

# Отчёт по лабораторной работе №8

# Шифрование (кодирование) различных исходных текстов

1. Для выполнения лабораторной работы был использован язык Python.
2. Для реализации алгоритма в начале опишем необходимые функции:
  - create\_key: создание ключа
  - hexadical\_form: перевод ключа в 16-ичную форму
  - gamming: гаммирование
3. Зададим в качестве примера два текста и реализуем алгоритм. Листинг программы выглядит следующим образом:

# Импортируем необходимые библиотеки

```
import random as rnd import string as str
```

# Пишем необходимые функции

```
def create_key(size=6, chars=str.ascii_letters +  
str.digits): return ''.join(rnd.choice(chars) for _ in  
range(size))
```

```
def hexadical_form(s): return '  
' + ''.join("{:02x}".format(ord(c)) for c in s)
```

```
def gamming(fst_text, sec_text): fst_text_ascii =  
[ord(i) for i in fst_text] sec_text_ascii = [ord(i) for  
i in sec_text] return ''.join(chr(s^k) for s,k in  
zip(fst_text_ascii, sec_text_ascii))
```

# Выполним шифрование

```
P1, P2 = 'ПримерТекста1',  
'ПримерДругогоТекста' print('Исходные  
тексты:') print(P1) print(P2)  
key=create_key(len(P1)) print('для  
кодирования текстов:', create_key(len(P1)))  
print('Шестнадцатиричный ключ для  
кодирования текстов:', hexadecimal_form(key))  
print('для открытого текста 1 и ключа:',  
gammimg(P1, key)) print('Шифротекст для  
открытого текста 2 и ключа:', gammimg(P2,
```

# Результат выполнения кода:

Исходные тексты:

ПримерТекста1

ПримерДругогоТекста

Ключ для кодирования текстов: 3Uugw4iH5z8aA

Шестнадцатичный ключ для кодирования текстов: 36 41 35 32 47 78 4a 63 59 66 4a 41 44

Шифротекст для открытого текста 1 и ключа: ЩЁЙЎθиѡіЪЧЈψи

Шифротекст для открытого текста 2 и ключа: ЩЁЙЎθиѡѣУKsVθ○

Получим тексты путём гаммирования двух шифровок и исходного текста:

6A52GxJcYfJAD

6A52Gx| 6Вы470

Рис. 1. Итоговый результат

# Выводы

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