Отчет по лабораторной работе №7

Элементы криптографии. Однократное гаммирование

Легиньких Галина Андреевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Выполнение лабораторной работы

1. Написала функцию для генерации рандомного ключа(состоит из случайно последовательности символов. (рис. 2.1)

```python
import random

def generate_key(word):
    key = ""
    for _ in range(len(word)):
        key += random.choice("0123456789abcdef") # Шестнадцатеричная сисктма
    return key
```

Рис. 2.1: Ключ

2. Функция шифрования. В основе используется XOR. (рис. 2.2)

```python
def encrypt(plaintext, key):
    chiphertext = ""
    for i in range(len(plaintext)):
        char = plaintext[i]
        key_char = key[i%len(key)]
        encrypted_char = chr(ord(char) ^ ord(key_char)) # XOR операция
        chiphertext += encrypted_char
    return chiphertext
```

Рис. 2.2: Шифрование

3. Аналогичный принцип для дешифрования. (рис. 2.3)

```python
def decript(chiphertext, key):
    decriptedtext = ""
    for i in range(len(chiphertext)):
        char = chiphertext[i]
        key_char = key[i%len(key)]
        decrypted_char = chr(ord(char) ^ ord(key_char)) # XOR операция
        decriptedtext += decrypted_char
    return decriptedtext
```

Рис. 2.3: Дешифрование

4. Функция нахождения ключей для фрагмента. (рис. 2.4)

```python
def find_possible_key(chiphertext, fragment):
    possible_keys = []
    for i in range(len(chiphertext) - len(fragment)+1):
        possible_key = " "
        for j in range(len(fragment)):
            char = chiphertext[i+j]
            fragment_char = fragment[j]
            key_char = chr(ord(char) ^ ord(fragment_char))
            possible_key += key_char
        possible_keys.append(possible_key)
    return possible_keys
```

Рис. 2.4: Фрагмент

5. Основной кусок кода, где задается строчка и вызов всех функций. (рис. 2.5)

```python
text = "С Новым Годом, друзья!"
key = generate_key(text)
encrypted_text = encrypt(text, key)
decrypted_text = decript(encrypted_text, key)
fragment = "С Новым"
possible_keys = find_possible_key(encrypted_text, fragment)
print("Ключ -", key)
print("Зашифрованный текст -", encrypted_text)
print("Дешифрованный текст -", decrypted_text)
print("Вызможные ключи -", possible_keys)
```

Рис. 2.5: Вызов функций

6. После запуска программы мы получим следующее. (рис. 2.6)

```
Ключ - 336033fa5298c0bcc0d6d1
Зашифрованный текст - В‼ЫЎЁ0ђАЦЌЙIџВїУѳѓQЫ►
Дешифрованный текст - С Новым Годом, друзья!
Вызможные ключи - [' 336033f', ' вЋ\x13?J\x11ỏ', ' \nЮ\x1cFhЊ\x1a', ' /Cedɵm0', '  jGѿ\x14G1', ' YОќ\x18>F:', ' {a;2?Mc', ' ѠI\x1134\x14P', ' \x07b\x
108mïѾ', ' -Э\x1baЮЉk', ' ,ЦВТΨ\x1c\x1f', " 'ѿЁỎehO", ' ~<џi\x118o', ' нbJ\x1dA\x18F', ' ѣѶ>Ma1\x17', ' vѓnmH`ь']
```

Рис. 2.6: Вывод

3 Вывод

Освоила на практике применение режима однократного гаммирования.