

Отчёт по лабораторной работе №7

Дисциплина: Основы информационной безопасности

Барсегян Вардан Левонович НПИбд-01-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Контрольные вопросы	10
4	Выводы	12

Список иллюстраций

2.1	Функция <code>encrypt()</code>	7
2.2	<code>decrypt()</code> с тем же ключом	8
2.3	<code>decrypt()</code> со случайным ключом	8

Список таблиц

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Выполнение лабораторной работы

1. Создаю функцию encrypt(), которая будет шифровать заданный текст с помощью гаммирования. Также можно подать на вход определенный ключ шифрования. Если ключа нет, то он генерируется случайно. Сначала исходный текст и ключ шифрования преобразуются в 16-ную СС, затем, применяется операция XOR для каждого элемента ключа и текста. Полученный шифротекст декодируется из 16-ной СС и получается набор из символов.

```
def encrypt(text: str, key: list = None):  
    """  
    Выводит шифротекст для заданного текста.  
    Если ключа нет, то генерируется случайный ключ  
    """  
  
    text_16 = [char.encode(encoding='cp1251').hex().upper() for char in text]  
    if not key:  
        key = generate_key(length=len(text))  
  
    print(f"Ключ шифрования:", ' '.join(str(s) for s in key))  
    print(f"Исходный текст:", ' '.join(text_16))  
    encrypted_text = []  
    for i in range(len(text)):
```

```

xor_char = int(text_16[i], 16) ^ int(key[i], 16)
encrypted_text.append(int2hex(xor_char))

encrypted_text = validate(encrypted_text)
ciphertext = bytes.fromhex("".join(encrypted_text)).decode('cp1251')
print(f'Шифротекст: {ciphertext}\n\n')

return {
    'key': key,
    'ciphertext': ciphertext
}

```

2. Результат работы функции encrypt() (рис. 2.1)



Рис. 2.1: Функция encrypt()

3. Далее, создаю функцию decrypt(), которая по заданному шифротексту выводит исходный текст. Также можно опционально задать ключ дешифровки, или же он будет сгенерирован автоматически. Функция преобразует шифротекст в 16-ную СС и применяет XOR для шифротекста и ключа

```

def decrypt(ciphertext: str, key: list = None):
    """

    ciphertext_16 = [char.encode('cp1251').hex().upper() for char in ciphertext]
    if not key:
        key = generate_key(length=len(ciphertext))

    print(f'Ключ шифрования:', ' '.join(str(s) for s in key))
    print(f'Исходный шифротекст:', ciphertext)

```

```

decrypted_text = []
for i in range(len(ciphertext)):
    xor_char = int(ciphertext_16[i], 16) ^ int(key[i], 16)
    decrypted_text.append(int2hex(xor_char))

decrypted_text = validate(decrypted_text)
decrypted_text = bytes.fromhex("".join(decrypted_text)).decode('cp1251')
print('Расшифрованный текст: ', decrypted_text)
return {
    'key': key,
    'text': decrypted_text
}

```

4. Результат работы функции decrypt() с тем же ключом, что и в шифровании (рис. 2.2)

```

PS D:\Рабочий стол\university\sem4\lab5\lab7> & C:\Users\Admin\AppData\Local\Programs\
abs\lab7\gamma.py
Ключ шифрования: 38 19 61 3A 3D B5 0A 77 8C 97 AC D6 42 03 16 DF F2 EE FF 21 E5 FE
Исходный текст: 01 20 CD EE E2 F8 EC 20 C3 EE E4 EE EC 20 E4 F8 F3 E7 FC FF 21
Шифротекст: 88-89M0JNB/6;Ф+3-8
Ключ шифрования: 38 19 61 3A 3D B5 0A 77 8C 97 AC D6 42 03 16 DF F2 EE FF 21 E5 FE
Исходный шифротекст: 88-89M0JNB/6;Ф+3-8
Исходный текст: С Новым Годом, друзья!
Расшифрованный текст: С Новым Годом, друзья!
PS D:\Рабочий стол\university\sem4\lab5\lab7>

```

Рис. 2.2: decrypt() с тем же ключом

5. Результат работы функции decrypt() со случайным ключом (рис. 2.3)

```

PS D:\Рабочий стол\university\sem4\lab5\lab7> & C:\Users\Admin\AppData\Local\Programs\
abs\lab7\gamma.py
Ключ шифрования: 0A 6C 4C 55 E9 15 1E E4 6A 1D 20 B1 99 BE EC 98 0A 07 1F 5A B5 B4
Исходный текст: 01 20 CD EE E2 F8 EC 20 C3 EE E4 EE EC 20 E4 F8 F3 E7 FC FF 21
Шифротекст: 88-89M0JNB/6;Ф+3-8
Ключ шифрования: 0D 5F 05 58 F9 D6 07 55 85 A2 8F 14 6C F0 EC 38 F7 6E 2F 75 DF AF
Исходный шифротекст: 88-89M0JNB/6;Ф+3-8
Исходный текст: 88-89M0JNB/6;Ф+3-8
Расшифрованный текст: 88-89M0JNB/6;Ф+3-8
PS D:\Рабочий стол\university\sem4\lab5\lab7>

```

Рис. 2.3: decrypt() со случайным ключом

6. Функция find_key() вызывает функцию decrypt() до тех пор, пока расшифрованный и исходный текст не совпадут, т.е. пытается подобрать ключ для расшифровки


```

def find_key(text):
    """
    Подбирает ключ, с помощью которого сообщение было зашифровано
    """
    decrypted_text = ""
    encryption = encrypt(text)
    while decrypted_text != text:
        decryption = decrypt(encryption['ciphertext'])
        decrypted_text = decryption['text']
        print(f"Полученный текст: {decrypted_text}")
    print(f"Ключ успешно подобран! {decryption['key']}")

```

3 Контрольные вопросы

1. Поясните смысл однократного гаммирования

Гаммирование – выполнение операции XOR между элементами гаммы и элементами подлежащего сокрытию текста. Если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте.

2. Перечислите недостатки однократного гаммирования

Шифр абсолютно стойкий только тогда, когда ключ сгенерирован из случайной двоичной последовательности

3. Перечислите преимущества однократного гаммирования

Это симметричный способ шифрования; алгоритм не дает никакой информации об исходном сообщении; шифрование/дешифрование может быть применено одной программой (в обратном порядке)

4. Почему длина открытого текста должна совпадать с длиной ключа?

Если ключ длиннее, то часть текста (разница между длиной ключа и открытого текста) не будет зашифрована. Если же ключ короче, то однозначное дешифрование невозможно

5. Какая операция используется в режиме одноразового гаммирования, назовите её особенности?

операция XOR (сложение по модулю 2), её особенность - симметричность, т.к. если ее применить 2 раза, то вернется исходное значение

6. Как по открытому тексту и ключу получить шифротекст?

Сначала исходный текст и ключ шифрования преобразуются в 16-ную СС, затем, применяется операция XOR для каждого элемента ключа и текста. Полученный шифротекст декодируется из 16-ной СС и получается набор из символов.

7. Как по открытому тексту и шифротексту получить ключ?

Применить операцию XOR для каждого элемента шифротекста и открытого текста: $\text{key}[i] = \text{crypted}[i] \text{ XOR } \text{text}[i]$

8. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра?

Необходимые и достаточные условия абсолютной стойкости шифра:

- полная случайность ключа
- равенство длин ключа и открытого текста
- одноразовое использование ключа

4 Выводы

Я узнал о схеме однократного гаммирования и научился ее применять на практике