Презентация лабораторной работы 7

ТЕМА «**Элементы криптографии. Однократное гаммирование**»

Выполнил:
Студент группы НПИбд-02-21
Студенческий билет № 1032205421
Стелина Петрити

Цель работы

Освоить на практике применение режима однократного гаммирования

Последовательность выполнения работы

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования.

Приложение 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 .

```python
import re
russian_alf= [  'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т',
    'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', ' ', '!', '?', '.', ',', '«', '»']

def encryption (txt, gama):
  txtlen =len(txt)
  gamalen = len(gama)
  keytxt = []
  for i in range(txtlen // gamalen):
      keytxt.extend(gama)
  for i in range(txtlen % gamalen):
      keytxt.append(gama[i])
  code=[]
  for i in range(txtlen):
      if txt[i] in russian_alf:
          code.append(russian_alf[(russian_alf.index(txt[i]) + russian_alf.index(keytxt[i])) % len(russian_alf)])
      else:
          code.append(txt[i])
  return ''.join(code)
encryptmsg = encryption('С Новым Годом, друзья!', 'ААЪАЙАААААААААААААААА')
print(encryptmsg)
```

```
С Бовым Годом, друзья!
```

рис. 1 Encryption

```python
def decryption(ciphertext, gama):
    txtlen = len(ciphertext)
    gamalen = len(gama)
    keytxt = []

    # Растягиваем ключ (гамму) до длины текста
    for i in range(txtlen // gamalen):
        keytxt.extend(gama)
    for i in range(txtlen % gamalen):
        keytxt.append(gama[i])

    plaintext = []

    for i in range(txtlen):
        if ciphertext[i] in russian_alf:
            # Вычитание индекса ключа из индекса символа шифротекста
            plaintext.append(russian_alf[(russian_alf.index(ciphertext[i]) - russian_alf.index(keytxt[i])) % len(russian_alf)])
        else:
            plaintext.append(ciphertext[i])

    return ''.join(plaintext)

# Пример использования
decryptmsg = decryption(encryptmsg, 'ААЪАЙАААААААААААААААА')
print(decryptmsg)
```

```
С Новым Годом, друзья!
```

рис. 2 Decryption

```python
def derivekey(ciphertxt, plaintxt):
    if len(ciphertxt) != len(plaintxt):
        raise ValueError("Ciphertext and plaintext must be of the same length.")

    key = []

    for i in range(len(ciphertxt)):
        if ciphertxt[i] in russian_alf and plaintxt[i] in russian_alf:
            # Вычисляем индекс символа ключа
            key_index = (russian_alf.index(ciphertxt[i]) - russian_alf.index(plaintxt[i])) % len(russian_alf)
            key.append(russian_alf[key_index])
        else:
            # Если символ не в алфавите (например, запятая или пробел), просто добавляем его как есть
            key.append(ciphertxt[i])

    return ''.join(key)

# Пример использования
ciphertxt = "С Бовым Годом, друзья!"
plaintxt = "С Новым Годом, друзья!"

# Нахождение ключа
derivedkey = derivekey(ciphertxt, plaintxt)
print("Derived Key:", derivedkey)
```

```
Derived Key: ААЪовымААодомААдрузьяА
```

рис. 3 Drive Key

Вывод

В результате выполнения работы я освоила на практике применение режима однократного гаммирования