Лабораторная работа №8

Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

Монастырская Кристина Владимировна

Содержание

Цель работы 4

Теоретическое введение 5

Выполнение лабораторной работы 6

Выводы 10

Список литературы 11

Список иллюстраций

1 Шифровка текстов при известном ключе 7
2 Дешифровка сообщений без ключа 8
3 Вывод 9

Цель работы

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

Два текста кодируются одним ключом (однократное гаммирование).

P1 = НаВашисходящийот1204

P2 = ВСеверныйфилиалБанка

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, а также должно дешифровать один из исходных текстов без использования ключа.

1. Реализовала функцию шифровки двух текстов сгенерированным в прошлом пункте ключом (рис. 1). Использовала функцию, написанную в предыдущей работе и отредактировала её.

```python
def encryption(P, K):
    hex_P = P.encode('utf-8').hex()
    bin_P = toBinary(hex_P)
    hex_K = K.replace(' ', '').lower()
    bin_K = toBinary(hex_K)

    bin_C = ""
    if len(bin_P) == len(bin_K):
        for i in range(len(bin_P)):
            if bin_P[i] == ' ':
                bin_C += ' '
            elif bin_P[i] == bin_K[i]:
                bin_C += '0'
            elif bin_P[i] != bin_K[i]:
                bin_C += '1'
    bin_C = bin_C.split()
    C = ''
    for i in range(len(bin_C)):
        x = int(bin_C[i], 2)
        if len(hex(x)[2:]) < 2:
            C += '0'
        C += hex(x)[2:]
        C += ' '

    C = C.upper()
    return C

```

Рис. 1: Шифровка текстов при известном ключе

2. Реализовала функцию дешифровки сообщений без знания ключа(рис. 2). Функция совершает преобразование P1, C1 и C2 в двоичный код, после чего в P2 записывается результат побитовой операции сложения по модулю

2 с тремя элементами C1, C2, P1. Полученный битовый P2 переводится в шестнадцатиричную систему, а после в символьный текст.

```
def decryption(C1, C2, P1):
    hex_C1 = C1.replace(' ', '').lower()
    bin_C1 = toBinary(hex_C1)
    hex_C2 = C2.replace(' ', '').lower()
    bin_C2 = toBinary(hex_C2)
    hex_P1 = P1.encode('utf-8').hex()
    bin_P1 = toBinary(hex_P1)
    bin_P2 = ''
    for i in range(len(bin_P1)):
        if bin_P1[i] == ' ':
            bin_P2 += ' '
        else:
            bin_P2 += str(int(bin_C1[i]) ^ int(bin_C2[i]) ^ int(bin_P1[i]))

    bin_P2 = bin_P2.split()
    P2 = ''
    for i in range(len(bin_P2)):
        x = int(bin_P2[i], 2)
        if len(hex(x)[2:]) < 2:
            P2 += '0'
        P2 += hex(x)[2:]
        P2 += ''
    P2 = bytes.fromhex(P2).decode("ASCII")
    return P2
```

Рис. 2: Дешифровка сообщений без ключа

3. Итог:


venv library root
main.py
External Libraries
Scratches and Consoles
P1 = 'NaVahishodahiiot1204'
P2 = 'VSevernyifilialBanka'
key = "05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54"
print("Выражение 1:", P1)
print("Выражение 2:", P2)
print("Ключ:", key)
print()
C1 = encryption(P1, key)
C2 = encryption(P2, key)
print("Шифротекст 1:", C1)
print("Шифротекст 2:", C2)
print()
print("Расшифрованное выражение 2:", decryption(C1, C2, P1))
main
C:\Users\krist\PycharmProjects\infosec\venv\Scripts\python.exe C:/Users/krist/PycharmProjects/infosec/ma
Выражение 1: NaVahishodahiiot1204
Выражение 2: VSevernyifilialBanka
Ключ: 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54
Шифротекст 1: 4B 6D 41 1E 66 27 44 BA FB 74 68 46 4B 3E 90 BC 3A 80 40 60
Шифротекст 2: 53 5F 72 09 6B 3C 59 AB FD 76 60 42 4B 36 93 8A 6A DC 1B 35
Расшифрованное выражение 2: VSevernyifilialBanka

Рис. 3: Вывод

Выводы

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

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.