Презентация о выполнении лабораторной работы №7

Элементы криптографии. Однократное гаммирование

Евсеева Дарья Олеговна

22 октября, 2022

Российский Университет Дружбы Народов, Москва, Россия

Цель работы

Целью работы является освоение на практике применения режима однократного гаммирования.

Задачи выполнения

Необходимо разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования, которое должно:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

Результаты выполнения

В ходе работы были выполнены поставленные задачи:

- Разработано приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования

```python
import numpy as np
```

```python
def check_enc(array):
    arr = [i if len(i)==2 else "0"+i for i in array]
    return arr
```

```python
def generate_key(n):
    k = np.random.randint(0, 255, n)
    key = [hex(i)[2:] for i in k]
    key = check_enc(key)
    return key
```

```python
def encode_str(string):
    s = []
    for i in string:
        s.append(i.encode('cp1251').hex())
    return s
```

Figure 1: Вспомогательные функции

Результаты выполнения

```
def cipher_define(text, key):
    t = encode_str(text)
    c = []
    for i in range(len(t)):
        c.append(hex(int(t[i], 16) ^ int(key[i], 16))[2:])
    c = check_enc(c)
    cipher = bytearray.fromhex("".join(c)).decode('cp1251')
    return cipher
```

```
def key_define(text, cipher):
    t = encode_str(text)
    c = encode_str(cipher)
    k = []
    for i in range(len(t)):
        k.append(hex(int(t[i], 16) ^ int(c[i], 16))[2:])
    key = check_enc(k)
    return key
```

Figure 2: Основные функции

Результаты выполнения

```
initial_str = "С Новым Годом, друзья!"
```

```
key1 = generate_key(len(initial_str))
print(key1)
```

```
['76', '3a', '37', 'd5', 'ed', 'e7', '09', '5a', 'eb', '81', '35', '9b', '14', 'a2', 'd8', '36', '95', '4e', '61',
'f8', '00', '4d']
```

```
cipher1 = cipher_define(initial_str, key1)
print(cipher1)
```

```
§ъ;ez(oCuшЋшTeS†яl
```

```
key2 = key_define(initial_str, cipher1)
print(key2)
```

```
['76', '3a', '37', 'd5', 'ed', 'e7', '09', '5a', 'eb', '81', '35', '9b', '14', 'a2', 'd8', '36', '95', '4e', '61',
'f8', '00', '4d']
```

Figure 3: Проверка работы

Выводы

В результате проделанной работы мы освоили на практике применение режима однократного гаммирования.