密码学大作业实验报告

姓名：秦文康 班级：信息安全 1701 学号：201713158021

作业内容

利用古典密码算法的思想，设计一种新密码算法，使用编程语言对上述密码算法进行实现，并利用差分密码分析方法对设计的密码算法进行分析。

古典密码体制的设计与实现

程序设计语言为 Python3，测试用到的环境为 Linux(Ubuntu 18.04 LTS)，需要安装 `numpy` 库（用于矩阵运算）。

> pip3 install numpy

古典密码体制基类

首先实现对文本的古典密码体制，代码文件：`classical.py`，对所有古典密码体制有一个公共基类 `Classical`，此基类中按照密码体制的五元组定义了如下几种方法和成员：

- 成员变量 `__plain` 用于储存明文
- 成员变量 `__cipher` 用于储存密文
- 成员变量 `__key` 用于储存密钥
- 成员函数 `encode` 和 `decode` 分别用于实现对函数的加密和解密
- 成员函数 `get/setPlain/Cipher/Key` 分别用于获取和设置 明文/密文/密钥
- 成员函数 `__init__` 用于对类的初始化

实现的古典密码体制

对基类继承实现了如下几种古典密码体制：

类名	密码体制
Caesar	移位密码体制
Substitution	代换密码体制
Affine	仿射密码体制
Vigenere	维吉尼亚密码体制
Hill	希尔密码体制
Permutation	置换密码体制

对于以上所有密码体制类都对 26 个英语字母有效。

数学部分

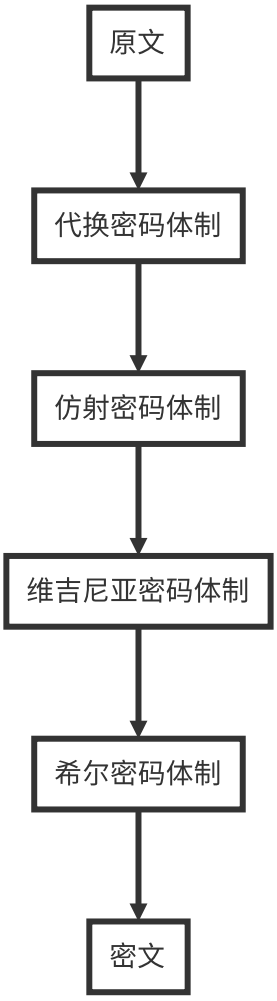
具体实现中所用到的数学操作实现位于文件 `cryptomath.py` 中，具体实现了两个函数，分别为：

- `ext_euclid` 为扩展欧几里德算法（用于仿射密码中求乘法逆元进行解密）
- `inverse_mat` 为求矩阵的逆（用于维吉尼亚密码中求矩阵的逆进行解密）

乘法密码体制的设计与实现

基于上述实现的古典密码学库，通过组合四种古典密码实现了一个乘法密码，实现位于文件 `multiply.py` 中，分别实现了该密码体制的加密函数 `multiply_encode` 和解密函数 `multiply_decode`。

具体实现框图如下：



四种密码体制所用到的密钥分别为：

代码密码所用到的密钥：

a	b	c	d	e	f	g	h	i	j	k	l	m
X	N	Y	A	H	P	O	G	Z	Q	W	B	T

n	o	p	q	r	s	t	u	v	w	x	y	z
S	F	L	R	C	V	M	U	E	K	J	D	I

仿射密码所用到的密钥：

a = 7, b = 3

维吉尼亚密码所用到的密钥：

分组长度为 8：(2, 8, 15, 7, 4, 17, 8, 7)

希尔密码所用到的密钥：

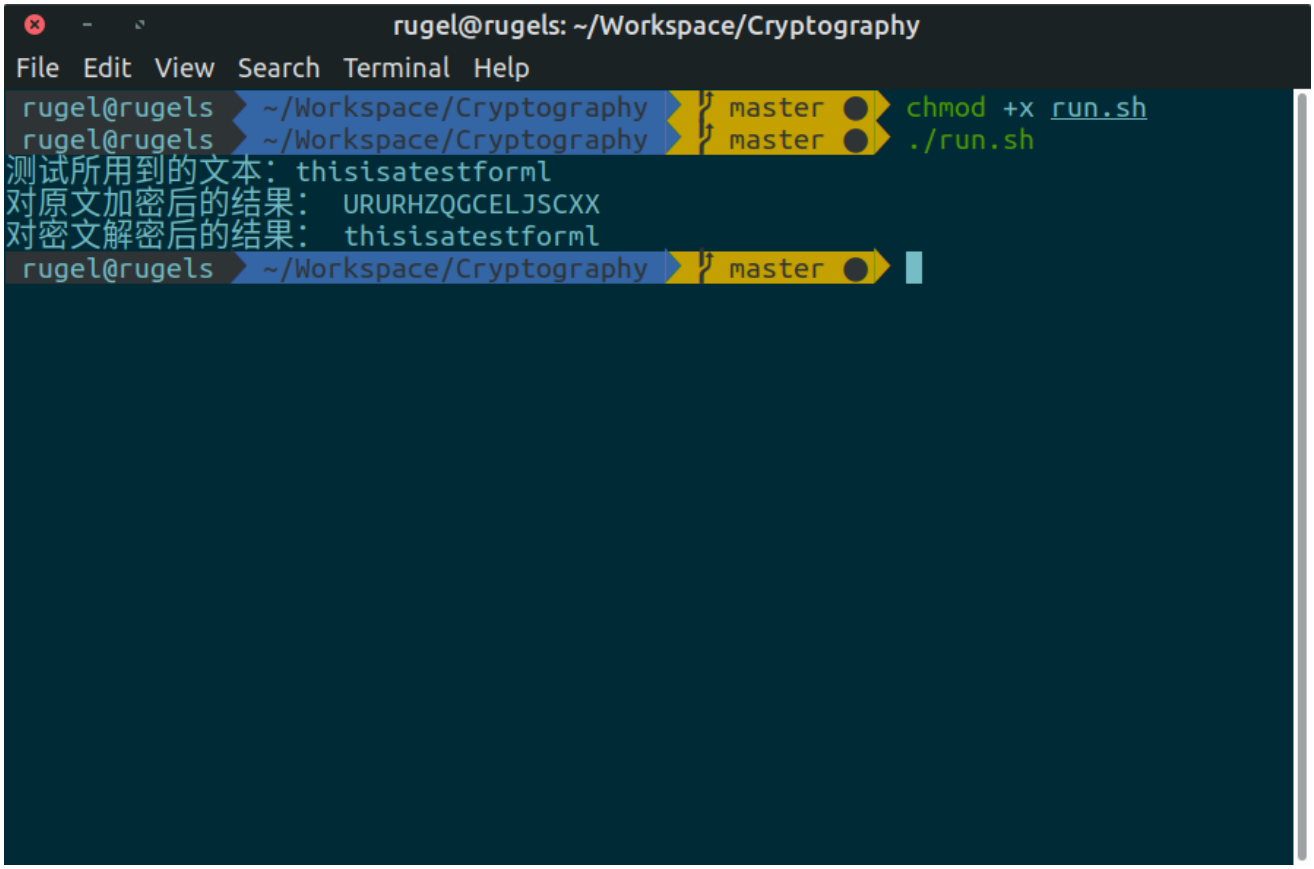
矩阵：$\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$

代码运行及结果

Linux环境下用 Python3 运行 `./src/multiply.py` 或者运行 `run.sh`：

> python3 ./src/multiply.py
>
> or
>
> chmod +x run.sh
>
> ./run.sh

得到的运行结果及截图：


rugel@rugels: ~/Workspace/Cryptography
File Edit View Search Terminal Help
rugel@rugels ~/Workspace/Cryptography master chmod +x run.sh
rugel@rugels ~/Workspace/Cryptography master ./run.sh
测试所用到的文本: thisisatestforml
对原文加密后的结果: URURHZQGCELJSCXX
对密文解密后的结果: thisisatestforml
rugel@rugels ~/Workspace/Cryptography master