Отчет по лабораторной работе №8

-

Овениязов Артур

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Задание

Исходные данные. Две телеграммы Центра: P1 = НаВашисходящийот1204 P2 = ВСеверныйфилиалБанка Ключ Центра длиной 20 байт: K = 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 OB B2 70 54 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

3 Теоретическое введение

Гамми́рование, или Шифр XOR, — 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. [1]

4 Выполнение лабораторной работы

Написал следующий код в файле kripto2.py.

```
Ввод [2]:
import random as rnd
import string as strg

abc = strg.ascii_letters.join(strg.digits)
abc

Out[2]: 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ2abcdefghijklmnopqrs
tuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ3abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ4abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMN
OPQRSTUVWXYZ5abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ6abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ7abcdefgh
ijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ8abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ9'

Ввод [10]:
p1 = "НаВашисходящийот1204"
p2 = "ВСеверныйфилиалБанка"

k = "".join(rnd.choice(abc) for i in range(len(p1)))
def xor_string(data, key):
    return "".join(chr(ord(x)^ord(y)) for x, y in zip(data, key))
c1 = xor_string(p1, k)

bytes(c1, "UTF-8").hex()

Out[10]: 'd19ed1b8d1a3d189d0a6d1b4d08dd0b7d1a9d196d097d0bfd1bdd19bd1bbd08a5444416c'

Ввод [11]:
c2 = xor_string(p2, k)
bytes(c2, "UTF-8").hex()

Out[11]: 'd191d1a9d184d18bd19bd08cd1b1d0b9d1aed0a6d1a0d18dd1bdd192d1bed199d195d18bd18bd1a8'

Ввод [12]:
p1_xor_p2 = xor_string(p1, p2)

bytes(p1_xor_p2, "UTF-8").hex()

Out[12]: '0f1127027d787c0e07707772000905 53d081d08fd08ad084'

Ввод [13]:
p2_found = xor_string(p1_xor_p2, p1)
p2_found

Out[13]: 'ВСеверныйфилиалБанка'

Ввод [14]:
p2_found == p2

Out[14]: True

Ввод [15]:
p1_found = xor_string(p1_xor_p2, p2_found)
p1_found
```

Рис. 4.1: Код kripto2.ipynb

```
Out[10]: 'd19ed1b8d1a3d189d0a6d1b4d08dd0b7d1a9d196d097d0bfd1bdd19bd1bbd08a5444416c'

Ввод [11]:
c2 = xor_string(p2, k)
bytes(c2, "UTF-8").hex()

Out[11]: 'd191d1a9d184d18bd19bd08cd1b1d0b9d1aed0a6d1a0d18dd1bdd192d1bed199d195d18bd18bd1a8'

Ввод [12]:
p1_xor_p2 = xor_string(p1, p2)

bytes(p1_xor_p2, "UTF-8").hex()

Out[12]: '0f1127027d787c0e07707772000905 53d081d08fd08ad084'

Ввод [13]:
p2_found = xor_string(p1_xor_p2, p1)
p2_found

Out[13]: 'ВСеверныйфилиалБанка'

Ввод [14]:
p2_found == p2

Out[14]: True

Ввод [15]:
p1_found = xor_string(p1_xor_p2, p2_found)
p1_found

Out[15]: 'НаВашисходящийот1204'

Ввод [16]:
p1_found == p1

Out[16]: True
```

Рис. 4.2: Результат выполнения kripto2.py

4.1. Ответы на вопросы

1. Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа?

С помощью формул режима однократного гаммирования получим шифротексты обеих телеграмм:

C1 = P1 xor K,

C2 = P2 xor K.

Задача нахождения открытого текста по известному шифротексту двух телеграмм, зашифрованных одним ключом, может быть решена. Сложим по модулю 2 оба равенства, получаем:

C1 xor C2 = P1 xor K xor P2 xor K = P1 xor P2.

имеем:

C1 xor C2 xor P1 = P1 xor P2 xor P1 = P2.

Таким образом, получаем возможность определить те символы сообщения P2, которые находятся на позициях известного сообщения P1. Догадываясь по логике сообщения P2, Имеем реальный шанс узнать ещё некоторое количество символов сообщения P2. Затем вместо P1 подставляя новоузнанные символы сообщения P2. И так далее. Действуя подобным образом, можно если даже не прочитает оба сообщения, то значительно уменьшит пространство их поиска.

2. Что будет при повторном использовании ключа при шифровании текста?

Если на сообщение наложить ключ дважды, мы получим исходное сообщение.

3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?

Один ключ накладываем на оба открытых текста и получаем два зашифрованных одним ключом шифротекста.

4. Перечислите недостатки шифрования одним ключом двух открытых текстов.

Зная текст одного из сообщения можно узнать текст второго, не зная кода.

5. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Нет необходимости каждый раз придумывать ключи для каждого сообщение.

5 Выводы

В ходе данной лабораторной работы я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом, разработал приложение, позволяющие шифровать и дешифровать различные тексты в режиме однократного гаммирования.

Список литературы

1. Дискреционное разграничение доступа Linux [Электронный ресурс]. Сайт, 2021. URL: http://debianinstall.ru/diskretsionnoe-razgranichenie-dostupa-linux/.