Лабораторная работа №7

Основы Информационной Безопасности

Александрова УВ

10 мая 2024

Российский университет дружбы народов, Москва, Россия

Цель работы

Цель работы

Освоить на практике применение режима однократного гаммирования.

Выполнение лабораторной работы

Я начала с разработки функции, которая будет подбирать ключ к открытому тексту по нескольким критериям: - полная случайность ключа; - равенство длин ключа и открытого текста; - однократное использование ключа.

```python
import random
import string

text = 'С Новым Годом, друзья!'

def key_gen(text):
    key = ''
    for i in range(len(text)):
        key += random.choice(string.digits+string.ascii_letters)
    return key

len(key_gen(text)) == len(text)
```

```
True
```

```python
key = key_gen(text)
print(key)
```

```
omyXcsGVZlDsoUDI9lnezj
```

Figure 1: функция key_gen

```python
def shifr(text, key):
    sh_text = ''
    for t,k in zip(text, key):
        xor = ord(t) ^ ord(k) # реализуем умножение по модулю два, при этом переводим элементы в биты
        sh_text += chr(xor)
    return sh_text

sh_text = shifr(text, key)
sh_text
```

```
'юМѬАёиǫѵщђѰэѓуdѼүѲљЩеК'
```

Figure 2: функция shifr

Чтобы не писать новые функции для поиска ключа, мы можем просто воспользоваться функцией **shifr**, которая по сути и сможет найти ключ по фрагменту и шифротекстсу, что следует из этого рассуждения

`Ci ? Pi = Pi ? Ki ? Pi = Ki`,

`Ki = Ci ? Pi`.

```
new_key = shifr(sh_text, text[15:21]) # находим ключ по той же функции, но меняем аргументы на шифротект и открытый текс
```

```
decode = shifr(new_key, sh_text) # расшифровывем. все работает!
```

```
print('Открытый текст: ', text, '\nИзвестный ключ к открытому тексту: ', key, '\nШифротекст: ', sh_text)
print('\n\nВозможный ключ по шифротексту и фрагменту: ', new_key, '\nРасшифрованный фрагмент: ', decode)
```

```
Открытый текст:  С Новым Годом, друзья!
Известный ключ к открытому тексту:  omyXcsGVZlDsoUDI91nezj
Шифротекст:  юМѬАёиφvщђѰэѓydȎѹѲљЩеК


Возможный ключ по шифротексту и фрагменту:  zЍ'Q□w
Расшифрованный фрагмент:  друзья
```

Figure 3: вывод программы