Отчёт по лабораторной работе №3. Шифрование гаммированием

Дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Леонова Алина Дмитриевна, 1032212306
Группа: НФИмд-01-21
Преподаватель: д-р.ф.-м.н., проф. Кулябов Дмитрий Сергеевич

27 ноября, 2021, Москва

Цель и задание работы

Цель работы

Целью данной работы является ознакомление с шифрованием гаммированием и реализация алгоритма на выбранном языке программирования.

Задание

Реализовать алгоритм шифрования гаммированием конечной гаммой программно.

Теоретическое введение

Шифрование гаммированием

Гаммирование, наложение гаммы или Шифр XOR ($\oplus$), – метод симметричного шифрования, заключающийся в наложении последовательности, состоящей из случайных чисел, на открытый текст.

Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных.

Суммирование обычно выполняется в каком-либо конечном поле. Например, суммирование может принимать вид операции исключающее ИЛИ / XOR / $\oplus$.

Шифрование гаммированием

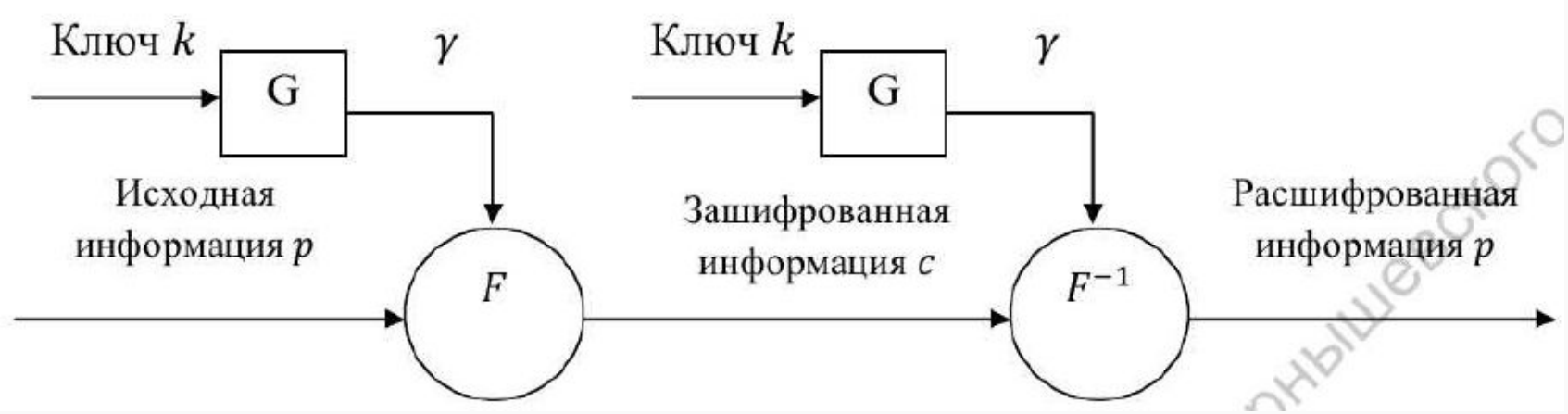


Figure 1: Схема гаммирования с использованием генератора псевдослучайных чисел

В качестве ключа используются либо истинно случайные гаммы, либо псевдослучайные гаммы — последовательности чисел, вычисленные по определённому алгоритму (рис. 1).

Сложение по модулю 2

Символы текста и гаммы представляются в двоичном виде, а затем каждая пара двоичных разрядов складывается по модулю 2:

$$C_i = P_i \oplus K_i$$

$$P_i = C_i \oplus K_i$$

где P_i, C_i – i-ый символ открытого и шифрованного сообщения;

K_i – i-ый символ гаммы (ключа).

Сложение по модулю N

При замене букв исходного сообщения и ключа на числа в рамках определённого алфавита процедуры шифрования и дешифрования выполняются по следующим формулам:

$$C_i = (P_i + K_i) \; mod \; N$$

$$P_i = (C_i + N - K_i) \; mod \; N$$

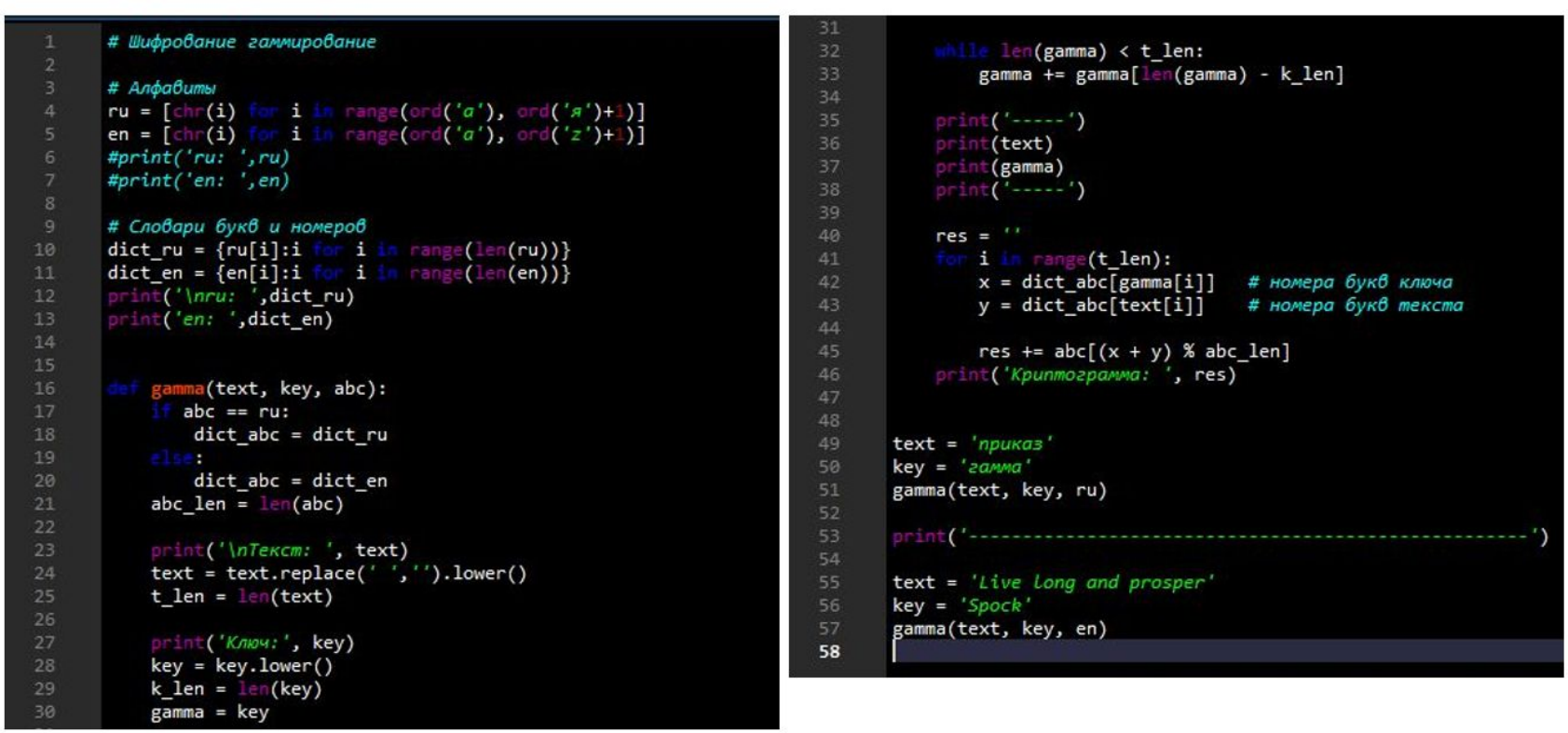
где P_i, C_i – i-ый символ открытого и шифрованного сообщения;

N – количество символов в алфавите;

K_i – i-ый символ гаммы (ключа).

Выполнение работы

Реализация шифрования гаммированием с конечной гаммой



```python
# Шифрование гаммирование

# Алфавиты
ru = [chr(i) for i in range(ord('а'), ord('я')+1)]
en = [chr(i) for i in range(ord('a'), ord('z')+1)]
#print('ru: ',ru)
#print('en: ',en)

# Словари букв и номеров
dict_ru = {ru[i]:i for i in range(len(ru))}
dict_en = {en[i]:i for i in range(len(en))}
print('\nru: ',dict_ru)
print('en: ',dict_en)


def gamma(text, key, abc):
    if abc == ru:
        dict_abc = dict_ru
    else:
        dict_abc = dict_en
    abc_len = len(abc)

    print('\nТекст: ', text)
    text = text.replace(' ','').lower()
    t_len = len(text)

    print('Ключ:', key)
    key = key.lower()
    k_len = len(key)
    gamma = key

    while len(gamma) < t_len:
        gamma += gamma[len(gamma) - k_len]

    print('-----')
    print(text)
    print(gamma)
    print('-----')

    res = ''
    for i in range(t_len):
        x = dict_abc[gamma[i]]   # номера букв ключа
        y = dict_abc[text[i]]    # номера букв текста

        res += abc[(x + y) % abc_len]
    print('Криптограмма: ', res)


text = 'приказ'
key = 'гамма'
gamma(text, key, ru)

print('----------------------------------------------------')

text = 'Live Long and prosper'
key = 'Spock'
gamma(text, key, en)
```

Figure 2: Файл L3_Leonova.py

Результат работы шифрования гаммированием с конечной гаммой

```
In [1]: runfile('E:/GitHub/1.2-IS/Lab_3/L3_Leonova.py',
wdir='E:/GitHub/1.2-IS/Lab_3')

ru:  {'а': 0, 'б': 1, 'в': 2, 'г': 3, 'д': 4, 'е': 5,
'ж': 6, 'з': 7, 'и': 8, 'й': 9, 'к': 10, 'л': 11, 'м':
12, 'н': 13, 'о': 14, 'п': 15, 'р': 16, 'с': 17, 'т':
18, 'у': 19, 'ф': 20, 'х': 21, 'ц': 22, 'ч': 23, 'ш':
24, 'щ': 25, 'ъ': 26, 'ы': 27, 'ь': 28, 'э': 29, 'ю':
30, 'я': 31}
en:  {'a': 0, 'b': 1, 'c': 2, 'd': 3, 'e': 4, 'f': 5,
'g': 6, 'h': 7, 'i': 8, 'j': 9, 'k': 10, 'l': 11, 'm':
12, 'n': 13, 'o': 14, 'p': 15, 'q': 16, 'r': 17, 's':
18, 't': 19, 'u': 20, 'v': 21, 'w': 22, 'x': 23, 'y':
24, 'z': 25}

Текст:  приказ
Ключ: гамма
-----
приказ
гаммаг
-----
Криптограмма:  трфцак
--------------------------------------------------

Текст:  Live long and prosper
Ключ: Spock
-----
livelongandprosper
spockspockspockspo
-----
Криптограмма:  dxjgvgcucxvefqchtf

In [2]:
```

Figure 3: Результат выполнения L3_Leonova.py

Выводы

Цель лабораторной работы была достигнута, алгоритм шифрования гаммированием с конечной гаммой был реализован на языке программирования Python.