Информация

Докладчик

- Надиа Эззакат
- студент группы НПМбд-02-20
- Факультет физико-математических и естественных наук
- Российский университет дружбы народов

Цель лабораторной работы

Освоить на практике применение режима однократного гаммирования

Выполнение лабораторной работы(1)

Сначала я импортировала необходимые библиотеки, как показано на шаге 43. Затем я создала функцию, выполняющую сложение по модулю два для двух строк, что является ключевой частью процесса шифрования, как описано на шаге 44. У меня также были открытые/исходные тексты, которые должны были быть одинаковой длины и готовы к шифрованию, как показано на шаге 45. Далее я создала ключ той же длины, что и открытые тексты, как указано на шаге 46. После подготовки ключа и открытых текстов я получила шифротексты с использованием ранее созданной функции, при условии, что у меня были и открытые тексты и ключ, как описано на шаге 47. Для завершения процесса мне также понадобился способ получить открытые тексты из шифротекстов с использованием того же ключа, для чего я обратилась к шагу 48. Эти шаги совместно позволили мне выполнять шифрование методом однократного гаммирования для двух текстов с общим ключом.

jupyter lab08 Last Checkpoint: last month

File Edit View Run Kernel Settings Help

Code

```
[43]: import random
      from random import seed
      import string
```

```
[44]: def cipher_text_function(text, key):
          if len(key) != len(text):
              return "ключ и текст должны быть одной длины!"
          cipher_text = ''
          for i in range(len(key)):
              cipher_text_symbol = ord(text[i]) ^ ord(key[i])
              cipher_text += chr(cipher_text_symbol)
          return cipher_text
```

```
[45]: text_1 = " С Новым Годом,друзья! "
      text_2 = " поздравляем с 8 марта!"
```

```
[46]: key = ''
      seed(23)
      for i in range(len(text_1)):
          key += random.choice(string.ascii_letters + string.digits)
      print(key)

      7X8s51fbLtByHwiUmrCaoND
```

```
[47]: cipher_text_1 = cipher_text_function(text_1, key)
      cipher_text_2 = cipher_text_function(text_2, key)
      print('Первый шифротекст:' , cipher_text_1)
      print('Второй шифротекст:' , cipher_text_2)

      Первый шифротекст:  ♈ ǯЋЃЭўlѦѼэѶыEѡЭбѵЭPod
      Второй шифротекст:  ѧIфЁѱiѐѷлѷxhжImМюѳCЭѾe
```

```
[48]: print('первый открытый текст:', cipher_text_function(cipher_text_1, key))
      print('второй  открытый текст:', cipher_text_function(cipher_text_2, key))

      первый открытый текст:  С Новым Годом,друзья!
      второй  открытый текст:  поздравляем с 8 марта!
```

Выполнение лабораторной работы(2)

В этом процессе, сначала было выполнено сложение по модулю два над двумя шифротекстами с использованием ранее разработанной функции, как указано в In[53]. Затем, на шаге In[50], я получила открытые тексты, применяя ту же функцию, при условии, что у меня были оба шифротекста и хотя бы один из открытых текстов. Этот шаг позволил восстановить исходные сообщения. В дальнейшем, на In[52], я извлекла часть второго текста, выравнивая его с позициями символов из первого открытого текста. Эта операция зависела от ранее созданной функции и знании обоих шифротекстов и сегменте первого открытого текста. Эти ключевые шаги позволили манипулировать и восстанавливать информацию в процессе шифрования.

[53]:
```
cipher_text_xor = cipher_text_function(cipher_text_1, cipher_text_2)
print('первый шифротекст XOR второй шифротекст:', cipher_text_xor)
```

```
первый шифротекст XOR второй шифротекст:  O*
Б Л\O}Ќ
```

[50]:
```
print('первый открытый текст:', cipher_text_function(cipher_text_xor, text_2))
print('второй открытый текст:', cipher_text_function(cipher_text_xor, text_1))
```

```
первый открытый текст:  С Новым Годом,друзья!
второй открытый текст:  поздравляем с 8 марта!
```

[51]:
```
text_1_ = text_1[3:6]
print('часть первого открытого текста:', text_1_)
```

```
часть первого открытого текста: Нов
```

[52]:
```
cipher_text_xor_ = cipher_text_function(cipher_text_1[3:6], cipher_text_2[3:6])
print('часть второго открытого текста:', cipher_text_function(cipher_text_xor_, text_1_))
```

```
часть второго открытого текста: здр
```

Вывод

В ходе выполнения данной лабораторной работы я освоила на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.