Лабораторная работа №8

Основы информационной безопасности

Павлюченков С.В.

07 сентября 25

Российский университет дружбы народов, Москва, Россия

Докладчик

- Павлюченков Сергей Витальевич
- Студент ФФМиЕН
- Российский университет дружбы народов
- 1132237372@pfur.ru
- https://serapshi.github.io/svpavliuchenkov.github.io/

Цель работы

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

Задание

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе ; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

Выполнение лабораторной работы

Ранее написанный код

Рис. 1: Код из 7 лабораторной

Шифрование

С помощью написаных ранее функций создаем уникальный ключ и шифрую им два предложения, после чего проверяю возможно обратно дешифрования

```
text1 = 'С новым годом!'
text2 = 'С новым годовщинрй товарищи!'
key = get_key(len(text))
key
```

```
'sFv39CxMp2tF0s'
```

```
encrypted1 = xor_crypt(text1, key)
encrypted2 = xor_crypt(text2, key)
print(encrypted1, encrypted2)
```

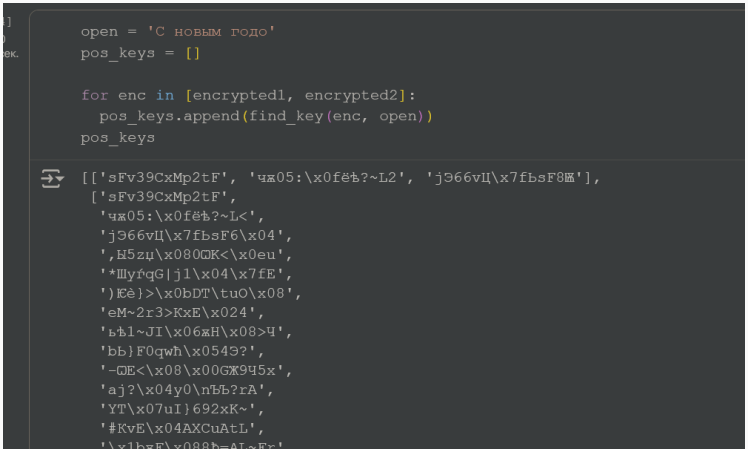
```
ђfыЙЋJфmуЌр0ґЌR ђfыЙЋJфmуЌр0ґЪкысжЊЏЁцЎрФьЏJR
```

```
for enc in [encrypted2, encrypted1]:
  print(xor_crypt(enc,  key))
```

```
С новым годовщинрй товарищи!
С новым годом!
```

Создание ключей

По открытому слову создаю список возможных ключей для каждого предложения, так как они зашифрованы одним ключом, то в теории при подборе может оказаться два одинаковых ключа в в двух массивах, и тогда это может быть истиным, но в этом случае такого не было

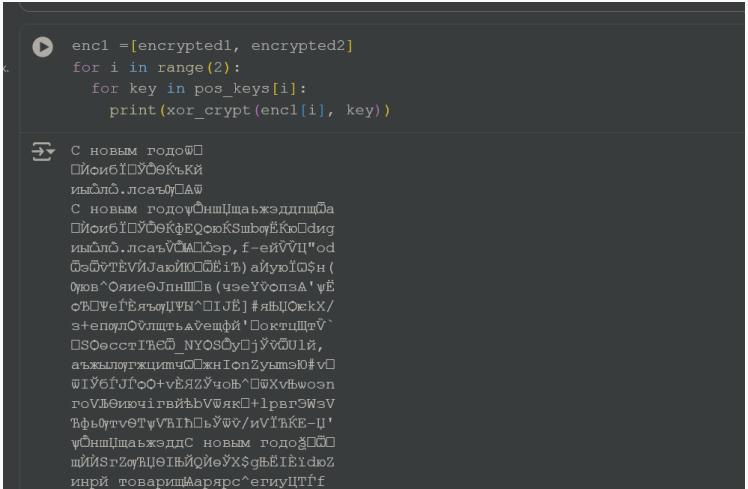


```
open = 'С новым годо'
pos_keys = []

for enc in [encrypted1, encrypted2]:
  pos_keys.append(find_key(enc, open))
pos_keys
```

```
[['sFv39CxMp2tF', 'чж05:\x0fёЪ?~L2', 'jЭ66vЦ\x7fЬsF8Ж'],
 ['sFv39CxMp2tF',
  'чж05:\x0fёЪ?~L<',
  'jЭ66vЦ\x7fЬsF6\x04',
  ',Ы5zц\x080ЖK<\x0eu',
  '*ШyґqG|j1\x04\x7fE',
  ')Юè}>\x0bDT\tuO\x08',
  'eM~2r3>KxE\x024',
  'ьЪ1~JI\x06жH\x08>Ч',
  'bЬ}F0qwh\x0549?',
  '-ЖE<\x08\x00GЖ9Ч5x',
  'aj?\x04y0\nЪЪ?rA',
  'YT\x07uI}692xK~',
  '#KvE\x04AXCuAtL',
  '\x1bжF\x088b=AL~Fr',
```

После чего я дешифрую предложения каждым возможным ключом, и получаю предложения, с некоторым количеством информации(рис. (fig:004?)).



Выводы

В данной лабораторной работе я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.