

Лабораторная Работа №8. Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

Основы информационной безопасности

Барсегян В.Л.

Российский университет дружбы народов им. Патриса Лумумбы, Москва, Россия

- Барсегян Вардан Левонович
- НПИБд-01-22
- Российский университет дружбы народов
- [1132222005@pfur.ru]
- <https://github.com/VARdamn/oib>

Вводная часть

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом

Выполнение лабораторной работы

Создаю функцию `encrypt()`, которая будет шифровать заданный текст с помощью однократного гаммирования. На вход функция получает открытый текст, также можно задать определенный ключ шифрования. Если ключа нет, то он генерируется рандомно. 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.

Функция encrypt()

```
def encrypt(text: str, key: list = None):
```

```
    """
```

```
    Выводит шифротекст для заданного текста.
```

```
    Если ключа нет, то генерируется случайный ключ
```

```
    """
```

```
    if not key:
```

```
        key = generate_key(length=len(text))
```

```
    text_16 = [ord(char) for char in text]
```

```
    key = [ord(el) for el in key]
```

```
    print(f"Ключ шифрования:", ' '.join(str(s) for s in key))
```

```
    print(f"Исходный текст:", text)
```

Генерация ключа, если он не задан, происходит в функции generate_key() из ascii-символов и цифр

```
def generate_key(length: int):  
    """  
    Генерация случайного ключа длины length  
    """  
    return random.sample(string.ascii_letters + string.digits, length)
```


Работа программы:

- сначала создается случайный ключ и с этим ключом шифруются тексты p1 и p1 (переменные c1 и c2)
- далее, шифротекст c1 шифруется по ключу c2
- полученный шифротекст c1_c2 шифруется по ключу открытого текста. в результате, получаем второй открытый текст, ранее неизвестный

```
0
1  p1 = 'НаВашисходящийот1204'
2  p2 = 'ВСеверныйфилиалБанка'
3  key = generate_key(20)
4
5  c1 = encrypt(p1, key=key)
6  c2 = encrypt(p2, key=key)
```

Полный вывод работы программы

```
PS D:\Рабочий стол\university\сем4\ои6\labs\lab8> & C:\Users\Admin\AppData\Local\Programs\Python\Python311/python.exe "d:/Рабочий стол/university/сем4/ои6/labs/lab8/gamma.py"
```

Ключ шифрования: 74 81 112 100 97 55 48 52 66 72 54 117 50 68 119 122 105 76 86 51

Исходный текст: НаВашисходящийот1204

Шифротекст: iWBeЩUщфОЮмнЪшХ~f

Ключ шифрования: 74 81 112 100 97 55 48 52 66 72 54 117 50 68 119 122 105 76 86 51

Исходный текст: ВСеверныйфилиалБанка

Шифротекст: jΨxієVЙиокЎюъУьлџѢf

Ключ шифрования: 1112 1136 1093 1110 1108 1143 1037 1151 1147 1036 1038 1102 1034 1140 1100 1131 1113 1137 1132 1027

Исходный текст: iWBeЩUщфОЮмнЪшХ~f

Шифротекст: ◀'0}x|rwг ♣\$EUE

Ключ шифрования: 1042 1072 1042 1072 1096 1080 1089 1093 1086 1076 1103 1097 1080 1081 1086 1090 49 50 48 52

Исходный текст: ◀'0}x|rwг ♣\$EUE

Шифротекст: ВСеверныйфилиалБанка

Ключ шифрования: 1042 1057 1077 1074 1077 1088 1085 1099 1081 1092 1080 1083 1080 1072 1083 1041 1072 1085 1082 1072

Исходный текст: ◀'0}x|rwг ♣\$EUE

Шифротекст: НаВашисходящийот1204

Figure 2: Работа программы

Я применил режим однократного гаммирования на примере кодирования различных исходных текстов одним ключом