Лабораторная работа №8: отчет.

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

Коне Сирики. Группа - НФИбд-01-20.

Содержание

# 1 Цель работы

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

# 2 Задание

1. Рассмотреть особенности и особенности кодирование однократного гаммирования с использованием одного ключа.
2. Создать код который будет показывать принцип работы нескольких шифротекстов с одним ключом и его взлом.
3. изучить способы взлома и декодирование шифротекста без ключа.

# 3 Указание к работе

## 3.1 Описание метода

Пример: 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 (смотреть лабораторную).

# 4 Выполнение лабораторной работы

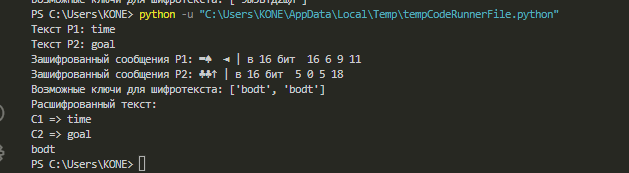
## 4.1 Условие задания

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.

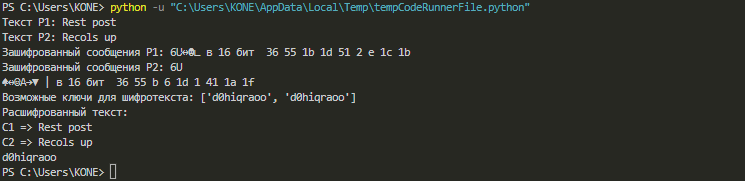
## 4.2 Код

import random  
import string  
  
def keyCreate(s, alf):  
 k = ''.join(random.choice(alf) for i in range(s))  
 return k  
  
def Hex\_coder(cod):  
 return ' '.join(hex(ord(i))[2:] for i in cod)  
  
def string\_coder(text, k, iter\_numb):  
 if iter\_numb == 1:  
 return ''.join(chr(ord(c) ^ ord(k)) for c, k in zip(text, k))  
 else:  
 return [''.join(chr(ord(c) ^ ord(k)) for c, k in zip(t, k)) for t in text]  
  
def find\_Key(cypher, texts, s):  
 possible\_keys = []  
 for f in range(len(texts)):  
 for i in range(len(cypher[f]) - s + 1):  
 key = [chr(ord(c) ^ ord(k)) for c, k in zip(cypher[f][i:i + s], texts[f])]  
 intact\_plaintext = string\_coder(cypher[f], key, 1)  
 if texts[f] in intact\_plaintext:  
 possible\_keys.append(''.join(key))  
 return possible\_keys  
  
  
P1 = input("Текст P1: ")  
P2 = input("Текст P2: ")  
if len(P1) != len(P2):  
 exit(0)  
size, char\_set = len(P1), string.ascii\_lowercase+string.digits  
C1, C2 = string\_coder([P1, P2], keyCreate(size, char\_set), 2)  
  
print(f"Зашифрованный сообщения P1: {C1} | в 16 бит {Hex\_coder(C1)}",  
 f"Зашифрованный сообщения P2: {C2} | в 16 бит {Hex\_coder(C2)}", sep="\n")  
  
possible\_keys = find\_Key([C1, C2], [P1, P2], size)  
print("Возможные ключи для шифротекста:", possible\_keys)  
  
D1, D2 = string\_coder([C1, C2], possible\_keys[-1], 2)  
print("Расшифрованный текст:", f"\nC1 => {D1}\nC2 => {D2}")  
  
print(string\_coder(C1, P1, 1))

## 4.3 Результат



Результат попытки 1



Результат попытка 2

# 5 Контрольные вопросы

1. Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа? Ответ: Это возможно сделать только в том случае если текст P1 и P2 одной длины и имеют общий ключ.
2. Что будет при повторном использовании ключа при шифровании текста? Ответ: Из-за одинаковости способа кодирование и декодирование после повторного использование слова и ключа даст нам шифротекст.
3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов? Ответ: Фактически следуя схеме 8.1 и принципу “шифра XOR” мы просто имеет два параллельных кодирование и декодирование с использованием одного ключа.
4. Перечислите недостатки шифрования одним ключом двух открытых текстов. Ответ: Если вспомнить тредования для абсолютной стойкости шифра расмотренных в предедущей лабораторной то можно сразу понять по первому пункту что если ключ не будет случайным и каждый раз новым для каждой строки то найдя пересечения или аналоги в шифротекстах можно определить одинаковые символы что может пошатнуть защиту текста даже если у вас нет ни одного исходного кода, а если и есть то определить другие слова легко.
5. Перечислите преимущества шифрования одним ключом двух открытых текстов. Ответ: На самом деле они есть, но они сомнительны: требуется передать один ключ что сделать проще и быстрее, при передаче большого количества шифротекста нет шанса запутаться в их порядке сочетания с ключами.

# 6 Выводы

Освоил на практике применение режима однократного гаммирования и возможных способах взлома при отсутствие ключа и наличие исходных текстов и шифротекстов.

# Список литературы

1. [Лабораторная](https://esystem.rudn.ru/pluginfile.php/2090286/mod_resource/content/2/008-lab_crypto-key.pdf)
2. [Описание методов](https://retrolib.ru/realizacziya-algoritma-shifrovaniya-gammirovanie-pascalpaskal-python-piton/)
3. [Другая лабораторная об материале](https://ami.nstu.ru/~gultyaeva/pszi/Materials/lab1.pdf)