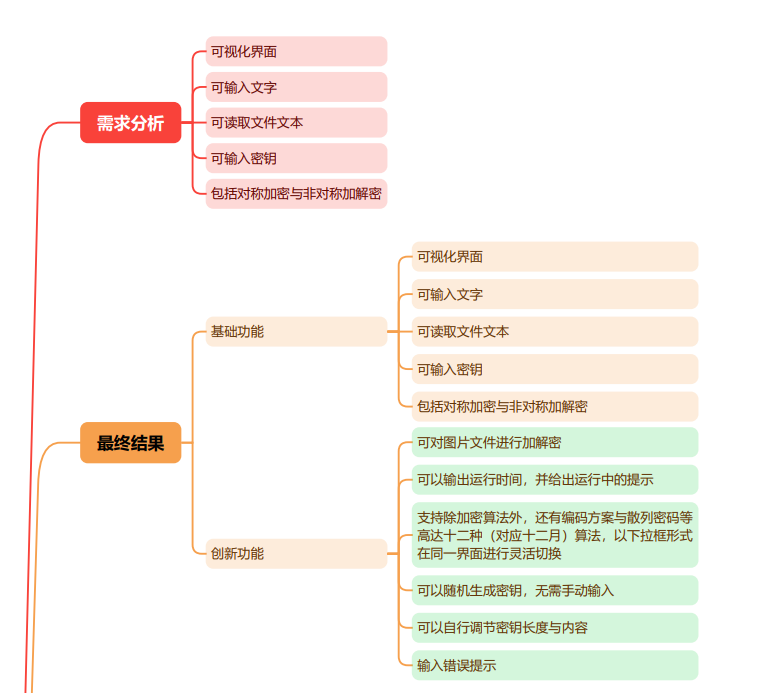
## 第1章 系统功能描述

本项目完成的是一个具有可视化功能的，集成多种密码算法的系统。如图1所示，不仅完成了基本的需求功能，还基于常人的使用习惯与应用场景，作出了更多的创新与优化。



系统功能

## 基础功能

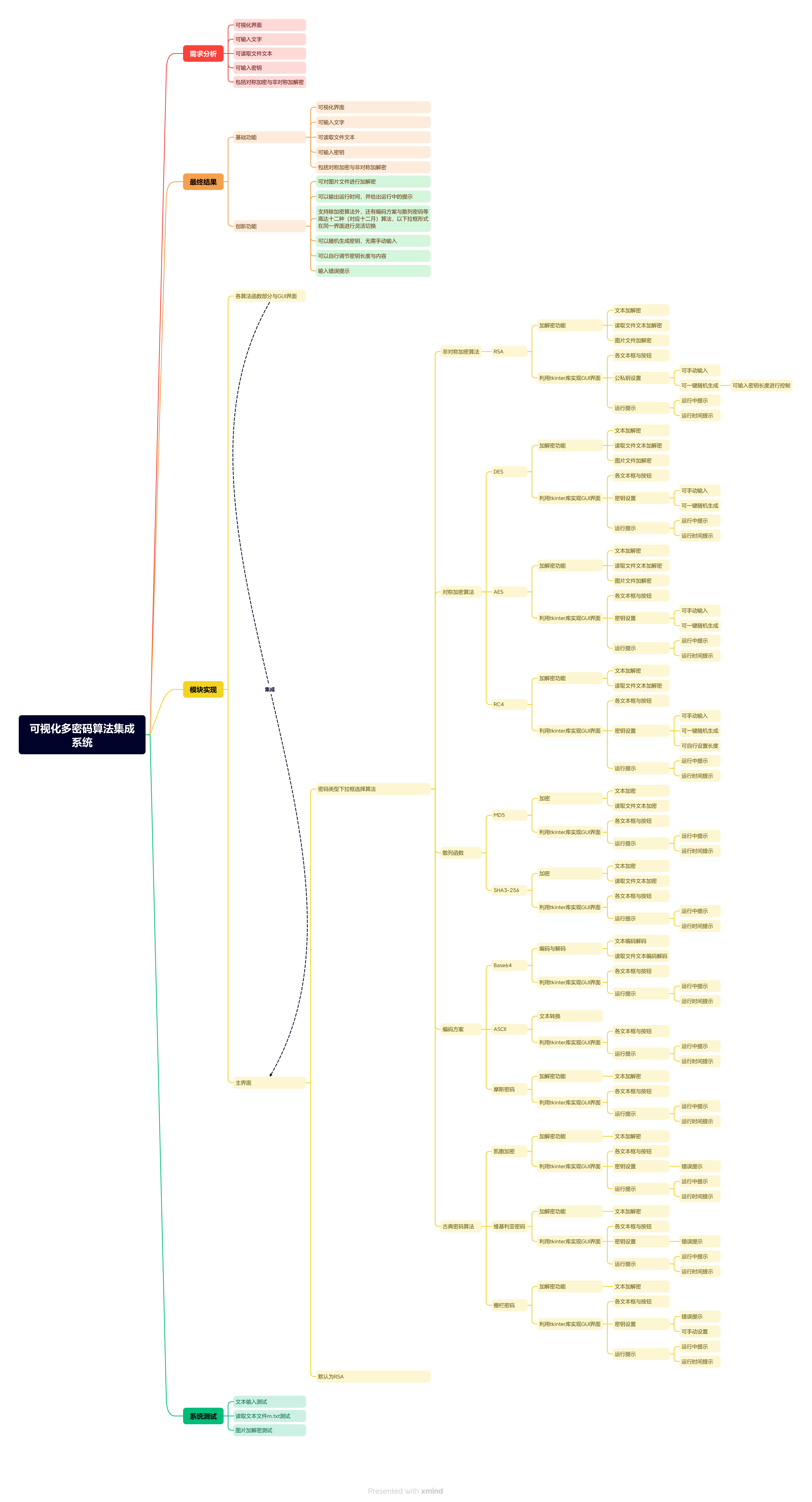
本项目基于需求，实现了用Python tk库编写的可视化界面，能够接受①直接输入文本②读取txt文件文本进行加解密，密钥的输入为用户自行输入，密码算法包括对称加密与非对称加密。

## 创新功能

本项目基于常人的使用习惯与应用场景，对基础功能进行扩展，作出了更多的创新与优化。扩展了加解密对象，增加了运行提示，扩展了支持的算法，优化了用户控制密钥生成，增添了错误提示。

## 第2章 系统设计与实现

一图一表胜千言，系统的详细设计与实现流程图如图2所示，亦可见附录pdf文件。总的来说，通过先写好十二种算法函数代码，然后再通过导入写好的函数代码，逐一通过tk库写好十二种算法各自对应的GUI界面，最后写一个主界面集成十二种GUI界面，通过密码类型下拉框选择并切换GUI界面，默认界面为RSA。最后支持对密钥进行各种控制，增添运行时间提示、错误提示等功能，通过对①直接输入文本②读取txt文件文本进行加解密③对图片文件进行加解密三种方式进行测试，测试通过。

****

系统设计与实现

## 创新功能

本项目基于常人的使用习惯与应用场景，对基础功能进行扩展，作出了更多的创新与优化。

### 加解密对象

除了对用户输入的文本以及读取文件文本进行加密外，本项目还实现了对图片进行加解密。用户可以选择本机上的图片文件对象，进行加密，也可以选择加密后的图片文件进行解密恢复，然后设置保存路径与文件名。

### 运行提示

我们使用系统的过程中，总希望知道该进程是否在运行，以及最后运行了多久。运行过程中若不进行显示，可能导致用户误关闭正在加解密的过程，所以运行提示是必要的。本项目运行中会提示“加/解密中，请稍后……”（虽然实际上该信息呈现时间很短，在非调试的情况下人眼几乎捕捉不了该信息），运行完毕会显示运行时间。

### 扩展支持的算法

本项目实现完毕于十二月，故支持十二种算法。涵盖了对称加密、非对称加密、散列函数、编码方案与古典密码算法。

### 密钥生成

在传统的密码算法系统中，密钥往往要用户自行输入，这还要用户上网查找符合格式的密钥，这是对用户不友好的。本项目支持一键生成随机密钥，并提供输入框支持用户自行输入密钥长度进行控制密钥长度（当然用户也可以手动输入他喜欢的符合格式的密钥）。

### 错误提示

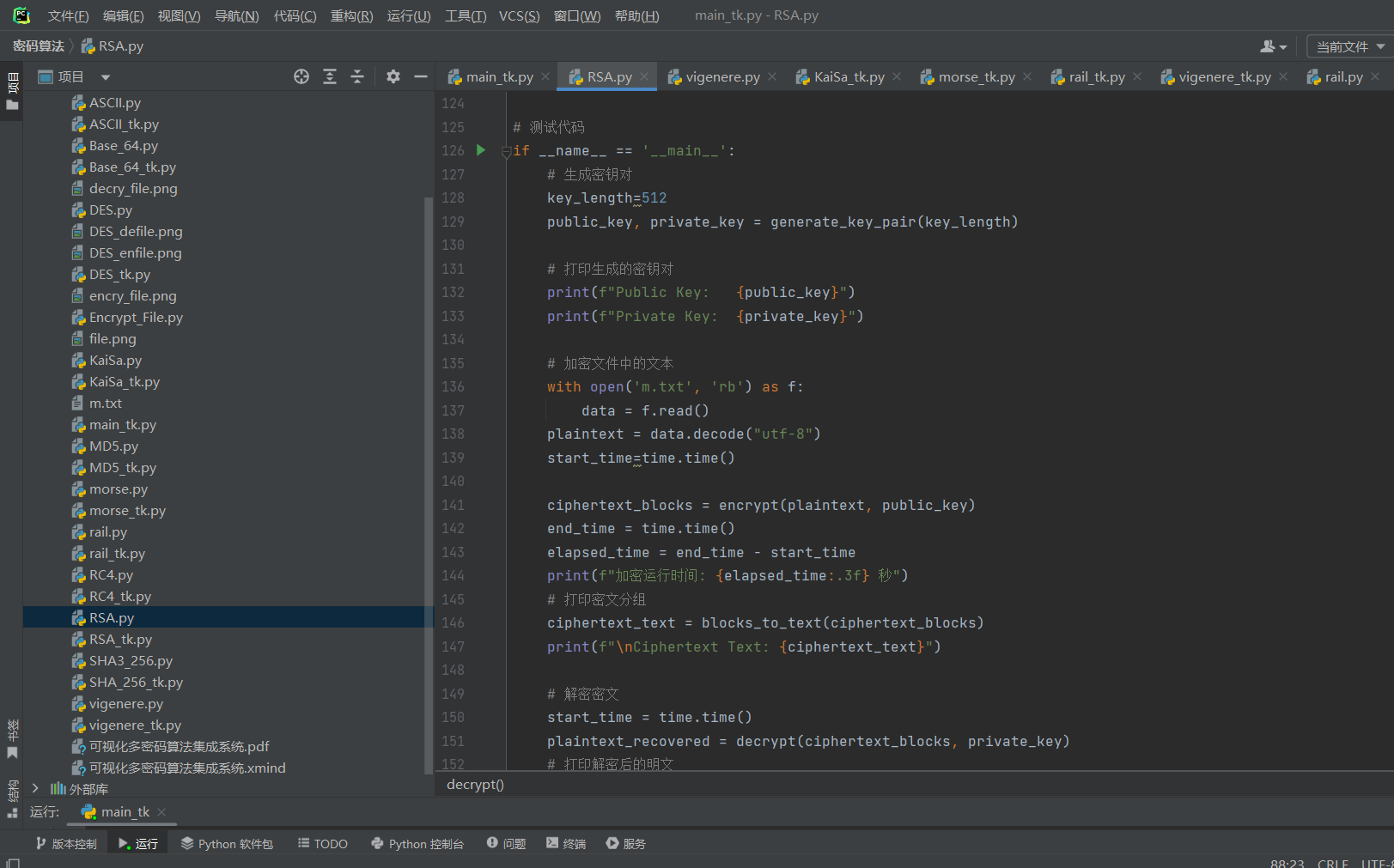
用户的输入有时候是不符合格式的，犹如在酒吧点了一碗扁肉，于是酒吧可能停止了运作或者炸了。这时候需要提示用户输入有误，并指导用户正确的输入格式。

## 系统模块设计

系统分为三个模块，先写好十二种算法函数，然后逐一写好各函数对应的GUI界面，最后在主界面通过密码类型下拉框的形式进行集成，用户可以进行灵活选择与切换。

### 十二种算法函数

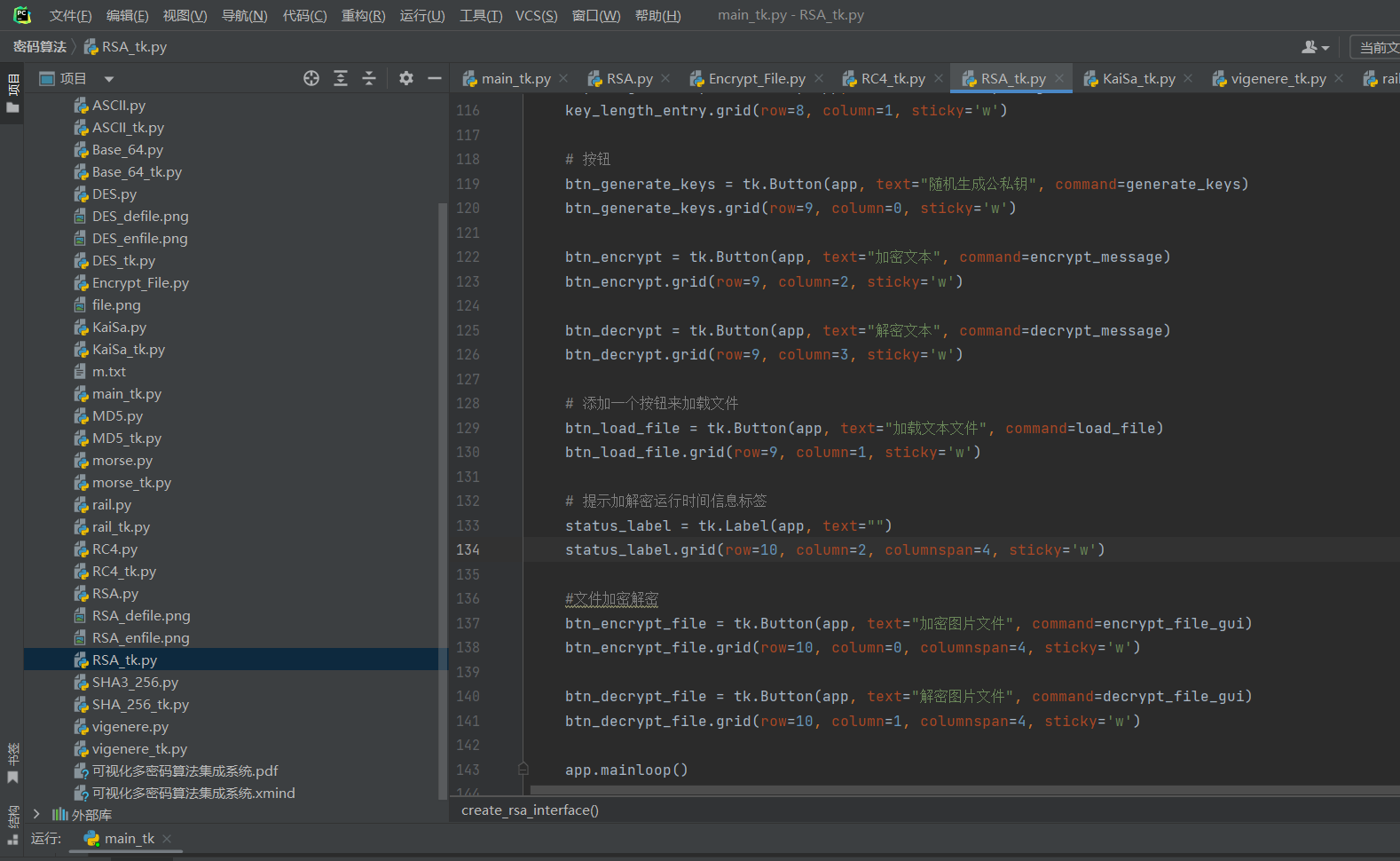
十二种算法包括RSA、DES、AES等主流算法，通过Python编程实现。如图3所示，该部分展示了RSA的部分代码，完全的代码见附录.py文件。



RSA代码

### 十二种算法GUI界面

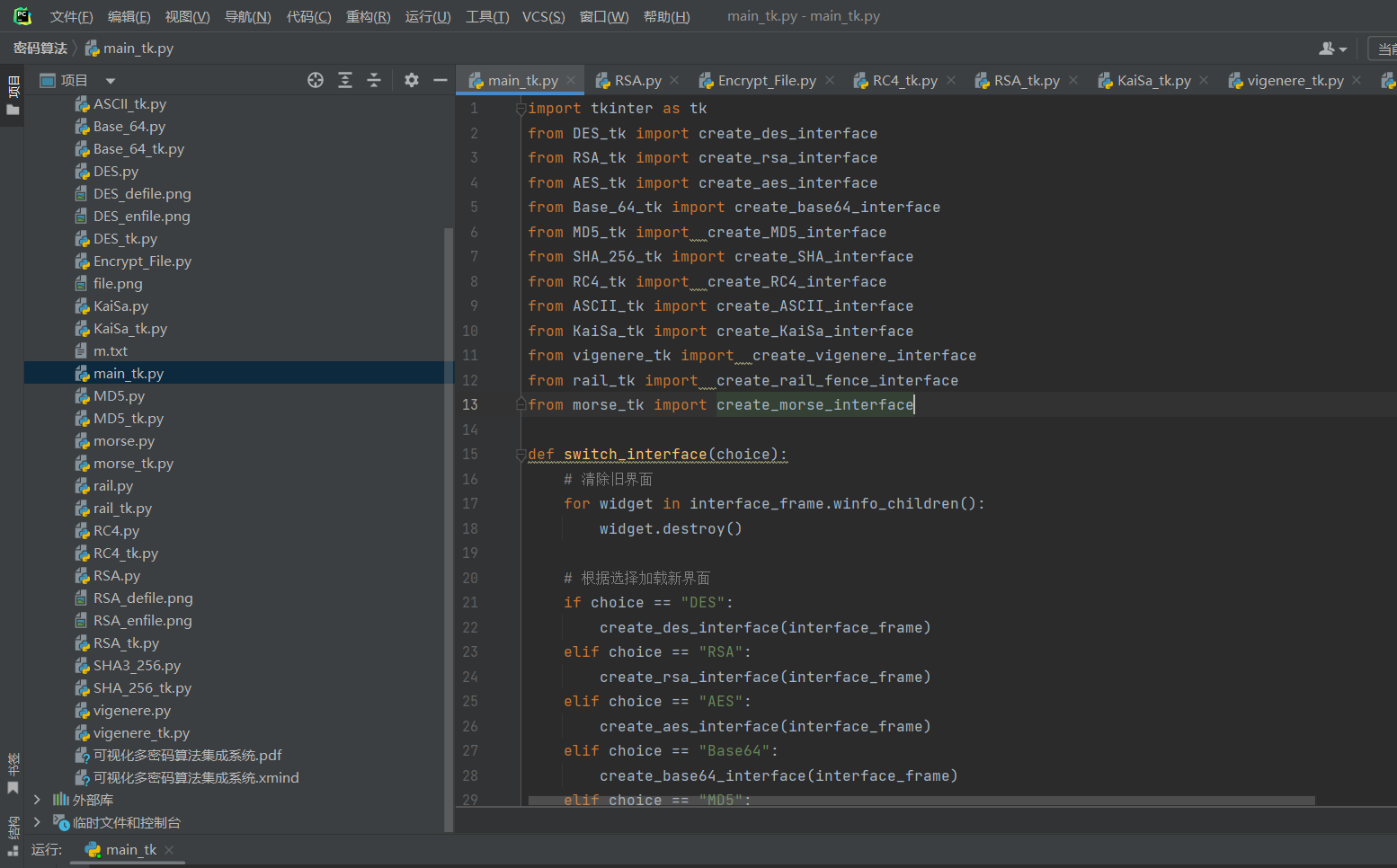
写好了2.2.1的十二种算法，便可以通过import导入算法函数，编写GUI界面。十二种算法GUI界面包括RSA、DES、AES等主流算法的GUI界面，通过Python的tk库编程实现。如图4所示，该部分展示了RSA\_tk的部分代码，完全的代码见附录.py文件。



RSA\_tk代码

### 主界面集成

写好了2.2.2的十二种算法GUI界面,便可以将他们通过一个密码类型下拉框集成，便于用户选择与切换，其中默认界面为RSA。



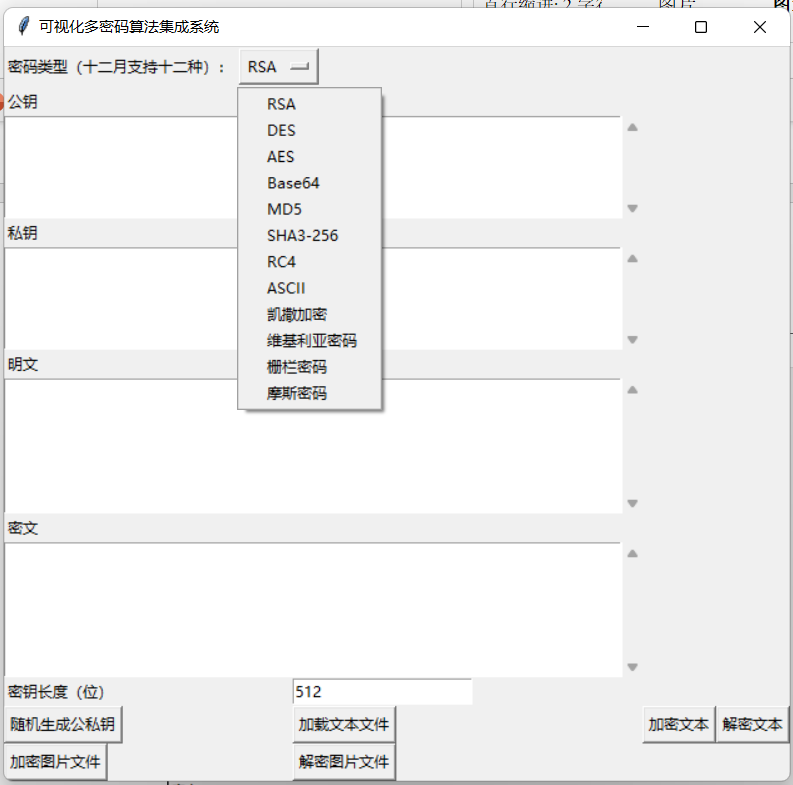
main\_tk代码

## 系统运行实现

本系统***一键运行***，在main\_tk代码点击运行即可。

### 界面预览

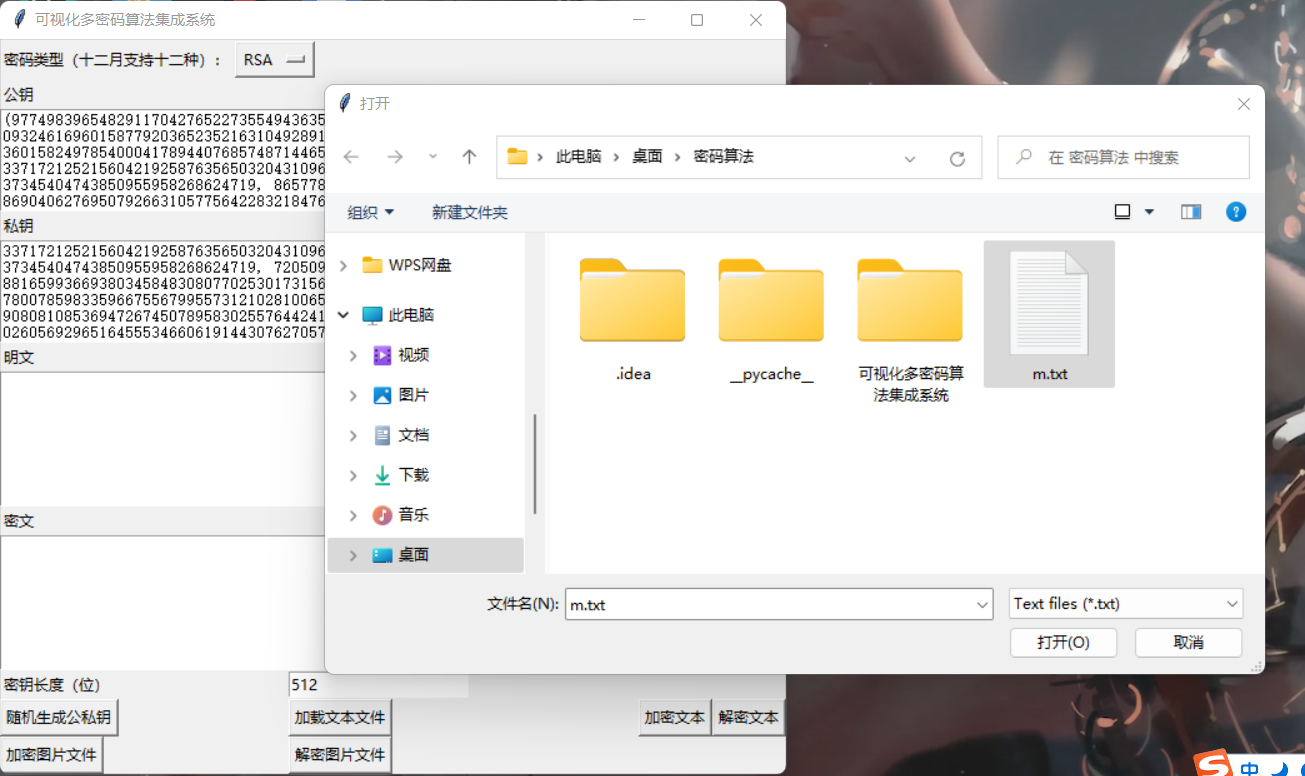
运行后映入眼帘的为默认的RSA算法GUI界面。可以在密码类型下拉框选择与切换其他密码算法。RSA默认密钥长度为512位，用户可以自行调节。



主界面预览

### 运行结果与功能展示

如图7所示，点击随机生成公私钥后系统会自动生成用户控制的密钥长度的公私钥，本图为512位，然后点击加载文本文件，可以读取本机上的txt文件中的文本。



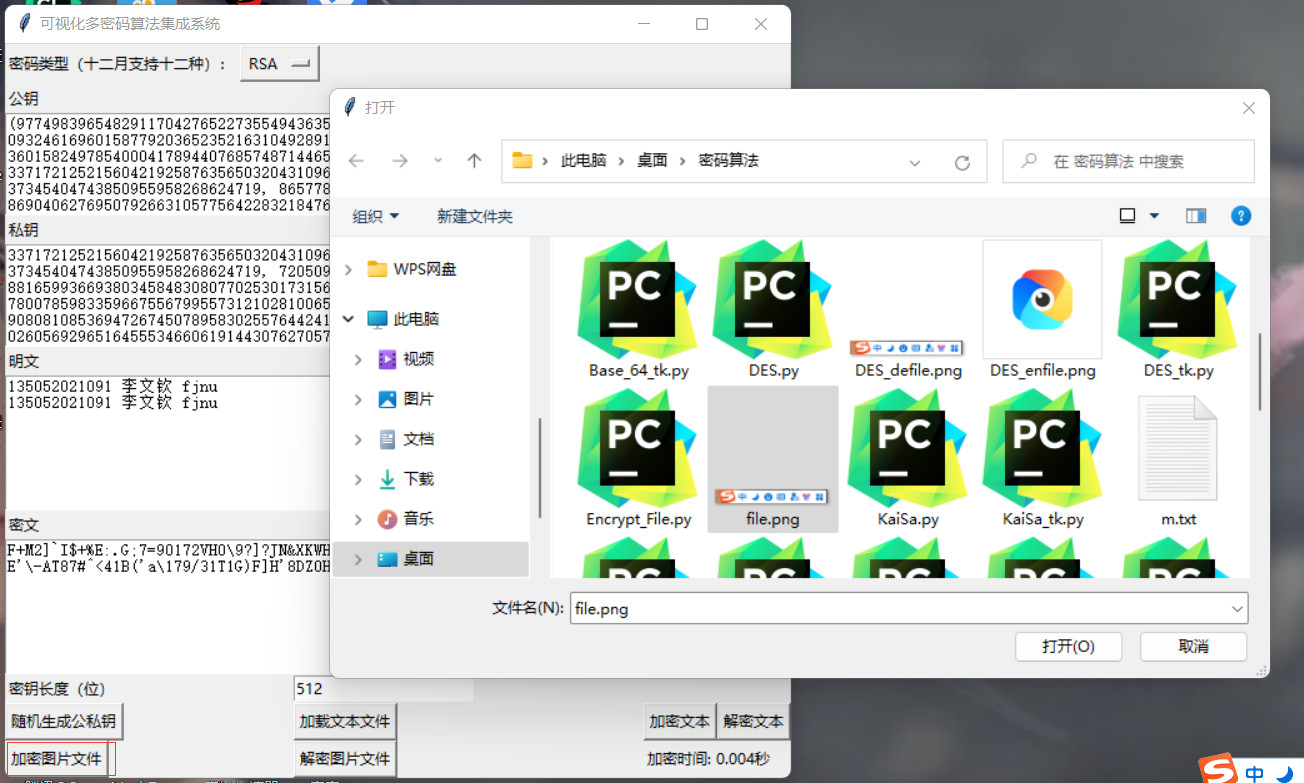
读取文件文本

如图8所示，成功读取了文件文本，并进行了加密，最后显示了加密时间为0.004秒。



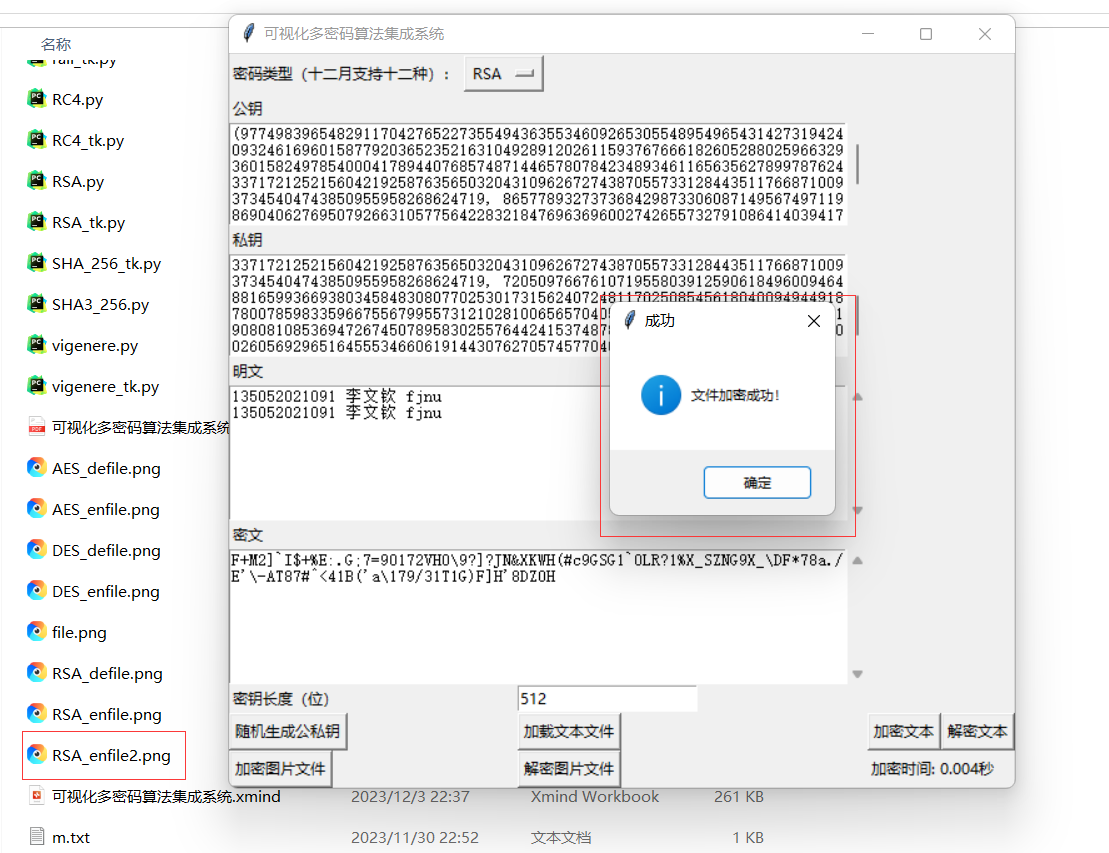
加密结果

如图9所示，点击加密图片文件按钮后，可以选择加密本机上的图片文件file.png。也可以看到以前加密过的文件DES\_enfile.jpg是不可视状态，加密成功。



文件加密

如图10所示，图片文件加密成功，显示了对话框，并在指定路径生成了指定文件名的加密后的RSA\_enfile2.png文件。



文件加密结果

## 第3章 实训总结与心得体会

本项目遇到的最大的困难便是前端GUI界面的实现。由于以往课程与实验未涉及前端界面，所以本项目的前端界面是利用Python的tk库现学实现的，也因此导致本项目的前端界面并不是很成熟与美观。不过很重要的一点是，能通过本门课程学到了以往课程未涉及的前端界面编程实现，是我在本门课程中收获的最大的一点。以往实现项目基本最好只能实现后端算法，但前端界面无法实现，也有畏难情绪不愿意实现。经过本门课程，也算赶鸭子上架勉强学会了一些前端算法，以后能将后端与前端结合，实现真正的“系统”，而不是只有算法了!