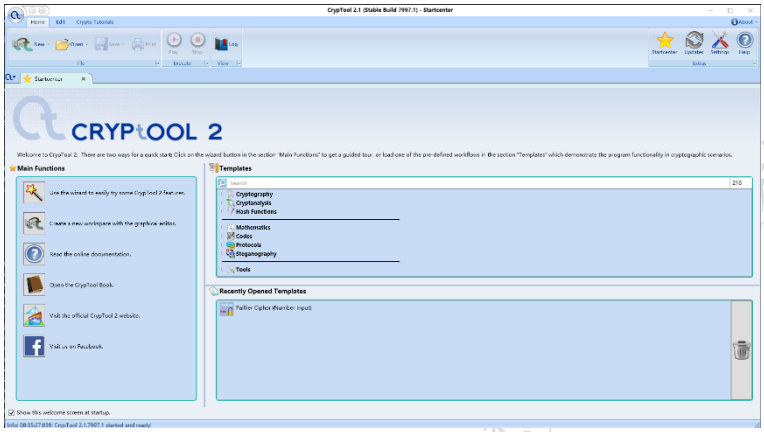
1 古典密码应用与分析实验

主要内容：凯撒密码开发；栅栏密码开发；古典密码分析（基于 Cryptool2.0）

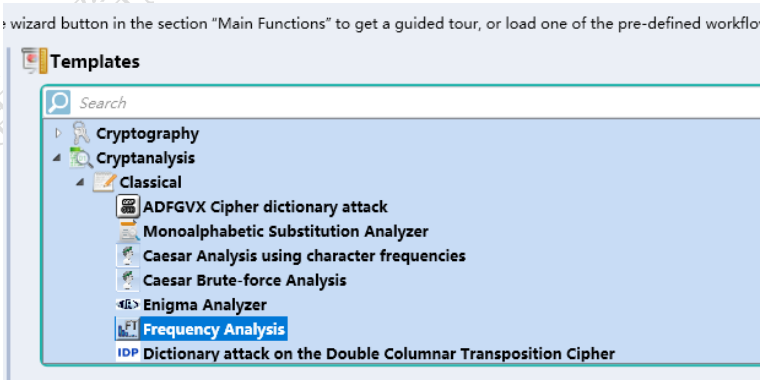
1.1实验环境部署与认识

1.1.1界面功能认知



1.2基于 Cryptool 的字母频率分析实验

1.2.1菜单选项位置



利用模板选项中字母频率分析分别对自己选定的有意义短文本、有意义长文本、随机文本进行频率分析，判断统计频率上的差别

1.2.2填写频率分析情况汇总表

统计频率最高的三个字母
分析哪一种文本更接近与语言字母统计频率，简述原因

1.2.3字母频率分析实验报告

1）标题

- 实验名称、班级、姓名、完成度

2）简述实验步骤

3）汇总表

4）分析与结论

1.3基于 Cryptool 的古典密码加解密与分析实验

1.3.1主要内容

恺撒密码（C 码）、维吉尼亚密码（V 码）、恩尼格玛密码（E 码）的加密解密、雪崩分析、破解实验

1.3.2私密消息设定

自行设定三则私密信息（如姓名、班级、嗜好、银行卡开户行等，注意英文或字母方式给出）

1.3.3利用向导选项中 Caesar 密码加密，并对三则密文分别分析字母频率、解密、破译

1）菜单选项位置

2）设置加密参数

- 移位值
- 明文字符串

3）设置解密参数

- 移位值
- 前面加密后的密文字符串

4）利用字母频率分析模板对密文分析字母统计频率

- 设置频率分析参数
  - 密文字符串
- 注意不同密钥加密的密文不要一起统计

5）设置破译参数

- 密文字符串
- 语言

6）填写加密、解密、分析、破译情况汇总表

- 加密明文、密文、移位值
- 解密密文、明文、移位值
- 密文中统计频率最高的三个字母

- 破译的明文、密文、移位值、是否破译成功，简要分析原因

1.3.4利用向导选项中 Vigenere 密码加密，并对三则密文分别分析字母频率、解密、破译

1）菜单选项位置

2）单词密钥加解密、分析

- 设置加密参数
  - 选用某一单词为密钥
  - 明文字符串
- 设置解密参数
  - 设定与加密密钥相同的解密密钥
  - 密文字符串
- 利用字母频率分析模板对密文分析字母统计频率
  - 设置频率分析参数
    - 密文字符串
  - 注意不同密钥加密的密文不要一起统计

- 与 Caesar 密码加密的相同消息的密文统计频率进行对比，有什么不同

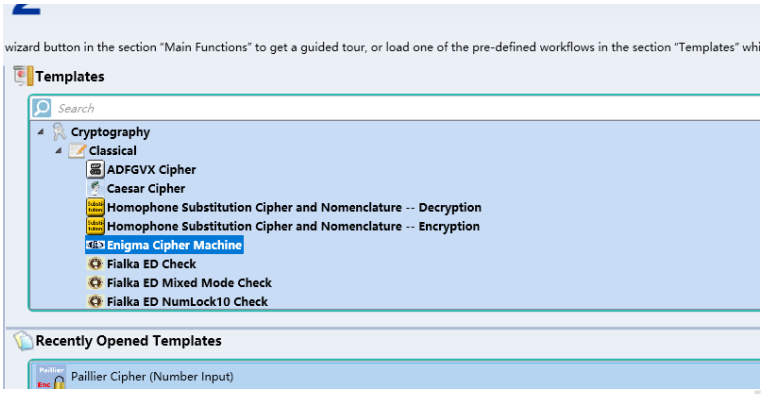
- 设置破译参数
  - 前面加密后的密文字符串
  - 语言
- 填写加密、解密、分析、破译情况汇总表
  - 加密明文、密文、密钥
  - 解密密文、明文、密钥
  - 密文中统计频率最高的三个字母
  - 破译的密文、明文、时长、密钥长度、密钥，破译是否成功，简要分析原因

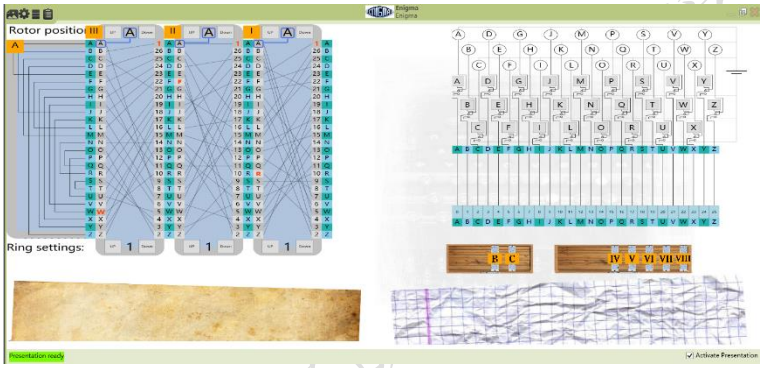
3) 长随机密钥加解密、分析

- 设置加密参数
  - 与明文相同长度的随机密钥
  - 明文字符串
- 设置解密参数
  - 设定与加密密钥相同的解密密钥
  - 密文字符串
- 利用字母频率分析模板对密文分析字母统计频率
  - 设置频率分析参数
    - 密文字符串
  - 注意不同密钥加密的密文不要一起统计
  - 与 Caesar 密码加密、单词密钥 V 码加密的相同消息的密文统计频率进行对比，有什么不同
- 填写加密、解密、分析、破译情况汇总表
  - 加密明文、密文、密钥
  - 解密密文、明文、密钥
  - 密文中统计频率最高的三个字母
    - 与单词密钥加密后密文分析比较有什么不同
  - 破译的密文、明文、时长、密钥长度、密钥，破译是否成功，简要分析原因

1.3.5利用模板选项中 Enigma 密码机加密，追踪加密过程，并对三则密文分别分析字母频率、解密

1）菜单选项位置

- 

- 

2）设置加密参数

- 明文字符串
- 初始参数：插线板、转轮初始位置、初始密钥

3）设置解密参数

- 密文字符串
- 设定与加密密钥相同的解密密钥

4）利用字母频率分析模板对密文分析字母统计频率

- 设置频率分析参数
  - 密文字符串
- 注意不同密钥加密的密文不要一起统计

- 与 Caesar 密码加密、单词密钥 V 码加密的相同消息的密文统计频率进行对比，有什么不同

5) 填写加密、解密情况汇总表
   - 加密明文、密文、密钥
   - 解密密文、明文、密钥
   - 密文中统计频率最高的三个字母
     - 与单词密钥加密后密文分析比较有什么不同

1.3.6 古典密码加解密与分析实验报告

1) 标题
   - 实验名称、班级、姓名、学号
2) 利用向导选项中 Caesar 密码加密，并对三则密文分别分析字母频率、解密、破译
   - 简述实验步骤
   - 汇总表
   - 分析与结论
3) 利用向导选项中 Vigenere 密码加密，并对三则密文分别分析字母频率、解密、破译
   - 简述实验步骤
   - 汇总表
   - 分析与结论
4) 利用模板选项中 Enigma 密码机加密，追踪加密过程，并对三则密文分别分析字母频率、解密
   - 简述实验步骤
   - 汇总表
   - 分析与结论

1.4古典密码编程（以Python为例，编程语言不限）

1.4.1置换密码编程实现

1) 复习恺撒密码的基本原理和推广出的置换密码
2) 绘制置换密码加解密流程图
    - 获取a到z的顺序字符表
    - 输入待处理消息
    - 加密
        - 建立随机化的密文字符表（随机置换表）
        - 逐个字符进行查表替代
        - 输出明文、密文
    - 解密
        - 输入密文字符表
        - 逐个字符进行查表替代
        - 输出密文、明文
3) 工程的建立
4) 相关包
    - numpy
        - random.permutation
5) 编写置换密码加密程序
6) 编写置换密码解密程序
7) 运行与调试
8) 自由优化（可以从交互输入输出、密钥的使用、密码本的生成等方面考虑，目标是更方便、更安全、更快速）

1.4.2栅栏密码编程实现

1) 复习栅栏密码的基本原理
2) 绘制栅栏密码的加解密流程图

- 输入待处理消息
- 加密
  - 对待处理消息形成数组或列表
  - 生成密文空间
  - 读取 0、2、4、6...数据到密文空间的 0、1、2、3,...round(n/2)-1
  - 读取 1、3、5、7...数据到密文空间的 round(n/2)、round(n/2)+1、round(n/2)+2...n-1
  - 输出明文、密文
- 解密
  - 生成明文空间
  - 读取密文的 0、1、2、3...round(n/2)-1 到明文空间的 0、2、4、6、...
  - 读取密文的 round(n/2)、round(n/2)+1、round(n/2)+2...n-1 到明文空间的 1、3、5、7...
  - 输出密文、明文
- 注意：方法不止一种，比如还可以先建立栅栏字符串数组，然后把消息写入，最后读出明文或者密文

3) 工程的建立

4) 编写栅栏密码加密程序

5) 编写栅栏密码解密程序

6) 运行与调试

7) 自由优化（可以从交互输入输出、密钥的使用、栅栏深度等方面考虑，目标是更方便、更安全、更快速）

1.4.3实验报告要求

1) 置换密码

- 给出加密、解密详细流程图
- 给出执行结果截图
- 给出代码工程压缩包

2) 栅栏密码

- 给出加密、解密详细流程图
- 给出执行结果截图
- 给出代码工程压缩包

1.4.4（选作）Python3 的简要入门

1) 语言简介、风格

- 有 c、c++基础学习起来很容易
- 基本内容涉及数据结构知识，用起来会更顺手（可以达到知其然，知其所以然）

2) Python 的安装部署

- python only
- python+pycharm
- anaconda+pycharm

3) 开发环境熟悉

- python in cmd
- jupiter
- pycharm

4) Python 简要语法

- 缩进
- 控制语句
  - 判断
    - if (condition):...else...elif
    - is in/not in
  - 循环
    - in X
      - 遍历 X
    - for
      - 与 in 结合，形成循环范围

        - while
    - 异常
        - try.....except Exception as...
        - raise
        - assert(condition)
    - 过程
        - with operation as X:
            - 保证操作产生的对象 x 能够在结束 with 过程时被正确清理
        - yield
            - 生成器（迭代器的一种）中用于返回数据
        - pass
- 函数（包）
    - def func():
    - import/ from...import.../import...as...
    - pip install、conda install
- 表达式
    - Python 的表达式写法与 C/C++类似
    - 列表推导式
        - sum(x * x for x in range(10))
- 类与对象
    - class
        - 构造函数（初始化函数）为 def __init__（...）
        - 成员函数的第一个参数为对象自身的引用，名称自定
        - 成员变量以 self.x 形式表述，self 名称自定
    - 构造对象
    - 调用
        - classname.method
        - instance.method
- 数据类型

- 动态类型系统，运行时检查确定类型
- 固定数据类型
  - str
    - "string"
    - upper
    - find
    - len
    - string[index]
  - bytes
    - b"string"
  - int
    - 精度不限！，对于密码如 RSA 等非常适用
  - float
    - 精度受系统影响
  - complex
  - bool
- 可变数据类型
  - 有序的
    - list
    - 列表，元素类型可以不同，有序，可以改变
    - [....., ..., ...]
    - append(element)、del (index)、pop#最后元素#、remove(value)
    - tuple
    - 元组，元素类型可以不同，有序，不可改变
    - array
    - 数组，元素类型相同，有序，可以改变
  - 无序的
    - set
    - 集合，元素类型可以不同，无序，可以改变

- frozenset
- 冷冻集合，元素类型可以不同，无序，不可改变
- dict
- 键值对集合，元素类型可以不同，无序，可以改变
- 与 map-reduce 的要求一致

- 操作
  - 所有多元素的数据类型，下标都是从 0 开始
  - element is in /is not X：在 X 中查找 element

5) 简单 Python 编程与调试

- hello world

6) 实验报告要求

- 按照以上顺序简要介绍实验内容
- 自选 10 项不同内容，给出具体代码和正确输出结果