

Лабораторная работа №3

Лабораторная работа №3

Цель работы

Задание

Теоретическое введение

Оборудование

Выполнение лабораторной работы

Выводы

Список литературы

Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом¹

Задание

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 прочесть оба текста, не зная ключа и не стремясь его определить.

Теоретическое введение

Предложенная Г. С. Вернамом так называемая «схема однократного использования (гаммирования)» является простой, но надёжной схемой шифрования данных. Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. [\[1\]](#).

Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования. В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте.[\[2\]](#)

Гаммирование является симметричным алгоритмом. Поскольку двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, шифрование и дешифрование выполняется одной и той же программой [\[3\]](#).

Оборудование

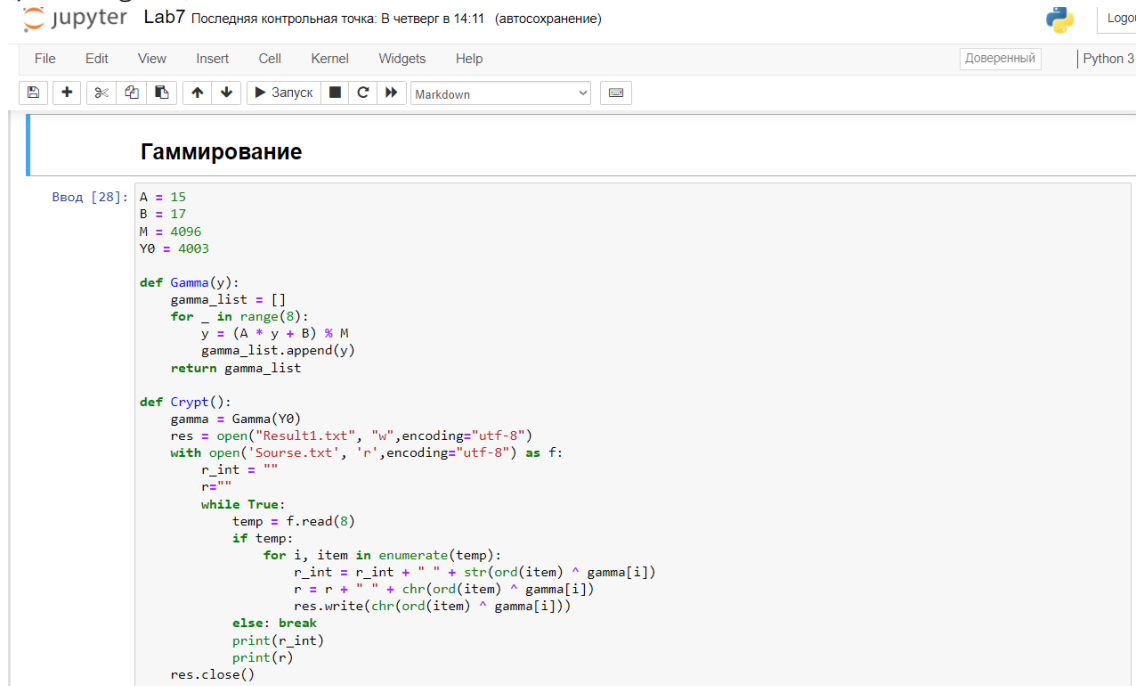
Лабораторная работа выполнялась дома со следующими характеристиками техники:

- Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.81GHz
- ОС Майкрософт Windows 10
- VirtualBox верс. 6.1.26

Выполнение лабораторной работы

1. Разработала приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования.

(рис. -@fig:001)



```
def Gamma(y):
    gamma_list = []
    for _ in range(8):
        y = (A * y + B) % M
        gamma_list.append(y)
    return gamma_list

def Crypt():
    gamma = Gamma(Y0)
    res = open("Result1.txt", "w", encoding="utf-8")
    with open('Source.txt', 'r', encoding="utf-8") as f:
        r_int = ""
        r = ""
        while True:
            temp = f.read(8)
            if temp:
                for i, item in enumerate(temp):
                    r_int = r_int + " " + str(ord(item) ^ gamma[i])
                    r = r + " " + chr(ord(item) ^ gamma[i])
                    res.write(chr(ord(item) ^ gamma[i]))
                else: break
            print(r_int)
            print(r)
        res.close()
```

Приложение написано на python 3. Я запускала его через jupyter Notebook. В данном коде имеется 2 основные функции. 1 - сложение по модулю 2, 2 - представление в байтовом виде.

2. Определим вид шифро-текстов C1 и C2 обоих текстов P1 и P2 при известном ключе. Для того запустим данную часть кода в нашем ноутбуке.

(рис. -@fig:002)

```
def DeCrypt():
    gamma = Gamma(Y0)
    res = open("NewResult.txt", "w", encoding="utf-8")
    with open("Result1.txt", 'r', encoding="utf-8") as f:
        r_int = ""
        r = ""
        while True:
            temp = f.read(8)
            if temp:
                for i, item in enumerate(temp):
                    r_int = r_int + " " + str(ord(item) ^ gamma[i])
                    r = r + " " + chr(ord(item) ^ gamma[i])
                    res.write(chr(ord(item) ^ gamma[i]))
                else: break
            print(r_int)
            print(r)
        res.close()
```

С помощью функции `byte_print()` шифруем оба текста p1 и p2 с помощью одного заранее созданного ключа k. `byte_print()` вызывает `xor_string()`, которая складывает два предложения по модулю 2. Сохраняем полученные зашифрованные тексты в переменные c1 и c2. Далее функция представляет их в буквенном виде и показывает нам.

3. Далее воспользуемся способом, который даст нам разгадать оба зашифрованных текста, без использования нашего ключа.

Следуем дальше этой схеме, представленной в инструкции к лабораторной работе.

(рис. -@fig:003)

```
Ввод [34]: Crypt()
3807 2926 464 3377 885 2191 2749 3677
Г Ì Œ , Ñ S
3807 2926 464 3377 885 2191 2749 3677 3754 2925 466
Г Ì Œ , Ñ S Ñ 9 ö
```

Складываем 2 зашифрованных текста по модулю. Получившийся результат складываем с одним из расшифрованных текстов p_1 . Выводим получившийся результат сложения c_1 , c_2 и p_1 . У нас получается расшифрованный текст p_2 , который мы сохраняем. Теперь уже получившийся расшифрованный текст складываем по модулю 2 с c_1 и c_2 . Теперь же у нас получается расшифрованный p_1 .

(рис. -@fig:004)

```
Ввод [35]: DeCrypt()
1089 1085 1086 1074 1099 1084 1075 1086 1076 1086 1084
СНОВЫМГОДОМ
```

Т.е. мы можем расшифровать любой текст, если у нас имеется два зашифрованных текста одной гаммой, и один расшифрованный текст. Т. е. нам даже не нужно знать шифровальный ключ для данных операций.

Выводы

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

Список литературы

1. Шифры замены и табличного гаммирования // Хабр URL: <https://habr.com/ru/post/583616/> (дата обращения: 10.12.2021).
2. Лабораторная работа № 7. Элементы криптографии. Однократное гаммирование // Туис URL: https://esystem.rudn.ru/pluginfile.php/1198312/mod_resource/content/2/007-lab_crypt_o-gamma.pdf (дата обращения: 9.12.2021).
3. Простейшие методы шифрования с закрытым ключом // НОУ ИНТУТ URL: <https://intuit.ru/studies/courses/691/547/lecture/12373?page=4> (дата обращения: 9.12.2021).