Отчет по лабораторной работе № 8

По дисциплине Информационная безопасность

Выполнил: Максимов Алексей Александрович

Группа: НПИ-бд-01-20

Российский Университет Дружбы Народов #### г. Москва

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

Пишем программу для кодирования и декодирования сообщений

```python
def encode_val(word):
    d = form_dict()
    return [k for c in word for k, v in d.items() if v == c]
# Разбираем слово и ключ из ASCII
4 usages
def decode_val(value):
    d = form_dict()
    decode_v = ''
    for i in value:
        decode_v = decode_v + d[i]
    return decode_v
1 usage
def comparator(value, key):
    return dict([(index, list(character))
                 for index, character in enumerate(zip(value, itertools.cycle(key)))])
# сложение по модулю
2 usages
def full_encode(value, key):
    d = comparator(value, key)
    l = len(form_dict())
    return [(v[0] ^ v[1]) % l for v in d.values()]


2 usages
def full_encode3_text(C1, C2, Pn):
    ans = []
    for i in range(len(Pn)):
        ans.append((Pn[i] ^ C1[i] ^ C2[i]))
    return ans
```

Image1

```
P1 = 'S novim godom, druzia!'
P2 = 'Schactlivogo Rozdestva'
key = 'pvnquervjttjvhkunuziiu'
encode_p1 = encode_val(P1)
encode_p2 = encode_val(P2)
# кодируем начальные тексты с помощью ключа
encode_c1 = full_encode(encode_p1, encode_val(key))
encode_c2 = full_encode(encode_p2, encode_val(key))
# переводим в текстовый вид
C1 = decode_val(encode_c1)
C2 = decode_val(encode_c2)
print('P1: ', "\n", P1, "; len:", len(P1))
print('C1: ', "\n", C1, "; len:", len(C1))
print('P2: ', "\n", P2, "; len:", len(P2))
print('C2: ', "\n", C2, "; len:", len(C2))
# декодируем сообщение без ключа
print("декодируем сообщение без ключа")
P1_decoded_no_key = full_encode3_text(encode_c1, encode_c2, encode_p2)
P2_decoded_no_key = full_encode3_text(encode_c1, encode_c2, encode_p1)
# переводим в текстовый вид
P1_1 = decode_val(P1_decoded_no_key)
P2_1 = decode_val(P2_decoded_no_key)
print('P1_1: ', "\n", P1_1, "; len:", len(P1_1))
print('P2_1: ', "\n", P2_1, "; len:", len(P2_1))
```

Image2

Проверяем результат

```
C:\Users\maksi\PycharmProjects\lab7_infoBez\venv\Scripts\python.exe C:\Users\maksi\PycharmProjects\lab7_infoBez\main.py
P1:
 S novim godom, druzia! ; len: 22
C1:
DK T ; len: 22
P2:
 Schactlivogo Rozdestva ; len: 22
C2:
 #V:
 ; len: 22
декодируем сообщение без ключа
P1_1:
 S novim godom, druzia! ; len: 22
P2_1:
 Schactlivogo Rozdestva ; len: 22

Process finished with exit code 0
```

Image3

Выводы

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