывает содержимое секретного сообщения, стеганография скрывает сам факт его существования. Как правило, сообщение будет выглядеть как что-

либо иное, например, как изображение, статья, список покупок, письмо или sudoku. Стеганографию обычно используют совместно с методами криптографии, таким образом, дополняя её.

Преимущество стеганографии над чистой криптографией состоит в том, что сообщения не привлекают к себе внимания. Сообщения, факт шифрования которых не скрыт, вызывают подозрение и могут быть сами по себе уличающими в тех странах, в которых запрещена криптография. Таким образом, криптография защищает содержание сообщения, а стеганография защищает сам факт наличия каких-либо скрытых посланий.

### Классификация стеганографии

Выделилось несколько направлений стеганографии:

- Классическая стеганография
- Компьютерная стеганография
- Цифровая стеганография

### Классическая стеганография

Одним из наиболее распространенных методов **классической стеганографии** является использование симпатических (невидимых) чернил. Текст, записанный такими чернилами, проявляется только при определенных условиях (нагрев, освещение, химический проявитель и т. д.)

Существуют также чернила с химически нестабильным пигментом. Написанное этими чернилами выглядит как написанное обычной ручкой, но через определенное время нестабильный пигмент разлагается, и от текста не остается и следа. Хотя при использовании обычной шариковой ручки текст можно восстановить по деформации бумаги, этот недостаток можно устранить с помощью мягкого пишущего узла, наподобие фломастера.

**Компьютерная стеганография** - направление классической стеганографии, основанное на особенностях компьютерной платформы. Примеры - стеганографическая файловая система

ма StegFS для Linux, скрывание данных в неиспользуемых областях форматов файлов, подмена символов в названиях файлов, текстовая стеганография и т. Д

**Цифровая стеганография** - направление классической стеганографии, основанное на сокрытии или внедрении дополнительной информации в цифровые объекты, вызывая при этом некоторые искажения этих объектов. Но, как правило, данные объекты являются мультимедиа-объектами (изображения, видео, аудио, текстуры 3D-объектов) и внесение искажений, которые находятся ниже порога чувствительности среднего человеческого человека, не приводит к заметным изменениям этих объектов. Кроме того, в оцифрованных объектах, изначально имеющих аналоговую природу, всегда присутствует шум квантования; далее, при воспроизведении этих объектов появляется дополнительный аналоговый шум и нелинейные искажения аппаратуры, все это способствует большей незаметности сокрытой информации.

### **Другие стеганографические методы:**

Во время Второй мировой войны активно использовались микроточки - микроскопические фотоснимки, вклеиваемые в текст писем.

Также существует ряд альтернативных методов сокрытия информации:

- запись на боковой стороне колоды карт, расположенных в условленном порядке;
- запись внутри варёного яйца;
- «жаргонные шифры», где слова имеют другое обусловленное значение;
- трафареты, которые, будучи положенными на текст, оставляют видимыми только значащие буквы;
- узелки на нитках и т. д.

В настоящее время под стеганографией чаще всего понимают сокрытие информации в текстовых, графических либо аудиофайлах путём использования специального программного обеспечения.

<https://dic.academic.ru/dic.nsf/ruwiki/30097>

## **Практика**

Для того, чтобы реализовать вышеописанные методы, мною было написана программа “stegano.py”, листинг которых можно найти в приложении к данному отчету, которое приведено после списка литературы.

Для написания программы использовался язык Python 3 с помощью Visual Studio Code, так как мне нравится работа со строками в данном языке.

Программа требует выбрать метод для сокрытия информации, у нас есть три метода:

1. Метод замены символов, этот метод заменяет буквы “o” и “p” на английском языке на символы “o” и “p” на русском языке, с условием, что символ “o” (RU) соответствует биту 0, а символ “p” (RU) соответствует биту 1.

2. Метод с использованием дополнительных хвостовых пробелов. Этот метод добавляет 1 пробел сразу до символа “\n”, соответствующей биту 1, добавляет 2 пробела сразу до символа “\n”, соответствующего биту 0.

3. Метод с добавлением служебных символов. Этот метод добавляет 1 нулевой символ “\0” сразу до символа “\n”, соответствующего биту 0, добавляет 2 нулевых символа сразу до символа “\n”, соответствующего биту 0.

После ввода метода программа запрашивает текст, который мы хотим скрыть.

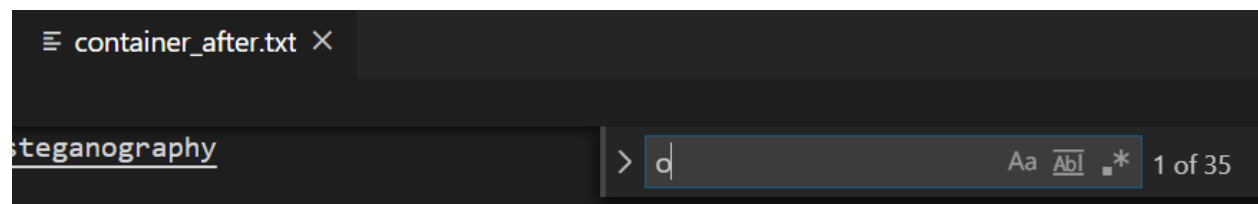
Эта программа читает файл “container\_before.txt” потом выводит файл “container\_after.txt”, содержащий скрытую информацию.

### Метод 1: Метод замены символов

Запустить программу

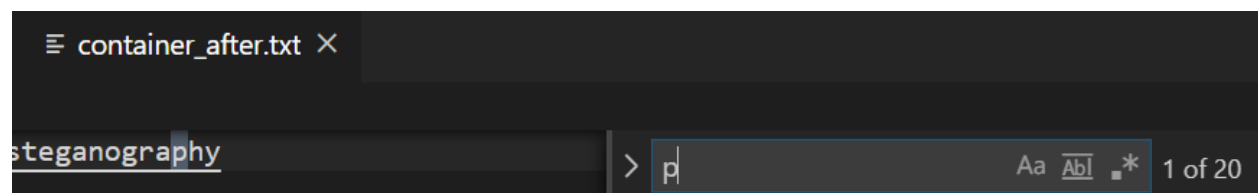
```
PS E:\Documents\ITMO\stegano\lab_1> py .\stegano.py
1. Replace 'o' and 'p' in English with 'o' and 'p' in Russian
2. Add 1 or 2 spaces before '\n'
3. Add 1 or 2 null characters before '\n'
Choose one method you want to use: 1
Current working directory: E:\Documents\ITMO\stegano\lab_1
Enter your secret message: коронавирус
Hiding secret message into the container_before file...
Extract secret message from the container_after file...
коронавирус
PS E:\Documents\ITMO\stegano\lab_1> █
```

В файле “container\_after.txt” содержится 35 символов “o” и 20 символов “p”



container\_after.txt X

steganography > o Aa AbI .\* 1 of 35



container\_after.txt X

steganography > p Aa AbI .\* 1 of 20

## Метод 2: Метод с использованием дополнительных хвостовых пробелов

Запустить программу

```
PS E:\Documents\ITMO\stegano\lab_1> py .\stegano.py
1. Replace 'o' and 'p' in English with 'o' and 'p' in Russian
2. Add 1 or 2 spaces before '\n'
3. Add 1 or 2 null characters before '\n'
Choose one method you want to use: 2
Current working directory: E:\Documents\ITMO\stegano\lab_1
Enter your secret message: мыпобедим
Hiding secret message into the container_before file...
Extract secret message from the container_after file...
мыпобедим
PS E:\Documents\ITMO\stegano\lab_1>
```

Открываем файл “container\_after.txt” для просмотра. Мы увидим на конец каждой строки есть один или два пробела.

```
e: > Documents > ITMO > stegano > lab_1 >  container_after.txt


1  https://searchsecurity.techtarget.com/definition/steganography
2  .
3  1. Steganography
4  |
5  Steganography is the technique of hiding secret data within an ordinary,
```

### Метод 3: Метод с добавлением служебных символов

Запустить программу

```
PS E:\Documents\ITMO\stegano\lab_1> py .\stegano.py
1. Replace 'o' and 'p' in English with 'o' and 'p' in Russian
2. Add 1 or 2 spaces before '\n'
3. Add 1 or 2 null characters before '\n'
Choose one method you want to use: 3
Current working directory: E:\Documents\ITMO\stegano\lab_1
Enter your secret message: ястудент
Hiding secret message into the container_before file...
Extract secret message from the container_after file...
ястудент
PS E:\Documents\ITMO\stegano\lab_1> █
```

Открываем файл “container\_after.txt” для просмотра. Мы увидим на конец каждой строки есть один или два пробела. На самом деле это нулевой символ “\0”, отображаемый как пробел

 container\_after.txt - Notepad

File Edit Format View Help

<https://searchsecurity.techtarget.com/definition/steganography>

#### 1. Steganography

Steganography is the technique of hiding secret data within an ordinary file. The use of steganography can be combined with encryption as an extra security measure. The word steganography is derived from the greek words steganos (meaning hidden) and grapho (meaning writing). Steganography can be used to conceal almost any type of digital content. The content to be concealed through steganography -- called hidden text -- can be encrypted or not. If not encrypted, the hidden text is commonly processed in some way to make it look like random data. Steganography is practiced by those wishing to convey a secret message. While there are many legitimate uses for steganography, malware developers have also used it. Forms of steganography have been used for centuries and include almost anything that can be hidden. For example, using invisible ink to hide secret messages in otherwise ordinary documents.

#### 2. Steganography techniques

In modern digital steganography, data is first encrypted or obfuscated before being hidden.

Мы видим, что программа работает корректно, обеспечивая встраивание стего в текст. Теперь необходимо проанализировать результат файл до и после встраивания. Для того, я

написал программа “analyzer.py”, запускаем программу для анализа файлов по каждому методу, получаем результаты:

Результат первого метода:

```
PS E:\Documents\ITMO\stegano\lab_1> py .\analyzer.py
Length before: 6202 bytes
Length after: 6205 bytes
Difference: 0.05 %
```

Результат второго метода:

```
PS E:\Documents\ITMO\stegano\lab_1> py .\analyzer.py
Length before: 6202 bytes
Length after: 6215 bytes
Difference: 0.21 %
```

Результат третьего метода:

```
PS E:\Documents\ITMO\stegano\lab_1> py .\analyzer.py
Length before: 6202 bytes
Length after: 6210 bytes
Difference: 0.13 %
```

Мы видим, что первый метод имеет самое низкое изменение данных, поэтому этот метод очень сложно обнаружить, с помощью первого метода мы также скрываем больше данных, потому что в английском тексте есть много букв “o” и “p”.

Второй и третий методы менее эффективны, мы скрываем данные в тексте добавлением символов до символа “\n”, поэтому скрываем меньше данных, потому что в тексте очень мало символов “\n”. Эти методы также увеличивают изменение текстового содержимого, потому что они «добавляют» вместо «заменяют» символы, поэтому этот метод НЕ очень сложно обнаружить.

### **Вывод**

После выполнения методов в первой лаборатории, и в то же время я исследовал и оценил надежность и безопасность методов, я сделал следующие выводы:

Метод замены символов одного типа весьма полезен и его трудно обнаружить обычным текстовым редакторам. Метод добавления пробела перед переносом строки легко обнаружить, приведенное выше изображение является примером того, когда его можно узнать в Word. Метод добавления специальных символов увеличивает объем текста, поэтому его легко обнаружить.

Поэтому, на мой взгляд, самый эффективный метод - это первый, который не меняет размер файла и его трудно обнаружить.

### Список Литературы

1. What is steganography? - <https://searchsecurity.techtarget.com/definition/steganography>
2. Wikipedia - <https://en.wikipedia.org/wiki/Steganography>
3. AH4S: an algorithm of text in text steganography using the structure of omega network - <https://onlinelibrary.wiley.com/doi/full/10.1002/sec.1752>

### Приложение

Программа для всех методов: stegano.py

```
import os
import codecs # For handling unicode characters when reading file containing unicode
characters to list

# Encode text (our secret message in clear text) to binary string, using the defined
dictionary in the main program
def encode(secret_message):
    binary_secret_message = ''
    for character in secret_message:
        # Method dict.get(key) returns a value for a given key in the dictionary 'dic
t'
        binary_secret_message += dictionary.get(character)
    return binary_secret_message

# Decode binary string to text (our secret message in clear text), using the defined
dictionary in the main program
def decode(binary_secret_message):
    secret_message = ''
    for i in range(0, len(binary_secret_message), 5):
        for key, value in dictionary.items():
            if value == binary_secret_message[i: i+5]:
                secret_message += key
    return secret_message

# Hide secret message into the container file, output to another file
def hide_secret_message(container_file_in, secret_message, container_file_out):
    # Read all characters from file 'container_file_in' into a list 'character_list'
    character_list = [char for char in open(container_file_in).read()]
    binary_secret_message = encode(secret_message)
    number_of_replacement = 0
    j = 0
    for i in range(0, len(character_list), 1):
```



```

        if character_list[i] == pair[0][0] and binary_secret_message[j] == '0':
            character_list[i] = pair[0][1]
            number_of_replacement += 1
            j += 1
        if character_list[i] == pair[1][0] and binary_secret_message[j] == '1':
            character_list[i] = pair[1][1]
            number_of_replacement += 1
            j += 1
        # If number_of_replacement == len(secret_message_in_binary), then break the loop
        if number_of_replacement == len(binary_secret_message):
            break
    # Write an updated character list elements after hiding secret message
    f = open(container_file_out, "wb")
    for character in character_list:
        f.write(character.encode('utf8'))
    f.close()

# Unhide secret message from a container file, return secret message in clear text
def unhide_secret_message(container_file, method):
    # Read all characters from the container file into a list
    character_list = []
    with codecs.open(container_file, encoding='utf-8') as f:
        for line in f:
            for character in line:
                character_list.append(character)
    binary_secret_message = ''
    if method == '1':
        for i in range(0, len(character_list), 1):
            if character_list[i] == pair[0][1]:
                binary_secret_message += '0'
            elif character_list[i] == pair[1][1]:
                binary_secret_message += '1'
    elif method == '2' or method == '3':
        for i in range(0, len(character_list), 1):
            if character_list[i] == '\n':
                if (character_list[i - 2] + character_list[i - 1]).count(pattern) == 1:
                    binary_secret_message += '0'
                elif (character_list[i - 2] + character_list[i - 1]).count(pattern) == 2:
                    binary_secret_message += '1'
        return decode(binary_secret_message)

# The main program

```

```

if __name__ == "__main__":
    # Create a dictionary for storing {character:binary} pairs
    dictionary = {}
    for i in range(1072, 1104, 1):
        # Add a new key:value pair
        dictionary.update( {chr(i) : bin(i-1072)[2:].zfill(5)} )
        # Method dict.update({key: value}) appends a key-
value pair to the dictionary 'dict'
        # Method chr(int) returns a character from an integer 'int'
        # Method bin(int)[2:] return the binary equivalent string of a given integer
'int'
        # Method binary_string.zfill(int) pads binary_string on the left with zeros t
o fill width 'int'

    print("1. Replace 'o' and 'p' in English with 'o' and 'p' in Russian")
    print("2. Add 1 or 2 spaces before '\\n'")
    print("3. Add 1 or 2 null characters before '\\n'")
    method = input("Choose one method you want to use: ")
    # The pair dictionary containing 2 tuples as its values
    pair = {}
    if method == '1':
        pair[0] = ('o', chr(1086)) # Function chr(1086) returns character 'o' in Rus
sian, bit 0
        pair[1] = ('p', chr(1088)) # Function chr(1088) returns character 'p' in Rus
sian, bit 1
    elif method == '2':
        pair[0] = ('\\n', ' \\n') # Add 1 space right before \\n character, bit 0
        pair[1] = ('\\n', '  \\n') # Add 2 spaces right before \\n character, bit 1
        # The following variable is used for function unhide_secret_message()
        pattern = ' '
    elif method == '3':
        pair[0] = ('\\n', '\\x00\\n') # Add 1 NULL character right before \\n charac
ter, bit 0
        pair[1] = ('\\n', '\\x00\\x00\\n') # Add 2 NULL characters right before \\n chara
cter, bit 1
        # The following variable is used for function unhide_secret_message()
        pattern = '\\x00'

    print("Current working directory:", os.getcwd())
    secret_message = input("Enter your secret message: ")

    print("Hiding secret message into the container_before file...")
    hide_secret_message('container_before.txt', secret_message, 'container_after.txt'
)

```

```
print("Extract secret message from the container_after file...")
print(unhide_secret_message('container_after.txt', method))
```

Программа для анализа: analyzer.py

```
# Function to count how many bytes in a file
def count_bytes(file):
    count = 0
    with open(file, 'rb') as f:
        while True:
            byte = f.read(1)
            if not byte:
                break
            count += 1
    return count

# The main program
if __name__ == "__main__":
    length_before = count_bytes('container_before.txt')
    length_after = count_bytes('container_after.txt')
    print("Length before: {} bytes".format(length_before))
    print("Length after: {} bytes".format(length_after))
    print("Difference: {:.2f} %".format((abs(length_after - length_before) / length_
before) * 100))
```