**Отчет по лабораторной работе №8**

**Основы информационной безопасности**

Авдадаев Джамал. Группа: НКАбд-01-23

Содержание

[1 Цель работы 1](#_Toc5316)

[2 Задание 1](#_Toc5317)

[3 Теоретическое введение 1](#_Toc5318)

[4 Выполнение лабораторной работы 2](#_Toc5319)

[5 Ответы на контрольные вопросы ........................................................................................................... 3](#_Toc5320)

[6 Выводы 5](#_Toc5321)

[Список литературы 5](#_Toc5322)

# Цель работы

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

# Задание

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.

# Теоретическое введение

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

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

𝐶1 = 𝑃1 ⊕ 𝐾,

𝐶2 = P\_2 ⊕ K$. (8.1)

Открытый текст можно найти, зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства (8.1) складываются по модулю 2. Тогда с учётом свойства операции XOR

1 ⊕ 1 = 0,1 ⊕ 0 = 1(8.2)

получаем:

𝐶1 ⊕ 𝐶2 = 𝑃1 ⊕ 𝐾 ⊕ 𝑃2 ⊕ 𝐾 = 𝑃1 ⊕ 𝑃2.

Предположим, что одна из телеграмм является шаблоном — т.е. имеет 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:

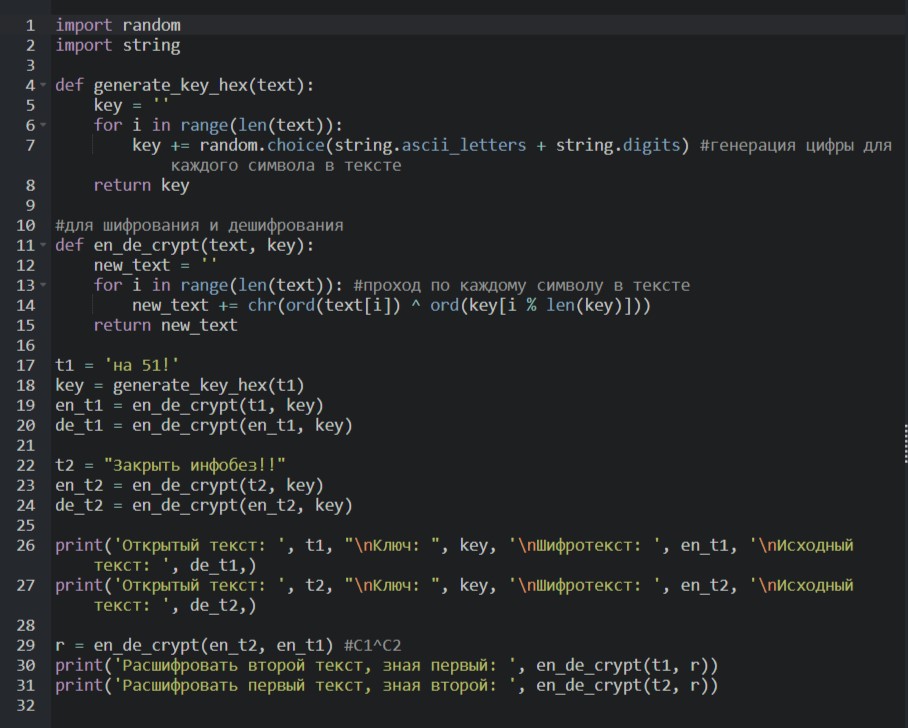
𝐶1 ⊕ 𝐶2 ⊕ 𝑃1 = 𝑃1 ⊕ 𝑃2 ⊕ 𝑃1 = 𝑃2. (8.3)

Таким образом, злоумышленник получает возможность определить те символы сообщения 𝑃2, которые находятся на позициях известного шаблона сообщения 𝑃1. В соответствии с логикой сообщения 𝑃2, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения 𝑃2. Затем вновь используется (8.3) с подстановкой вместо P1 полученных 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. [1]

# Выполнение лабораторной работы

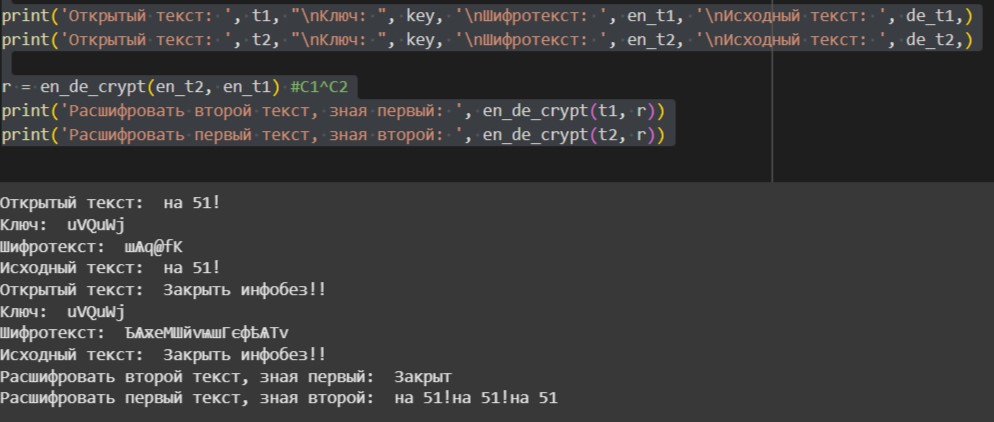
Я выполнил свою лабораторную работу на языке программирования Python, используя функции, реализованные в лабораторной работе №7.

Используя функцию для генерации ключа, генерирую ключ, затем шифрую два разных текста одним и тем же ключом (рис. 1).



## Рис. 1: Шифрование двух текстов

*Расшифровываю оба текста сначала с помощью одного ключа, затем предполагаю, что мне неизвестен ключ, но извествен один из текстов и уже расшифровываю второй, зная шифротексты и первый текст (рис. 2).*



## Рис. 2: Результат работы программы

**Листинг программы 1**

**import random import string def generate\_key\_hex(text):**

**key = '' for i in range(len(text)):**

**key += random.choice(string.ascii\_letters + string.digits) #генерация цифры для каждого символа в тексте return key**

**#для шифрования и дешифрования def en\_de\_crypt(text, key):**

**new\_text = '' for i in range(len(text)): #проход по каждому символу в тексте new\_text += chr(ord(text[i]) ^ ord(key[i % len(key)])) return new\_text**

**t1 = 'на 51!' key = generate\_key\_hex(t1) en\_t1 = en\_de\_crypt(t1, key) de\_t1 = en\_de\_crypt(en\_t1, key)**

**t2 = "Закрыть инфобез!!" en\_t2 = en\_de\_crypt(t2, key) de\_t2 = en\_de\_crypt(en\_t2, key)**

**print('Открытый текст: ', t1, "\nКлюч: ", key, '\nШифротекст: ', en\_t1,**

**'\nИсходный текст: ', de\_t1,)**

**print('Открытый текст: ', t2, "\nКлюч: ", key, '\nШифротекст: ', en\_t2, '\nИсходный текст: ', de\_t2,)**

**r = en\_de\_crypt(en\_t2, en\_t1) #С1^C2 print('Расшифровать второй текст, зная первый: ', en\_de\_crypt(t1, r)) print('Расшифровать первый текст, зная второй: ', en\_de\_crypt(t2, r))**

# Ответы на контрольные вопросы

1. Как, зная один из текстов (𝑃1 или 𝑃2), определить другой, не зная при этом ключа? - Для определения другого текста (𝑃2) можно просто взять зашифрованные тексты 𝐶1 ⊕ 𝐶2, далее применить XOR к ним и к известному тексту: 𝐶1 ⊕ 𝐶2 ⊕ 𝑃1 = 𝑃2.
2. Что будет при повторном использовании ключа при шифровании текста? - При повторном использовании ключа мы получим дешифрованный текст.
3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов? - Режим шифрования однократного гаммирования одним ключом двух открытых текстов осуществляется путем XOR-ирования каждого бита первого текста с соответствующим битом ключа или второго текста.
4. Перечислите недостатки шифрования одним ключом двух открытых текстов - Недостатки шифрования одним ключом двух открытых текстов включают возможность раскрытия ключа или текстов при известном открытом тексте.
5. Перечислите преимущества шифрования одним ключом двух открытых текстов - Преимущества шифрования одним ключом двух открытых текстов включают использование одного ключа для зашифрования нескольких сообщений без необходимости создания нового ключа и выделения на него памяти.

# Выводы

В ходе лабораторной работы я освоил навыки применения режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

# Список литературы

1. Кулябов Д. С. Г.М.Н. Королькова А. В. Лабораторная работа № 8. Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом [Электронный ресурс]. 2023. URL: [https://esystem.rudn.ru/pluginfile.php/2293724/mod\_resource/content/2/008-lab\_cryptokey.pdf.](https://esystem.rudn.ru/pluginfile.php/2293724/mod_resource/content/2/008-lab_crypto-key.pdf)