

Лабораторная работа №7

Элементы криптографии. Однократное гаммирование

Го Чаопэн

21 октября 2023

Российский университет дружбы народов, Москва, Россия

Информация

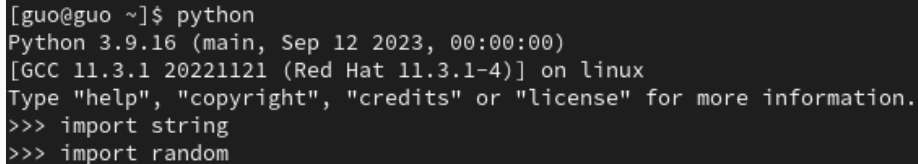
- Го Чаопэн
- студент НФИбд-02-20
- Российский университет дружбы народов
- 1032194919@pfur.ru
- <https://github.com/LIONUCKY>

Вводная часть

- Атрибуты файлов
- Дистрибутив Rocky
- Дискреционное разграничение доступа

Освоить на практике применение режима однократного гаммирования

Импортируем необходимые модули (@fig:001).

A terminal window with a dark background and light gray text. The text shows the execution of the 'python' command, the Python version (3.9.16), the GCC version (11.3.1), and the operating system (linux). It then shows the execution of 'import string' and 'import random' in a Python shell.

```
[guo@guo ~]$ python
Python 3.9.16 (main, Sep 12 2023, 00:00:00)
[GCC 11.3.1 20221121 (Red Hat 11.3.1-4)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import string
>>> import random
```

Рис. 1: Импорт модулей

Создадим функции для преобразования данных в шестнадцатеричный формат, генерации ключа и кодирования, декодирования данных (@fig:002).

```
>>> def to_hex(text):  
...     return " ".join(hex(ord(i))[2:] for i in text)  
...  
>>> def generate_key(size):  
...     key = "".join(random.choice(string.ascii_letters + string.digits) for _ in range(size))  
...     return key  
...  
>>> def encoder(text, key):  
...     return "".join(chr(a^b) for a, b in zip(text, key))  
...  
...
```

Рис. 2: Функции

Закодируем и декодируем строку "С Новым годом, друзья!" (@fig:003).

```
...
>>> mess = "С Новым годом, друзья!"
>>> key = generate_key(len(mess))
>>> hex_key = to_hex(key)
>>>
>>> enc_text = encoder([ord(i) for i in mess], [ord(i) for i in key])
>>> hex_text = to_hex(enc_text)
>>> decr_text = encoder([ord(i) for i in enc_text], [ord(i) for i in key])
>>>
>>> print("Ключ: ", hex_key)
Ключ:  39 54 41 71 50 64 6e 72 72 30 69 38 79 4e 62 4c 62 4d 37 44 70 6f
>>> print("Зашифрованное сообщение: ", hex_text)
Зашифрованное сообщение:  418 74 45c 44f 462 42f 452 52 441 40e 45d 406 445 62 42 478 422 40e 400 408 43f 4e
>>> print("Расшифрованный текст: ", decr_text)
Расшифрованный текст:  С Новым годом, друзья!
>>>
```

Рис. 3: Кодирование и декодирование строки

Получим ключ, с помощью которого получим сообщения “С Новым годом, семья”, “С Новым годом, учителя” вместо “С Новым годом, друзья!” при декодировании. Воспользуемся симметричностью кодирования(@fig:004).

```
>>> new_mess = "С Новым годом, семья!"
>>>
>>> key = encoder([ord(i) for i in enc_text], [ord(i) for i in new_mess])
>>> print("Ключ: ", to_hex(key))
Ключ:  39 54 41 71 50 64 6e 72 72 30 69 38 79 4e 62 39 17 32 4c 47 41e
>>>
>>> new_mess = "С Новым годом, учителя!"
>>>
>>> key = encoder([ord(i) for i in enc_text], [ord(i) for i in new_mess])
>>> print("Ключ: ", to_hex(key))
Ключ:  39 54 41 71 50 64 6e 72 72 30 69 38 79 4e 62 3b 65 36 42 3d 4 401
>>> █
```

В рамках данной лабораторной работы было освоено на практике применение режима однократного гаммирования