

Лабораторная работа № 7

Элементы криптографии. Однократное гаммирование

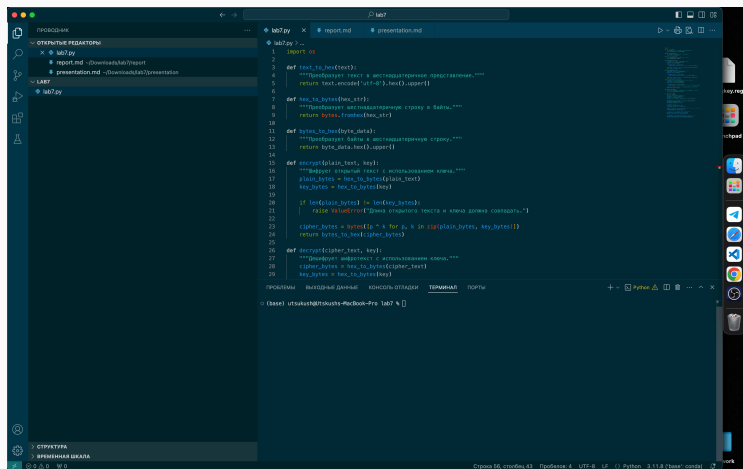
Морозов М. Е.

12 октября 2024

Российский университет дружбы народов, Москва, Россия

Выполнение лабораторной работы

Напишем функции на python и зададим переменные. (рис. 1)



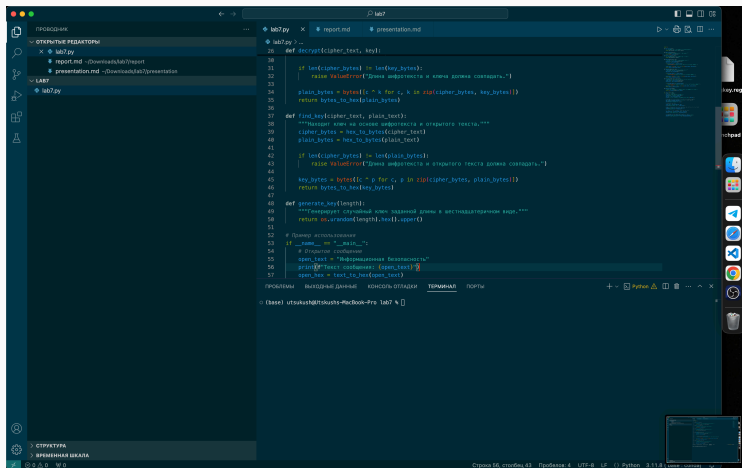
The screenshot shows a code editor with a dark theme. The left sidebar displays a file explorer with a project named 'lab7' containing files 'report.md' and 'presentation.md'. The main editor area shows a Python file 'lab7.py' with the following code:

```
1 import os
2
3 def text_to_hex(text):
4     """Преобразует текст в шестнадцатеричное представление."""
5     return text.encode('utf-8').hex().upper()
6
7 def hex_to_bytes(hex_str):
8     """Преобразует шестнадцатеричную строку в байты."""
9     return bytes.fromhex(hex_str)
10
11 def bytes_to_hex(byte_data):
12     """Преобразует байты в шестнадцатеричную строку."""
13     return byte_data.hex().upper()
14
15 def encrypt(plain_text, key):
16     """Шифрует открытый текст с использованием ключа."""
17     plain_bytes = hex_to_bytes(plain_text)
18     key_bytes = hex_to_bytes(key)
19
20     if len(plain_bytes) != len(key_bytes):
21         raise ValueError("Длина открытого текста и ключа должна совпадать.")
22
23     cipher_bytes = bytes([p ^ k for p, k in zip(plain_bytes, key_bytes)])
24     return bytes_to_hex(cipher_bytes)
25
26 def decrypt(cipher_text, key):
27     """Дешифрует шифротекст с использованием ключа."""
28     cipher_bytes = hex_to_bytes(cipher_text)
29     key_bytes = hex_to_bytes(key)
```

The bottom of the editor shows a terminal window with the command: `(base) utskush@itskushs-MacBook-Pro: lab7 %`. The status bar at the bottom indicates 'Страна 00, строка 43 Пробелы 4 UTF-8 LF Python 3.11.8 (tags/cpython)'.

Figure 1: Код. Часть 1

Напишем функции на python и зададим переменные. (рис. 2)



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'lab7' with files 'report.md' and 'presentation.md'. The code editor shows the following Python code:

```
def decrypt(cipher_text, key):
    if len(cipher_bytes) != len(key_bytes):
        raise ValueError("Длина шифротекста и ключа должны совпадать.")
    plain_bytes = bytes([c ^ k for c, k in zip(cipher_bytes, key_bytes)])
    return bytes_to_hex(plain_bytes)

def find_key(cipher_text, plain_text):
    """Находит ключ на основе шифротекста и открытого текста."""
    cipher_bytes = hex_to_bytes(cipher_text)
    plain_bytes = hex_to_bytes(plain_text)
    if len(cipher_bytes) != len(plain_bytes):
        raise ValueError("Длина шифротекста и открытого текста должны совпадать.")
    key_bytes = bytes([c ^ p for c, p in zip(cipher_bytes, plain_bytes)])
    return bytes_to_hex(key_bytes)

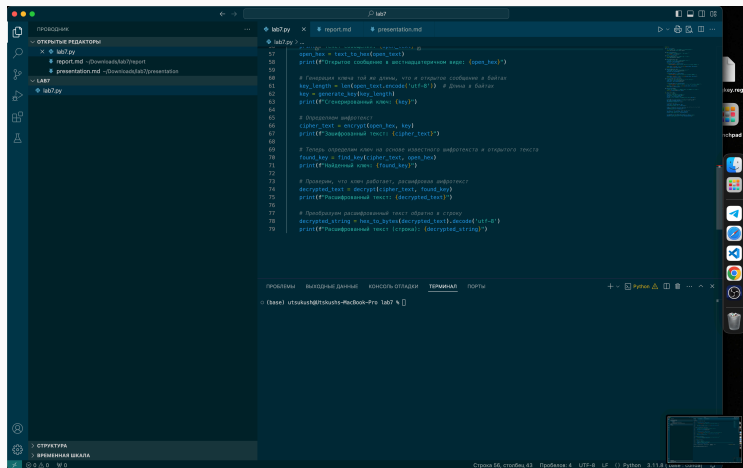
def generate_key(length):
    """Генерирует случайный ключ заданной длины в шестнадцатеричном виде."""
    return hex_to_bytes(random(length).hex()).upper()

# Пример использования
if __name__ == "__main__":
    # Открытие сообщения
    open_text = "Информационная безопасность"
    print(f"Текст сообщения: {open_text}")
    open_hex = text_to_hex(open_text)
```

The code editor also shows a terminal window at the bottom with the command 'python lab7.py' and the output 'Text message: Информационная безопасность'.

Figure 2: Код. Часть 2

Напишем функции на python и зададим переменные. (рис. 3)



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'lab7' with files 'report.md' and 'presentation.md'. The code editor shows a Python script named 'lab7.py' with the following code:

```
57 def main():  
58     open_hex = text_to_hex(open_text)  
59     print(f"Открытое сообщение в шестнадцатеричном виде: {open_hex}")  
60  
61     # Генерация ключа той же длины, что и открытое сообщение в байтах  
62     key_length = len(open_text.encode('utf-8')) # Длина в байтах  
63     key = generate_key(key_length)  
64     print(f"Сгенерированный ключ: {key}")  
65  
66     # Шифруем открытый текст  
67     cipher_text = encrypt(open_hex, key)  
68     print(f"Зашифрованный текст: {cipher_text}")  
69  
70     # Теперь определим ключ на основе известного шифротекста и открытого текста  
71     found_key = find_key(cipher_text, open_hex)  
72     print(f"Найденный ключ: {found_key}")  
73  
74     # Проверим, что ключ работает, расшифруем шифротекст  
75     decrypted_text = decrypt(cipher_text, found_key)  
76     print(f"Расшифрованный текст: {decrypted_text}")  
77  
78     # Преобразуем расшифрованный текст обратно в строку  
79     decrypted_string = hex_to_string(decrypted_text).decode('utf-8')  
80     print(f"Расшифрованный текст (строка): {decrypted_string}")
```

The terminal at the bottom shows the command 'python lab7.py' being executed.

Figure 3: Код. Часть 3

Применим написанные функции для создания ключа, шифрования текста и дешифрования. (рис. 4)

```
• (base) utsukush@Utskushs-MacBook-Pro lab7 % /opt/anaconda3/bin/python /Users/utsukush/Desktop/lab7/lab7.py
Текст сообщения: С Новым Годом, друзья!
Открытое сообщение в шестнадцатеричном виде: D0A120D09DD0BED0B2D18BD0BC20D093D0BED0B4D0BED0BC2C20D0B4D180D183D0B7D18CD18F21
Сгенерированный ключ: 200D582B4EA35C845D2A402C0BC4EB1F7406F2E8FB401E213A5C64315FC806FF5E38421D1894F7
Зашифрованный текст: F0AC78FBD373E254EFFBCBFCB7E43B8CA4B8225C2BFCE9D167CB4858E48D77C8E8F9391CA1BD6
Найденный ключ: 200D582B4EA35C845D2A402C0BC4EB1F7406F2E8FB401E213A5C64315FC806FF5E38421D1894F7
Расшифрованный текст: D0A120D09DD0BED0B2D18BD0BC20D093D0BED0B4D0BED0BC2C20D0B4D180D183D0B7D18CD18F21
Расшифрованный текст (строка): С Новым Годом, друзья!
○ (base) utsukush@Utskushs-MacBook-Pro lab7 %
```

Figure 4: Получили “С новым годом, друзья!”

Освоили на практике применение режима однократного гаммирования.

1. Гаммирование [Электронный ресурс]. Wikimedia Foundation, Inc., 2024. URL: https://en.wikipedia.org/wiki/XOR_cipher.