Отчёт по лабораторной работе №8

Дисциплина: Основы информационной безопасности

Барсегян Вардан Левонович НПИбд-01-22

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

1. Создаю функцию encrypt(), которая будет шифровать заданный текст с помощью однократного гаммирования. На вход функция получает открытый текст, также можно задать определенный ключ шифрования. Если ключа нет, то он генерируется рандомно. Сначала исходный текст и ключ шифрования преобразуются в 16-ную СС, затем, применяется операция XOR для каждого элемента ключа и текста. Полученный шифротекст декодируется из 16-ной СС и получается набор из символов.

def encrypt(text: str, key: list = None):  
 '''  
 Выводит шифротекст для заданного текста.  
 Если ключа нет, то генерируется случайный ключ  
 '''  
 if not key:  
 key = generate\_key(length=len(text))  
  
 text\_16 = [ord(char) for char in text]   
 key = [ord(el) for el in key]  
  
 print(f"Ключ шифрования:", ' '.join(str(s) for s in key))  
 print(f"Исходный текст:", text)  
  
 encrypted\_text = []  
 for i in range(len(text)):  
 encrypted\_text.append(text\_16[i] ^ key[i])  
  
 ciphertext = ''.join([chr(i) for i in encrypted\_text])  
 print(f'Шифротекст: {ciphertext}\n\n')  
  
 return ciphertext

1. Генерация ключа, если он не задан, происходит в функции generate\_key() из ascii-символов и цифр

def generate\_key(length: int):  
 '''  
 Генерация рандомного ключа длины length  
 '''  
 return random.sample(string.ascii\_letters + string.digits, length)

1. Работа программмы: (рис. 1)

* сначала создается случайный ключ и с этим ключом шифруются тексты p1 и p1 (переменные c1 и c2)
* далее, шифротекст с1 шифруется по ключу c2
* полученный шифротекст c1\_c2 шифруется по ключу открытого текста. в результате, получаем второй открытый текст, ранее неизвестный

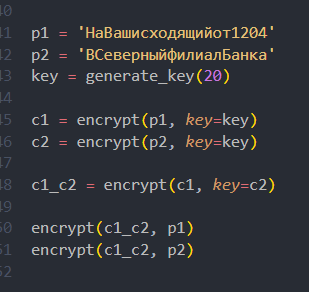


Рис. 1: Работа программы

1. Полный вывод работы программы (рис. 2)

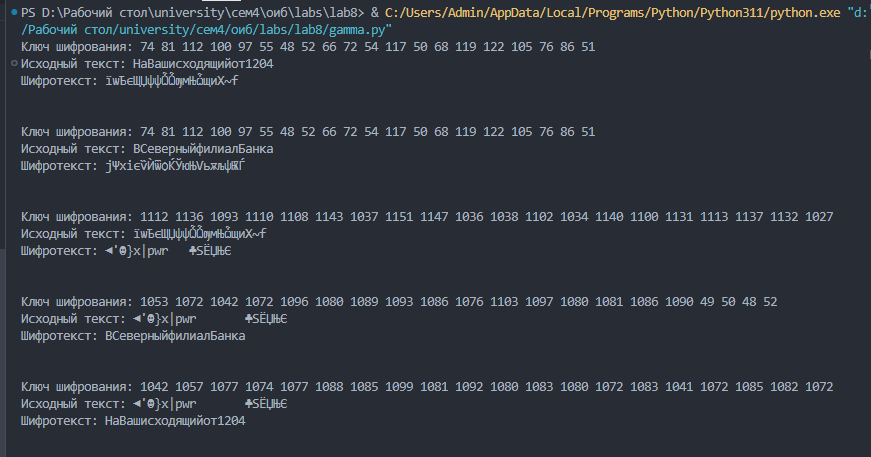


Рис. 2: Работа программы

# 3 Контрольные вопросы

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

Нужно применить XOR для двух шифротекстов, а к полученному результату применить XOR с ключом, равным известному открытому тексту. Тогда результатом будет второй открытый текст

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

Шифрование будет небезопасным, т.к. с помощью шифротекстов и одного открытого текста можно дешифровать другой текст

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

Каждый текст шифруется однократным гаммированием отдельно с использованием этого ключа

1. Перечислите недостатки шифрования одним ключом двух открытых текстов

Главный недостаток - можно дешифровать открытый текст без знания ключа

1. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Т.к. ключей используется меньше, то тратится меньше памяти на хранение и передачу ключей

# 4 Выводы

Я применил режим однократного гаммирования на примере кодирования различных исходных текстов одним ключом