Правительство Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

(НИУ ВШЭ)

Московский институт электроники и математики им. А.Н. Тихонова

ОТЧЕТ

О ПРАКТИЧЕСКОЙ РАБОТЕ № 3

по дисциплине «Криптографические методы защиты информации»

ТЕМА РАБОТЫ

***Подстановочные шифры***

|  |
| --- |
| Студент гр. МКБ231  Е.В. Шараев  «27» мая 2024 г. |
| Руководитель  Заведующий кафедрой информационной безопасности киберфизических систем  канд. техн. наук, доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_О.О. Евсютин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г. |

Москва 2024

**СОДЕРЖАНИЕ**

[1 Задание на практическую работу 3](#_Toc155608317)

[2 Краткая теоретическая часть 4](#_Toc155608318)

3[Описание программной реализации 4](#_Toc155608319)

3.1 Шифр простой замены ……………….………………………………………………...5

3.2 Аффинный шифр ...………………….………………………………………………….7

3.3 Аффинный рекуррентный шифр …………………………….....………………….…..9

4 Примеры зашифрования / расшифрования …………………………………………….…..11

5 Криптоанализ подстановочных шифров ………………………….…………….………….11

6[Выводы о проделанной работе](#_Toc155608320) 11

7[Приложение 1](#_Toc155608321) 11

8[Приложение 2](#_Toc155608321) 12

9[Приложение 3](#_Toc155608321) 19

# 1 Задание на практическую работу

Целью данной работы является приобретение навыков программной реализации и ручного шифрования подстановочных шифров.

В рамках практической работы необходимо выполнить следующее:

1. написать программную реализацию следующих шифров:

- шифр простой замены

- аффинный шифр

- аффинный рекуррентный шифр

1. изучить методы криптоанализа моноалфавитных подстановочных шифров с использованием дополнительных источников
2. провести криптоанализ данных шифров;
3. подготовить отчет о выполненной работе

Аффинный шифр и аффинный рекуррентный шифр должны быть реализованы одним из двух рассмотренных выше способов:

1. в кольце классов вычетов ;
2. в поле Галуа .

Студент самостоятельно выбирает способ реализации.

Программа должна обладать следующей функциональностью для каждого из реализованных в ней шифров:

* принимать на вход произвольную последовательность символов, вводимую пользователем в качестве открытого текста или шифртекста;
* принимать на вход секретный ключ вида, соответствующего конкретному шифру;
* осуществлять зашифрование или расшифрование введенного текста по выбору пользователя.

Отчет должен содержать следующие составные части:

* раздел с заданием;
* раздел с краткой теоретической частью;
* раздел с двумя-тремя примерами «ручного» шифрования для произвольных последовательностей символов;
* раздел с результатами работы программы для тех же последовательностей символов, что и в предыдущем разделе;
* раздел с примерами криптоанализа реализованных шифров;
* раздел с выводами о проделанной работе.

# **2** **Краткая теоретическая часть**

**2.1 Шифр простой замены**

Шифр простой замены, простой подстановочный шифр, моноалфавитный шифр — класс методов [шифрования](https://ru.wikipedia.org/wiki/Шифр), которые сводятся к созданию по определённому алгоритму таблицы шифрования, в которой для каждой буквы открытого текста существует единственная сопоставленная ей буква шифр-текста. Само шифрование заключается в замене букв согласно таблице. Для расшифровки достаточно иметь ту же таблицу, либо знать [алгоритм](https://ru.wikipedia.org/wiki/Алгоритм), по которому она генерируется.

К шифрам простой замены относятся многие способы шифрования, возникшие в древности или средневековье, как, например, Атбаш (также читается как этбаш) или [шифр Цезаря](https://ru.wikipedia.org/wiki/Шифр_Цезаря). Для вскрытия подобных шифров используется [частотный криптоанализ](https://ru.wikipedia.org/wiki/Частотный_криптоанализ).

**2.2 Аффинный шифр**

В аффинном шифре каждой букве алфавита размера m ставится в соответствие число из диапазона 0… m-1. Затем при помощи специальной формулы, вычисляется новое число, которое заменит старое в шифртексте. где x — номер шифруемой буквы в алфавите; m — размер алфавита; a, b — ключ шифрования. Этот шифр, также как и шифр простой замены поддается простому частотному криптоанализу.

**2.3 Аффинный рекуррентный шифр**

Аффинный рекуррентный шифр похож на простой Аффинный, но в рекуррентном шифре для каждой буквы, начиная с третьей, ключи составляются новые. Новые ключи рассчитываются по формуле:

Таким образом для зашифрования каждого символа в аффинном рекуррентном шифре используется свой уникальный ключ. По этой причине данный шифр взлому частотным анализом не поддается так как один и тот же символ может быть зашифрован разными символами шифртекста и наоборот, два разных символа могут быть зашифрованы одним и тем же.

#### **3. Описание программной реализации**

Все файлы данной практической работы я опубликовал в своем гитхаб репозиторииhttps://github.com/Djoongaar/simple\_cipher

Программная реализация целиком находится в файле https://github.com/Djoongaar/simple\_cipher/blob/master/simple\_cipher.py

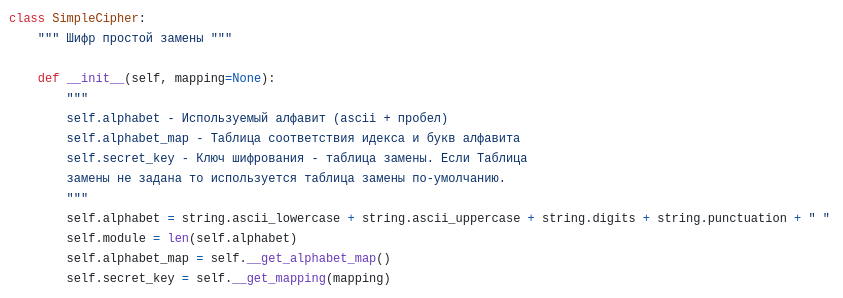
Максимально подробные комментарии к коду я постарался оставить в самих блоках кода, а здесь лишь описание основных методов класса.

Демонстрационный юпитер ноутбук (пдф файл этого ноутбука приложен к данному отчету) <https://github.com/Djoongaar/simple_cipher/blob/master/demo.ipynb>

Юпитер ноутбук с примерами криптоанализа (пдф файл этого ноутбука приложен к данному отчету)https://github.com/Djoongaar/simple\_cipher/blob/master/decrypt.ipynb

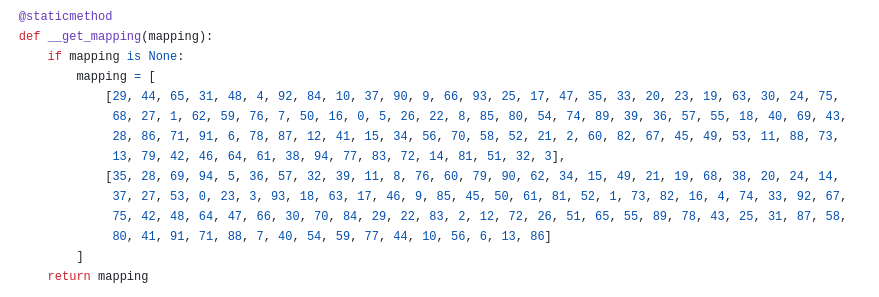
#### **3.1 Программная реализации шифра простой замены**

Шифр простой замены реализован в классе SimpleCipher в файле simple\_cipher.py

 В качестве алфавита шифрования (self.alphabet) был установлен алфавит ascii в нижнем регистре, верхнем регистре, цифры, знаки препинания и пробел.

Модулем алгоритма (self.module) взят размер алфавита.

В качестве ключа шфирования принята таблица замены (self.alphabet\_map) - два массива (прямой и инверсионный), которые зашифровывают порядковый номер символа открытого текста в шифртекст. Класс подразумевает также возможность передачи ему на вход своей таблицы замен. Размерность такой таблицы должна при этом соответствовать модулю алгоритма (self.module).



Зашифрования и расшифрование описано в методах encrypt и decrypt. По сути они просто проверяют порядковый номер элемента шифртекста или открытого текста и возвращают номер элемента в который он должен быть трансформирован. Затем склеивают все символы в одну строку и возвращают её.



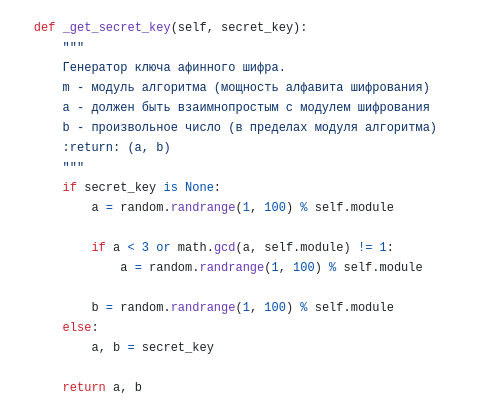
\*

#### **3.2 Программная реализации аффинного шифра**

Аффинный шифр реализован в классе AffineCipher, который наследуется от класса SimpleCipher, однако при инициации переопределяет параметр ключа шифрования self.secret\_key, так как он у Аффинного шифра откличается.

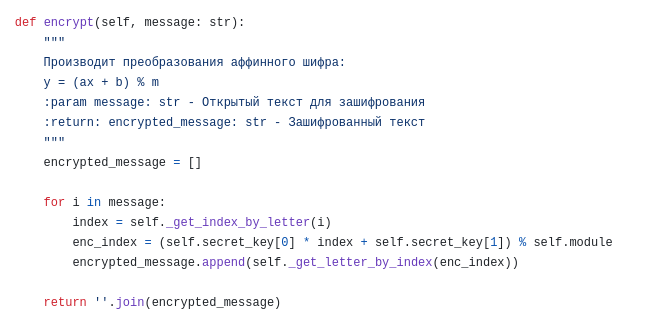


Ключ шифрования в Аффинном шифре должен состоять из пары целочисленных значений (a, b), причем `а` должно быть взаимнопростым с модулем алгоритма , `b` - производльное число. Оба значения должны быть меньше модуля алгоритма.

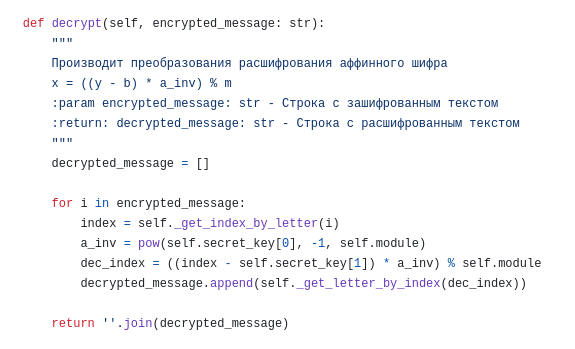


Зашифрование по сути является вычислением порядкового номера элемента шифртекста в алфавите. Порядковый номер вычисляется по формуле:

Затем все элементы шифртекста конкатенируются в одну строку и возвращаеются в виде результата.



Расшифрование представляет собой процедуру обратную зашифрованию. Из порядкового номера элемента шифртекста вычитается `b`, затем результат делится на `a`. Поскольку в полях классов вычетов операция деления не предусмотрена, она заменяется на умножение на обратный элемента `a\_inv`



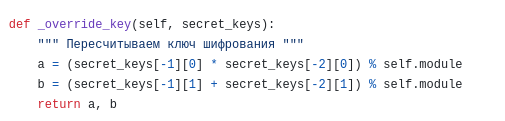
Полученное число — это номер элемента в алфавите. Затем все элементы склеиваются в строку и возвращается расшифрованный текст.

#### **3.3 Программная реализации аффинного рекуррентного шифра**

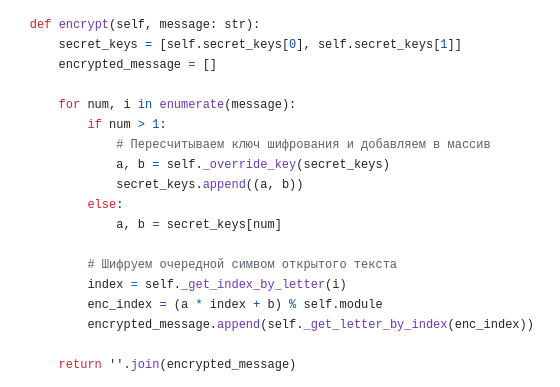
Аффинный рекуррентный шфир релизован в классе AffineRecurrentCipher, унаследованного от класса AffineCipher (аффинный шифр). В рекуррентном аффинном шифре ключь шифрования также представлен парой цифр (a,b), однако ключей шифрования два. Они используются для шифрования первого и второго символов открытогт текста.



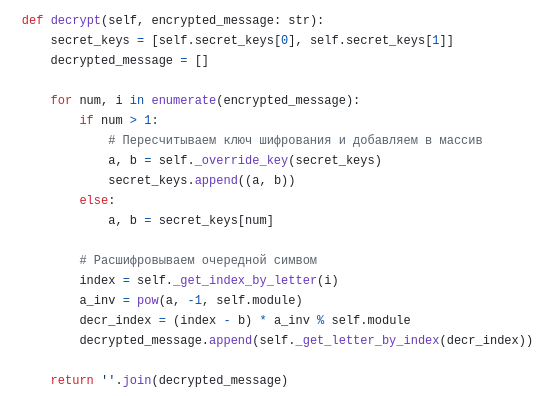
Далее, начиная с третьего элемента, `a` и `b` пересчитываются, и получается для для шифрования каждого элемента открытогт текста расчитываются своя пара ключей. Логика пересчета ключей реализована в методе \_override\_key и она выполняется на каждом шаге шифрования начиная с третьего элемента



Зашифрование реализовано также как и в Аффинном шифре, за исключением генерации нового ключа на каждом шаге начиная с третьего.



Расшифрование в рекуррентном аффинном шифре выполняется так же как и в аффинном шифре, за исключением генерации нового ключа на каждом шаге расшифрования начиная с третьего элемента.



#### **4 Примеры зашифрования / расшифрования**

Примеры зашифрования и расшифрования программным способом и вручную приведены в Приложении 2

**5 Криптоанализ подстановочных шифров**

Примеры дешифрации (криптоанализа) рассматриваемых подстановочных шифров приведены в Приложении 3

**6. Выводы**

Глубоко изучил алгоритмы подстановочных шифров, методы их криптоанализа, а также получил практические навыки по программной реализации таких методов шифрования.

# ПРИЛОЖЕНИЕ 1. Список использованных источников

1. Исходный код использованный программы — URL: <https://github.com/Djoongaar/simple_cipher/blob/master/simple_cipher.py>

2. Исходный код демонстрации работы программы и ручного примера зашифрования / расшифрования — URL: https://github.com/Djoongaar/simple\_cipher/blob/master/demo.ipynb

3. Исходный код примеров криптоанализа шифртекстов шифров замены — URL: https://github.com/Djoongaar/simple\_cipher/blob/master/decrypt.ipynb

4. Статья Википедии о Аффинном шифре — URL: [https://ru.wikipedia.org/wiki/%D0%90%D1%84%D1%84%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9\_%D1%88%D0%B8%D1%84%D1%80](https://ru.wikipedia.org/wiki/Аффинный_шифр)

5. Статья Википедии о Аффинном шифре — URL: https://studbooks.net/1786378/informatika/affinnyy\_rekurrentnyy\_shifr