Computer Science and Application 计算机科学与应用, 2019, 9(6), 1095-1103
Published Online June 2019 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2019.96123

Hans汉斯

Design and Implementation of Synchronized Password Management System for Individual Users with Multiple Terminals

Yu Liu, Youhui Su, Zhengzhi Xu

School of Mathematics and Physics, Xuzhou University of Technology, Xuzhou Jiangsu
Email: suyh02@163.com, ly-yuaoh@qq.com

Received: Jun. 2nd, 2019; accepted: Jun. 14th, 2019; published: Jun. 24th, 2019

Abstract

In order to enable individual users to manage their passwords on multiple platforms more safely and conveniently, we have developed a password management system for individual users, using the python technology. The unique features of personal computers and custom encryption logic are used to encrypt and store the users' passwords, which are managed in a unified way so that users can manage passwords more conveniently and safely. The advantage of this system is that the storage option of remote servers is added on the basis of local operation, which is convenient for users to transfer or synchronize passwords at multiple terminals. At the same time, plug-in sub-modules and self-setting encryption logic are adopted to improve the security of password storage of users.

Keywords

Python, Encrypted Storage, Plug-In Sub-Modules, User-Defined Encrypted Logic

个人用户多个终端同步密码管理系统的设计与实现

刘　禹，苏有慧，徐正知

徐州工程学院数理学院，江苏　徐州
Email: suyh02@163.com, ly-yuaoh@qq.com

收稿日期：2019年6月2日；录用日期：2019年6月14日；发布日期：2019年6月24日

文章引用: 刘禹, 苏有慧, 徐正知. 个人用户多个终端同步密码管理系统的设计与实现[J]. 计算机科学与应用, 2019, 9(6): 1095-1103. DOI: 10.12677/csa.2019.96123

摘 要

为了使个人用户更安全便捷地管理自己在多个平台的密码，借助于python技术开发了一类个人用户密码管理系统，利用个人电脑的唯一特征和自定义加密逻辑对用户的密码进行加密存储，并统一管理，让用户能更便捷，放心地管理密码。本系统的优点是在本地运行的基础上添加了远程服务器的存储选项，方便用户转移或在多个终端同步密码，同时采用插件化的子模块和自设定加密逻辑等方式来提高用户的密码存储安全性。

关键词

Python，加密存储，插件化子模块，自定义加密逻辑

1. 引言

随着大量的便捷网站和 APP 应用的使用，我们需要不断地注册账户。然而随着某些软件长时间不用，就会忘记一些账户信息，尤其是用户名和密码。但是记录在本子或电脑上，又是十分不安全而且容易丢失的。所以，有个能保证用户账户信息安全的应用来帮我们随时随地管理这些个人账户是必须的。

随着计算机科学技术的发展，账户密码管理方式主要分为文本记录，浏览器自动记录，在线用户密码管理和本地化用户密码管理四种策略。文本记录主要是用户通过文本文件或者 word 文档以明文记录自己的账户密码信息。这是一种最简单便捷的方法，缺点在于数据安全性极差，这些文件遗失或者泄露都会造成个人利益损失。浏览器自动记录密码和在线用户密码管理两种方式都是使用云端应用保存的典型方式，不同之处在于浏览器自动记录密码在本地留有备份，由用户选择是否上传云端。使用云端管理密码的方式的缺点在于数据由认可的第三方统一保存。因为统一由第三方保存，加密方式和存储都具备相同点，如果认可的第三方应用或者数据中心遭受网络黑客的攻击，就会造成大量用户密码的泄露。

现在，LastPass 是国内外应用最为广泛的在线用户账号密码管理工具。关于密码学见文献[1] [2]，python，代码重用，软件架构见文献[3]-[7]，LastPass 软件，keepass 软件相关内容见文献[8] [9] [10]。虽然在很多地方比浏览器有了很大的进步，但是它同样也有安全方面的隐患。假如有人恶意攻击他的服务器，或者盗去人们用户的 LastPass 密码，那么就泄漏了用户的账户信息。因此用户信息泄露的危险也容易发生在 LastPass 中。本地化用户密码管理基本解决了在线管理的安全方面的问题，其中最受欢迎的无疑就是 KeePass。在将账户信息保存到数据库里面去之前，先用加密类算法对其加密，通过移动端和电脑端本地数据库储存用户的账户信息，不管是用户的电脑被攻击，还是有人偷了用户的电脑，他们看到的账户信息也是经过加密的。然而这种管理工具就只能够在用户的某一台电脑或移动端上，非常不便于用户使用。从上述描述可知，这四种账户密码管理方式都有其不好之处。

鉴于上述原因，本系统综合考虑了现在已有的账户密码管理系统的各类弊端和安全问题，针对有一定编程能力的使用者设计了一个基于 python 的跨平台的轻量级且安全便捷的密码管理软件。它在本地化密码管理的基础上增加了两方面的设定，分别是用户自定义加密模块，加密逻辑和多终端转移，从而提高了密码管理的安全性并且便于用户进行多终端间的账户信息转移。

本系统通过用户自定义加密模块和加密逻辑让每个用户都可以有独立的账户信息加密过程，从而减少应用被攻击后大批量泄露密码的可能。而多终端转移则是借鉴了云端应用存储的经验并结合了当前租用小型云服务器便宜便捷的社会现实，使用远程服务器作为中端媒介进行多平台间的账户信息转移，并且提供官方服务器和自设定服务器的选项，让对账户信息安全性要求更高的用户多一种选择。本系统的优点是在本地运行的基础上添加了远程服务器的存储选项，方便用户转移或在多个终端同步密码，同时采用插件化的子模块和自设定加密逻辑等方式来提高用户的密码存储安全性。

2. 系统总体设计

2.1. 系统架构设计

系统的架构模式非常重要，适合需求的系统架构模式能对信息传递起到关键影响。因此在系统设计之前，必须先确定它的架构模式。本系统采用 C/S 架构和模块化设计，根据需求进行设计，从而提高了信息传递的效率，并且保障了程序本身的鲁棒性，健壮性和可拓展性。

本系统有 7 个模块，分别是启动模块，控制模块，密码管理模块，用户信息管理模块，加密管理模块，工具模块，配置模块，如图 1 所示。

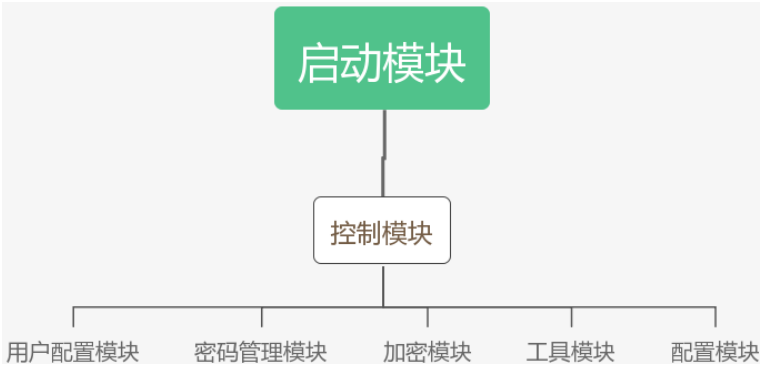

Figure 1. System architecture diagram
图 1. 系统架构图

2.2. 系统架构介绍

本系统的开发架构采取了 C/S 架构。C/S 架构是 Client/Server 的缩写，客户端需要安装专用的客户端软件，每个客户端的软件都可以向一个服务器发出请求。C/S 架构对信息安全的控制能力很强，符合本系统对安全的需求。同时 C/S 架构更加注重流程，对运行速度较少考虑，但是本系统是一个轻量级的密码管理系统，架构对运行速度的影响极小。

2.3. 系统开发工具

根据系统的功能要求和架构设计，系统选定远程服务器的操作系统为 linux，采用 python 语言编写系统的客户端，利用 openssh 与远程服务器连接。

3. 系统功能模块设计

3.1. 启动模块

启动模块主要用于用户的初始化和程序的启动、退出管理。用户的初始化中包含了创建新用户和远程旧用户信息下载，其中旧用户信息下载功能用户在更换个人设备后转移账户信息的关键功能，组织结

构如图 2 所示。

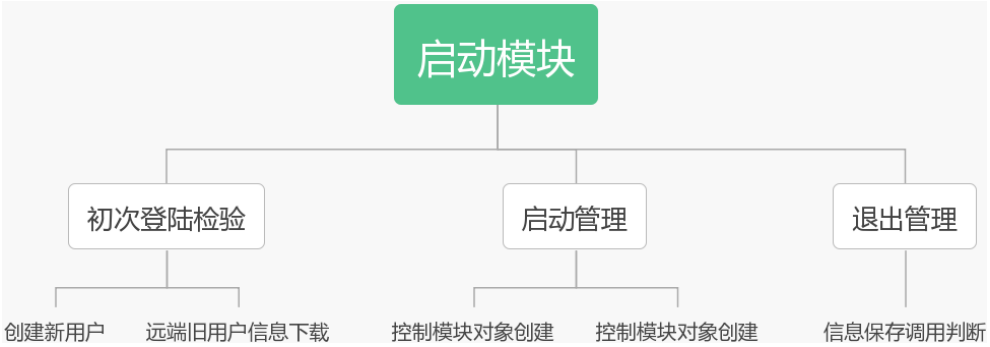

Figure 2. Start up module
图 2. 启动模块

启动模块是程序的入口点，判断是否是首次登陆并从命令行中获取参数进行处理。通过对软件的默认存储区进行检测存在用户文件和密钥文件，从而判断用户是否是第一次登陆。如果不是第一次登陆，则将参数处理后传入控制器进行逻辑判断和参数处理，主要处理逻辑如图 3 所示。

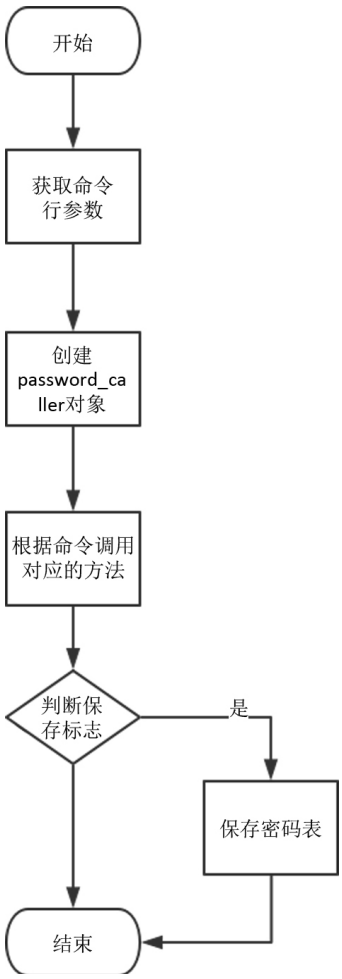

Figure 3. Start up module flowchart
图 3. 启动模块流程图

3.2. 控制模块

控制模块主要用于获取本地用户信息与解析用户的命令参数，根据命令进行功能调用或者错误处理。其中本地用户信息主要有三个部分(登陆状态，用户配置，用户密码表)组成。其中登录状态项在拦截了非本人用户的访问，提供了最基础的安全保障的同时，通过保存一定时间的登陆缓存认证提高使用体验。控制模块程序结构如图 4。

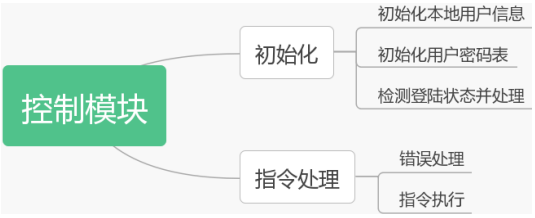

Figure 4. Control module
图 4. 控制模块

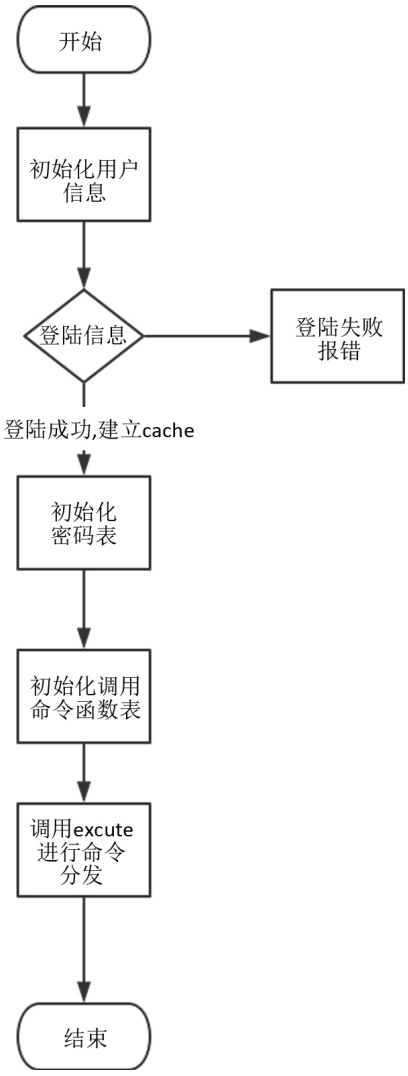

Figure 5. Control module flowchart
图 5. 控制模块流程图

控制模块首先加载软件配置，然后通过一级密钥读取并且解密出本地的用户信息，并检测用户的登陆状态(如果用户的登陆状态失效，即距离上次登陆超过 8 小时，则要求用户重新登陆)。用户登陆状态有效时，进行密码列表的加载，并且根据指令进行对应的处理。

控制模块命令执行方法，传入命令参数，类型是字符串，根据固定的分隔符划分出对应的函数名和参数，然后根据 switcher 的字典进行跳转，并且在参数或者不是对应格式时显示帮助文档，模块主要代码逻辑如图 5。

3.3. 用户配置模块

用户配置模块主要用于为控制模块解析用户配置并向控制模块提供关于用户配置的操作接口，同时额外提供远程服务器的存储选项，方便在多个终端间转移用户配置。用户配置的解析是根据用户机器独有的 mac 地址，利用默认的加密逻辑进行加解密，因为 mac 地址不会重复且长度为 16 位，所以常用的破解密码逻辑爆破密钥只具备理论的实现可能性，与密码管理模块共同组成用户信息安全的第二层安全保障，用户配置模块程序结构如图 6。

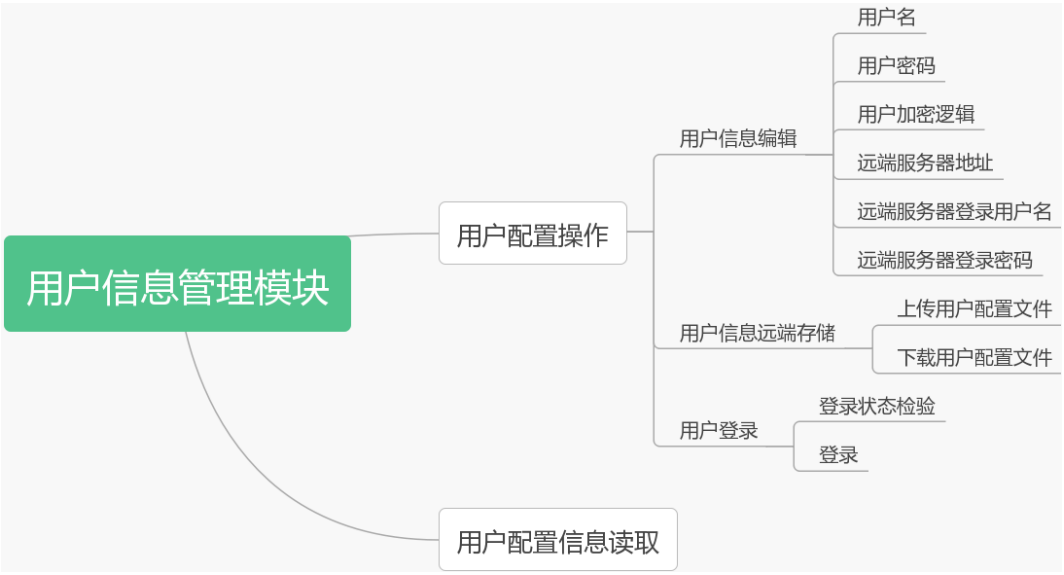

Figure 6. User information management module
图 6. 用户信息管理模块

用户信息由姓名，密码，加密逻辑，服务器地址，用户名，密码等部分组成，由一级密钥 mac 地址进行保护，其中姓名和密码登陆后从用户信息组合出二级密钥来根据加密逻辑解密用户的密码列表，而服务器地址，用户名，密码则是远程服务器的下载和上传功能的配置信息。

用户的一级密钥是 mac 地址的一部分，使用 uuid 进行获取，用“:”作为分割符号存入文件。使用 configparser 作为和配置文件读写的类接口，在读取用户存储路径下的用户文件后，用一级密钥进行解密后存入缓存文件，然后读取用户信息，并删除缓存文件，用户配置模块主要代码如图 7 所示。

3.4. 密码管理模块

密码管理模块主要用于为控制模块解析用户密码表并且为控制模块提供关于用户密码表的操作接口。用户密码表的解析在用户配置解析的基础上进行，利用用户设定的密钥和加密逻辑，使用插件化的加密模块进行加解密，密码管理模块如图 8 所示。

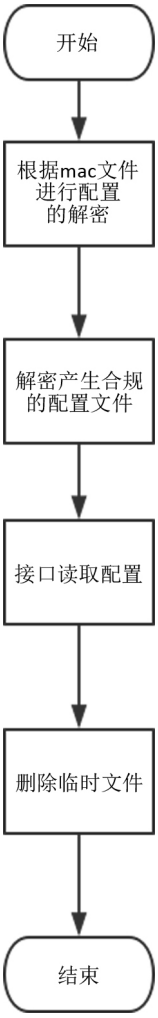

Figure 7. User information management module flowchart
图 7. 用户信息管理模块流程图

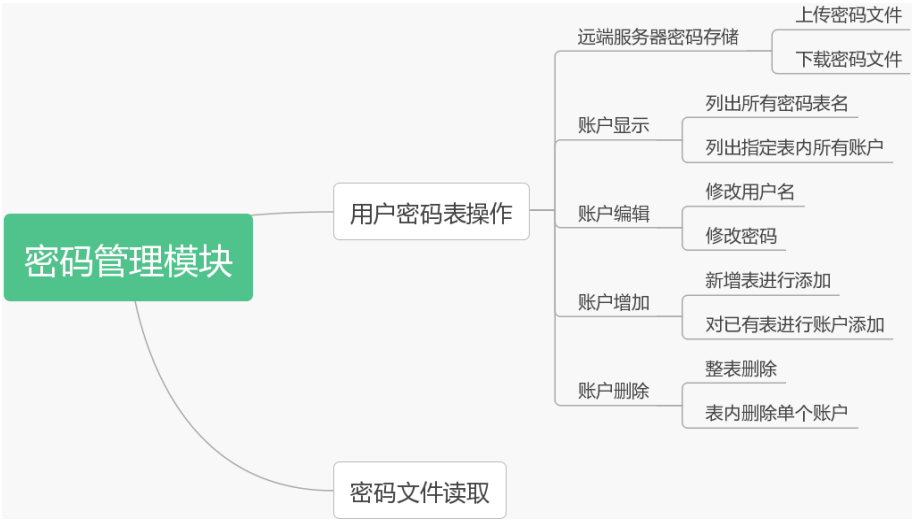

Figure 8. Password management module
图 8. 密码管理模块

密码管理模块在初始化传入用户配置，根据用户配置得到处理后的用户名和加密逻辑，将用户名转换为 Byte 数组后设定为 key，设定好加密模块对应的方法后，根据用户定义加密逻辑调用加密模块。

远端服务器密码存储功能，使用 ssh 和服务器进行连接，用用户设定的账号和密码登陆服务器，并开启一个 sftp 连接进行文件上传和下载。默认的密码文件由二级密钥(用户密钥)进行加密，以密文形式进行存储。上传功能是在远端服务器创建程序目录，并上传本地的用户文件和密码文件，下载功能则是从远端服务器下载旧用户文件和密码文件来覆盖本地文件。

3.5. 加密模块

加密模块主要用于给用户配置模块和密码管理模块提供便捷的加解密接口。加密模块在使用便捷的同时具备可拓展性，能够让有技术实力的用户便捷地添加自定义的加密子模块，从而提供独有的安全保障。这是用户安全的第三层保障，加密模块主要结构如图 9 所示。

Figure 9. Encryption management module
图 9. 加密管理模块

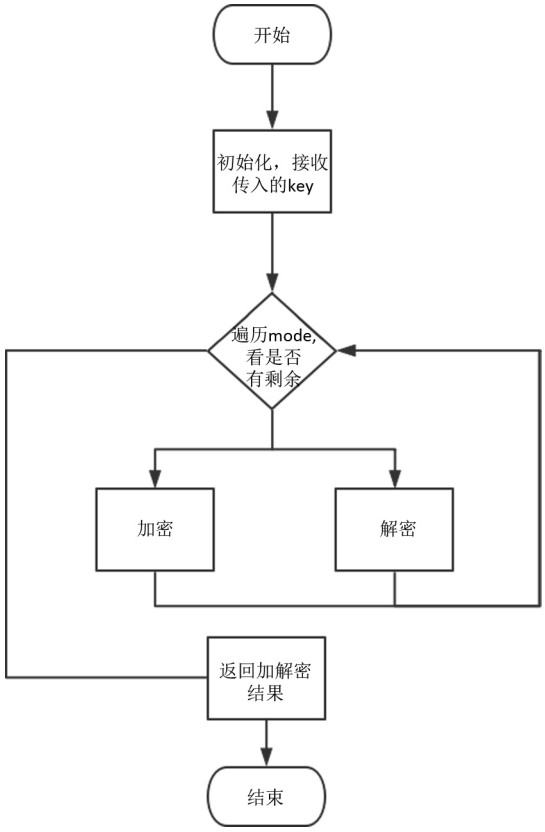

Figure 10. User configuration parsing flowchart
图 10. 用户配置解析流程图

加密模块的特点是它是一个标准化的加密模块，用户继承并实现其中的__init__，encode 和 decode 方法后，配置进密码管理模块即可使用。配置过程是将类文件放置在 cipher 目录下，并且修改 cipher_master 中的 self。cipher_dict 将类名和类的构造方法对应设置成 key-value 键值对。

3.6. 工具模块

工具模块[8]主要用于提供多个模块都需要使用的功能接口。工具模块的存在提高了代码复用率，优化了程序的代码结构。

在工具模块中实现了自定义的读写文件方法，输入方法和获取 sftp 连接的方法，从而统一了各个模块对文件读写的模式和标准。

3.7. 配置模块

配置模块主要用于提供多个模块都需要使用的固定变量和为多模块间通信提供一个媒介。例如，用户配置模块解析出的用户信息，需要通过配置模块来传递给密码管理模块。

配置模块中设定了本系统的名称，基础路径，并且动态获取了用户基础路径，根据用户基础路径生成了密码路径，缓存路径，密钥路径和临时文件路径，并且设定了远程服务器配置的存储路径。在这些内容之外，配置模块在运行过程中添加临时变量 user 作为传输媒介。获得用户配置的解析代码如图 10 所示。

4. 总结

考虑到现有的账户密码管理方式的各类弊端和安全问题，本文提出了一个新的解决思路针对有一定编程能力的使用者设计了一个基于 python 的跨平台的轻量级且安全便捷的账户密码管理软件。在本地化密码管理的基础上，通过自定义加密模块，加密逻辑降低应用或数据中心被攻击可能造成的损失，通过设定用户独有的远程服务器，从而在保证安全性的前提下让用户可以便捷地在多个终端间转移密码。

基金项目

江苏省大学生创新计划项目(201811998038Y)。

参考文献

[1] Zeriouh, M., Chillali, A. and Boua, A. (2019) Cryptography Based on the Matrices. *Boletim da Sociedade Paranaense de Matemática*, **37**, 75-83. https://doi.org/10.5269/bspm.v37i3.34542

[2] 范明钰, 何新民. 密码领域专用语言研究[J]. 计算机科学与应用, 2019, 9(1): 157-165.

[3] 姜文泽. 面向对象技术在软件开发中的应用[J]. 电子技术与软件工程, 2018(20): 63-64.

[4] Manteuffel, C., Avgeriou, P. and Hamberg, R. (2018) An Exploratory Case Study on Reusing Architecture Decisions in Software-Intensive System Projects. *Journal of Systems and Software*, **144**, 60-83. https://doi.org/10.1016/j.jss.2018.05.064

[5] 邱菊, 叶志锋, 赵永平. 基于 Python 语言的余度特性分析与应用[J]. 南京师范大学学报(工程技术版), 2018, 18(4): 80-85.

[6] 钟睿. 设计模式在软件设计中的应用[J]. 电子技术与软件工程, 2018(14): 27-27.

[7] 许福, 郝亮, 陈飞翔, 李冬梅, 崔晓晖. 面向开源代码复用的程序比对分析方法[J/OL]. 计算机工程: 1-8.

[8] Cui, H.-T., Xu, L. and Wang, G.-D. (2011) Application of Last Pass Force Lock-on Method to Rolling Schedule Calculation of Medium Plate. *Iron & Steel*, **46**, 53-55.

[9] Maurya, R.K. (2015) Using a Password Manager to Access Credentials. *Pc Quest*, **30**, 59-59.

[10] Stuart, R. (2013) Read Passwords More Easily. *Computer Active*, **28**, 46.

知网检索的两种方式：

1. 打开知网页面 http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD
   下拉列表框选择：[ISSN]，输入期刊 ISSN：2161-8801，即可查询
2. 打开知网首页 http://cnki.net/
   左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：http://www.hanspub.org/Submission.aspx
期刊邮箱：csa@hanspub.org