Отчёт по лабораторной работе №8

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

Аминов Зулфикор Мирзокаримович

Содержание

# 1 Цель работы

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

# 2 Указание к работе

Исходные данные.

Две телеграммы Центра:

P1 = НаВашисходящийот1204

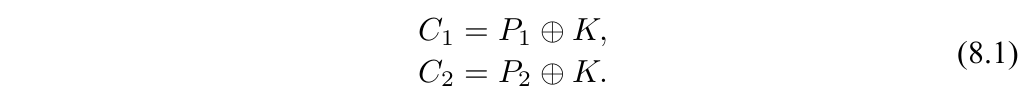
P2 = ВСеверныйфилиалБанка

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

Режим шифрования однократного гаммирования одним ключом двух видов открытого текста реализуется в соответствии со схемой, приведён- ной на рис. 8.1.

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:



Открытый текст можно найти в соответствии с (8.1), зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства (8.1)

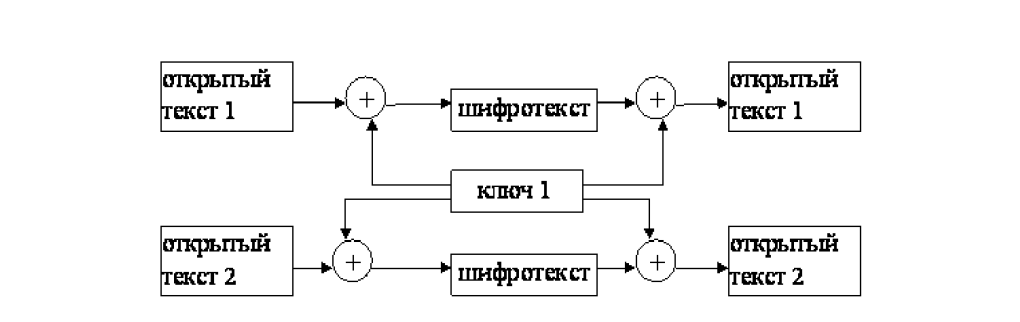
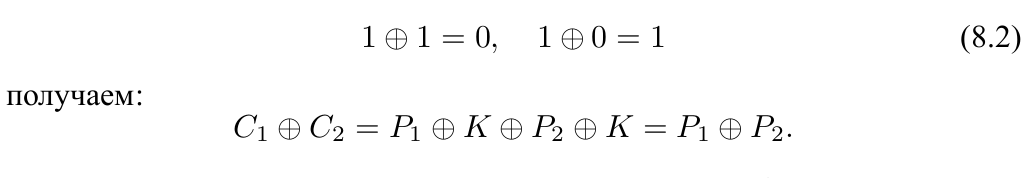
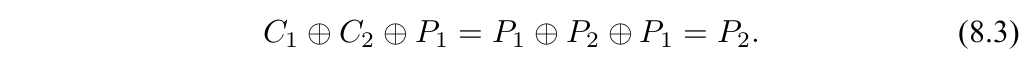


Figure 1: Общая схема шифрования двух различных текстов одним ключом

складываются по модулю 2. Тогда с учётом свойства операции XOR



Предположим, что одна из телеграмм является шаблоном — т.е. име- ет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар C1 ⊕ C2 (известен вид обеих шифровок). Тогда зная P1 и учитывая (8.2), имеем:



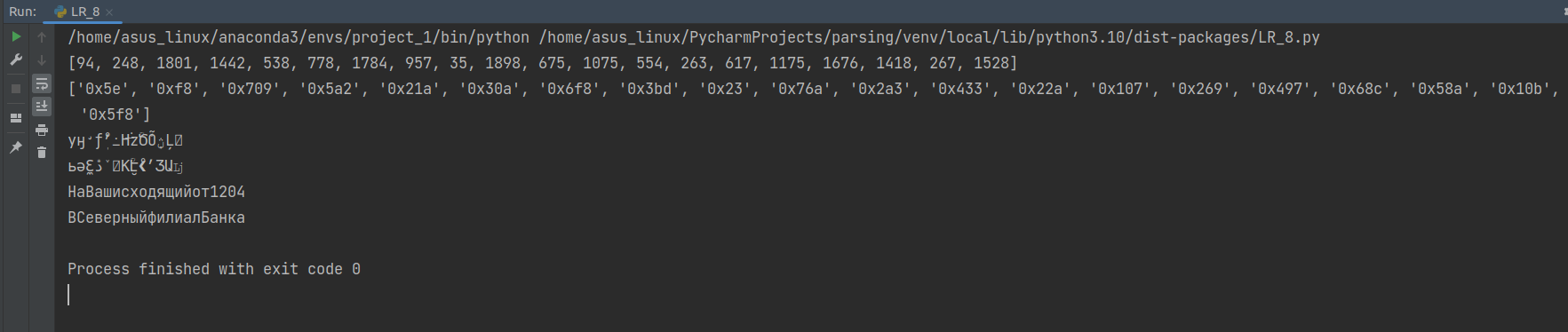
Таким образом, злоумышленник получает возможность определить те символы сообщения P2 , которые находятся на позициях известного шаб- лона сообщения P1 . В соответствии с логикой сообщения P2 , злоумышлен- ник имеет реальный шанс узнать ещё некоторое количество символов сооб- щения P2 . Затем вновь используется (8.3) с подстановкой вместо P1 полу- ченных на предыдущем шаге новых символов сообщения P2 . И так далее. Действуя подобным образом, злоумышленник даже если не прочитает оба сообщения, то значительно уменьшит пространство их поиска.

# 3 Выполнение работы

## 3.1 Программа на python

import random  
  
P\_1 = "НаВашисходящийот1204"  
P\_2 = "ВСеверныйфилиалБанка"  
  
def key(text):  
 K = []  
 for i in range(len(text)):  
 K.append(random.randint(0, 2000))  
 return K  
  
def hex\_key(key):  
 h\_k = []  
 for i in key:  
 h\_k.append(hex(i))  
 return h\_k  
  
def encod\_and\_decod(text, key):  
 c = ""  
 for i in range(len(text)):  
 c += chr(ord(text[i]) ^ key[i])  
 return c  
  
k = key(P\_1)  
print(k)  
print(hex\_key(k))  
c\_1 = encod\_and\_decod(P\_1, k)  
c\_2 = encod\_and\_decod(P\_2, k)  
print(c\_1)  
print(c\_2)  
print(encod\_and\_decod(c\_1, k))  
print(encod\_and\_decod(c\_2, k))

## 3.2 Результат работы



# 4 Выводы

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