

Лабораторная работа №7

Элементы криптографии. Однократное гаммирование

Пономарева Лилия Михайловна

Содержание

Цель работы	4
Теоретическое введение	5
Выполнение лабораторной работы	6
Выводы	10
Список литературы	11

Список иллюстраций

1	Представление данных в двоичном виде	6
2	Реализация сложения по модулю	7
3	Представление данных в двоичном виде	7
4	Получение ключа	8
5	Ключ сообщения	8
6	Получение изначального ключа	9

Цель работы

Освоить на практике применение режима однократного гаммирования. [1]

Теоретическое введение

Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования. В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте. Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) между элементами гаммы и элементами подлежащего сокрытию текста. Такой метод шифрования является симметричным, так как двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, а шифрование и расшифрование выполняется одной и той же программой. Открытый текст имеет символьный вид, а ключ — шестнадцатеричное представление. Ключ также можно представить в символьном виде, воспользовавшись таблицей ASCII-кодов.

Выполнение лабораторной работы

Разработала приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования.

Приложение может: 1. Определять вид шифротекста при известном ключе и известном открытом тексте. 2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

Для реализации первого пункта реализовала функцию `ciphertext` ((рис. 1), в которой с помощью сложения по модулю 2 нашла шифротекст при известном ключе и открытом тексте((рис. 2).

```
def ciphertext(text, key):
    encoded_bytes = text.encode('utf-8', 'replace')
    x = binascii.hexlify(encoded_bytes)
    y = str(x, 'utf-8')
    binary = lambda x: " ".join(reversed(
        [i + j for i, j in zip(*[f"{0:04b}".format(int(c, 16)) for c in reversed("0" + x)][n::2] for n in [1, 0]])))
    x1 = binary(y)

    key1 = bytearray.fromhex(key)
    x_key = binascii.hexlify(key1)
    y_key = str(x_key, 'utf-8')
    x1_key = binary(y_key)
```

Рис. 1: Представление данных в двоичном виде

```

text2 = ""
for i in range(len(x1)):
    if x1[i] == ' ':
        text2 += ' '
    elif x1[i] == x1_key[i]:
        text2 += '0'
    elif x1[i] != x1_key[i]:
        text2 += '1'

text2_ = text2.split()
crypt = ' '
for n in text2_:
    crypt += hex(int(n, 2))[2:] + ' '
return crypt

```

Рис. 2: Реализация сложения по модулю

Для реализации второго пункта реализовала функцию getkey ((рис. 3), в которой нашла ключ преобразующий шифротекст в какой-либо другой((рис. 4).

```

def getkey(cipher, text):
    encoded_bytes = bytearray.fromhex(cipher)
    x = binascii.hexlify(encoded_bytes)
    y = str(x, 'utf-8')
    binary = lambda x: " ".join(reversed(
        [i + j for i, j in zip(*[f["{0:04b}"].format(int(c, 16)) for c in reversed("0" + x)][n::2] for n in [1, 0]])))
    x1 = binary(y)

    text1 = text.encode('utf-8', 'replace')
    x_text = binascii.hexlify(text1)
    y_text = str(x_text, 'utf-8')
    x1_text = binary(y_text)

```

Рис. 3: Представление данных в двоичном виде

```

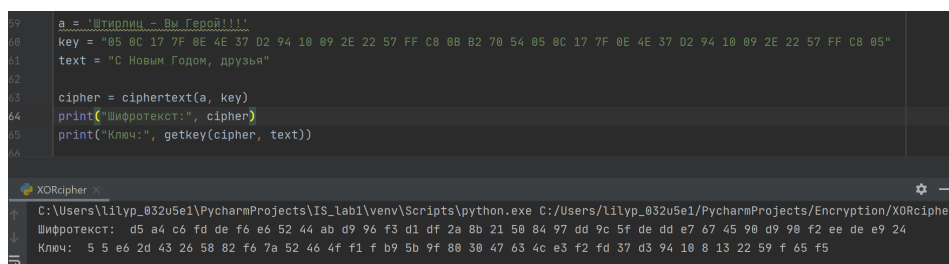
key = ""
for i in range(len(x1)):
    if x1[i] == ' ':
        key += ' '
    elif x1[i] == x1_text[i]:
        key += '0'
    elif x1[i] != x1_text[i]:
        key += '1'

key_ = key.split()
pos_key = ' '
for n in key_:
    pos_key += hex(int(n, 2))[2:] + ' '
return pos_key

```

Рис. 4: Получение ключа

Подобрала ключ, чтобы получить сообщение «С Новым Годом, друзья». ((рис. 5)



```

59 a = 'Штирлиц - Вы Герой!!!'
60 key = "05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 05"
61 text = "С Новым Годом, друзья"
62
63 cipher = XORCipher(a, key)
64 print("Шифротекст:", cipher)
65 print("Ключ:", getkey(cipher, text))
66
XORcipher
C:\Users\lilyp_032u5e1\PycharmProjects\IS_lab1\venv\Scripts\python.exe C:/Users/lilyp_032u5e1/PycharmProjects/Encryption/XORcipher
Шифротекст: d5 a4 c6 fd de f6 e6 52 44 ab d9 96 f3 d1 df 2a 8b 21 50 84 97 dd 9c 5f de dd e7 67 45 90 d9 90 f2 ee de e9 24
Ключ: 5 5 e6 2d 43 26 58 82 f6 7a 52 46 4f f1 f b9 5b 9f 80 30 47 63 4c e3 f2 fd 37 d3 94 10 8 13 22 59 f 65 f5

```

Рис. 5: Ключ сообщения

Также произвела проверку работы функции getkey ((рис. 6)


```
59 a = "Штирлиц - Вы Герой!!!"
60 key = "05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 05"
61 text = "С Новым Годом, друзья"
62
63 cipher = cipherText(a, key)
64 print("Шифротекст:", cipher)
65 print("Ключ:", getKey(cipher, a))
66
```

XORcipher X

C:\Users\lilyp_032u5e1\PycharmProjects\IS_Lab1\venv\Scripts\python.exe C:/Users/lilyp_032u5e1/PycharmProjects/Encryption/XORcipher

Шифротекст: d5 a4 c6 fd de f6 e6 52 44 ab d9 96 f3 d1 df 2a 8b 21 50 84 97 dd 9c 5f de dd e7 67 45 90 d9 90 f2 ee de e9 24

Ключ: 5 c 17 7f e 4e 37 d2 94 10 9 2e 22 57 ff c8 b b2 70 54 5 c 17 7f e 4e 37 d2 94 10 9 2e 22 57 ff c8 5

Рис. 6: Получение изначального ключа

Выводы

Освоить на практике применение режима однократного гаммирования.

Список литературы

1. Основы безопасности информационных систем : Учеб. пособие для студентов вузов, обучающихся по специальностям “Компьютер. безопасность” и “Комплекс. обеспечение информ. безопасности автоматизир. систем” / Д.А. Зегжда, А.М. Ивашко. - М. : Горячая линия - Телеком, 2000. - 449, [2] с. : ил., табл.; 21 см.; ISBN 5-93517-018-3.