Лабораторная работа

Хохлачева Яна

Элементы криптографии.

Однократное гаммирование

Цель работы

- Освоить на практике применение режима однократного гаммирования.

Функция кодирования

```
Открытый текст:  С Новым Годом, друзья!

Открытый текст в шестнадцатиричной системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

Ключ в шестнадцатиричной системе:  ['ac', 'd9', '8', '32', '79', '39', '5c', '8', 'fd', 'dc', '18', '8', '99', '76', '1a', '47', '63', '31', 'd7', '7f', '76', '57']

Шифротекст в шестнадцатиричной системе:  ['7d', 'f9', 'c5', 'dc', '9b', 'c2', 'b0', '28', '3e', '32', 'fc', 'e6', '75', '5a', '3a', 'a3', '93', 'c2', '30', '83', '89', '76']

Шифротекст:  }щЕЬ›В°(>2ьжuZ:J"В0ѓ‰v
```

Результат выполнения функции

```
Открытый текст:  С Новым Годом, друзья!

Шифротекст:  }щЕЬ›В°(>2ьжuZ:J“B0f‰v

Открытый текст в шестнадцатиричной системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

Шифротекст в шестнадцатиричной системе:  ['7d', 'f9', 'c5', 'dc', '9b', 'c2', 'b0', '28', '3e', '32', 'fc', 'e6', '75', '5a', '3a', 'a3', '93', 'c2', '30', '83', '89', '76']

Найденный ключ в шестнадцатиричной системе:  ['ac', 'd9', '8', '32', '79', '39', '5c', '8', 'fd', 'dc', '18', '8', '99', '76', '1a', '47', '63', '31', 'd7', '7f', '76', '57']
```

Функция декодирования

```
if key_find == r:
    print("Ключ найден верно!")
else:
    print("Ключ найден неверно!")
```

```
Ключ найден верно!
```

Результат выполнения функции

```
# функция декодирования
def decoding(text, encrypted_text):
    print("Открытый текст: ", text)
    print("\nШифротекст: ", encrypted_text)

    new_text = []
    for i in text:
        new_text.append(i.encode("cp1251").hex())
    print("\nОткрытый текст в шестнадцатиричной системе: ", new_text)

    temporary_text = []
    for i in encrypted_text:
        temporary_text.append(i.encode("cp1251").hex())
    print("\nШифротекст в шестнадцатиричной системе: ", temporary_text)

    xor_text = [hex(int(r,16)^int(t,16))[2:] for (r, t) in zip(new_text, temporary_text)]
    print("\nНайденный ключ в шестнадцатиричной системе: ", xor_text)
    return xor_text
```

```
key_find = decoding(message, et)
```

Проверка

```python
# импортируем необходимые библиотеки
import numpy as np
import operator as op
import sys
```

```python
# сообщение на вход
message = "С Новым Годом, друзья!"
```

```python
# функция шифрования
def encryption(text):
    print("Открытый текст: ", text)
    new_text = []
    for i in text:
        new_text.append(i.encode("cp1251").hex())
    print("\nОткрытый текст в шестнадцатиричной системе: ", new_text)
    # генерируем ключ
    k = np.random.randint(0, 255, len(text))
    key = [hex(i)[2:] for i in k]
    new_key = []
    for i in key:
        new_key.append(i.encode("cp1251").hex().upper())
    print("\nКлюч в шестнадцатиричной системе: ", key)
    # получение зашифрованного сообщения
    xor_text = []
    for i in range(len(new_text)):
        xor_text.append("{:02x}".format(int(key[i], 16) ^ int(new_text[i], 16)))
    print("\nШифротекст в шестнадцатиричной системе: ", xor_text)
    # Зашифрованное сообщение
    encrypted_text = bytearray.fromhex("".join(xor_text)).decode("cp1251")
    print("\nШифротекст: ", encrypted_text)
    return key, xor_text, encrypted_text
```

Вывод

Вывод

- Освоила на практике применение режима однократного гаммирования.