大作业作业报告

87-代码一遍队

版本：1.0

日期：2025 年 6 月 30 日

1.程序功能介绍

我们的程序完成的是一个密码管理与加密工具。此工具基于 Qt 开发，旨在为用户提供一个安全、便捷的平台来统一管理各类账户密码、加密敏感文本及文件。程序通过设置一个主密码来保护所有存储的数据。

1.1 注册与登录功能

在打开程序后的第一个界面中，用户可以自行选择注册新账号或者根据已有的用户登录。登录后用户的敏感信息会被加载，未登录的用户的敏感信息不会被加载到内存中。登录界面如图所示。

1.2 密码生成功能

在登录后，共有四个可用的界面，利用顶端的四个按钮来进行页面切换。最左侧的界面是密码生成界面。此界面可以帮助用户生成随机的，强度较高的密码。界面中部最下方的复选框和输入栏分别代表使用的字符和密码长度。选定之后点击“生成”即可生成由所选字符构成的制定长度的密码。点击“应用到…”即可把密码快速移动到密码管理界面的密码输入栏中（下述）。

1.3 密码管理功能

登陆后的第二个可用界面是密码管理界面。密码管理界面可以帮助您安全地管理您的密码。左侧的密码搜索框允许用户搜索匹配的站点，只需要向搜索框输入站点的名称（可以是部分名称）并点击搜索按钮即可实现。下方的栏目中会列出符合匹配的站点名称，用户名和密码。不输入任何文字直接按搜索会显示所有密码。

显示密码后，密码单元格的背景颜色会提示您密码的强度，红色代表弱，绿色代表强，而黄色代表中等。

在右侧输入站点，用户名和密码，按下添加，更新和移除三个键即可添加，更改与删除存储的密码。

下方的导出按钮是用来将密码以明文形式导出的，“导入”按钮则可以将 JSON 格式的密

码以明文导入。在此界面上，也可以将以明文导出的密码文件拖入以实现更多密码的导入。

1.4 文本加密功能

文本加密界面会使用独属于用户的主密钥加密文本，其他人的密钥无法解密您的密钥加密后的文本。

将待加密的文本输入左侧的文本框中，点击加密即可在右侧获得加密后的文本。同样，将加密后的文本输入右侧的文本框中，点击解密可以在左侧获得解密的文本。

1.5 文件加密功能：

文件加密页面会使用独属于用户的主密钥加密用于加密文件的密钥。用户需要向“待处理文件路径”一栏输入待处理文件的路径，向“处理后文件路径”一栏输入处理后文件的路径（处理后的文件会自动创建），以及向“标识符“一栏输入您为这个文件提供的内部名称。提供后内部名称和生成的密钥将隐式存于密码管理系统中。

点击加密按钮会将待处理文件加密，生成加密后的文件；点击解密按钮会将使用您提供的标识符对应的内部密码将这一文件进行解密。

用户也可以点击右侧的选择文件（以及选择/创建目标文件）来选择您需要处理的文件和文件处理后的路径。标识符右侧的自动生成只在加密时起作用。在加密时，文件的路径会作为标识符。

将文件拖入此界面下的窗口可以自动将拖入的文件作为待处理文件进行选择。

1.6 桌面助手功能

最顶端的菜单栏中包含 Assistant Mode 的选项，用户通过选择 Assistant Mode 栏下的 Switch to table assistant 选项可以将加密器切换为桌面助手选项。桌面助手会以紫色圆角矩形的形态显示在屏幕的右上角，会同时显示此时登录的用户名。把鼠标放在桌面助手上，会出现一个下拉菜单，下拉菜单中有几种快捷功能。

第一个快捷功能是自动密码填充。将站点名称复制到剪贴板上，再点击菜单上的快速填充密码，可以将剪贴板上的内容替换为此站点的密码。

第二个快捷功能是快速文本加密（解密）。此功能可以通过点击菜单完成，也可以通过 Ctrl+Shift+E(Ctrl+Shift+D)完成。仍然将待加密（解密）的文本复制到剪贴板上，点击菜单或者调用快捷键即可把剪贴板上的文本替换为加密（解密）后的文本。

第三个快捷功能是快速文件加密。将文件拖入桌面助手内，文件将被自动加密至设置的默认路径（通常是 C 盘下的[电脑用户名称]/Appdata/Local/PasswordKeeper/[在软件

中注册的用户名]）在默认路径中可以取出加密后的文件。

菜单中还包含帮助界面，讲解了桌面助手的可用功能。最后，桌面助手在双击时可以回到主窗口。

1.7 其它功能

主界面顶端的菜单栏中，Setting 一栏中包含四个选项。第一个是 Change Password，用户可以在此更改自己的密码。第二个是 Exit 选项，点击后即可退出程序。第三个是 Change File Path 选项。在此用户可以更改自己的默认文件路径。第四个是 Minimize to Tray 选项，用户可以把窗口缩小到系统托盘。

顶端菜单栏的 Help 一栏中包含了帮助事项。用户可以参考 Help 一栏进行操作。

2.项目各模块与类设计细节。

下面将分为后端部分和前端部分分别介绍项目的模块与类设计细节。

2.1 后端部分（后端除 aes 算法实现之外，其余部分的类都包含在对应类名.h 与.cpp 文件中）

2.1.1 底层加密库：aes.c，aes.h

此库提供 C 语言实现的 AES 算法，是加密体系的核心部分。此库并不是我们的工作。仓库地址：https://github.com/kokke/tiny-AES-C/。

2.1.2 密码学工具类 CryptoUtils

此类是一个工具类，负责提供密钥派生，密码哈希与 AES 封装的功能，可以形成高强度的派生密钥，计算密码的哈希值以及将 aes.c 中的功能进行封装。

2.1.3 加密处理器 EncryptionHandler

此类封装了一个加密/解密的接口，使用一个私有成员 QByteArray m_masterKey 持有主加密密钥，为上层模块提供统一的加密与解密接口。

2.1.4 登录管理器 LoginManager

此类负责管理用户的完整生命周期，包括创建，登录和主密码变更。

构造函数将用户凭据文件存储在 QStandardPaths::writableLocation(QStandardPaths::AppLocalDataLocation)处，以 JSON 格式存储；saveCredentials 和 loadCredentials 函数用于将盐和哈希值以 Base64 的字符串形式存储目标 JSON 对象。

createUser：用于创建用户，按照生成盐 -> 哈希密码 -> 派生密钥 -> 保存凭据的顺序执行，以创建用户。

login：用于判断密码是否成功登录。通过一个不提前退出的循环对哈希值进行比较，使得正确的密码和错误的密码比较操作的耗时基本一致，有效防范时序攻击。

changeMasterPassword：通过此函数可以更改当前用户的密码。此函数按照以下流程处理以确保安全性。1) 验证旧密码的正确性；2) 生成一个全新的盐和新的密码哈希；3) 从新密码派生出新的主密钥；4) 调用 passwordManager 的 reEncryptAllData 方法（之后阐述），用新旧密钥重新加密整个密码库；5) 只有在第 4 步成功后，才更新实例内部的盐、哈希和主密钥；6) 最后调用 saveCredentials() 将新的凭据持久化到磁盘。这个流程最大限度地保证了操作的原子性。

2.1.5 密码管理器 PasswordManager

此类是后端最重要的类，负责管理密码库的密码条目和文件密钥，并处理密码库的加密存储功能。成员 m_entries（QMap<QString, SiteEntry>类型）负责存储站点，站点用户名和密码（加密后）构成的密码条目（以及文件密钥条目）。

此类使用 SiteEntry 结构体存储密码条目。SiteEntry 通过 toJson() 和 fromJson()两个辅助方法与 QJsonObject 相互转换（二进制数据编码为 Base64 字符串）。包含 siteName，username，fileKeyIdentifier，encryptedPassword，encryptedFileKey 五个成员变量，前三者为 QString 类型，后二者为 QByteArray 类型。当用于密码加密时，启用 siteName，username 和 encryptedPassword，用于文件密钥储存时，启用 siteName，username 和 encryptedFileKey。

PasswordManager 类的成员函数则如下方所述。

saveVault：此函数通过将内存中读取或修改的 m_entries 内的所有 SiteEntry 转换为 QJsonArray，再调用 m_encryptionHandler 加密整个 JSON 文档，最后写入文件。写入前，会使用 QDir::mkpath 确保目标目录存在。

loadVault：执行与 saveVault()相反的解密和反序列化过程。

addEntry，updateEntry，removeEntry：用于增加，更新与删除条目。使用 m_encryptionHandler（PasswordManager 类内置的 EncryptionHandler 对象）加密用户的输入，并合并到 m_entries 中。

getDecryptedPassword：用于从 m_entries 中获取解密后的密码。先从 m_entries 中获取对应条目，再解密获得密码。

getEntryDetails：用于获取站点对应的用户名（查询 m_entries）。

getAllSiteNames：用于获取所有的站点名称。

storeEncryptedFileKey/getDecryptedFileKey：管理文件密钥，对文件的密钥进行加密或者解密（仍然查询 m_entries）。

reEncryptAllData：用于在修改密码时重新加密所有数据。

2.1.6 文件加密器 FileEncryptor

此类实现文件的具体加密功能，通过 encryptFile 和 decryptFile 函数进行加密和解密。此类包含一个 PasswordManager 类型的引用 m_passwordManager，负责和用户的主要 PasswordManager 进行连接，以将文件密钥和其它同一用户的密码统一存储。

encryptFile 的加密过程：CryptoUtils::generateSalt(CryptoUtils::AES_KEY_SIZE) 为当前文件生成一个一次性的、高强度的随机文件密钥 (fileKey)。2) 使用此 fileKey 加密文件内容。3) 调用 m_passwordManager.storeEncryptedFileKey()，将这个明文的 fileKey 连同其标识符一起存入密码库（PasswordManager 内部会用主密钥对其加密）。4) 最后将加密后的文件内容写入磁盘。

decryptFile 的解密过程：1) 根据用户提供的文件标识符，调用 m_passwordManager.getDecryptedFileKey() 从密码库中取出并解密出对应的 fileKey。2) 使用此 fileKey 解密文件内容。

2.1.7 密码生成器 PasswordGenerator

此类负责根据用户的规则生成随机密码。调用 generatePassword 即可获得随机密码。该成员函数的工作逻辑为：先强制从每一个选定的字符集各选一个放入一个临时的 QVector，然后再随机从选定的所有字符集中随机选取字符，填满剩余长度；最后对整个 Vector 进行随机置换，确保最终密码的随机性。

2.1.8 密码强度检测器 PasswordStrengthChecker

此类负责检查用户的密码的强度。强度根据密码长度和包含的字符种类数量以及是否是常见密码进行评分，评分最终映射到<3,<5<>=5 中，返回 weak，medium，strong 的分类。

2.2 前端部分：

前端部分包含一个主窗口类（MainWindow）和一个桌面助手类（DesktopPet）。

2.2.1 主窗口类 MainWindow

此类是程序的主窗口和 UI 总控制器类。

UI 方面，此类包含一个 ui 类型的对象指针，可以调用编辑器中编辑好的 ui。UI 的主要部分是一个 stackedWidget 对象，通过切换页面来进入不同的功能页面。

MainWindow 类将不同的后端类的指针以及所需的数据按照功能分在不同的结构体

中，按需调用。此外，MainWindow 类还带有 DesktopPet 和 QSystemTrayIcon 类指针各一个，用于切换到系统托盘和桌面助手模式。

除了槽函数外，MainWindow 类还包含以下成员函数：

preLogin 和 login：前者在触发登录事件时调用，用于判断是否登录成功；后者在登陆后调用，用于初始化用户。

loginStatusShow：用于为用户返回登录信息。

clear：用于清除用户信息。

createInputDialog/createReadOnlyTextDialog：用于快速创建小型输入对话框或者只读的提供信息的对话框

pushInfo：向用户提供信息。

showTopBar/hideTopBar：显示/隐藏顶部菜单栏/功能切换按钮

importPasswords/exportPasswords：导入/导出密码（在密码管理页面使用）

dragEnterEvent/dropEvent：拖入/放下事件（此函数为重载的基类同名函数）

createTrayIcon：创建系统托盘图标

2.2.2 桌面助手类 DesktopPet

桌面助手创建时

桌面助手包含的成员函数如下：

createContextMenu：创建下拉菜单

updateAppearance：更新桌宠外观

showPet/hidePet：显示/隐藏桌宠

setPasswordFillCallback/setFileEncryptCallback/setTextEncryptCallback/setTextDecryptCallback：设置回调函数（以便同主窗口更好地协作，完成功能）

setUsername：设置用户名

createReadOnlyTextDialog：功能同主窗口同名成员，用于创建小型信息对话框。

以及其它槽函数和重载的事件函数。

3.小组分工

郭梓墨（组长）负责前端代码（上述前端类）与 UI 设计。

李牧之负责后端代码（上述后端类）与部分前端成员函数（主窗口的 exportPasswords，桌宠的 setTextEncryptCallback/setTextDecryptCallback）

唐睿琪负责界面的 QSS 设计和素材搜集。

4.项目总结与反思

4.1 项目总结：

本项目成功地实现了一个功能全面、安全性高的密码加密助手。项目包含从加密工具类到上层清晰的 C++/Qt 业务逻辑封装，再到创新的 UI 交互的一套完整内容。项目组织相对模块化，各个类各司其职，有效封装，让前端和后端的合作非常轻松。桌面助手的 UI 设计大大提升了项目的便捷性，便于用户日常使用。

4.2 项目反思与可改进之处

本项目仍有许多待改进之处。在错误处理与恢复机制上，如果程序在保存新凭据前崩溃，则会将用户数据永久锁定，未来可以尝试引入事务性写入或者备份机制。在 MainWindow 类中，多个 struct 管理状态的模式相对臃肿，之后可以考虑把它们提升为独立的类，以使程序结构更清晰易懂。在依赖管理方面，直接包含 aes.c 和 aes.h 的设计可以改进为使用 CMake 等工具构建系统。