cryptFileTool设计文档

# 修改历史

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修改日期** | **修改人** | **备注** |
| V1.0.0 | 2017-05-13 | WuJinlei | create |
| V1.0.1 | 2017-05-27 | WuJinlei | 完善各设计图；丰富开发阶段； |

目录

[cryptFileTool设计文档 1](#_Toc483667922)

[修改历史 1](#_Toc483667923)

[1.目标 3](#_Toc483667924)

[2.场景 3](#_Toc483667925)

[3.UI 3](#_Toc483667926)

[4.上下文图 4](#_Toc483667927)

[5.序列图 5](#_Toc483667928)

[6.模块图 5](#_Toc483667929)

[6.1.UI模块 6](#_Toc483667930)

[6.2.通信层 6](#_Toc483667931)

[6.3.后台处理模块 6](#_Toc483667932)

[7.类图 6](#_Toc483667933)

[8.内存及线程 6](#_Toc483667934)

[9.头文件 6](#_Toc483667935)

[10.开发阶段及计划 6](#_Toc483667936)

[10.1.V1.x.x系列 6](#_Toc483667937)

# 1.目标

cryptFileTool定位为一个对文件进行加密、解密的工具，可以自行选择一个(或几个)算法，对给定文件执行加密、解密操作，以保护公共场合文件内容的安全性。

本工具在客户端(PC/移动终端)运行，加解密过程封装在工具本身，无需任何网络操作，不存储任何用户信息，因此也无需数据库操作。

计划做两个版本线：

1. V1.x.x:PC上的客户端程序，使用QT做UI界面；
2. V2.x.x:移动设备上的app程序，基于android，暂不考虑ios(因为没钱买设备测试…..);

# 2.场景

涉及到的场景有两个：加密文件和解密文件；

加密文件的场景如下：

1. 选择要进行加密的文件；
2. 选择要使用的加密算法；
3. 输入加密使用的key；
4. 点击“加密”按钮：
   1. 如果没有选中源文件、加密算法没选择、输入的key不合法，弹出对话框，提示用户“必须选择要加密的文件！”、“必须选择一种加密算法”、“输入的key不合法！(不合法的原因简述)”；

一旦用户点击“好，朕知道了”按钮，不清空刚才的内容，回到初始页面；

* 1. 如果以上三个信息都有效，弹出对话框：“源文件：xxx；加密算法：xxx;密钥：xxx；大王，信息无误的话，小的开始加密啦~~”，用户选择“好，加密吧”后，进入step5；选择“等会，我要改点东西”，不清空刚才的内容，回到初始页面；

1. 弹出一个模态对话框，标题是“玩命加密ing~~~”，有进度条提示加密进度，主页面上的日志框里有日志信息显示:
   1. 如果加密正确结束，提示“大王，小的加密完啦！”，用户点击“确定”，回到主页面；
   2. 如果加密出现错误，提示“不好拉，大王，加密出错了！出错原因是：xxx；”，用户点击“确定”，回到主页面；

至此，一轮加密结束；

解密流程与加密雷同，不做赘述；

# 3.UI

UI界面设计草图如下：

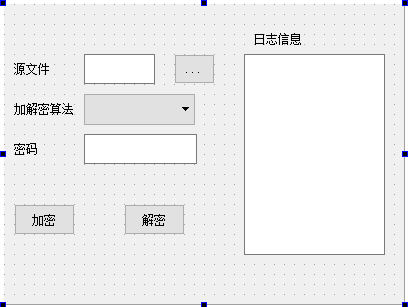


图3.1.UI界面草图

“源文件”后面的lineEdit中，记录了源文件的绝对路径，可以手动输入，也可以点击后面“…”按钮，选中文件，一次可选择多个；

目标文件无需指定，将会在源文件后面，增加“.cftCipher”或“.cftPlain”作为标记；

# 4.上下文图



图4.1. 上下文图

UI显示部分，依赖于QT和Android，这个显而易见；

工具的核心功能，加解密的功能，依赖于moCrypt实现的加解密算法；

文件的相关操作，依赖于moUtils提供的File操作；

日志记录，依赖于moLogger；

# 5.序列图

图5.1.序列图

图5.1所示，为加密/解密从UI发起请求、到进度条展示进度、到加密/解密完成的流程；

# 6.模块图



图6.1.模块图

约定本工具的缩写，取每个独立单词的首字母，为cft(CryptFileTool);

分为3个模块：

## 6.1.UI模块

约定本模块的缩写，为：cftUi；

本模块主要负责UI展示；

本模块只与“通信模块”交互；

## 6.2.通信层

约定本模块的缩写为：cftProt；

本模块负责接收“UI模块”发送来的加解密请求，并将请求发送给“后台处理模块”，并将后台处理模块的处理结果，发送给“UI模块”展示；

对于一次要求加解密多个文件的请求，将其加入到本地缓存中，逐个请求执行；

## 6.3.后台处理模块

约定本模块的缩写，为：cftBgCrypt；

接收“通信层”发送来的请求信息，并处理加解密请求；

# 7.类图

无父类、子类；

类图略；

# 8.内存及线程

“UI模块”无法控制，忽略；

“通信层”需要一个queue，用来存储所有的加解密请求；需要1个线程，用来取出queue中的任务并发送给“后台处理模块”；

“后台处理模块”无需额外的内存和线程；

# 9.头文件

# 10.开发阶段及计划

## 10.1.V1.x.x系列

V1.x.x系列，以QT为UI界面，“通信层”和“后台处理模块”以动态库的形式体现，实现最终效果。

版本号中，V1不可更改，代表以QT+C++的模式开发的PC版的应用程序；V1.x的第一个子版本号x，伴随加解密算法的增加等业务功能的完善，自加；V1.x.x的第二个子版本号x，随bug修改增加；

V1.0.0版本作为第一个功能版本，分如下几个开发阶段：

1. UI界面为主，“通信层”和“后台管理模块”以桩体现；
2. “通信层”和“后台管理模块”功能实现；
3. 整体功能实现；