# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

Дисциплина: Информационная безопасность

**Студент: Ду Нашсименту Висенте Феликс Жозе**

**Группа: НкНбд01-20**

Москва 2023

# 7.1. Цель работы

Освоить на практике применение режима однократного гаммирования

# 7.2. Порядок выполнения работы

у Меня есть скрипт на языке Python, который выполняет шифрование и дешифрование для заданного сообщения MS. Скрипт использует случайно сгенерированный ключ, выполняет побитовые операции XOR на ASCII-значениях символов и затем выводит зашифрованное сообщение, а также расшифрованное сообщение. Кроме того, скрипт вычисляет исходный ключ, выполняя XOR-операции над зашифрованным сообщением и зашифрованным ключом.

Давайте пройдемся по коду и предоставим краткое описание:

1. funcao1(text):

- Преобразует символы входного текста в их шестнадцатеричное представление.
- Используется для отображения ключа и зашифрованного сообщения в шестнадцатеричном формате.

2. funcao2(size):

- Генерирует случайный ключ указанного размера, состоящий из заглавных букв и цифр.
- Используется в качестве ключа шифрования.

3. funcao3(text, key):

- Выполняет побитовую операцию XOR над ASCII-значениями символов во входном тексте и ключе.
- Используется как для шифрования, так и для дешифрования.

4. funcao4(text, incrypt):

- Аналогично funcao3, выполняет операцию XOR, но на этот раз используется для вычисления исходного ключа из зашифрованного сообщения и зашифрованного ключа.

Вот краткое описание выполнения скрипта:

- Исходное сообщение: "С Новым Годом,друзья!" Генерируется случайный ключ с помощью funcao2.
- Ключ преобразуется в шестнадцатеричный формат с использованием funcao1.
- Исходное сообщение шифруется с использованием сгенерированного ключа и выводится в шестнадцатеричном формате.
- Зашифрованное сообщение расшифровывается обратно с использованием того же ключа и выводится.
- Исходный ключ вычисляется из зашифрованного сообщения и зашифрованного ключа.
- Исходный ключ и вариант расшифрованного сообщения (вычисленный из зашифрованного сообщения и вычисленного ключа) выводятся на экран.

```
[108] import random
      import string
```

```
[109] def funcao1(text):
        t =''.join(hex(ord(i))[2:] for i in text )
        return t
```

```
def funcao2(size):
  return ''.join(random.choice(string.ascii_letters+string.digits)for _ in range(size))
```

+ Code + Text

```
[111] def funcao3(text, key):
        tk= ''.join(chr(a^b) for a,b in zip (text, key))
        return tk
```

```
def funcao4(text,incrypt):
  inc = ''.join(chr(a^b) for a,b in zip (text,incrypt))
  return inc
```

```
[113] MS = 'С Новым Годом,друзья!'
```

```
[114] key = funcao2(len(MS))
```

```
[115] xkey= funcao1(key)
```

```
[116] print("Ключ: {}".format(key))
```

```
[116] print("Ключ: {}".format(key))

      Ключ: pbuF3XWf4QaFy780c5sCk
```

```
[117] print("Ключ шестнадцатиричном виде: {}".format(xkey))

      Ключ шестнадцатиричном виде: 706275463358576634516146793738306335734 36b
```

```
[118] incrypt= funcao3([ord(i) for i in MS], [ord(i) for i in key])
```

```
[119] x_crypt = funcao1(incrypt)
      print("Зашифрованное сообщение: {}".format(x_crypt))

      Зашифрованное сообщение: 45142468478401413 46b4642746f4554784451b40c47042040243f40c4a
```

```
[120] decrypt = funcao3([ord(i) for i in incrypt], [ord(i) for i in key])
      print("Расшифрованное сообщение: {}".format(decrypt))

      Расшифрованное сообщение: С Новым Годом,друзья!
```

```
[125] computed_key = funcao4([ord(i)for i in MS], [ord(i)for i in incrypt])
      descypt_Ck = funcao4([ord(i)for i in incrypt], [ord(i) for i in key])
      print("Исходный Ключ: {}".format(key))
      print("вариантов прочтения открытого текста: {}".format(descypt_Ck))


      Исходный Ключ: pbuF3XWf4QaFy780c5sCk
      вариантов прочтения открытого текста: С Новым Годом,друзья!
```

# Выводы

Этот код выполняет простую форму шифрования XOR. Это базовая демонстрация XOR-шифрования и не должен использоваться для безопасных коммуникационных целей. Кроме того, код предполагает, что входное сообщение и ключ находятся в той же кодировке символов. Если это не так, результаты могут не соответствовать ожиданиям.