2020 年全国大学生信息安全竞赛

作品报告

作品名称：击键识人—基于键盘与鼠标击键行为的认证系统

电子邮箱：1404574966@qq.com

提交日期：2020 年 8 月 13 号

填写说明

1. 所有参赛项目必须为一个基本完整的设计。作品报告书旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。

2. 作品报告采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。

3. 作品报告中各项目说明文字部分仅供参考，作品报告书撰写完毕后，请删除所有说明文字。(本页不删除)

4. 作品报告模板里已经列的内容仅供参考，作者可以在此基础上增加内容或对文档结构进行微调。

5. 为保证网评的公平、公正，作品报告中应避免出现作者所在学校、院系和指导教师等泄露身份的信息。

目　　录

摘　要

当前信息安全局势变得越来越严峻，例如QQ被盗号、用户名密码被窃取等现象层出不穷；而身份认证技术作为信息系统安全的第一道屏障，它发挥着非常重要的作用。传统的基于知识或令牌的身份认证技术，存在着用户名密码、令牌等容易被窃取等缺点，而指纹识别、人脸识别等新兴技术需要部署一些昂贵的传感器；最重要的是，他们都是一些单次认证技术。针对以上问题，我们设计的基于鼠标与键盘击键行为的身份认证系统，无需额外的辅助设备，仅依靠用户产生的鼠标与键盘的击键特征，便可对用户使用信息系统的每一个过程进行认证，时刻防止系统被非法人员使用。

经过我们调查，发现当前一些已有的击键认证研究或作品，他们存在着认证准确率有待提高、认证方案设计不合理、认证时间过长、没有采用安全可靠的信息传输方式等缺点。为克服这些缺点，我们设计了我们的作品，创新性如下。

一、首个面向用户使用系统全过程的基于鼠标与键盘击键行为的认证作品。相比于传统的认证作品，本作品不仅可以实现在登录时对系统进行认证，而且还在用户使用系统的过程中实现对系统的监控，创新性地实现了对系统全方位全过程的保护。

二、基于加权贝叶斯分类与欧式距离结合算法的静态认证模块。该算法不仅考虑到了不同特征的重要程度不同，而且还可以控制认证系统的可用性与安全性，根据需要去改变。最终算法分类准确率达到了97.27%。

三、创新性地设计了基于信任得分机制的持续认证方案。克服了传统的滑动窗口式认证方案的弊端，快速准确的识别用户的击键行为并给出结果，当用户信任得分低于阈值则强制退出系统。核心算法最小二乘支持向量机的分类准确率达到了96.50%。

四、首个基于国密SM9算法与套接字的认证系统分布式智能终端网络。基于套接字进行网络传输，使用SM9对数据进行加解密，保证了网络传输数据的安全性与可靠性。除此之外为了方便用户与管理员使用，我们还开发了微信小程序——**击键认证小助手**，以及**后台管理员网站**，更加方便了用户与管理员的使用。

本作品以贝叶斯算法、支持向量机、国密算法为基础，设计了基于鼠标与键盘击键行为的认证作品，为击键认证领域提供了一个新的解决方案和设计思路。

关键词：击键认证，全过程认证，贝叶斯分类，支持向量机，国密SM9算法

第一章 作品概述

本章从作品背景分析、相关工作、研究目标、特色描述和应用前景分析五个方面进行介绍。

1.1 背景分析

随着时代与信息技术的发展，互联网已经深刻的影响到了生活的方方面面，信息安全已经成为国家安全的重要组成部分，信息安全若得不到充分保障，军事、政治、经济等领域都将受到严重危害。据 CNNIC 最新发布的 2020 年第 45 次《中国互联网络发展状况统计报告》，截至 2020 年 3 月，中国网民规模达 9.04 亿，互联网普及率为 64.5%。IT 技术的蓬勃发展不仅使得人们生活质量得到提高，更为计算机信息安全带来了机遇与挑战。2016 年 9 月，雅虎 5 亿账户信息泄露，包括用户号码、出生日期等重要信息。2016 年 10 月，美国东海岸出现大面积断网，推特等重要网站无法访问。2017 年 5 月至 7 月，美国信用机构 Equifax 遭到黑客入侵，近 1.43 亿名用户的个人隐私被窃取。在这个高度信息化的社会，网络犯罪对公民资产与社会安全构成了严重的威胁。**有效的身份认证是保障数据信息安全的第一道防线，具有举足轻重的地位。**，如图 1-1 展示了我国互联网发展状况。

图 1-1 截止 2020 年 3 月我国网民规模图

在信息系统中，身份认证是对用户身份进行确认的行为，是通往信息系统的一道门，确保当前用户具备对资源进行访问或执行其他操作的权限，能够防止非法用户的入侵、窃取、破坏等恶意行为，是保障信息系统访问控制策略能够有效执行的先决条件，为信息系统的安全提供了保护[1]。我国近些年来网络安全事件频发，如图 1-2 所示。

图 1-2 近年来的安全事件

用户身份的认证一般分为三类：基于知识的认证方法、基于令牌的认证方法与基于生物特征的认证方法[2]。

基于知识的认证方法：例如密码（口令）、PIN 码、个人信息等；根据用户所知信息对其身份进行认证，缺点是**有关信息容易被黑客盗取利用**，从而产生不可估量的损失。

基于令牌的认证方法：需要用户提供相应物品作为认证依据，例如智能卡、信用卡或者迷你设备等；令牌不易被伪造，但是携带令牌增加了用户的负担，同时**令牌也易于丢失或被窃取**。

基于生物特征的认证方法：是根据用户自身的生物特征进行身份鉴别，例如指纹、虹膜、声音等。人们所固有的生物特征**具有不会丢失、难伪造、无需携带**等优势，因此生物认证相比于传统的密码或令牌认证更为安全可靠，能够满足时代的需求。例如指纹认证、面部识别等功能已经逐渐投入商用，并为广大用户所接受。同时，像步态

识人，击键识人这种生物行为特征认证方式，目前也在探索之中。

如今，个人计算机已经广泛普及，键盘与鼠标成为人们非常重要的输入设备，不同用户使用键盘与鼠标的行为会受到生理因素以及行为习惯的影响而体现出差性。所以，以用户的击键行为与鼠标行为作为认证依据具有一定的优势。相比于其他通过生理特征的生物认证方式，比如指纹识别，人脸识别等；**通过键盘与鼠标认证，无需部署生物传感器之类的设备，具有低成本、高可用、易普及、用户交互性良好等特性，**使之在近年逐渐发展起来。从安全性、低成本和通用性等方面考虑，研究击键认证技术是一个必然趋势[1]。

通过键盘与鼠标认证，主要包含两个部分，一个是在登录时，依据固定文本进行**静态认证**。另一个是登录之后，用户使用时，依据自由文本对用户进行**持续认证（动态认证）。**

1.2 相关工作

在基于鼠标行为与击键行为的身份认证研究领域，国内外对此均进行了相关的研究，在近几年，主要包括：

相关论文分析：

（1）2018 年，陈功[1]等人提出了同时依据用户击键与鼠标行为进行动态认证的方案，他们利用核主成分分析法从众多特征抽取了数十个特征作为模型的输入，利用最小二乘支持向量机与加权极限学习机作为分类模型，最后得到了 99.76%的准确率，但是认证时间确需要将近 20 秒。

（2）2019 年，刘梦昕[2]针对依据用户击键行为进行静态认证的问题，使用卷积神经网络，在公开数据集上达到了 96.8%的准确率；针对依据用户击键行为进行静态认证的问题，指定奖惩机制，对用户每个动作计分，使用决策树模型，达到了 93.94%的平均准确率。

（3）2018 年，郑航[3]等人使用加权贝叶斯算法与欧氏距离算法结合，针对依据用户击键行为进行静态认证的问题，错误拒绝率达到 3.38%，错误接受率 2.64%。

（4）2017 年，Mondal[4]等人针对击键行为作为持续认证方案的问题，提出了 Dynamic Trust Model（DTM）模型，对每一个用户设计相应的信任得分，用户击键的每一个动作会使得信任得分增加或减少，若分数低于某一个阈值时则强制退出系统。

相关作品分析：

目前，市面上与击键认证或鼠标认证的相关产品，都还处于探索或测试优化阶段，典型的如下所示：

（1）Bio Tracker 识别系统[5]:2017 年开始，美国国防部计划采用该系统作为身份验证方法。它通过记录用户的击键速度，击键风格以及鼠标的使用进行生物验证。但目前该系统还处于测试阶段，以辅助认证的方式出现。

（2）Keyin 识别系统[6]：由集赢智能公司研发，通过研究用户击键模式，来完成对用户的认证，目前，通过对 5000 份样本测试，验证率达到了 95.7%。目前该系统还在测试中，并未商业化落地。

如图 1-3 所示，展示了一些典型的鼠标键盘图案。

图 1-3 鼠标键盘图

通过对相关产品或相关论文的调研分析发现，击键认证或鼠标认证系统存在如下问题：

（1） 认证准确率有待提高

对于固定文本的静态认证准确率最高达到了 96%左右，而动态认证准确率最高只达到 93%左右（有的甚至不足 80%），低于欧盟规定的商用标准。这个我们应该通过精准选取输入特征与优化分类模型或算法来提高。

（2） 认证时间过长

目前，大多研究使用的动态认证方案都是**滑动窗口**[1, 3]式的，意味着模型需要多

个特征同时作为输入，模型才可给出结果；这表示，**如果一个用户使用时，长时间没有产生某个击键或鼠标特征，则系统长时间无法对用户进行认证。**同时，有的系统由于分类模型设置的过于复杂，也导致认证时间过长。

（3） 动态认证方案不合理

目前的动态认证方案，只要模型有一次判定为非正常值，就强制用户退出系统，这样偶然性因素太大；事实上，我们可以奖惩计分[4]的方式，持续多次认证，会更客观一些。

（4） 击键或鼠标行为特征选取不够精准

在进行动态认证时，应该选取具有足够的区分度，能最好的区分不同用户的特征，同时特征不适合选取过多，这样会导致时间过长。而如今，有的研究为了提升准确率，而选取了数十个特征，这样导致认证时间过长。有的研究选取为了提升速度，选取特征过少，导致模型精确率下降。

（5） 没有采用安全可靠的信息传输方式

当前，一些系统传输信息时简单的通过套接字，没有加密[2]或采用国外的加密方式，使得算法设计和算法实现方面安全性，稳定性低，做不到平台的自主自控。

综上所述：本作品通过采用目前准确率较高，处理速度快的**最小二乘支持向量机模型与加权贝叶斯分类模型**，来进行分类，从而提升认证准确率；动态认证时，采用了**基于奖惩机制的认证模型**，精准地选取了例如双键敲击时间间隔等**用户间差异化巨大的特征**，对用户的每一个常规动作进行判断，大大减少了认证等待时间与认证时间，更为贴近了实际生活。同时，编写了底层的**钩子程序**来收集有关鼠标和击键信息，作为数据收集的工具。同时，数据传输时使用 **SM9 国产加密算法**，保障了数据传输时的安全，增强了系统安全性。**最终，设计成了一个贴近用户实际使用情况，对用户使用信息系统全过程进行认证，安全高效，简单便捷的基于击键与鼠标特征的认证系统。**

1.3 研究目标

本作品的目的是构建一个基于鼠标与键盘击键行为特征的、对用户使用信息系统地全过程进行认证的身份认证系统。该系统可以通过用户的键盘与鼠标击键行为，不仅在用户登录系统时，也在用户使用系统的过程中对用户进行认证。

以静态认证作为用户的登录验证方式，针对固定文本（用户设定的密码），根据用户输入密码的方式对用户进行击键认证；以动态持续认证作为用户使用电脑过程中的

认证方式，对用户输入进行有效的键盘行为或鼠标行为击键特征提取，建模并对用户进行持续的身份认证，一旦认证失败，则返回系统登录界面。同时，无需提供任何昂贵的辅助设备，只是常用的键盘与鼠标，即可完成对用户的身份认证。

具体如下：

（1） 编写钩子程序，完成对鼠标行为数据与击键行为数据的捕获的预处理。

（2） 用户登录时，建立静态认证模型，准确地对用户进行认证。

（3） 用户通过登录验证后，使用信息系统时，建立基于奖惩机制的持续动态认证模型，准确、便捷、快速对用户进行认证。一旦认证失败，则返回系统登录界面。

（4） 设计基于套接字与 SM9 国密算法的信息传输方式，确保系统安全通信与自主可控。

1.4 特色分析

本作品从实际需求出发，提出了一种新型的基于鼠标行为与键盘行为的身份认证系统，并且结合了当前的一些先进技术，并对其进行创新与融合，以下为本作品的几大特色：

（1）首次提出了基于鼠标与键盘击键行为的全过程身份认证系统。现在市面上的一些认证系统几乎全部都是一些单次认证的系统，即只在登录系统时认证，登入之后便不再认证，如果用户密码被盗或者用户短暂地离开了所使用的信息系统，入侵者可以无差别地使用用户的信息系统。而本作品，不仅在登录时对用户验证，还在使用时对用户认证，而且以鼠标与键盘击键行为这一新型的方式作为认证手段，保证了信息系统使用全过程的安全，**成本低廉，安全便捷**。

（2）**采用了先进高效的模型算法对数据进行分类，大大提升了认证准确率**。静态认证时，本作品使用加权贝叶斯分类与欧氏距离结合的模型进行分类；持续动态认证时，使用最小二乘支持向量机作为分类模型。实验证明，综合考虑认证精确度与认证效率的情况下，与现有的其他方案比对，本作品均为最优。

（3）**创新性地提出对每一个击键或鼠标动作进行判断的持续认证方案，采用奖惩机制，突破了传统的滑动窗口式的方案，大大提升了模型的实用性，显著地提升了作品的认证效率。**由于传统的持续认证模型需要多个特征作为输入，这使得认证过程非常漫长，需要很长的时间等待用户输入（如果用户不产生某个特征，则永远无法认

证）。本作品精准地选取了部分特征，几乎为用户的每一个动作进行认证，**实用高效**。

（4）**首个基于 SM9 与套接字的认证系统数据安全传输方式**。使用国密算法，安全自主，保障了数据传输的安全。

1.5 应用前景分析

随着信息技术的发展，为保障信息系统的安全，身份认证变得越来越重要。传统基于知识或令牌的身份认证方式具有容易被窃取等弱点，而基于生物生理特征的指纹识别与人脸识别等需要部署专门的设备，成本较为高昂。

基于鼠标与键盘击键行为的认证方式，具有部署成本低，使用便捷，不易丢失等特点，且可以对用户使用全过程进行认证。具有一定的应用前景。

2017 年，美国国防部计划逐渐采用 Bio Tracker 这一基于鼠标行为与击键行为的系统进行认证，去取代传统的基于卡片的识别访问控制的身份验证解决方案。相关开发公司 Plurilock 公司表示，BioTracker 是一个行为生物识别解决方案，是检测员工或入侵者盗用登录凭证的最佳方式。即使入侵者获取了国防部员工的密码，要复制击键风格和速度也是不可能的[5]。由此可见基于鼠标行为与击键行为的认证方式具有相当地应用前景。但由于目前该项技术还处于测试阶段，以辅助认证的方式接入。

1.6 小结

在本章，对作品做了一个全面整体的概述，包括作品背景分析、相关工作、研究目标、特色描述和应用前景分析等。着重对作品创新处进行了说明，大致说明了本作品的主要内容与优势。为后续作品设计与实现奠定基础。

第二章 作品设计

本章从本作品的整体方案设计、架构设计、关键技术和功能模块设计四个方面进行阐述。

2.1 系统方案设计

我们设计的基于击键行为与鼠标行为的身份认证系统主要分为五个角色，分别是PC客户端，数据收集模块，信息传输加密模块，静态认证模块，持续认证模块。本作品的总体流程设计如图2-1所示。

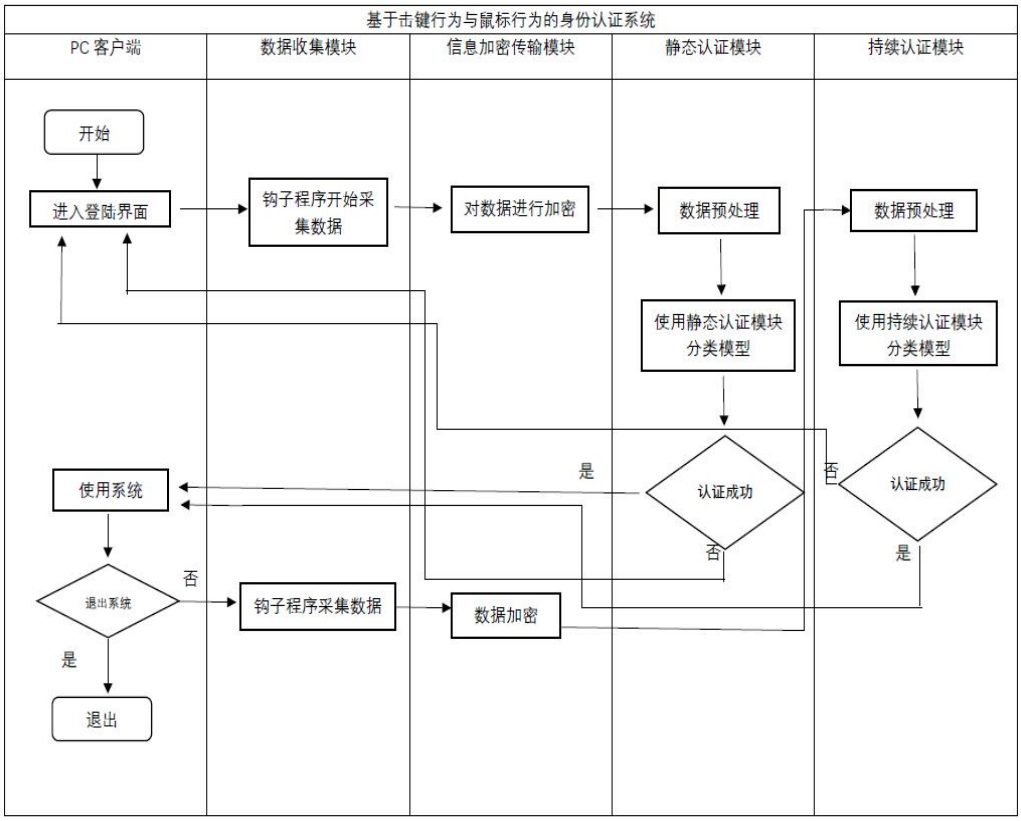

图2-1 基于键盘与鼠标击键行为的认证系统的流程设计

（1）PC客户端：主要完成对用户的可视化图形界面显示，注册登录时，给出相应的界面；认证失败时，给出相应提示。

（2）数据收集模块：使用Windows编程的HOOK技术编写对应数据收集的钩子程序，完成对用户的击键行为与鼠标行为的数据收集。

（3）信息传输加密模块：使用SM9加密体系与套接字技术，完成对数据的可靠安

全传输，确保传输的过程中不会被他人窃取。

（4）静态认证模块：该模块主要用于用户登录时的认证，当用户输入一串密码时，通过提取用户的击键特征，调用分类模型去判断是否认证成功，并反馈结果。同时，在分类模型训练成功之前，需要收集有关数据完成对模型的训练。

（5）持续认证模块：该模块主要用于用户使用过程中的认证，用户使用系统时会产生相应的鼠标与击键行为动作，通过提取用户的动作特征，调用分类模型去判断是否认证成功，并在基于奖惩机制的认证模型，扣除或增加相应分数，如果最终低于阈值，则强制返回登录界面，反之，继续使用系统。同时，在分类模型训练成功之前，需要收集有关数据完成对模型的训练。

除了以上模块,为了考虑进一步方便用户和系统后台管理员对系统的控制与使用，在系统开发的过程中，基于微信这一被广泛使用的社交工具，我们还开发了**微信小程序—击键认证小助手**，方便用户对自己的PC电脑的认证过程进行管理，实现的功能有用户登录、更改登录密码、强制进入系统、强制用户退出系统、查看我的系统使用情况等功能；同时为了便于后台管理员，管理庞大的用户数据，我们还开发了**后台管理员网站**,实现对系统中所有用户有关信息的管理、数据库的增删改查等。

2.2 系统总体架构设计

基于击键行为与鼠标行为的身份认证系统总体架构设计由网络架构设计和软件结构设计两部分组成。

2.2.1 网络架构设计

网络架构上,基于击键行为与鼠标行为的身份认证系统采用分布式终端与C/S结合的体系架构。

PC客户端，指的是被实施击键认证的对象（被保护的对象），主要完成击键行为与鼠标行为的数据采集工作，并将有关数据加密传输给服务器端。同时，用户可以仅仅根据自己的击键行为与鼠标行为的完成认证，防止有人非法登录个人的信息系统；或者是防止在登录之后，用户中途有事离开，攻击者非法使用自己的信息系统。

服务器端，解密并处理从服务器端发来的击键行为与鼠标行为的数据采集，调用静态认证或持续认证模型判断并反馈结果。在分类模型建立之前，则不断收集数据完成对模型训练工作。同时，根据微信小程序—击键认证小助手、后台管理员网站发来的有关命令与请求，对用户数据库进行有关处理，控制对应的PC端的行为。

微信小程序—击键认证小助手，输入对应的用户民与密码，登录我们的击键认证小助手，会根据用户的有关要求，发送有关命令给服务器，进而控制自己的PC客户端。借助微信这一工具，在移动手机端上就可控制自己的PC电脑的使用，简单、方便、灵活，极大地增进了用户体验。

后台管理员网站端，后台管理员可以登录，从而管理整个系统，查看修改所有的用户信息，从而可以进行增删改查等有关操作。

本作品的主体网络架构图（PC客户端与服务器），如图2-2所示。

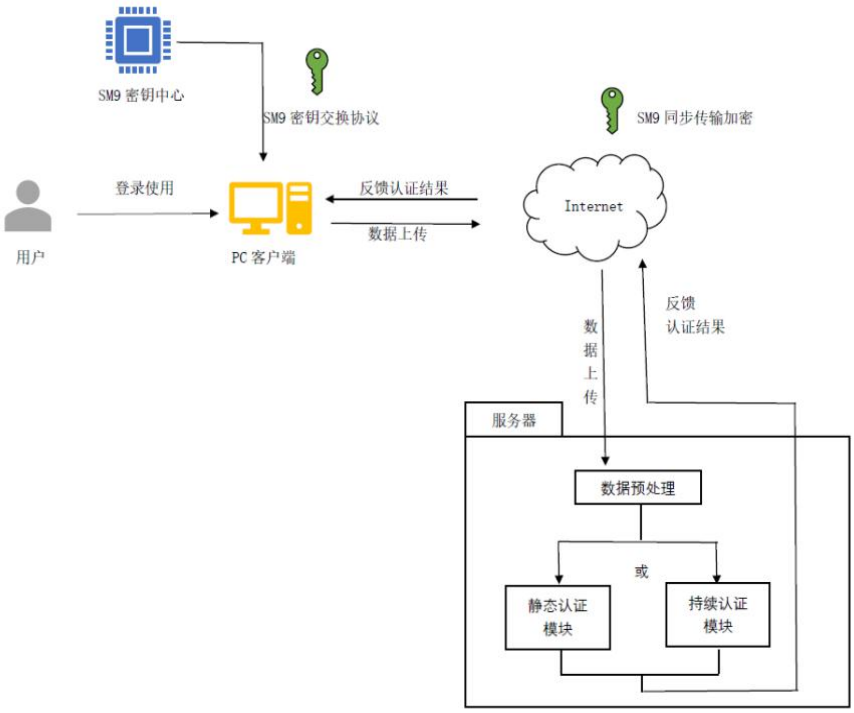

图2-2 基于鼠标与键盘击键行为的身份认证系统的软件架构设计

（1）客户端通过产生与收集击键或鼠标行为数据去认证

用户使用时，马上开启钩子程序进行全程监听并收集有关数据发给服务器端，首先是一个登录界面，用户输入密码进行认证；如果认证成功，用户开始使用电脑，如果在使用过程中，用户击键或鼠标行为产生了一些不合“常规”的动作，使得用户的认证得分低于阈值，则用户被强制退出系统；反之，则继续正常使用。

（2）服务器端调用对应模型去判断是否认证成功并反馈结果。

服务器，接收客户端钩子程序发来的数据，并进行处理传入对应模块。登录时，用户根据密码正确与否以及击键特征正确与否进行双重认证。用户使用系统时，调用分类模型判断用户是否认证成功，并根据这个结果修改用户的认证得分，如果修改后，

认证得分低于阈值，则强制退出系统。同时，还要根据微信小程序一击键认证小助手、后台管理员网站发来的有关命令与请求，对用户数据库进行有关处理，控制对应的PC端的行为。

（3）微信小程序端根据自己的需求管理自己的PC电脑。

用户可以通过微信小程序一击键认证小助手,查看PC的使用情况,如发现有人正在使用，可以点击“强制退出系统”按钮，强制关闭系统，让它回到登录界面上来；如果发现系统正常登录不进去或是其他需求，可以点击“强制进入系统”，免除登录时的认证；同时还可以进行修改密码、设置认证强度等其他操作。

（4）后台管理员网站端实现对整个系统所有用户数据的管理。

后台管理员可以登录对应的网站，管理整个系统的数据，进行对数据的增删改查等操作，控制有关用户行为，是维护整个系统的重要手段，保障整个系统各方面的正常运行。

2.2.2 软件架构设计

基于击键行为与鼠标行为的身份认证系统软件结构设计分为五个层次，应用层，接口处，逻辑层，网络层以及存储层。

本作品的软件架构，如图2-3所示。

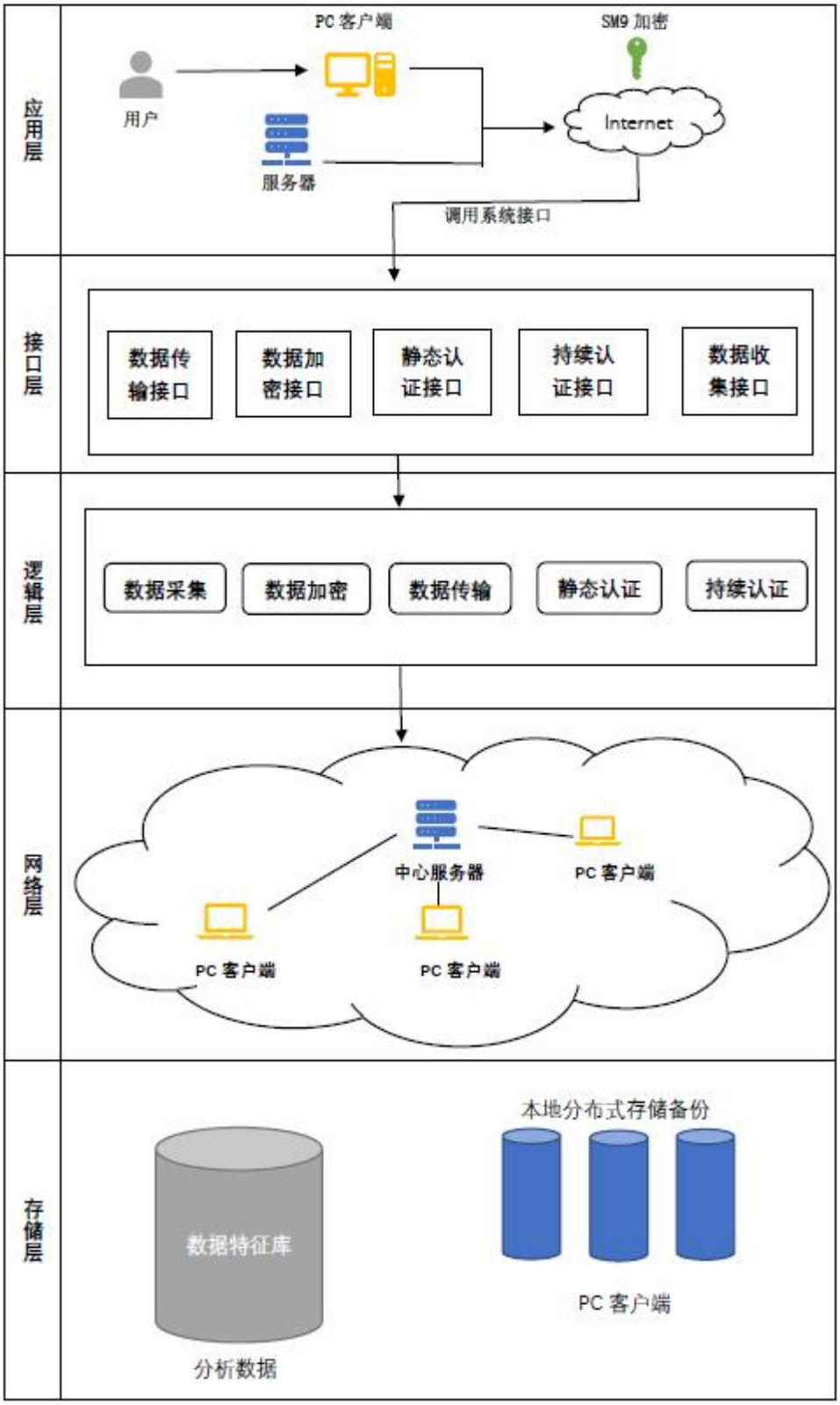

图2-3 基于鼠标与键盘击键行为的身份认证系统的软件架构设计

（1）应用层

应用层由分布式客户端与服务器组成，通过调用接口层的接口，与网络层进行通信，一些关键数据由SM9加密，保护数据安全。客户端部署在PC机上，保障PC系统使用的安全。

（2）接口层

通过实现一些接口以供上层调用，包含了数据传输接口，数据加密接口，静态认证接口，持续认证接口，数据收集接口。

（3）逻辑层

负责实现一些具体的逻辑功能，由系统的主要功能组成，比如:数据采集，数据加密，数据传输，持续认证，动态认证等逻辑功能。

（4）网络层

本作品网络层主要采用的是中心化的C/S架构，客户端收集有关数据，服务器进行处理并反馈结果。

（5）存储层

服务器存储一些有关认证数据，而在模型未建立时，还要存储模型的训练数据。为了防止中心化的一系列弊端，客户端也做了有关数据的备份。

2.2.3 基于套接字的数据传输过程设计

为了实现客户端与服务器的通信，需要用到socket技术。Socket又可称为套接字，应用程序常常通过套接字，来完成发送网络请求或应答网络请求。而套接字又可分为流式套接字（基于TCP）与数据报套接字（基于UDP）等，为了实现稳定可靠传输，本作品选用流式套接字。具体的过程如下：

（1）服务器端监听：一开始，服务器创建的监听套接字，等待客户端发来连接。同时，为了能够让客户端连接到，服务器的套接字地址应该是公开的。而客户端的套接字一般是不公开的，这也是服务器需要等待连接的原因。

（2）客户端发送连接请求：客户端创建套接字，由于服务端监听套接字公开，客户端可以直接向服务端发起连接。

（3）服务器响应连接：服务器的监听套接字，创建一个子线程，在子线程中，再创建一个连接套接字，负责专门处理与该客户端的连接请求，同时将该新套接字的描述发往客户端。

（4）开始通信：客户端收到后，意味着与服务器的连接建立成功，以后双方可以进行数据传送了。同时，服务器的监听套接字继续监听，等待其他客户端的连接。

客户端套接字传输过程，如图2-4所示。

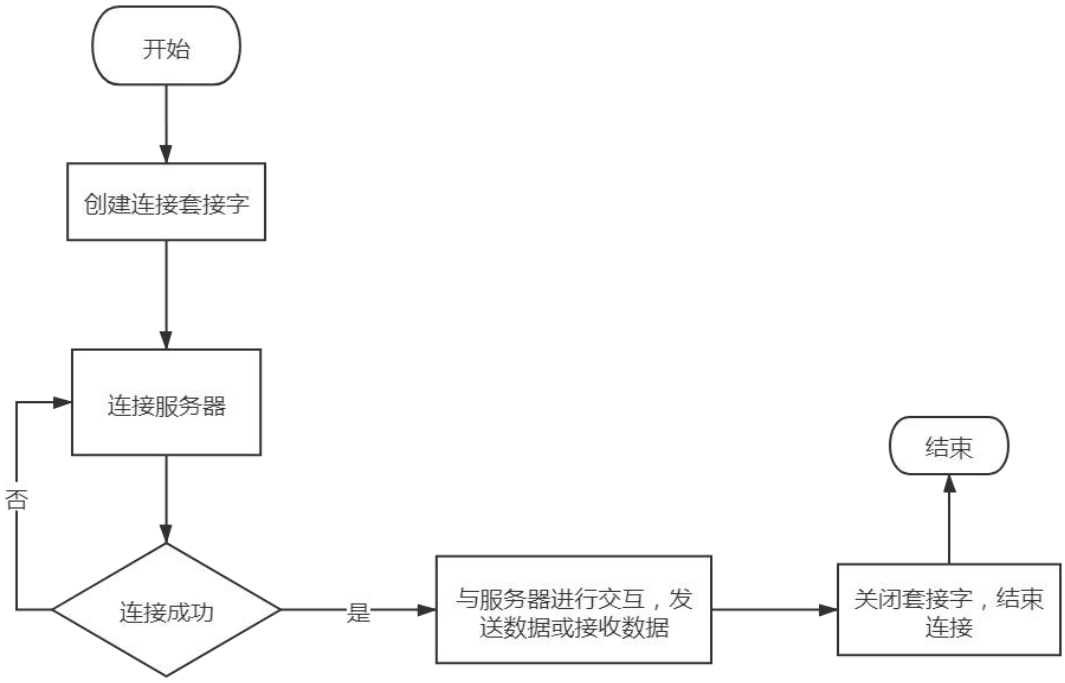

图2-4 客户端套接字传输过程

服务器套接字传输过程，如图2-5所示。

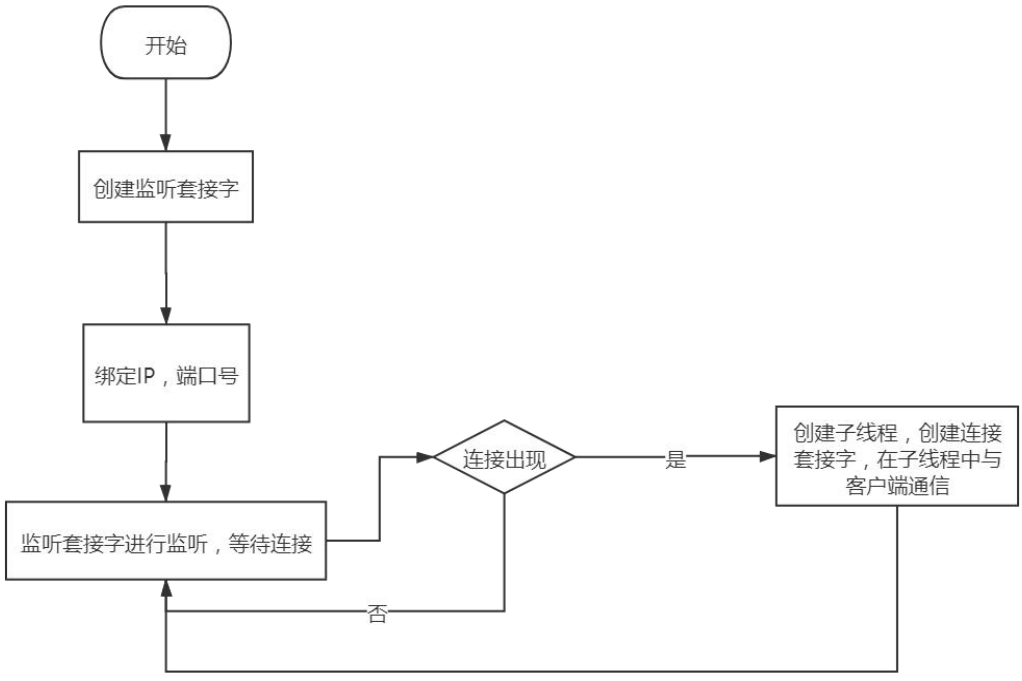

图2-5 服务器套接字传输过程

2.2.4 基于SM9的数据加密过程设计

基于SM9的数据加密过程，分为密钥申请，数据加密（发送时），数据解密（接收时）。

2.2.4.1 密钥申请

（1）PC 客户端首先向服务器发送接入请求消息，服务器向用户返回请求响应消息，该消息中包含随机生成的用于认证加密的挑战字等信息，PC 客户端在收到响应消息之后，将挑战字、设备号、IP 地址和 MAC 地址等数据打包并加密该数据，然后将加密数据发送给身份认证中心，认证中心接收并解密数据，认证合法性。若用户通过认证，则将准许接入的消息返回给 PC 客户端。

（2）PC 客户端向服务器发送申请通信私钥消息。

（3）服务器在收到申请消息后，加上数字签名并转发给密钥中心。

（4）密钥中心接收消息并确认信息来源，生成用户的公钥与私钥对，在一个安全会话环境下，将通信公钥与私钥对发送给 PC 客户端，PC 客户端接收通信密钥，并将其中私钥保证安全存放。至此，后面的工作过程已不再需要密钥中心，密钥中心可以离线[7]。

密钥申请过程，如图 2-6 所示。

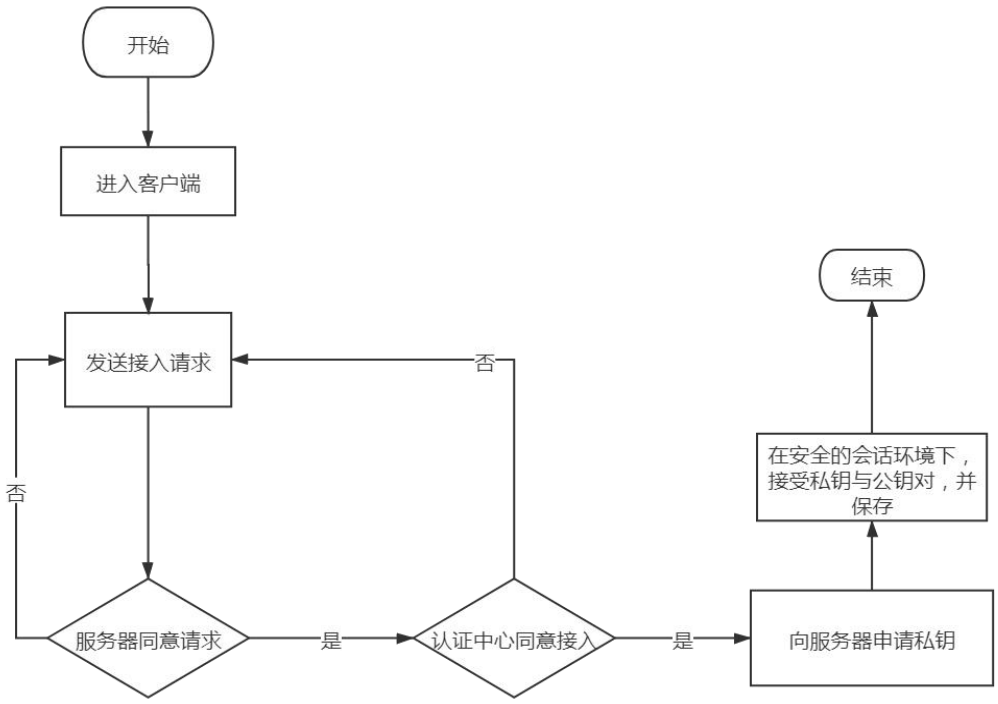

图 2-6 密钥申请流程图

2.2.4.2 数据加密（发送时）

在完成密钥申请的基础之上，基本过程是将钩子程序收集的数据，先用SM9密钥加密，最后通过套接字传输。

数据加密过程，如图2-7所示。

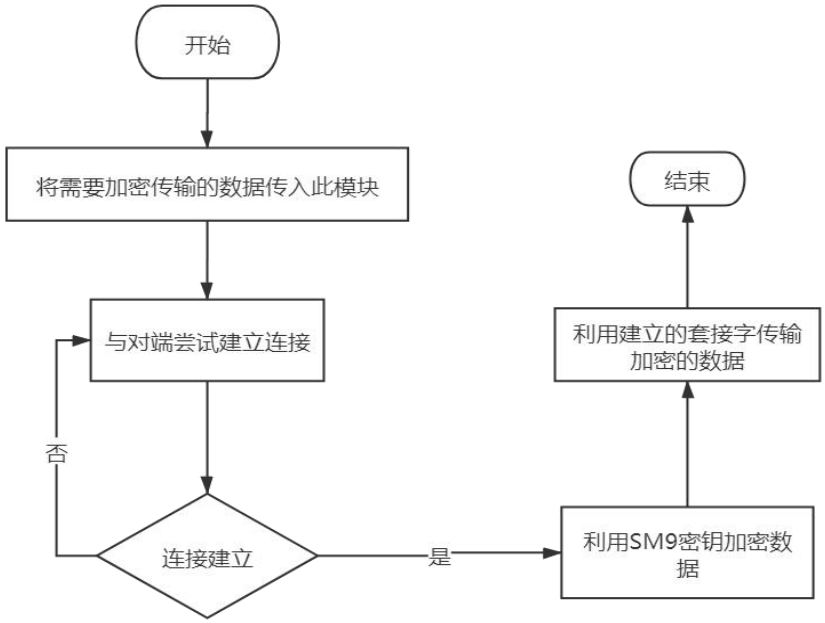

图 2-7 数据加密流程图

2.2.4.3 数据解密（接收时）

在完成密钥申请的基础之上，先用套接字接收数据，再用约定的SM9密钥解密，从而查看信息。

数据解密过程，如图2-8所示。

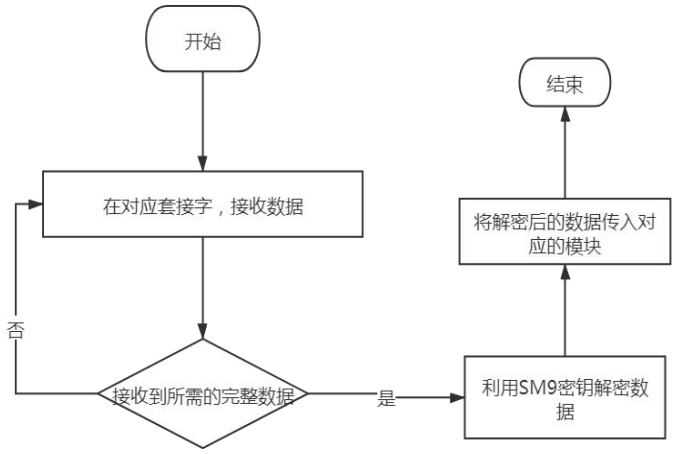

图 2-8 数据解密流程图

2.3 系统关键技术

2.3.1 基于HOOK技术的数据收集方法

2.3.1.1 HOOK技术基础知识

HOOK（钩子，挂钩）是一种实现 Windows 平台下类似于中断的机制。HOOK 机制允许应用程序拦截并处理 Windows 消息或指定事件，当指定的消息发出后，HOOK 程序就可以在消息到达目标窗口之前将其捕获，从而得到对消息的控制权，进而可以对该消息进行处理或修改，加入我们所需的功能。本作品中需要对鼠标键盘发出的消息进行捕获，用于后续的分析。

Windows 下操作系统，应用程序，计算机硬件关系的原理如图 2-9 所示。

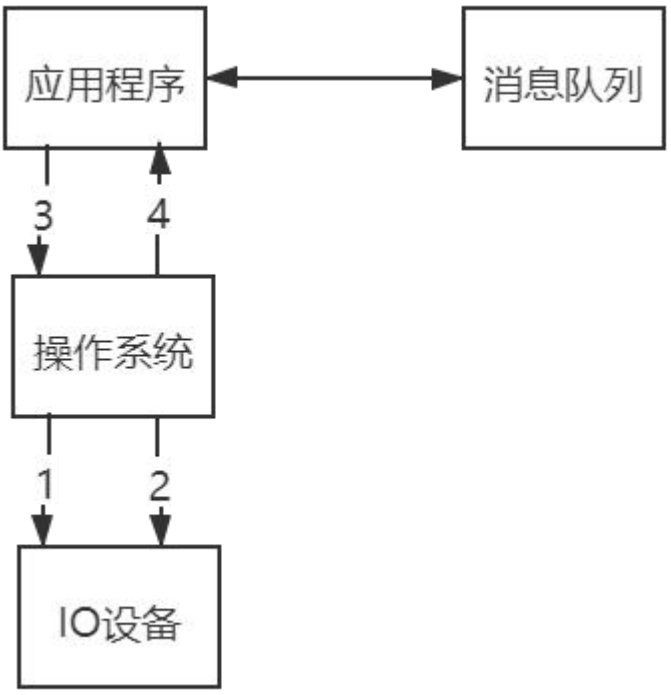

图 2-9 Windows 下操作系统，应用程序，计算机硬件关系图

钩子按使用范围分，可分为线程钩子和系统钩子，其中，系统钩子具有相当大的功能，几乎可以实现对所有 Windows 消息的拦截、处理和监控。这项技术涉及到两个重要的 API，一个是 SetWindowsHookEx，安装钩子；另一个是 UnHookWindowsHookEx，卸载钩子。

本作品使用的 HOOK API 技术，是指截获系统对某个 API 函数的调用，使得 API 的执行流程转向我们指定的代码段，从而实现我们所需的功能。Windows 下的每个进

程均拥有自己的地址空间，并且进程只能调用其地址空间内的函数，因此 HOOK API 尤为关键的一步是，设法将自己的代码段注入到目标进程中，才能进一步实现对该进程调用的 API 进行拦截。然而微软并没有提供 HOOK API 的调用接口，这就需要开发者自己编程实现，大家所熟知的防毒软件、防火墙软件等均采用 HOOK API 实现[13]。

以鼠标钩子为例，运行原理如图 2-10 所示。

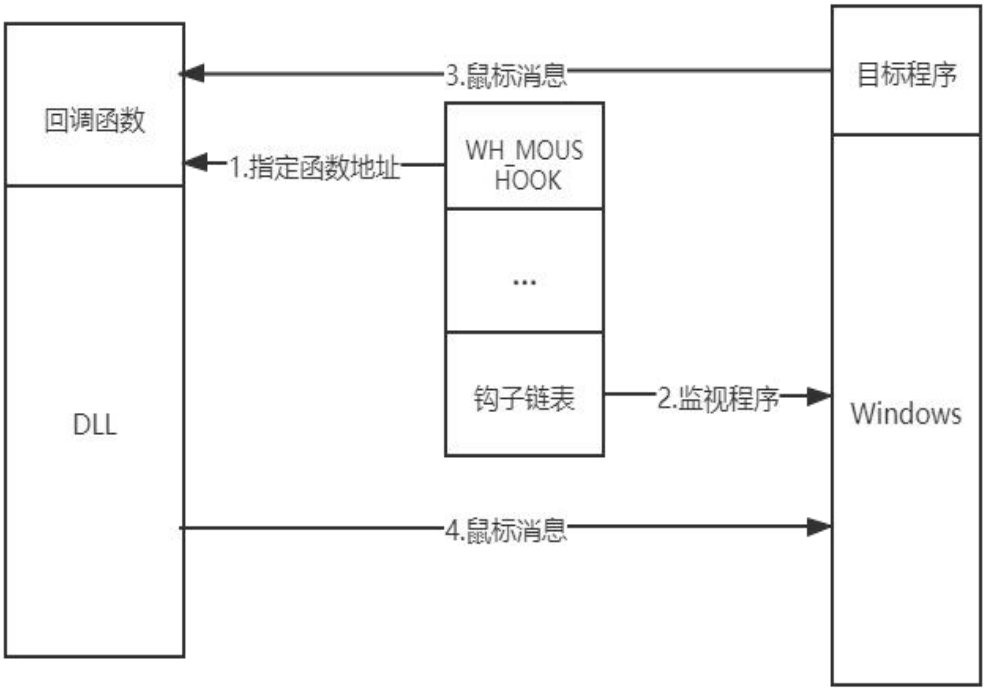

图 2-10 鼠标钩子的运行原理

一般来说，HOOK API 由两个组成部分，即实现 HOOK API 的 DLL 文件，和启动注入的主调程序。本文采用 HOOK API 技术对剪切板相关的 API 函数进行拦截，从而实现对剪切板内容的监控功能，同样使用该技术实现进程防终止功能。其中 DLL 文件支持 HOOK API 的实现，而主调客户端程序将在初始化时把带有 HOOK API 功能的 DLL 随着钩子的加载注入到目标进程中。

按事件分类，HOOK 钩子有如下的几种常用类型。

（1） 键盘钩子和低级键盘钩子可以监视各种键盘消息。

（2） 鼠标钩子和低级鼠标钩子可以监视各种鼠标消息。

（3） 外壳钩子可以监视各种 Shell 事件消息。比如启动和关闭应用程序。

（4） 日志钩子可以记录从系统消息队列中取出的各种事件消息。

（5） 窗口过程钩子监视所有从系统消息队列发往目标窗口的消息。

此外，还有一些特定事件的钩子提供给我们使用，不一一列举。

接下来再介绍几个重要的组件：

（1）SetWindowsHookEx()函数[8]：由windows API提供，它将一个应用程序定义的钩子进程安装到钩子链表中。当对应的钩子监视的事件发生时，系统就调用与这个钩子关联的钩子进程。

格式如下：

HHOOK WINAPI SetWindowsHookEx(

int idHook,

HOOKPROC lpfn,

HINSTANCE hMod,

DWORD dwThreadID);

其中，idHook指的是钩子的类型，也是他要处理的消息的类型，典型的有WH_KEYBOARD_LL（底层键盘钩子），WH_MOUSE_LL（底层鼠标钩子），WH_DEBUG（调试钩子）等等，而在本作品中由于要收集鼠标与键盘的消息，所以需要设置WH_KEYBOARD_LL(底层键盘钩子)与WH_MOUSE_LL(底层鼠标钩子)，用于捕获数据（系统按键）。

lpfn用于指向钩子子程，钩子捕获到任何函数都会调用这个函数，本作品设置的键盘lpfn命名为Keystroke,鼠标的为Mousestroke。

hMod是应用程序实例的句柄，标识包含lpfn所指子程的DLL。

dwThreadID是与安装的钩子子程相关联的线程描述符，当其为0时，为全局钩子，因此我们将其设置为0。

（2）钩子回调函数：即上文中的lpfn，以Keystroke为例，我们介绍一下该函数：

LRESULT WINAPI Keystroke(

int nCode,

WPARAM wParam,

LPARAM lParam)

其中，参数 wParam 和 lParam 包含所钩消息的信息，比如鼠标位置、状态，键盘按键值等。nCode 包含有关消息本身的信息，比如是否从消息队列中移出。在函数内部我们使用 GetTickCount()函数,它将返回从操作系统启动经过的时间,再经过处理，

我们可以收集有关按键的时间信息，完成后面的工作。

（3）卸载钩子：调用 UnHookWindowsEx()函数完成对鼠标，键盘钩子的卸载，从而结束相关数据收集工作。

2.3.1.2 本作品基于HOOK技术的数据收集方法

在本作品中，为了实现系统的功能，我们需要收集有关鼠标与键盘的有关操作行为。这个时候需要用到HOOK技术，编写钩子程序去实现。在实现时，我们将钩子程序封装在了动态链接库DLL中，从而方便将钩子程序载入系统。用户打开数据收集程序，即可完成钩子程序加载。从而正式开始收集数据。

具体步骤如下：

步骤一：创建DLL文件，在其中编写程序。

步骤二：调用SetWindowsHookEx函数分别设置鼠标，键盘钩子；即设置钩子类型为WH_KEYBOARD_LL（底层键盘钩子）、WH_MOUSE_LL（底层鼠标钩子），从而成功安装钩子。

步骤三：编写对应的鼠标钩子回调函数以及键盘钩子回调函数，用于对捕捉到的信息进行处理。

步骤四：调用UnHookWindowsEx()函数完成对鼠标，键盘钩子的卸载，从而结束相关数据收集工作。

步骤五：将以上涉及到的所有函数声明为外部可调用。

步骤六：对该DLL文件进行编译，生成对应的.h,.dll,.lib文件。

步骤七：创建目标程序，在该进程目录下，把上述DLL生成的.h,.dll,.lib文件放入同一目录即可。同时，在目标程序中调用对应的安装钩子函数，运行该程序，即可完成对整个系统的鼠标、键盘数据的收集。

通过上述步骤，可圆满的完成对指定数据的收集，其中的核心是钩子的回调函数，当我们捕捉到有关的鼠标、键盘信息时，进入该程序进行处理。通过编写回调函数里的内容，我们可以随心所欲的处理有关数据。

本作品的回调函数处理流程如下（以键盘钩子为例）：

步骤一：判断回调函数参数nCode是否为HC_ACTION,即是否发生键盘活动事件；若有，进入步骤二；否则，函数返回。

步骤二：判断回调函数参数lParam->vkCode是否为指定范围的值，即是否为我们

要收集的按键的信息；若有，进入步骤三；否则，函数返回。

步骤三：判断回调函数参数wParam是否为VM_KEYDOMN,即是否发生键盘按下活动；若有，先利用GetTickCount函数，返回从操作系统启动到当前所经过的时间，最后以（lParam->vkCode，KEYDOWN, GetTickCount）的形式存入指定文件，之后，函数返回；否则，进入步骤四。

步骤四：判断回调函数参数wParam是否为VM_KEYUP,即是否发生键盘抬起活动；若有，先利用GetTickCount函数，返回从操作系统启动到当前所经过的时间，最后以（lParam->vkCode，KEYUP, GetTickCount）的形式存入指定文件，之后，函数返回；否则，进入步骤五。

步骤五：函数返回。

以上便是本作品收集数据的原理与方法。

2.3.2 键盘与鼠标行为特征选取

为了完成对使用者的分类，区分合法用户与非法用户，我们需要从收集的有关鼠标键盘的数据上，提取有关特征。同时，我们需要精准选取特征，特征太多，速度过慢；特征过少，精确率低。我们应当选取一组能最大反映用户击键特征，同时不相互包含，数量不过多的特征组。

一些击键特征，具体如下图2-9所示：

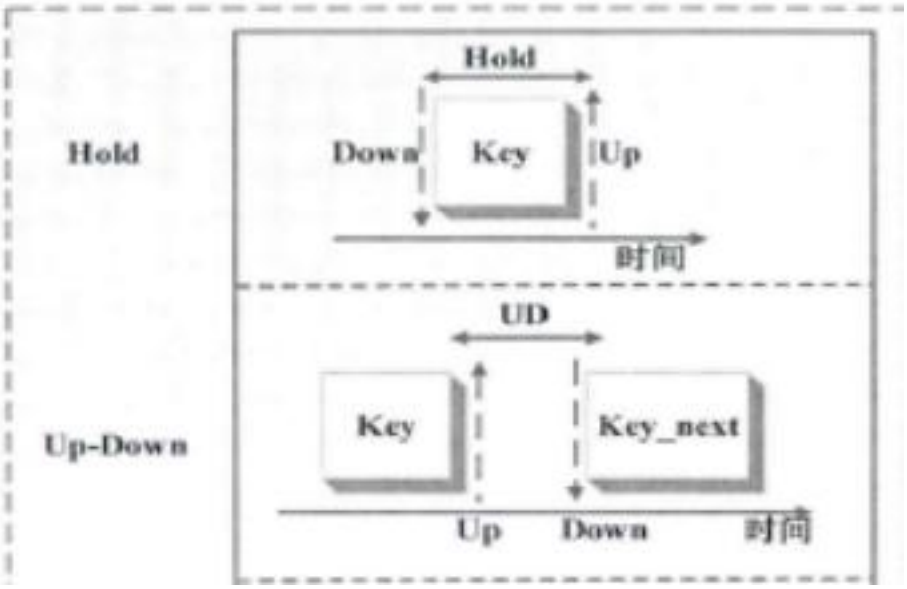

图2-11 击键特征

我们给出几个基本主流击键特征的定义：

（1） **击键延续时间（hold）**:表示击键的持续时间，也可称为延续时间。表示一个键从按下到抬起所经历的时间。

（2） **间隔时间（UD）**:即Up-Down。表示一个键松开到下一个键按下经历的时间。

这里我们需要强调的是，我们将对鼠标与键盘的敲击都看成是击键行为。

2.3.2.1 静态认证模块特征选取

静态认证主要完成系统登录时的认证，通过用户输入一段固定文本（密码），提取击键信息，从而完成认证。

在这里，我们以密码“hello”为例，我们选取的特征组为：

Hold:h,e,l,l,o（表示h,e,l,l,o五个键的持续时间）

UD：he,el,ll,lo（以he为例：表示从h键松开到e键按下的间隔时间）

总计9个特征作为分类模型的输入向量，同理如果密码长度为N，则输入向量的特征数为2*N-1。

2.3.2.2 持续认证模块特征选取

持续认证模块我们建立了十一个模型，前十个模型为常用的十个双键，最后一个为鼠标左键双击时间模型。

以双键an为例，输入向量为：

Hold:a,n与UD:an

共计三个特征，两个单键的持续时间，以及两者之间的间隔时间。我们通过数据采集程序，征集自愿者，经过统计，比较常用的十个双键如表2-1所示：

表2-1 双键统计

双键	出现次数
an	6036
in	5877
ai	4956
sh	4822
en	4819
cn	4752
zh	4689

wo	4587
ji	4322
ao	4210

同理，我们打开一个程序时，经常需要鼠标左键双击打开，我们选取的输入特征向量与上类似，分别是前后两次单机鼠标左键的持续时间，两次点击鼠标之间持续的时间，共计三个特征作为输入。

2.3.2.3 模型精准性说明

众多研究表明[1, 2, 3]，在鼠标与键盘众多击键特征之中，尤其是两次敲击间隔时间与敲击持续时间人与人之间差异非常巨大。

以单键平均持续时间为例，如表2-2所示：

表2-2 单键平均持续时间

按键	用户A/ms	用户B/ms
B	128.98	140.22
A	120.14	125.55
S	124.50	174.22
V	122.45	150.24

双键间隔时间差异则更加显著，如表2-3所示：

表2-3 双键平均间隔时间

按键	用户A/ms	用户B/ms
OP	150.26	782.04
SH	450.21	152.36
AN	458.21	654.27
CH	120.33	390.28

而与此同时，基于鼠标行为的研究[1, 3, 9]，鼠标左键的双击的间隔时间与持续时间组合判断，在一个分类效果不是特别好的分类模型之上可达88%以上的准确率，如果使用更好的分类模型，则结果更加可观。

综上所述，本作品所选取的特征具有一定合理性且能对准确率与认证效率产生一定的提示。

2.3.3 基于加权贝叶斯分类与欧氏距离的静态认证分类模型

2.3.3.1 朴素贝叶斯算法

贝叶斯分类算法是统计学的一种分类方法，它是一类利用概率统计知识进行分类的算法。在许多场合，朴素贝叶斯(Naïve Bayes，NB)分类算法可以与决策树和神经网络分类算法相媲美，该算法能运用到大型数据库中，而且方法简单、分类准确率高、速度快[14]。

先给出朴素贝叶斯算法，本作品选择的算法是在这基础之上进行改进的。

步骤如下：

步骤一：假设$x = \{t_1, t_2, \ldots, t_d\}$为一个待分类用户，其表示每个待分类用户都有着d个特征属性,即t_i。

步骤二：系统可能的用户集合为U=$\{U_1, U_2, \ldots, U_n\}$,表示有n个用户，每个用户存在d个特征属性。

步骤三：利用公示计算出x属于每个用户的U_i的概率，即$P(U_i|x)$。

步骤四：取 $P(U_i|x) = \max\{P(U_1|x), P(U_2|x), \ldots, P(U_n|x)\}$，将待分类用户x分入系统某个用户中。

该算法以贝叶斯定理为基础，计算待分类对象属于系统中每一个用户的概率，然后选择具有最大后验概率的类作为该对象所属的类，即通过对象先验概率计算其后验概率[3]。

其中贝叶斯定理如下：

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \tag{2-1}$$

其中的P(X),P(Y)为X,Y的先验概率，而P(X|Y)为X发生的情况下,Y发生的概率。

同理，对于用户身份认证，有：

$$P(U_k|X) = \frac{P(U_k)P(X|U_k)}{P(X)} \tag{2-2}$$

在估计所属用户的条件概率时，朴素贝叶斯方法均假设各变量独立，公式如下：

$$P(X|U_k) = \prod_{i=1}^{d} P(t_i|U_k) \tag{2-3}$$

$$P(U_k|X) = \frac{P(U_k)\prod_{i=1}^{d} P(t_i|U_k)}{P(X)} \tag{2-4}$$

由于P(X)与$P(U_k)$对于待分类的用户为固定的，所以为了找最大的$P(U_i|x)$，等价于找最大的$P(x|U_i)$。假设所选样本用户的特征服从高斯分布，则对应的概率密度为：

$$\int(t_i;\mu;\sigma)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(t_i-\mu_i)^2}{2\sigma_0^2}} \quad (2\text{-}5)$$

$$P(X|U_k)=\prod_{i=1}^{d}P(t_i|U_k)=\prod_{i=1}^{d}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(t_i-\mu_i)^2}{2\sigma_0^2}} \quad (2\text{-}6)$$

其中，待分类用户存在d个特征属性，在特征t_i下，均值为μ_i，标准差为σ_i。

2.3.3.1 本文算法

如上所述，在估计条件概率时，假设了每个变量完全独立，互不影响。事实上，每一个特征对用户的区分度是不一样的，如果都平均看待，那将会使数据的部分价值浪费掉。

所以为了体现出每个特征的不同，我们选择对特征进行加权。如果对于同一个特征，每个数据与其均值差异很大，那么意味着这组数据很不稳定，没有更好的反映出这个特征的实际情况，所以权重低。反之，如果对于同一个特征，每个数据与其均值差异很小，那么意味着这组数据很稳定，更好的反映出这个特征的实际情况，所以权重高。

综上所述，分类加权贝叶斯分类模型如下：

$$P(X|U_k)=\prod_{i=1}^{d}P(t_i|U_k)^{\alpha_i} \quad (2\text{-}7)$$

$$\alpha_i=\frac{\sqrt{(t_i-\mu_i)^2}}{\sum_{j=1}^{d}\sqrt{(t_j-\mu_j)^2}} \quad (2\text{-}8)$$

公式中，α_i代表对应于特征t_i的权值。

如上所示，分类之前需要确定用户集合，但是如果没有及时加入新用户，又有新用户出现，则模型判断效果不好。这个时候我们引入欧氏距离的概念，当已经通过加权贝叶斯算法后，划分到了一个用户后，再通过欧氏距离计算，如果大于阈值G则划分为该用户；反之，不属于该用户。欧氏距离公式如下：

$$D(t_i,u_i)=\sqrt{\sum_{i=1}^{d}\sqrt{(t_i-u_i)^2}} \quad (2\text{-}9)$$

其中，t_i为待分类对象的属性，u_i为系统用户的属性。

2.3.4 基于最小二乘支持向量机的持续认证分类模型

支持向量机（Support Vector Machine, SVM）作为一种监督学习模型，相比于传

统的神经网络，能够给出相当甚至更好的分类准确率，具有良好的泛化能力。支持向量机目前已经被广泛应用于各领域的分类问题上，例如文本挖掘、图像分类、物体识别等等。在持续身份认证的应用场景下，需要处理的用户行为数据量较大，并且经常需要根据正常用户的行为数据更新训练集并进行多次训练，因此不仅要求分类器具有较高的准确率，而且对分类器的学习速度有一定的要求。本文选取最小二乘支持向量机（Least Squares-Support Vector Machine，LS-SVM）作为基于用户击键行为的分类器。最小二乘支持向量机是对标准支持向量机的一种扩展，能够有效提高模型的运行效率[1]。

支持向量机的核心思想是在训练集中找出一系列特征向量作为支持向量，支持向量所构成的最优超平面能够将属于不同类的特征向量相互分隔并使分类间隔最大。进而完成对目的对象的分类。

支持向量机示意图如 2-12 所示。

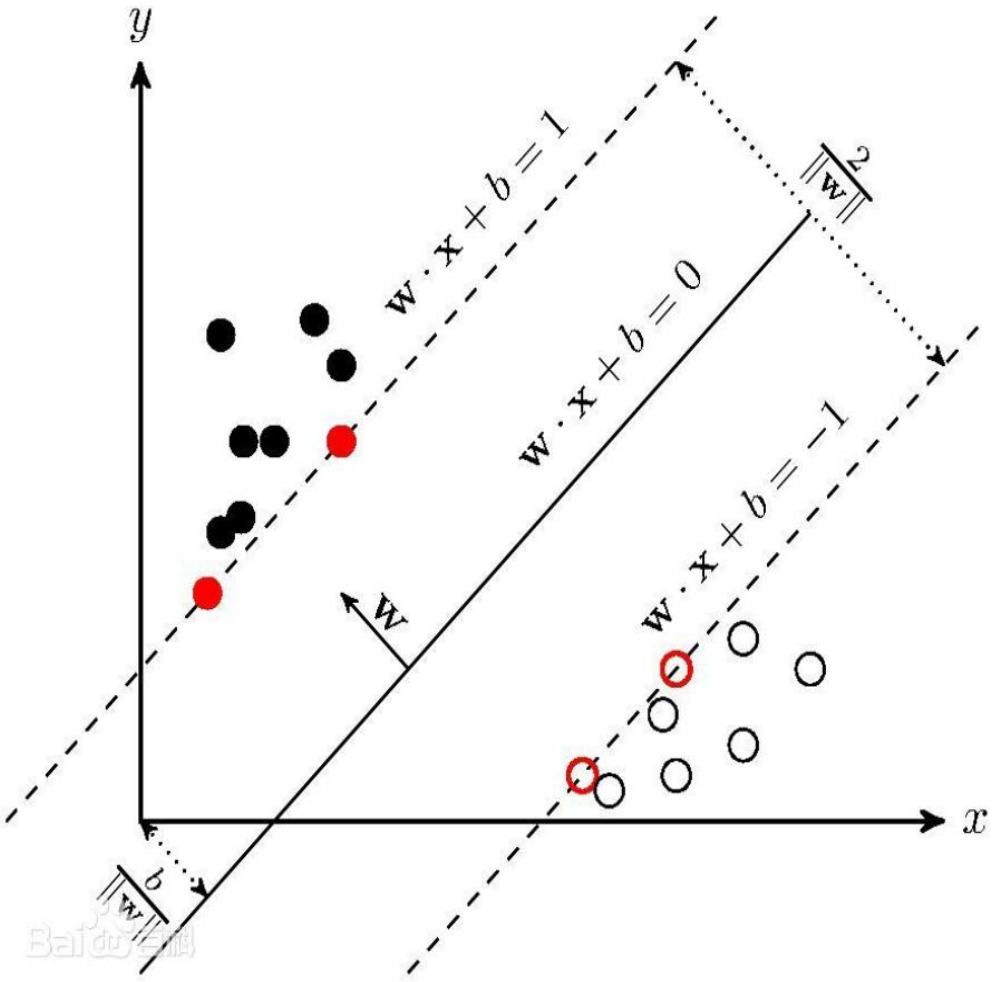

图 2-12 支持向量机原理

2.3.4.1 支持向量机模型

支持向量机核心想法是寻找一个最优超平面将不同类分割，并且使得分类间隔最大。

本作品的分类属于二分类问题，对于训练集D,样本数为m,样本的特征数为d，则该训练集可以表示为：

$$D = \{(x^{(i)},y^{(i)})\},x^{(i)} \in R^d,y^{(i)} \in \{-1,1\},i = 1,2,\ldots,m \quad (2\text{-}10)$$

其中，$x^{(i)}$为特征向量，$y^{(i)}$为类别标识，为1属于正类，为-1属于负类。

在d维空间下，超平面的方程为$w^T x = b$，w为d维常向量，b为标量。由于，支持向量机目的是寻找一个超平面使得区别两类样本，同时还要使样本的分类间隔最大。故，在支持向量机中，一个而分类的问题可转化为最优化问题，如下：

$$\begin{cases} min\frac{1}{2}\|w\|^2 \\ s.t.y^{(i)}(w^T x - b) \geq 1,i = 1,\ldots,m \end{cases} \quad (2\text{-}11)$$

约束条件保证样本分割，目标函数是让分类间隔最大化。基于拉格朗日对偶性，（2-11）的最优化问题可以转化为对偶性问题。

$$\begin{cases} \max \mathcal{L}(\alpha) = \alpha^T - \frac{1}{2}\alpha^T M\alpha \\ s.t\ \alpha^T y = 0 \end{cases} \quad (2\text{-}12)$$

其中，$\alpha^T = [\alpha_1,\ldots,\alpha_d]^T$为d维向量，$\alpha_i \geq 0$为拉格朗日乘子。M为d维方阵。（2-12）是一个二次规划问题，的解中非零元素α所对应的样本即为支持向量，因此可得支持向量机的决策函数为：

$$f(x) = sgn(\sum_{i=1}^{m} \alpha_i\, y_i(k(x^{(i)},x)) + b) \quad (2\text{-}13)$$

由此可以看出，有了核函数的支持，对非线性可分数据的分类依然可以在原空间内完成。

2.3.4.2 本文模型

由于实际中可能会存在离群值，会使得最佳分类间隔减小，甚至无法确定最优分类面，因此需要在一定程度上，减弱约束条件的约束强度，加入了惩罚因子C,所以在公式2-10的基础上，我们将模型修改为如下：

$$\begin{cases} min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \\ s.t.y^{(i)}(w^T x - b) \geq 1 - \xi_i,i = 1,\ldots,m \end{cases} \quad (2\text{-}14)$$

最终，还是一个带有分等式约束的QP问题，虽然决策函数为：

$$f(x) = sgn(\sum_{i=1}^{m} \alpha_i y_i \left(k\left(x^{(i)}, x\right)\right) + b) \quad (2\text{-}15)$$

在形式上与标准支持向量机类似，但此时对模型的求解可以转化为对线性方程组的求解，运算更快，适合处理大规模样本分类问题[1]。从而可以适应本作品的情境，完成快速分类的需求。

2.3.5 基于加权动态信任模型的持续认证方案

持续认证方案是本作品重要创新之处，也是本作品的核心。他与之前登录时的静态认证是不同的。

（1） 静态认证针对的是固定的字符串（密码），提取特征。而在持续认证时我们无法预估用户的输入。

（2） 静态认证是一次性认证技术，对于本作品，只在登录时有效；过后，则无效。而持续认证是会在用户登录之后，使用信息系统的全过程进行监督。防止有人当合法用户中途因故离开时，非法使用系统。

同时，本作品与其他研究的持续认证方案也不同，一些作品认证方案本质是依靠一次判定来确定是否为非法用户，这往往需要等待用户产生多个特征，认证时间长，同时，应为只要有一次异常就判断为非法用户，过于武断。

本文的思想来源于Monda1等人提出的动态信任模型[4]（DTM）,提出设定一个信任得分，为每一个产生的动作计分，在总信任得分上加上或减去一个分数，如果使得总信任得分低于阈值，则强制用户退出系统。

DTM具体算法如下图2-10所示：

变量说明：

sc_i 表示第 i 个动作的分类结果

A 表示产生奖惩的阈值

B 表示 sigmoid 函数的宽度

C 表示最大奖励

D 表示最大惩罚

$Trust_{i-1}$ 表示在第 i-1 次动作之后的系统信任得分

结果：

$Trust_i$ 为所需要的结果，即第 i 此动作之后的系统信任得分

公式：

$$\Delta_T(sc_i) = \min\{-D + (\frac{D \times (1 + \frac{1}{C})}{\frac{1}{C} + \exp(-\frac{sc_i - A}{B})}), C\}$$

$$Trust_i = \min\{\max\{Trust_{i-1} + \Delta_T(sc_i), 0\}, 100\}$$

图2-13 DTM算法

本文的算法与之不同之处在于：

（1） 针对的对象不同，DTM是针对每一个动作，本文设置了十一个模型，针对的是每一个模型中对应的动作（常用双键或鼠标双击等）。

（2） 初始得分，根据用户静态登录的得分确定，设置在0到10的区间。

（3） 总信任得分，最高分为10，最低分为0。

（4） 在具体的分类算法上，本作品使用速度快，分类效果好的最小二乘支持向量机。

（5） 具体总信任得分的加分与减分上。对于鼠标双键模型，由于其应用得最频繁，故每次认证失败或成功直接加2分或者减2分。而对于其他十个双键模型则需加权决定。假设只有两个双键an与cn，an使用次数为10次，cn使用次数为90次，则每次an分类后加上或减去0.1分，而每次cn分类后加上或减去0.9分。同理，有十对双键时也应当按比例加权扣除或增加相应的信任得

分。

2.3.6 国密算法SM9原理

SM9算法是一种基于双线性对的标识密码体制,是我国商用密码行业公钥密码算法的一种标准算法。SM9算法主要内容包括：数字签名算法、密钥交换协议、密钥封装机制和公钥加密算法、SM9加密算法等等。SM9密码算法的理论基础和数学工具是有限域群上椭圆曲线的点群运算的性质及双线性对运算特性[14]。

2.3.6.1 国密算法SM9基本技术

SM9密码算法涉及有限域和椭圆曲线、双线性对及安全曲线、椭圆曲线上双线性对的运算等基本知识和技术. 其中与双线性对运算直接相关的有Miller算法和BN曲线上R-rate对的计算方法.

以下是Miller算法介绍。

设$F_q k$上有椭圆曲线$E(F_q k)$的方程为:

$$y^2 = x^3 + ax + b \tag{2-16}$$

定义过$E(F_q k)$上点U和V的直线为:

$$g_{U,V}: E(F_q k) \to F_q k \tag{2-17}$$

若过U,V 2点的的直线方程为：

$$\lambda x + \delta y + \tau = 0 \tag{2-18}$$

则令函数

$$g_{U,V}(Q) = \lambda x_Q + \delta y_Q + \tau \tag{2-19}$$

其中$Q = (x_q, y_q)$,当$U = V$时，$g_{U,V}$定义为过点U的切线；若U和V中有一个无穷远点O，$g_{U,V}$就是过另一个点且垂直于x轴的直线。一般用g_U作为$g_{U,U}$的简写。

记

$$U = (x_U, y_U), V = (x_V, y_V), Q = (x_Q, y_Q) \tag{2-20}$$

$$\lambda_1 = (3x_V^2 + \alpha)/(2y_V) \tag{2-21}$$

$$\lambda_2 = (y_U - y_V)/(x_U - x_V) \tag{2-22}$$

则有以下性质：

$$g_{U,V}(Q) = g_{U,Q}(Q) = g_{Q,V}(Q) = 1 \quad (2\text{-}23)$$

$$g_{V,V}(Q) = \lambda_1(x_Q - x_V) - y_Q + y_V, Q \neq O \quad (2\text{-}24)$$

$$g_{U,V}(Q) = \lambda_2(x_Q - x_V) - y_Q + y_V, Q \neq O, U \neq \pm V \quad (2\text{-}25)$$

$$g_{V,V}(Q) = x_Q - x_V, Q \neq O \quad (2\text{-}26)$$

Miller算法是计算双线性对的有效算法。

输入：曲线E,E上2点P与Q，整数c;

输出：$f_{p,c}(Q)$.

（1）设c的二进制表示为$c_j \cdots c_1 c_0$,最高位c_j为1；

（2）置f=1,V=P;

（3）对i从j-1降至0，计算：

$$f = f^2 \cdot \frac{g_{V,V}(Q)}{g_{2V}(Q)} \quad (2\text{-}27)$$

$$V = [2]V \quad (2\text{-}28)$$

若$c_1 = 1$，则令：

$$f = f \cdot \frac{g_{P,V}(Q)}{g_{P+V}(Q)} \quad (2\text{-}29)$$

$$V = V + P \quad (2\text{-}30)$$

（4）输出f;

一般，称$f_{p,c}(Q)$.为Miller函数。

下面给出Miller算法在BN曲线R-ate对的计算方法。

输入：$P \in E(F_q)[r], Q \in E'(F_{q^2})[r], a = 6t + 2$;

输出：$R_d(Q,P)$

（1） 设：

$$a = \sum_{i=o}^{L-1} a_i 2^i \quad (2\text{-}31)$$

$$a_{L-1} = 1 \quad (2\text{-}32)$$

（2） 置$T = Q, f = 1$;

（3） 对i从L-2降至0，执行计算：

$$f = f^2 \cdot g_{T,T}(P) \quad (2\text{-}33)$$

$$T = [2]T \quad (2\text{-}34)$$

若$a_i = 1$，计算：

$$f = f \cdot g_{T,Q}(P) \quad (2\text{-}35)$$

$$T = T + Q \quad (2\text{-}36)$$

（4） 计算：

$$Q_1 = \pi_q(Q) \quad (2\text{-}37)$$

$$Q_2 = \pi_{q^2}(Q) \quad (2\text{-}38)$$

（5） 计算：

$$f = f \cdot g_{T,Q_1}(P) \quad (2\text{-}39)$$

$$T = T + Q_1 \quad (2\text{-}40)$$

（6） 计算：

$$f = f \cdot g_{T,-Q_2}(P) \quad (2\text{-}41)$$

$$T = T - Q_2 \quad (2\text{-}42)$$

（7） 计算：

$$f = f^{(q^{12}-1)/r} \quad (2\text{-}43)$$

（8） 输出f

2.3.6.2 国密算法SM9加密具体流程

设需要发送的消息为比特串M，mlen为M的比特长度，K1_len为对称密钥K1的比特长度，K2_len为函数MAC(K2,Z)中密钥K2的比特长度。

国密算法SM9加密具体流程如图2-14所示。

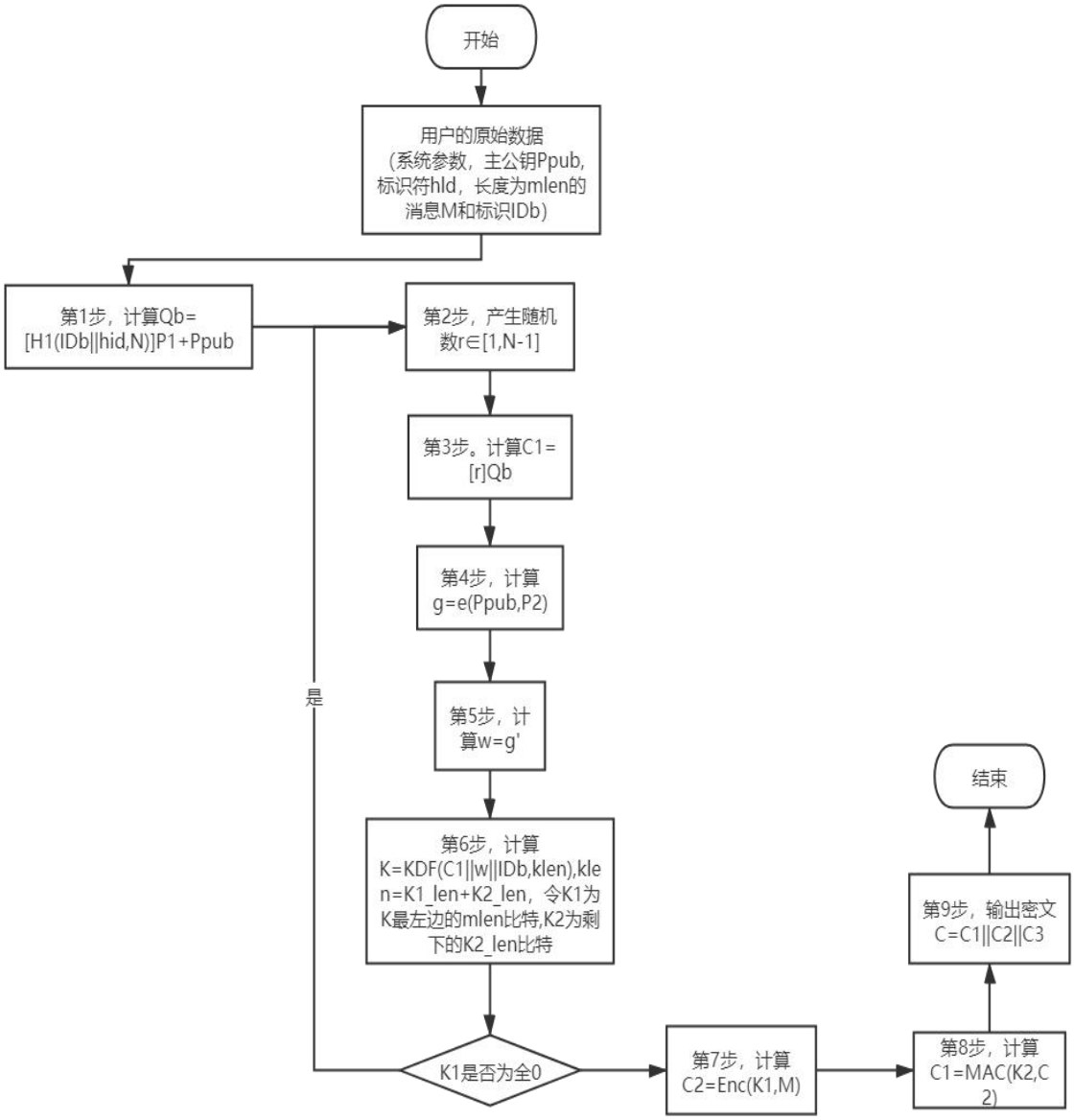

图2-14 SM9加密流程

2.3.6.3 国密算法SM9解密具体流程

设mlen为密文C=C1||C2||C3中C2的比特长度，K1_len为对称密钥算法中密钥K1的比特长度，K2_len为函数MAC(K2,Z)中密钥K2的比特长度。具体流程如下所示。

国密算法SM9解密具体流程如图2-15所示。

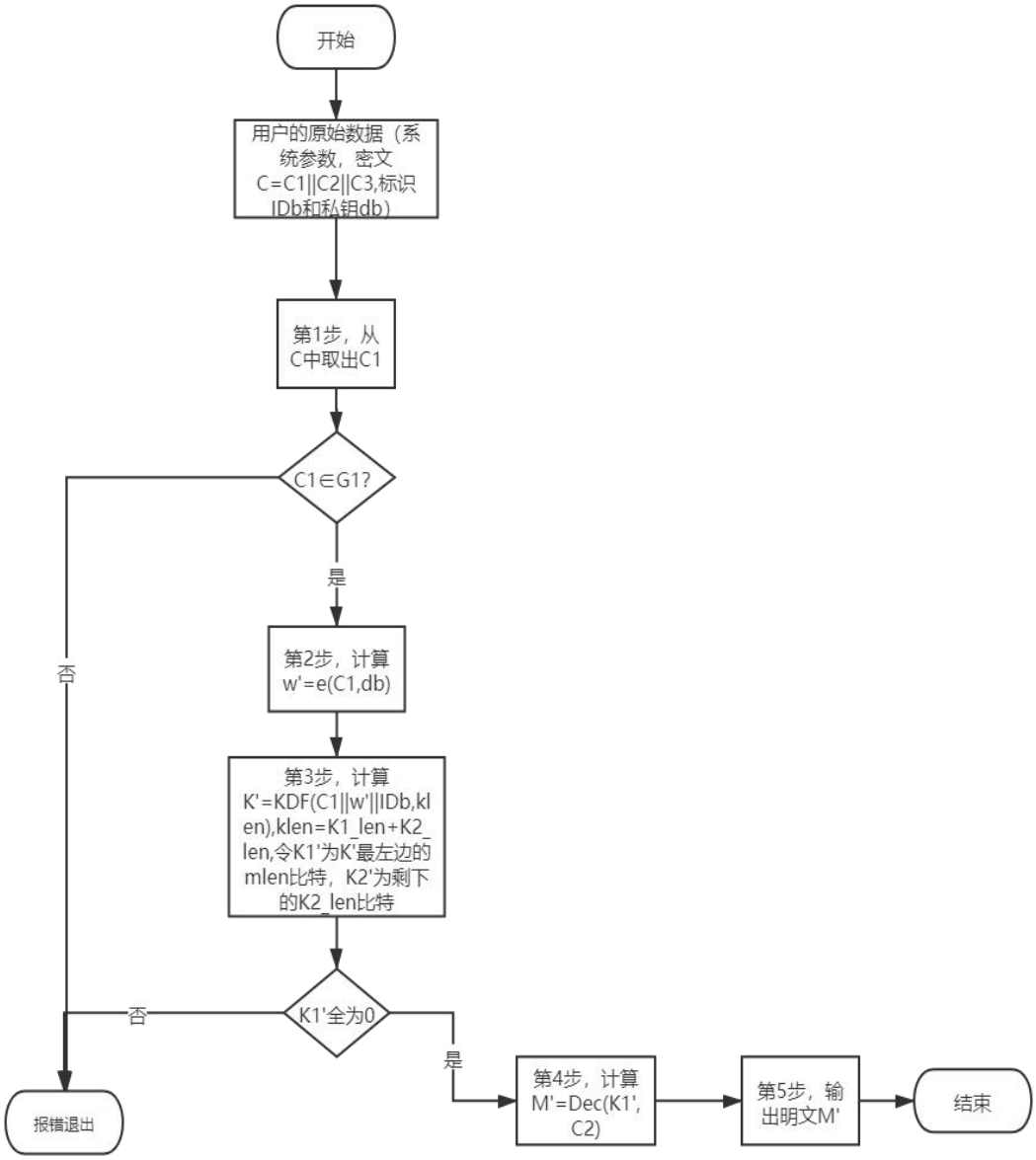

图2-15 SM9解密流程

2.3.7作品总体设计想法——系统可用性与安全性的辩证关系

对于一个具有安全功能的作品而言，往往作品的安全性与可用性存在着一定的冲突。对于作品的可用性，其实诸如人脸识别、指纹识别、包括本作品的击键识别等身份认证机制，本身对系统可用性就是一个 “损害”，因为用户不进行这些过程，用户依然可以使用系统的功能；如果进行这些认证过程，一定程度有损用户体验感。反过来，对于作品的安全性，则必然会对作品的可用性造成影响。强调绝对的安全与可用，本身是不太科学的。

为了进一步分析系统可用性与安全性的辩证关系，先引出两个概念。

（1）错误接受率（FAR）

FAR是系统错误接收的非法用户（即将非法用户认为合法用户）的比率，FAR越高，系统越不安全；FAR越低，系统更为安全（这意味觉大多数非法用户被识别出来），FAR代表着认证系统的安全性，其公式如4-1：

$$FAR = \frac{非法用户误认为合法用户的次数}{非法用户认证次数} \quad (2\text{-}44)$$

（2）错误拒绝率（FRR）

FRR是系统错误拒绝的合法用户（即将合法用户认为非法用户）的比率，FRR越高，系统可用性低；FAR越低，系统可用性高，用户体验也好。FRR代表着认证系统的可用性,其公式如4-2：

$$FRR = \frac{合法用户误认为非法用户的次数}{合法用户认证次数} \quad (2\text{-}45)$$

一个安全系统的FAR与FRR的关系如图2-16所示：

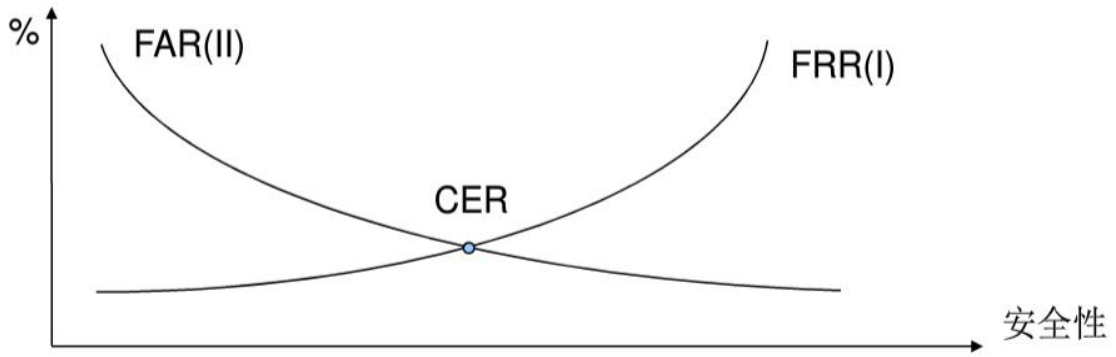

图2-16 安全系统的FAR与FRR关系

如图2-16所示，发现FAR与FRR在交点附近存在着一定的“反比”关系，即FRR数值大时，FAR数值小；FRR数值小时，FAR数值大。这代表着作品安全性与可用性存在着的冲突。（交点处代表着系统的整体准确率）

基于以上分析，我们发现可以改变认证算法的某个关键阈值，调节系统的可用性与安全性。从而使得本作品可以运用到更多的场合，增大作品的实用性，更好地推广作品。

（1） 对于一些重要商业系统、军事系统，更强调系统的安全性，这时候可以改变认证阈值，使得降低FAR，增进FRR，保障系统安全。

（2） 对于一些普通的个人电脑，我们在兼顾安全性的同时，也比较注重用户的使用体验感；所以，我们可以在用户接受的基础上，改变认证阈值，使得降低FRR，增进FAR，保障系统安全的同时，尽可能把对用户正常使用系统的影响降到最低。

在作品实际设计时，我们提供给了用户对应的可视化界面，自主的设定自己电脑的认证强度。

2.4 系统功能模块设计

本作品主要由四个模块组成，分别是数据收集与处理模块、静态认证模块、持续认证模块、信息安全传输模块。具体情况如图2-17所示。

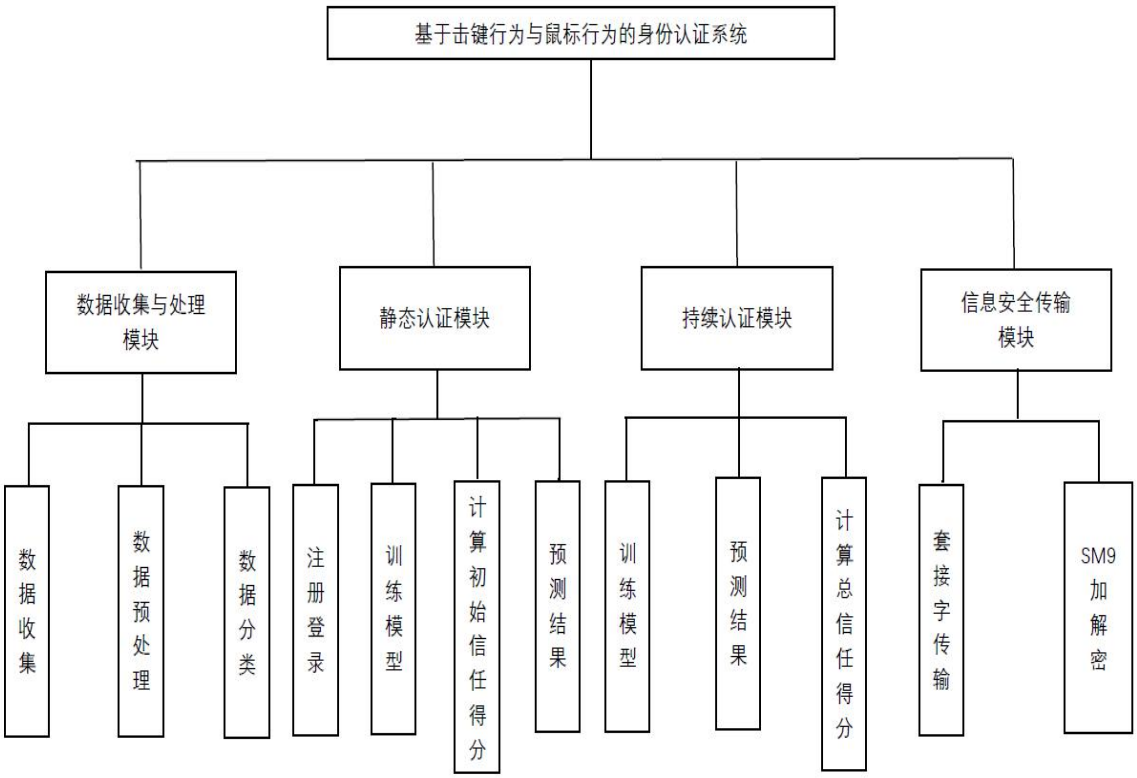

图2-17 系统功能模块图

2.4.1 数据收集与处理模块设计

本模块主要完成整个系统的数据工作，主要是通过钩子程序的收集数据，对于一些不合常理的值进行预处理，再将数据导入到对应的处理模块。

第一步我们要先开启钩子程序开始收集，第二步再把收集的数据进行预处理并送往所需的模块。

数据收集的具体流程如下图2-18所示：

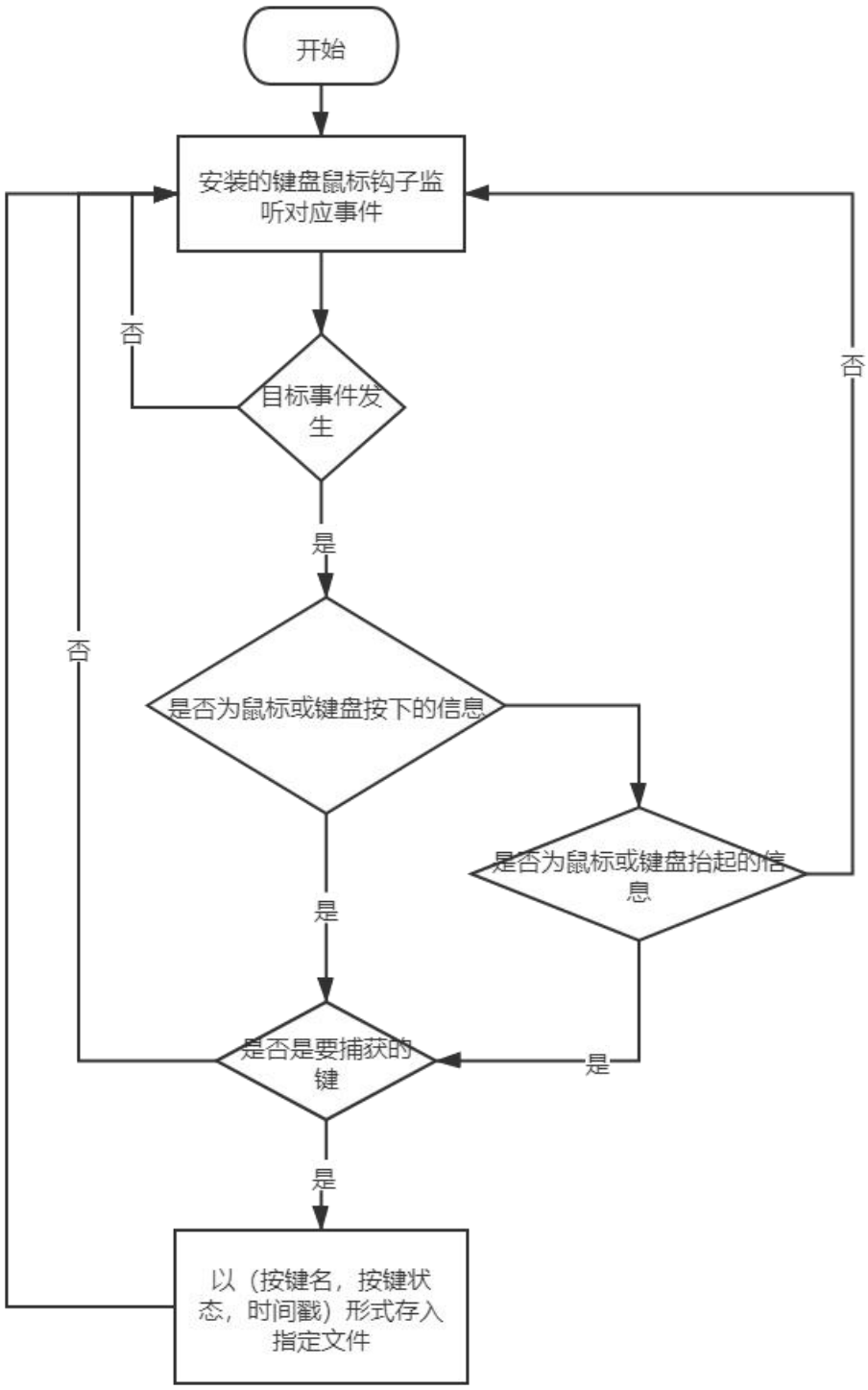

图2-18 数据收集模块流程图

数据处理的具体流程如下图2-19所示：

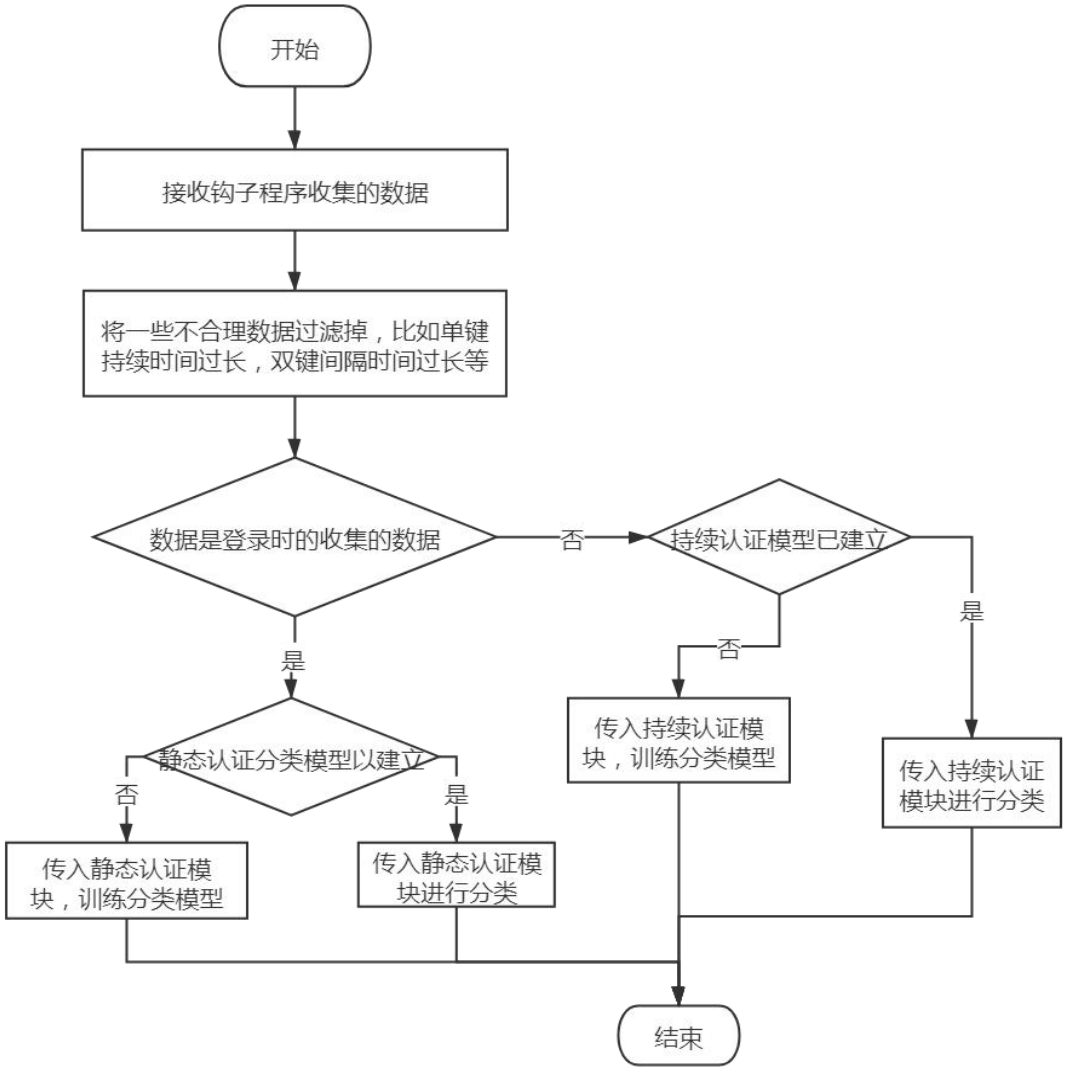

图2-19 数据处理模块流程图

2.4.2 静态认证模块设计

本模块主要解决用户注册登录时的问题，用户注册、登录、分类模型的训练、预测结果、以及根据登录时的认证情况，给出初始信任得分。

具体流程如下图2-20所示：

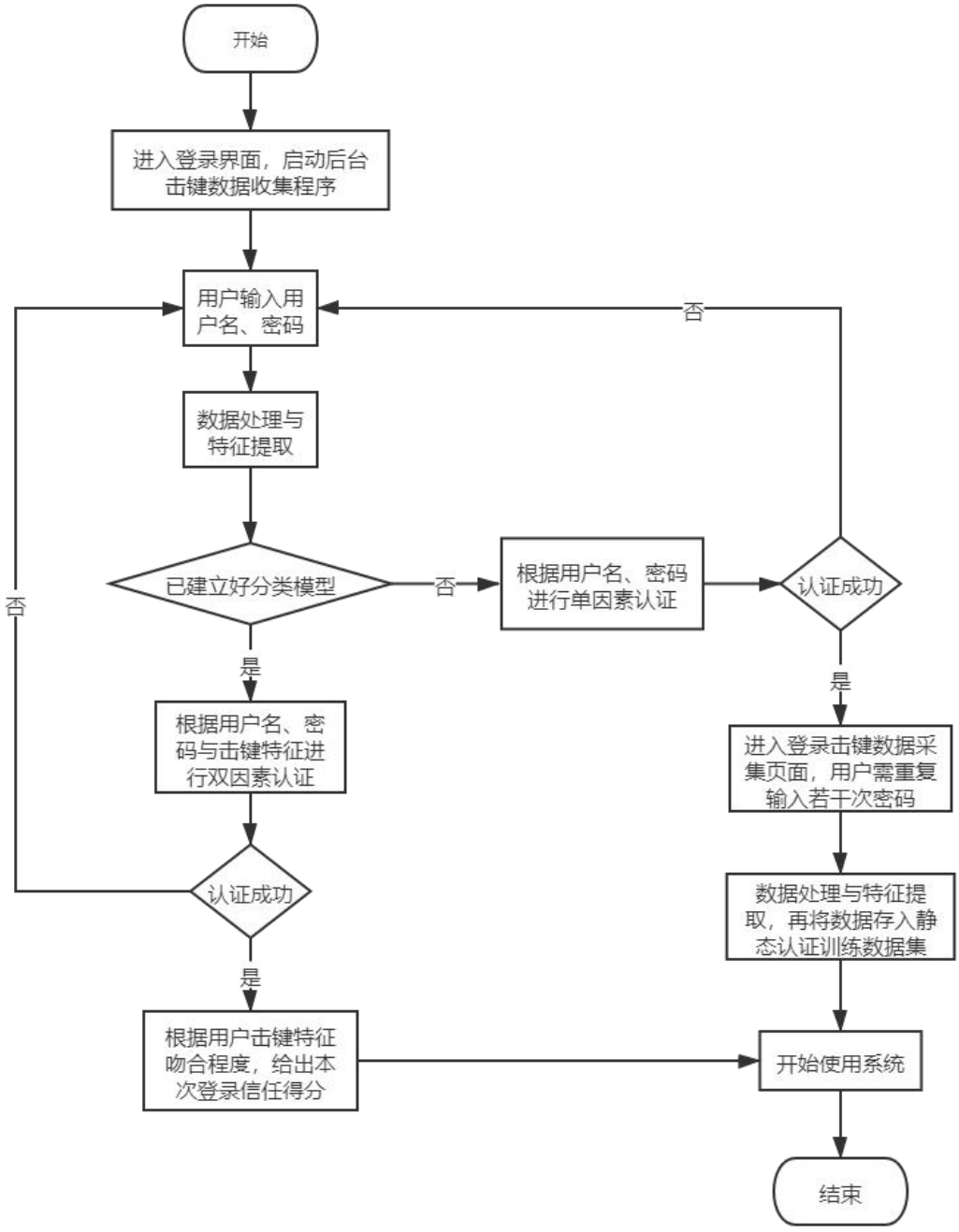

图2-20 静态认证模块流程图

2.4.3 持续认证模块设计

本模块主要解决用户登录后，使用电脑时的问题，监视用户的行为、分类模型的训练、预测结果、以及根据产生的动作更新总信任得分，若低于阈值，强制退出系统。

具体流程如下图2-21所示：

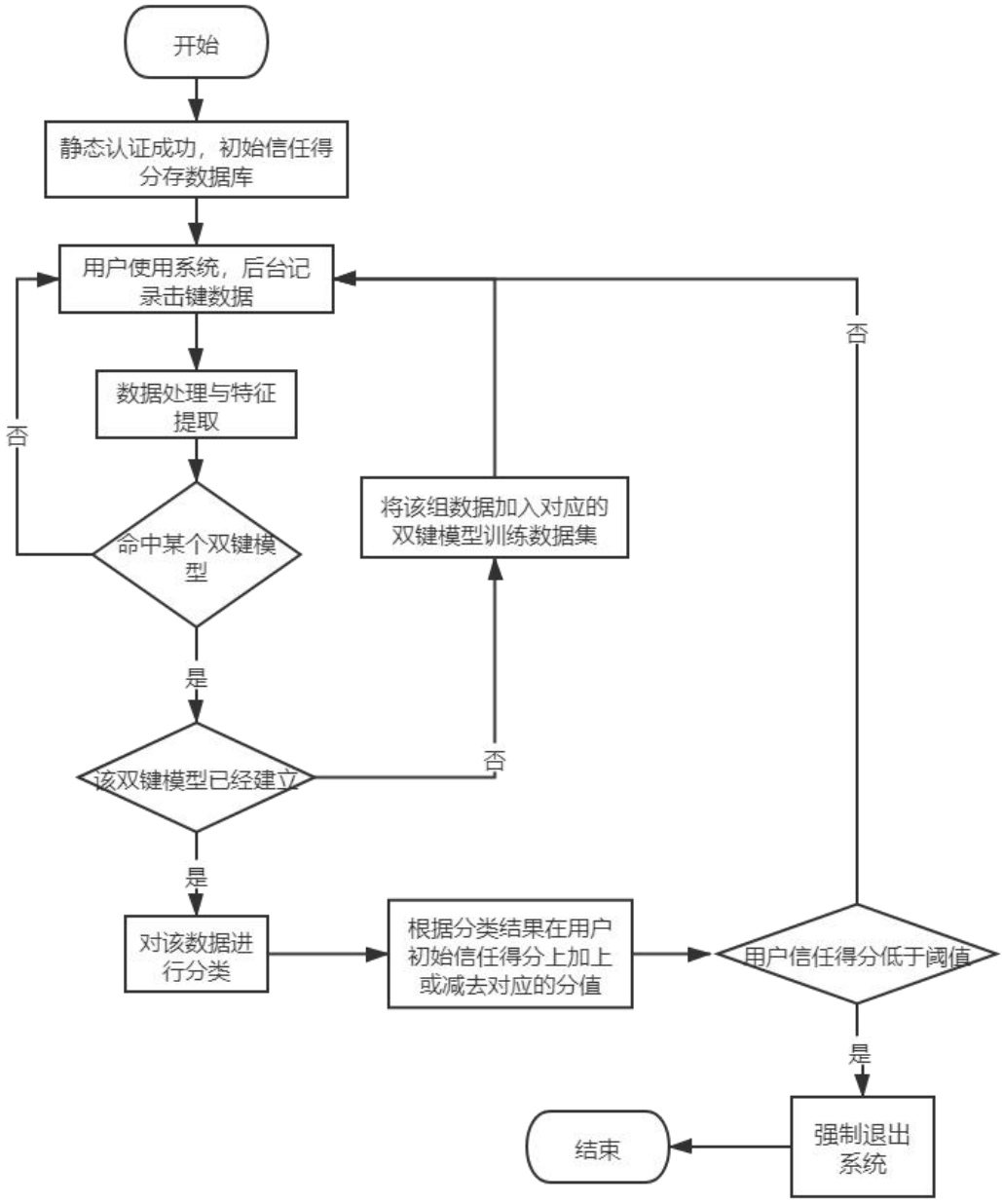

图2-21 持续认证模块流程图

2.4.4 信息安全传输模块设计

本模块信息在客户端与服务器的可靠安全加密传输。以数据发送过程为例，具体流程如下：

具体流程如下图2-22所示：

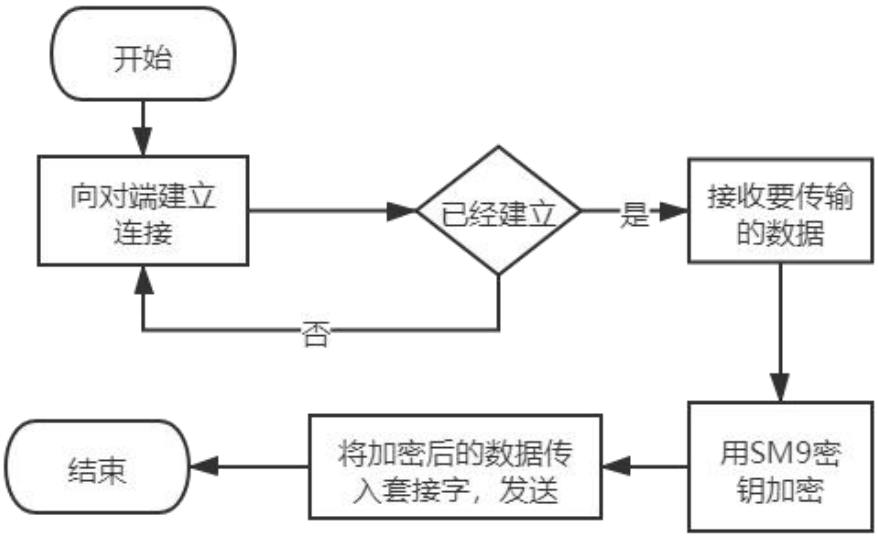

图2-22 信息安全传输模块流程图

2.5 小结

本章先对作品的整体流程进行了介绍，同时对基于击键行为与鼠标行为的身份认证系统的网络架构、软件架构进行了分析，同时结合套接字技术与SM9加解密技术，论述了本作品的信息安全传输方案。接着，介绍了系统实现所需要依赖的几种关键技术的原理，比如分类模型与算法，数据特征选取，数据收集方法，认证方案等等。最后，展开介绍了系统的各大模块，包括数据收集与预处理模块，持续认证模块，静态认证模块，信息安全传输模块。总结来说，本章从理论设计的层面，讲解了基于击键行为与鼠标行为的身份认证系统是如何实现的，为后续章节打下基础。

第三章 作品实现

本章将具体讲述各个模块的实现，具体包括实验环境介绍、客户端实现、服务端实现以及安全传输体系实现。

3.1 实现环境与开发工具介绍

在基于击键行为与鼠标行为的身份认证系统中，相关的重要环境或用到的工具如下表3-1所示：

表3-1 实验所用的环境或工具列表

名称	版本	描述
Windows操作系统	Windows 10	使用的实现设备是基于windows 10系统
CPU	酷睿i7-7500u	使用的实现设备的cpu
安卓操作系统	10	进行手机端的调试
Visual Studio	2019	用于制作C++程序与界面
PyCharm	2020.1.1	用于编写调试训练模型
SQL Server	2017	后台数据库
Tomcat	7.0.92	构建web服务器环境
Java ee	-	搭建服务器
Internet Information Services	7.0	搭建后台管理员网站服务器
JDK	1.8.0_261	构建java环境
微信开发者工具	-	开发微信小程序

3.2 PC客户端实现

客户端主要由用户可视化界面，数据收集程序组成。用户进入系统，首先看到的是可视化界面，根据提示，完成输入；同时后台的数据收集程序，收集有关信息并发送给服务器认证；服务器返回认证结果，客户端根据返回的结果，完成相应的操作。同时，为了减轻服务器负担，在PC本地也建立了一套认证机制，可以完成对用户行为的认证。

3.2.1 数据收集程序实现

3.2.1.1 编写对应的DLL

数据收集程序就是前面所说的钩子程序，部署在客户端收集用户的击键或鼠标行为数据。该程序主要使用windows提供的API来实现，最后封装成DLL,用户打开击键采集程序即可完成加载。

它先需要布置钩子程序SetWindowsHookEx()，编写回调函数KeyboardProc()，利用GetTickCount()记录时间，再调用Savelog()将击键数据存到指定文件中。最后，数据收集完成后，调用UnSetWindowsHookEx()消除钩子。

我们通过创建MFC共享DLL来实现。

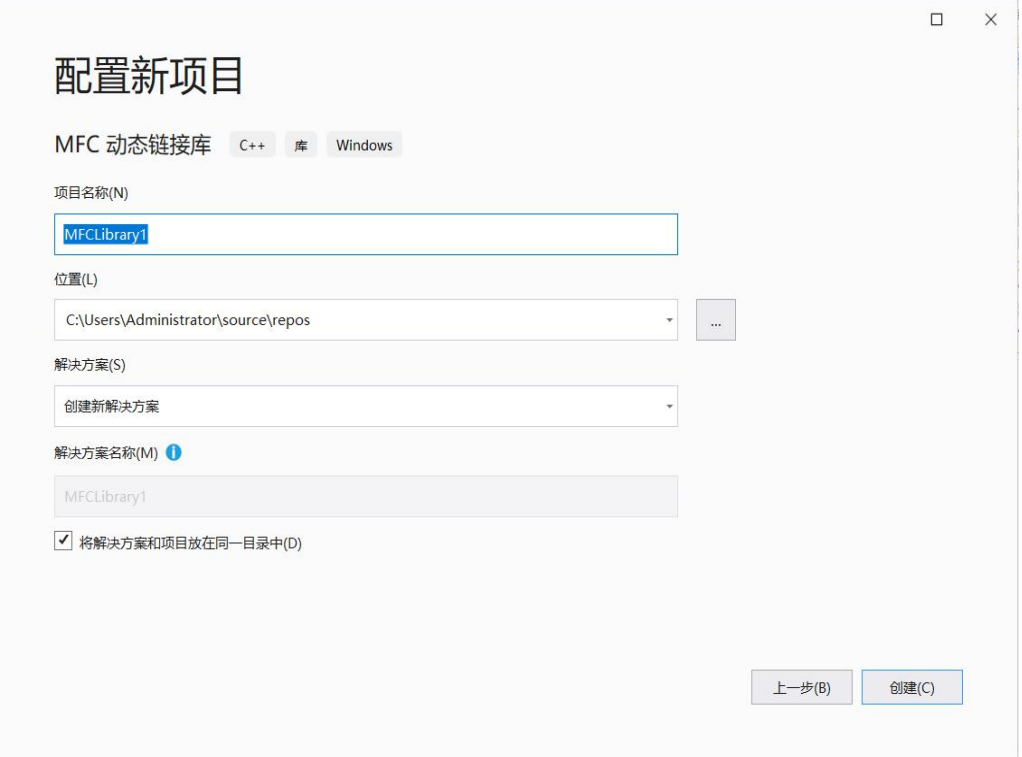

图3-1 创建MFC DLL

DLL中接口函数具体定义如表3-2所示。

表3-2 数据收集模块接口函数表

方法名字	返回类型	方法描述
installhook()	BOOL	设置钩子
uninstallhook()	BOOL	卸载钩子
KeyboardProc(int,wParam,lParam)	LRESULT	键盘钩子回调函

		数
MouseProc(int,wParam,lParam)	LRESULT	鼠标钩子回调函数
GetTickCount()	DWORD	记录击键时间戳
Savelog(char*)	BOOL	将击键数据记录到文本中

注：以上函数都需声明为_declspec(dllexport)，使得可以被调用。

再编译生成对应的DLL,LIB文件。如图3-2所示

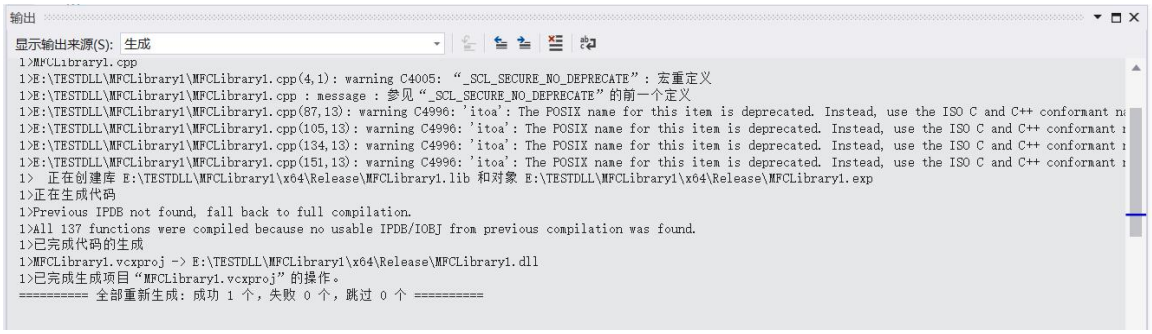

图3-2 生成MFC DLL

3.2.1.1 注入到对应进程

首先将上述生成的DLL,LIB,.h文件，放到目标进程目录下，如图3-3所示：

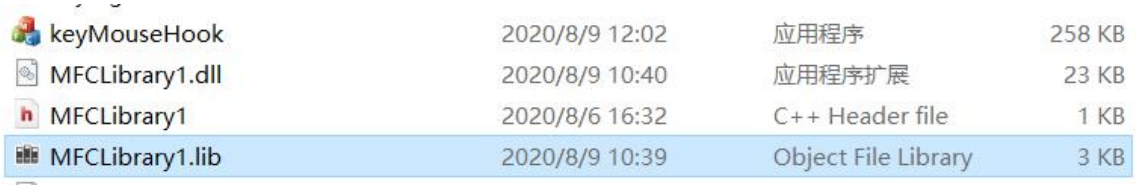

图3-3 DLL注入

最后，在目标程序中包含对应头文件，加上：

```
#pragma comment(lib,"MFCLibrary1.lib")
#include "MFCLibrary1.h"
```

再在目标程序中调用installhook()函数即可，同时，在目标程序结束时调用uninstall()函数，完成对钩子的卸载。

进行数据收集。点击keyMouseHook.exe，收集的结果如下图3-4所示。

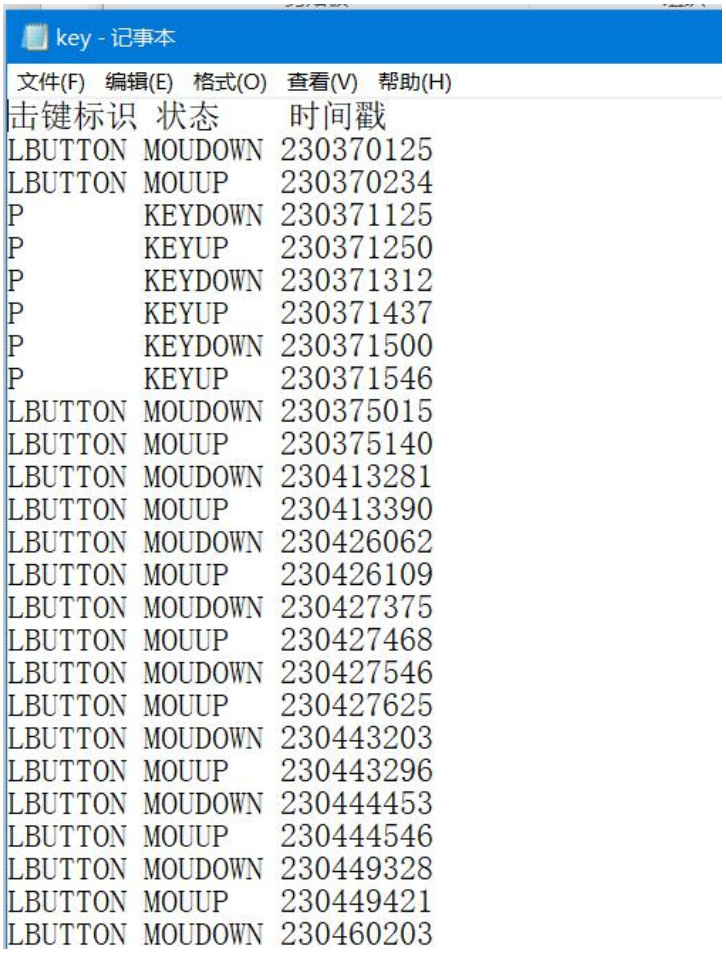

图3-4 收集结果

3.2.2 客户端界面及控制逻辑实现

客户端界面以及有关控制逻辑主要使用MFC实现，主要界面如下：

首先，用户需要进行注册，填写有关信息，设置用户名，密码等。注册之后，就可以正常登录了。如果在后面收集到了足够多的数据，训练好了分类模型之后，则可以开始正式认证过程。如果在使用时，有非法用户入侵，则弹出强制退出系统提示，最终返回到登录界面。

1、注册

第一步便是开始注册，注册不仅包括收集用户有关登录信息，同时还包括收集用户的击键习惯。对于静态认证，成功登录后，会弹出一个提示框，要求输入指定次数的密码（可以分多次输入）用于收集静态认证数据。对于持续认证，会在后台进行收集，即不会影响用户正常使用；后台运行着的程序监听击键事件，一旦发现命中了某个双键模型，则将有关数据增加到对应的训练数据集中。

注册界面如图3-5所示。

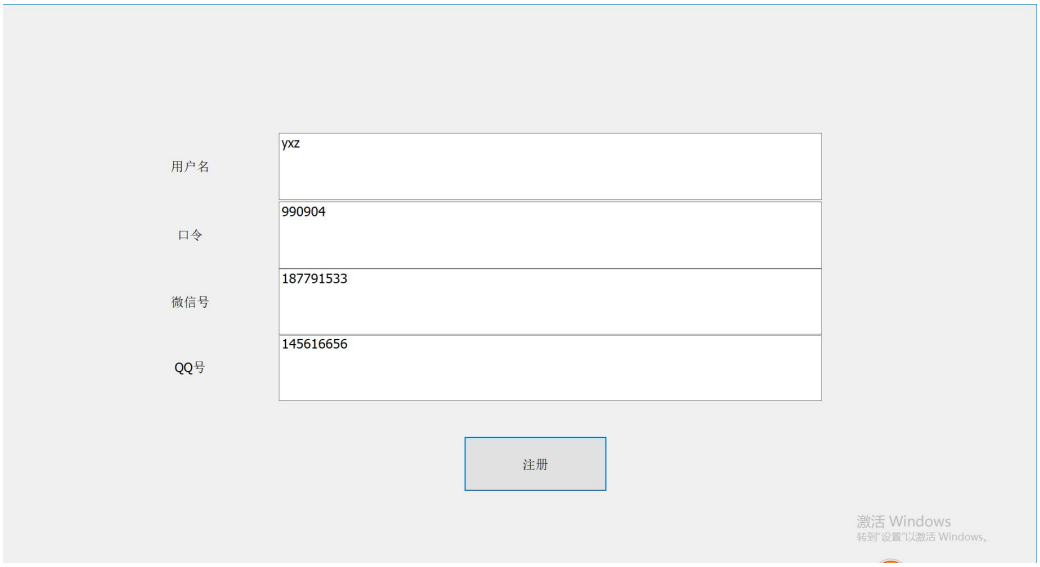

图3-5 注册界面

初始静态认证数据收集页面如图3-6所示。

图3-6 初始静态认证数据收集页面

注册阶段用到的程序接口如表3-3所示。

表3-3 注册阶段接口函数

方法名字	返回类型	方法描述
Regis(char*,char*)	BOOL	注册

Save_pwd(char*)	BOOL	收集静态认证数据
Readline(char*)	int	读取指定文件的函数
find_dy()	void	收集动态认证数据
Save_log(char*, char*, char*, char*, char*)	BOOL	将收集到的动态认证数据存到指定文件

2、登录（静态认证）

登录阶段不仅要核对用户名密码是否正确，还要进行击键静态认证（用户的击键习惯是否正确）。登录界面如图3-7所示。

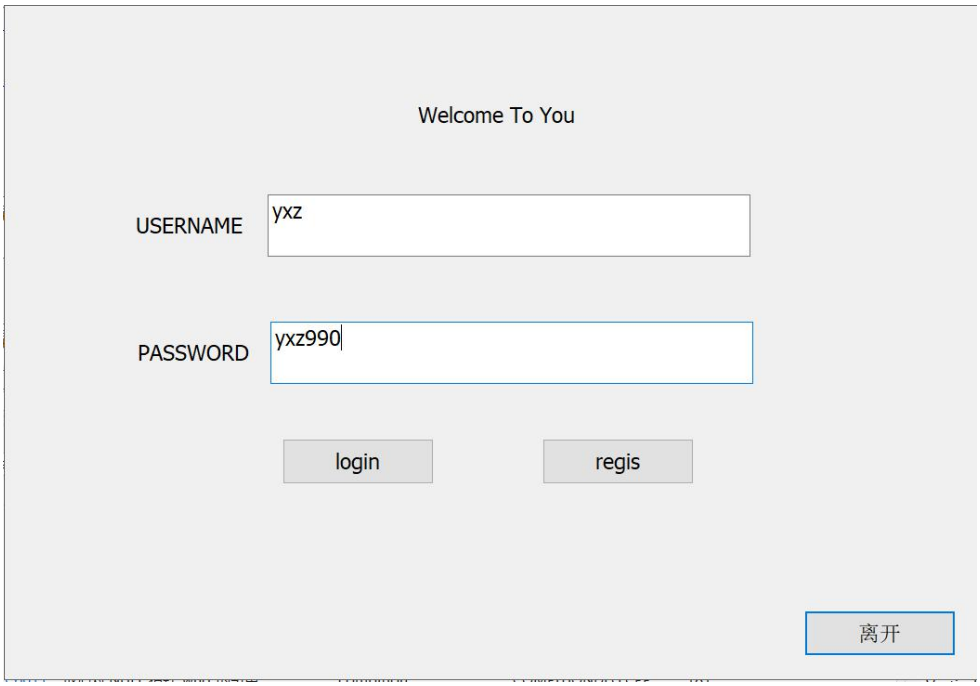

图3-7 登录界面

用户名密码不对弹出的提示如图3-8所示。

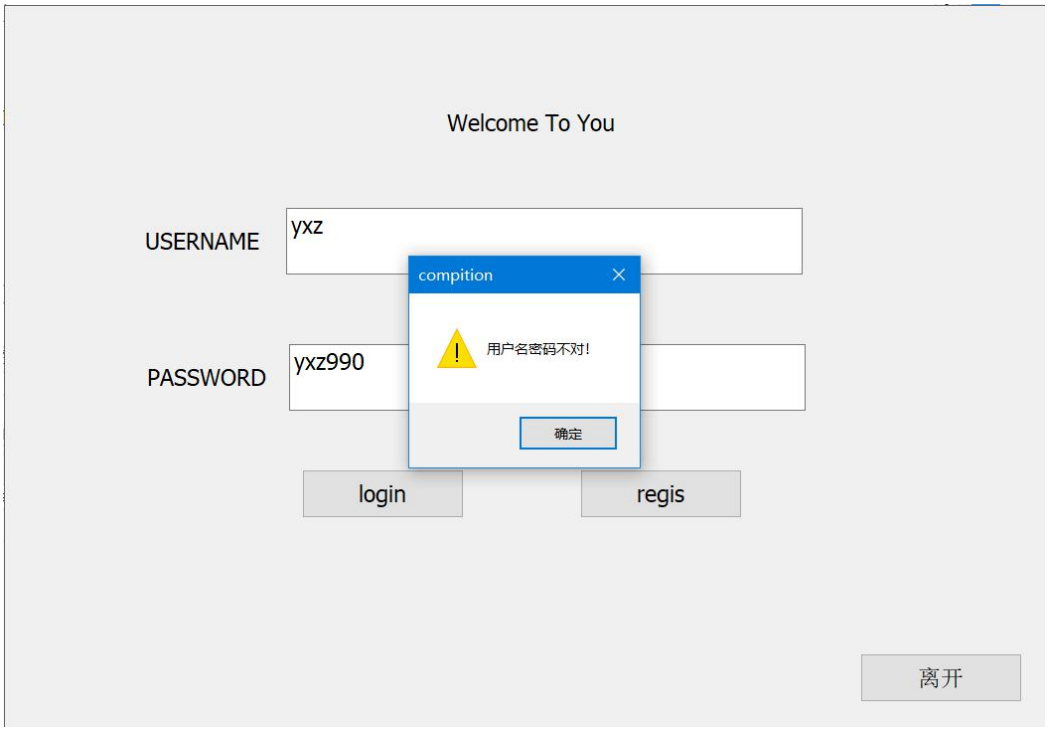

图3-8 用户名密码不对

击键习惯不对弹出的提示如图3-9所示。

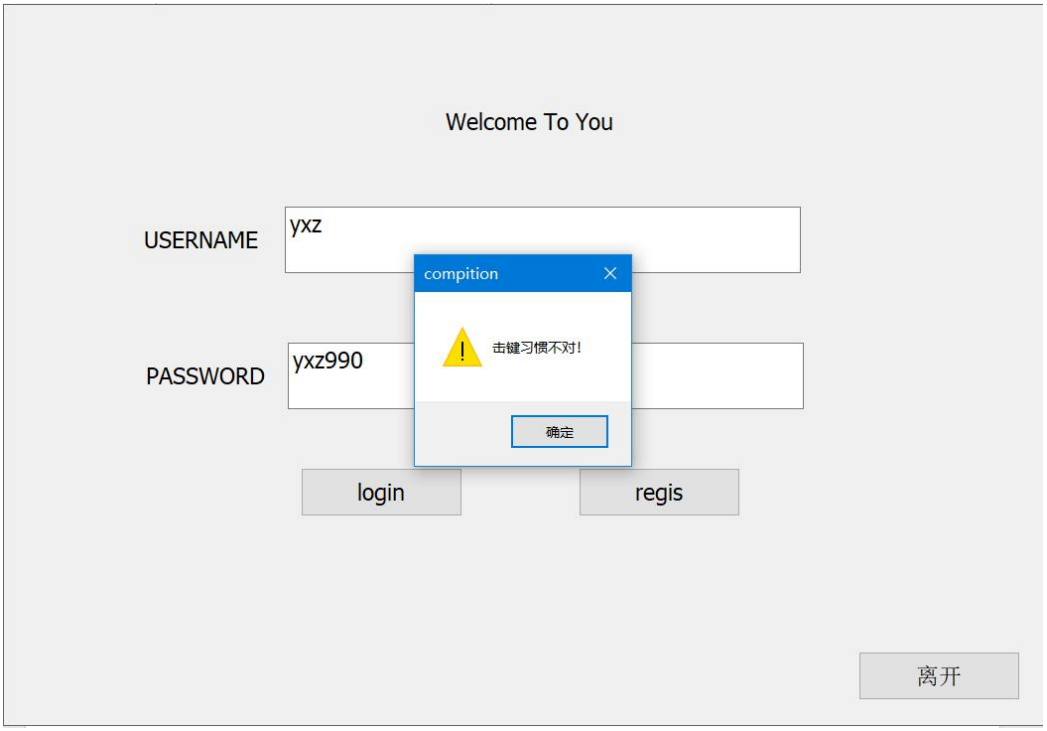

图3-9 击键习惯不对

登录阶段调用的函数接口如表3-4所示。

表3-4 登录阶段接口函数

方法名字	返回类型	方法描述
Login_userpwd(char*,char*)	BOOL	用户名密码登录认证
Login_pwd(char*,char*)	BOOL	击键习惯登录认证
Readinput(char*)	BOOL	收集用户输入密码时产生的击键信息，并存入指定文本中

3、持续认证

持续认证是在用户通过登录后，使用电脑时，监视用户的行为，进行实时动态认证。后台运行着对应的分类器一旦捕捉到了对应的信息，则启动进行分类；若分类为正常用户则增加认证得分，若分类为非正常用户则减少认证得分，若认证得分低于某个阈值，则强制退出系统。

事实上，软件发布时，我们只会在用户被强制退出系统时给出如图3-11的提示，其他提示（如图3-9,3-10）只是为了方便演示和测试而设置的。

持续认证成功的提示如图3-10所示（仅在测试阶段显示，方便测试与演示）。

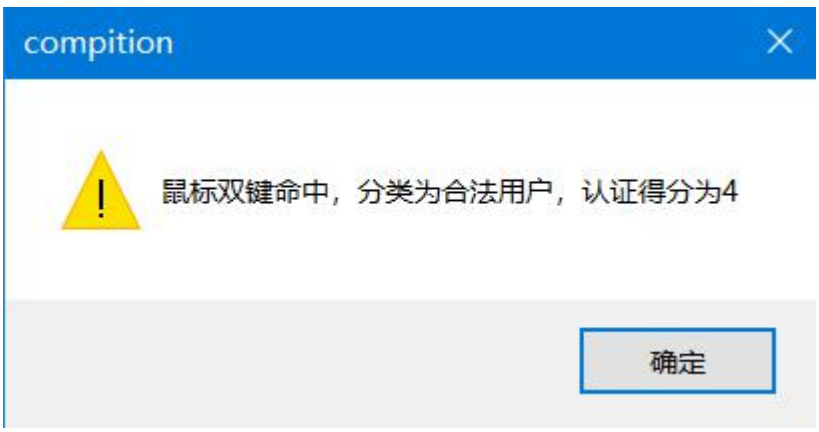

图3-10 持续认证成功

持续认证失败的提示如图3-11所示（在测试阶段显示，方便测试与演示）。

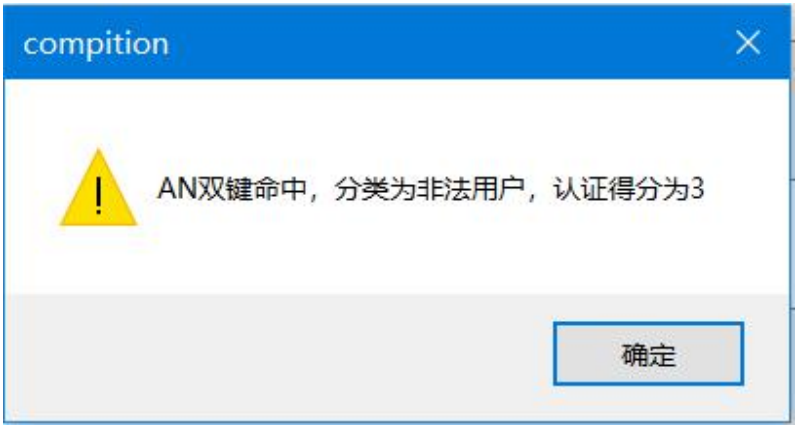

图3-11 持续认证失败

持续认证失败，被强制退出系统的提示如图3-12所示。

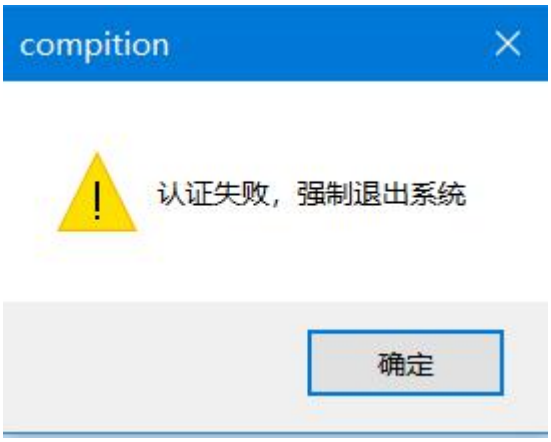

图3-12 被强制退出系统

调用的接口如表3-5所示。

表3-5 持续认证阶段接口函数

方法名字	返回类型	方法描述
Data_catcher()	void	监听是否产生对应双键
Save_catcher(char*, char*, char*, char*, char*)	BOOL	存储捕捉到的双键数据
Readlines(char*)	int	查看指定文件的行数

3.3 服务端实现

服务端主要处理客户端发来的信息，再经过处理之后，传入持续认证或静态认证模块处理，最终将反馈结果返回给客户端。或者根据击键认证小助手、后台管理员网站发来的命令和请求做出有关处理。

主要是基于java EE软件进行开发，并将它部署在Tomcat中，数据库采用SQL Server 2017。

同时为了提供给微信小程序、网站的服务，需要构建Web服务，开启Tomcat，如图3-13所示。

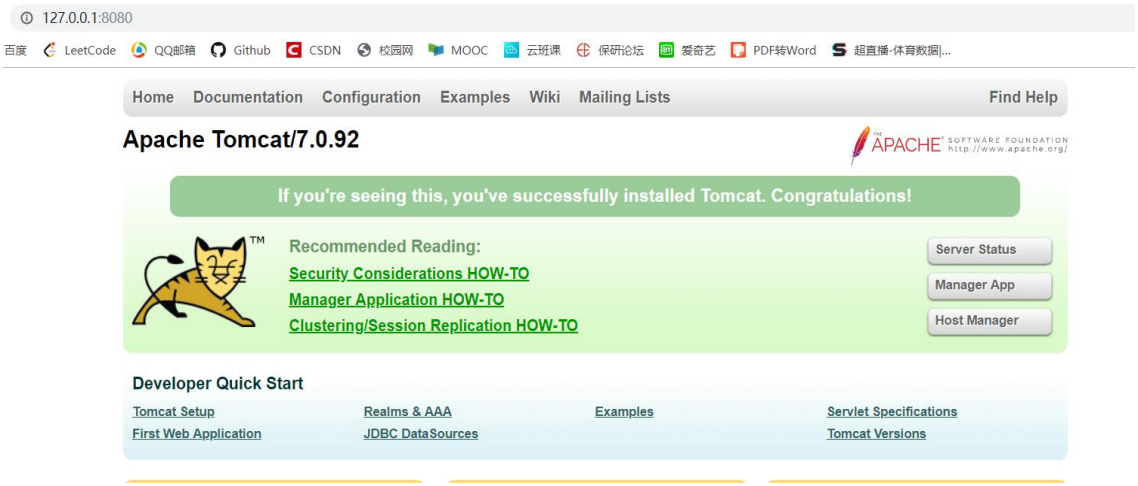

图3-13 Tomcat开启

由于所使用的分类器用python编写，还需要创建python环境，导入python36.dll。并且python代码被调用时，要按照以下格式。

以调用python文件hello.py中的hello()函数为例。

```
Py_Initialize();//初始化 python 环境
if (!Py_IsInitialized())//判断初始化是否成功
{
    AfxMessageBox(_T("ERROR INIT"));
}
PyRun_SimpleString("import hello");//导入对应 python 文件
PyRun_SimpleString("hello.hello()");//执行其中的函数
Py_Finalize();//结束 python 环境
```

3.3.1 数据处理功能实现

数据处理模块，主要是对传来的数据进行处理并对其分类，等待客户端数据的到来，将数据传入对应的模块。如表3-6所示。

表3-6 数据处理功能接口函数表

方法名字	返回类型	方法描述
Classify()	BOOL	对客户端发来的数据处理与分类
doGet(HttpServletResponse,HttpServletRequest)	void	处理http的Get请求
doPost(HttpServletResponse,HttpServletRequest)	void	处理Http的Post请求
Regis(string,string)	string	完成用户注册
Login(string, string)	string	完成用户登录
Recoder(string,string)	string	记录用户的操作记录
Send_recoder(string)	string	发送用户操作记录给客户端
Change_pwd(string)	string	修改密码
Set_classfier(string)	string	设置认证强度
GoSystem(string)	string	强制进入系统
OutSystem(string)	string	强制离开系统

3.3.2 静态认证功能实现

静态认证功能主要是由加权贝叶斯分类与欧式距离模型实现，在模型能给出预测结果之前，需要足够多的数据进行锻炼。用python语言在实现在PyCharm中实现。主要方法如表3-7所示：

表3-7 静态认证模块接口函数表

方法名字	返回类型	方法描述
loaddataset()	数组	加载数据与数据预处理
SeprateByclass(dataset)	dict,dict	划分不同的类
mean(number_list)	float	计算均值
var(number_list)	float	计算方差
summarizeAttribute(dataset)	数组	对数据集每一列求均值方差
SummarizeByClass(dataset)	dict	形成类的均值方差数组
CalulateClassPriorProb(dataset,dasetinfo)	数组	计算先验概率
CalulateProb(x,mean,var)	Float	计算后验概率
CalulateClassProb()	dict	计算属于每一个类的概率
BayessianPredictOneSample(input_data)	string	对样本进行预测

3.3.3 持续认证功能实现

持续认证功能主要是利用最小二乘支持向量机建立十一个模型，建立成功后，如果用户产生的动作命中了某个模型，则扣除或加上一定的信任得分，如果低于某个得分则强制退出登录。用python语言在实现在PyCharm中实现。主要方法如表3-8所示：

表3-8 持续认证模块接口函数表

方法名字	返回值	方法描述
loaddataset()	数组	导入数据
kerneltrans()	BOOL	训练数据
Leastsquares()	数组	最小二乘法
Predict()	1或-1	预测结果

3.4 击键认证小助手实现

击键认证小助手是基于微信开发者工具开发的小程序，他可以让用户便捷的去管理自己的PC系统，自主的控制PC系统的行为。

它与服务端之间主要通过http方式进行通信，使用wx.request方法，用于与服务端之间进行交流。

1.登录界面

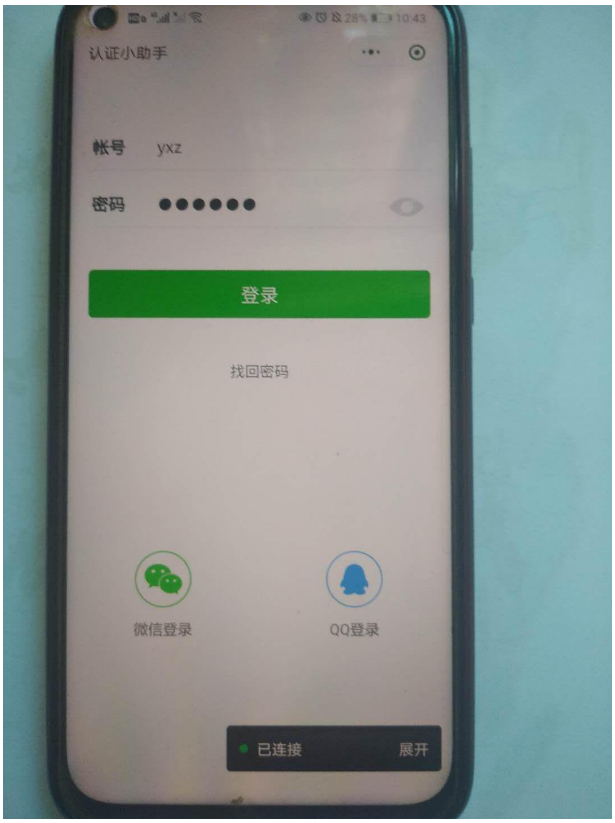

图3-14 微信小程序登录界面

登录界面主要是通过用户名密码对用户进行认证（与个人微信号或QQ号关联），通过之后可以进入功能主界面，进行各项操作。

2.功能主界面

图3-15 微信小程序功能主界面

功能主界面主要是五个按钮，对应五个功能。

1. 修改密码

完成对静态认证密码的修改，修改之后，原有的密码无效。

2. 强制退出系统

当在认证小助手上发现有人非法使用自己的电脑时，可以点击这个按钮，强制退出系统，回到登录认证界面。

3. 强制进入系统

主要是针对合法用户使用时的一些小概率异常情况，比如明明是合法用户，却无法进入系统，则可以点击此按钮，强子进入系统。

4. 查看使用记录

查看过去的电脑使用记录，实时查看电脑的使用情况。

5. 设置认证强度

针对用户对系统不同程度的安全性需求，用户可以设置高、中、低，三个认证强度以适应其需求。

图3-16 微信小程序设置认证强度

3.5 管理员后台网站的实现

管理员后台网站主要是为系统管理员提供的,便于管理员去管理庞大的用户数据，并进行有关处理。

管理员进入时首先要进行登录验证，通过验证之后可以查看所有用户使用系统的信息，以及用户个人的用户名、密码信息。除了查看之外还可以对用户的信息进行增删改等操作。

1.登录界面

网站的登录界面如图3-17所示。

图3-17 网站登录界面

2. 功能主界面

用户成功登录之后，接下来进入功能主界面，界面如图3-18所示。

图3-18 网站功能主界面

进入主界面之后，有如下功能可以选择。

（1） 管理所有用户的信息

可以查看用户的有关信息，包括用户名、密码、微信号、QQ号等信息。方便管理员预览；同时可以根据需求删除对应的用户。

对应的界面如图3-19所示：

图3-19 用户信息管理界面

（2） 管理所有系统使用信息

同时我们还可以管理所有用户对系统的使用情况。界面如图3-20所示。

图3-20 系统使用信息管理界面

（3） 查看所有系统认证强度

用此界面查看对应系统安全强度设置。界面如图3-21所示。

图3-21 系统认证强度界面

3.6 安全传输体系实现

安全传输体系分为套接字传输功能与SM9加密功能两个部分，实现语言都是C++。

3.6.1 套接字传输功能实现

本文使用的都是windows流式套接字，同时，客户端所调用的方法与服务器不同，故分两部分介绍。

客户端模块接口如表3-10所示。

表3-10 客户端套接字接口函数表

方法名字	返回类型	方法描述
Socket()	int	创建套接字
Connect()	int	连接服务器
Send()	int	发送数据
Recv()	int	接收数据

服务器模块接口如表3-11所示。

表3-11 服务器套接字模块接口函数表

方法名字	返回类型	方法描述
Socket()	int	创建套接字
Bind()	int	绑定地址
Send()	int	发送数据
Recv()	int	接收数据
Listen()	int	监听是否有传来的连接
Accept()	int	接受连接

3.6.2 SM9加解密功能实现

SM9算法是一种基于双线性对的标识密码算法，可以生成用户的公私密钥对，可用于数字签名、身份认证、数据加密等领域。本作品实现需要的接口如表3-12所示：

表3-12 SM9算法模块接口函数表

方法名字	返回类型	方法描述

Sign()	Signature	签名
verify()	BOOL	验证签名
Keyencap()	Keyencapsolution	密钥封装
Keydecap()	String	密钥解封
Encrypt()	String	加密
Decrypt()	String	解密
Keychange()	keyagreement	密钥交换

3.7 小结

本章主要根据我们的设计对作品进行了实现，包括环境建立与工具选择，客户端中数据收集模块与用户界面的实现，服务器中数据处理模块、持续认证模块与静态认证模块的实现，击键认证小助手的实现，后台管理员网站的实现，安全传输体系中套接字传输与SM9加密的实现等。

第四章 作品测试与分析

4.1 测试环境说明

为了测试本作品的功能，我们在本机上完成了配置，具体硬件测试设备与环境如表4-1所示.

表4-1 硬件测试环境

用途	环境	鼠标类型	键盘类型
服务器	Windows10+ Core i7	普通光电鼠标	QUERY 全键盘
客户端	Windows10+ Core i7	普通光电鼠标	QUERY 全键盘

由于分类模型训练与测试以及界面展示的必要，我们还要设置测试的软件环境。具体软件件测试设备与环境如表4-2所示.

表4-2 软件测试环境

用途	环境	版本号
分类模型的测试与训练	PyCharm	2020.1.1
界面展示	QT	4.5.0

4.2 测试数据说明

4.2.1 静态认证模块测试数据

（1）数据集的选取：为了方便与他人比较且数据收集同时满足静态认证的要求，静态认证数据模块我们选取的是最早出的CMU基准数据集，可从互联网中搜索得到。

CMU数据集总共包含51名用户的数据，对每一名用户收集了8轮，每次轮入50回“.tie5roanl”。这样就有了8*50*51共20400组数据。每一组数据包含了三个特征，第一类为Hold：即单个按键的持续时间；第二类为UD（Up-down）：即第一个键放下到第二个键抬起的时间，即为两个键之间的间隔时间；第三类为DD（Down-Down）：即第一个键按下到第二个键按下所经历的时间。目前多数作品将这个三个特征都纳入考虑，但显然由于DD=Hold+UD，所以，我们只考虑Hold与UD已经足够了。故本作品选择了单键持续时间与双键持续时间来训练我们的静态认证模型。

（2）训练方案：依次选择一个用户为合法用户，其余用户为非法用户，进行训练与测试。其中，合法用户数据的60%为训练数据，40%为测试数据；同理，其余50名非法用户也按照这种比例划分训练数据与测试数据。

4.2.2 持续认证模块测试数据

（1）数据集的选取：由于持续认证时，用户输入的是自由文本，没有找到公开数据集，加上时间的原因，我们征集了6个自愿者，收集针对11个模型的数据，每个模型共有50组数据。每组数据特征包含两个单键（这里我们将鼠标单击也看成是一个击键行为）的持续时间，以及两个单键之间的间隔时间。这6个志愿者的信息如下表4-3所示：

表4-3 志愿者信息

志愿者编号	年龄	性别	身份
1	20	男	大三学生
2	21	男	大三学生
3	20	男	大二学生
4	20	男	大三学生
5	21	女	大三学生
6	20	女	大二学生

（2）训练方案：与静态认证类似，依次选择一个用户为合法用户，其余用户为非法用户，对每一个模型进行训练与测试。其中，每一个合法用户数据的60%为训练数据，40%为测试数据；同理，其余5名非法用户也按照这种比例划分训练数据与测试数据。

4.3 算法测试

4.3.1 评价标准说明

由于涉及到对认证算法的测试，所以我们有必要给出相关评价标准。

（1）错误接受率（FAR）

这个指标内容是系统错误接收的非法用户（即将非法用户认为合法用户）的比率，FAR越高，系统越不安全；FAR越低，系统更为安全（这意味觉大多数非法用户被识别出来），其公式如4-1：

$$FAR = \frac{非法用户误认为合法用户的次数}{非法用户认证次数} \quad (4\text{-}1)$$

（2）错误拒绝率（FRR）

这个指标内容是系统错误拒绝的合法用户（即将合法用户认为非法用户）的比率，FRR越高，系统可用性低；FAR越低，系统可用性高，用户体验也好，其公式如4-2：

$$FRR = \frac{合法用户误认为非法用户的次数}{合法用户认证次数} \quad (4\text{-}2)$$

（3）准确率（Accuracy）

这个指标内容是系统认证正确的比率，综合评判系统的准确性，其公式如4-3：

$$\text{Accuracy} = \frac{系统认证正确的次数}{系统认证次数} \quad (4\text{-}3)$$

4.3.2 静态认证算法测试与对比

主要是对加权贝叶斯分类与欧式距离算法的测试，测试用例如表4-4所示：

表4-4 静态认证算法测试用例

用例编号	NO.1	模块名称	静态认证算法
测试方法	黑盒测试	测试日期	2020.5.21
测试说明	输入不同用户的击键数据，观察输出结果是否准确		
预置条件	无		
判断标准	对比输出与预期是否相符		
测试数据	CMU公开击键数据集		
测试输出	合法用户或非法用户		
测试评价	测试准确率符合要求，达到较高水准，测试通过		

利用的是PyCharm进行测试，通过改变阈值P，系统的平均FRR与FAR也随着改变，当P=0.07时,性能最优，FRR=3.27%,FAR=2.62。如表4-5所示：

表4-5 阈值与FRR,FAR对照表

阈值	FAR	FRR
0.09	17.97%	1.15%
0.08	9.85%	2.24%
0.07	2.62%	3.27%
0.06	1.24%	8.95%
0.05	0.88%	20.11%

当P=0.07± 0.001时，51名用户的准确率情况如图4-1所示：

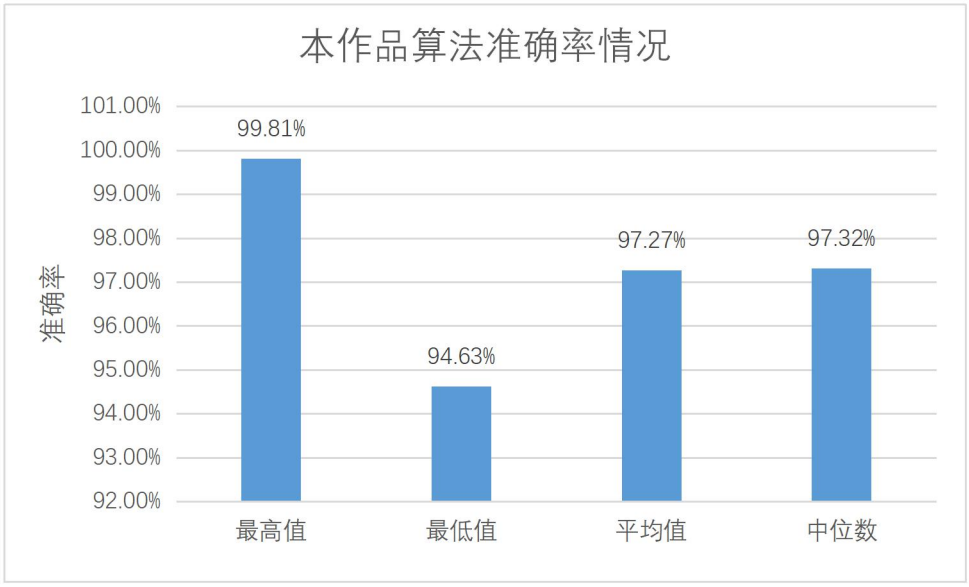

图4-1 用户准确率情况

最终平均准确率达到97.27%，51名用户中最低为94.63%，最高为99.81%，总体在一个比较高的水准上。再对比其他研究，如图4-2所示：

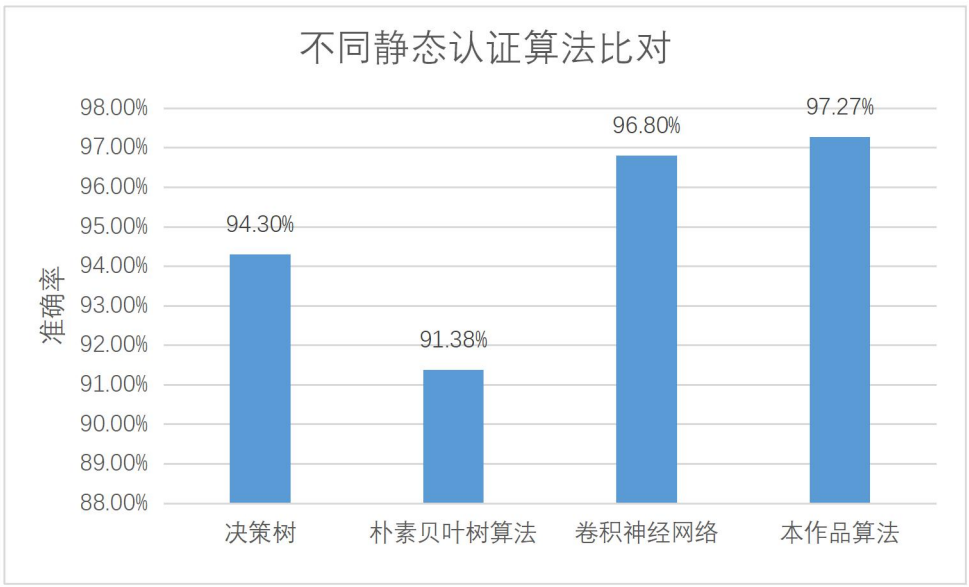

图4-2 不同算法准确率

如图4-2所示，以上算法都是在针对同一个数据集——CMU击键数据集所做的实验，无论是与决策树、朴素贝叶斯算法还是卷积神经网络算法，本作品的算法在认证准确率上均有一定优势。虽然卷积神经网络算法认证准确率与本作品相近，但是它在运算速度上相对较慢，需要花费更长的时间来认证。综上所述，本文算法在针对静态认证的问题上具有高准确性和优异性。

4.3.3 持续认证算法测试与对比

主要是对最小二乘支持向量机分类模型的测试，测试用例如图4-6所示。

表4-6 持续认证算法测试用例

用例编号	NO.2	模块名称	持续认证算法
测试方法	黑盒测试	测试日期	2020.5.22
测试说明	输入不同用户的击键数据，观察输出结果是否准确		
预置条件	无		
判断标准	对比输出与预期是否相符		
测试数据	收集6位志愿者的数据		
测试输出	合法用户或非法用户		
测试评价	测试准确率符合要求，达到较高水准，测试通过		

每一个志愿者的准确率变化如图4-3所示。

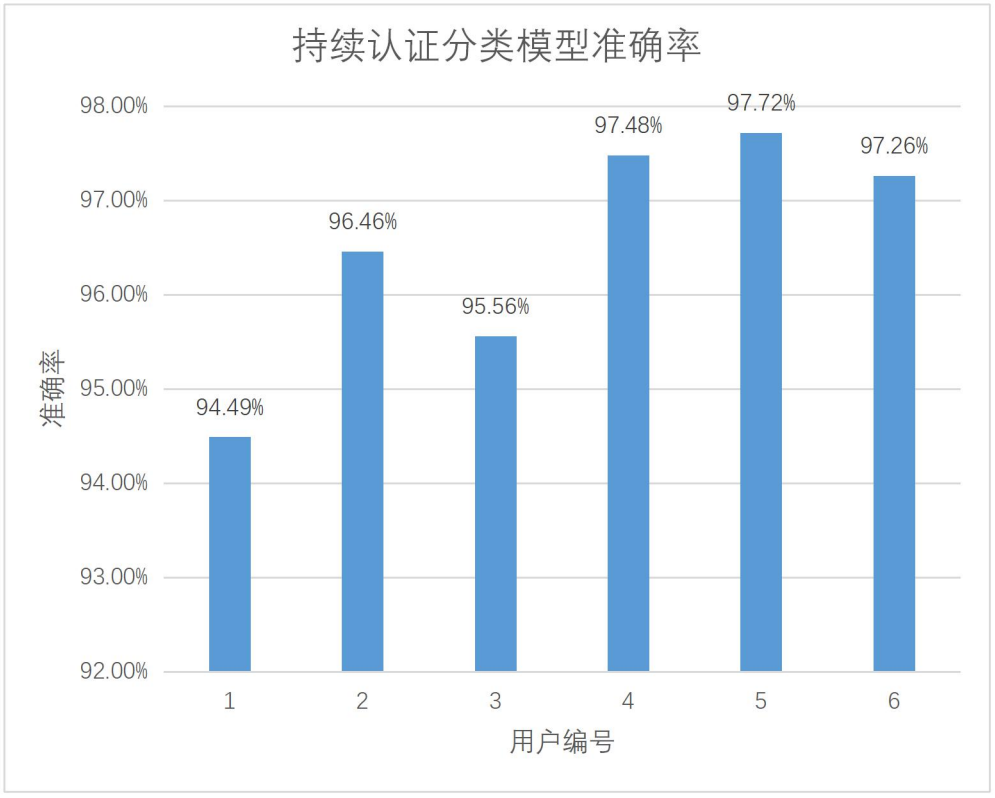

图4-3 志愿者准确率

细致对比每一个志愿者的FAR与FRR，基本维持在一个比较低的区间，证明我们的模型得到了一个比较满意的结果。

每个用户的具体FAR与FRR如表4-7所示：

表4-7 FAR与FRR情况

用户编号	FAR	FRR
1	5.64%	4.88%
2	3.21%	5.22%
3	4.62%	3.57%
4	2.24%	3.95%
5	1.88%	4.31%
6	2.58%	3.54%

对以上结果分析，发现平均准确率达到96.5%，与其他算法作对比，结果如图4-4所示：

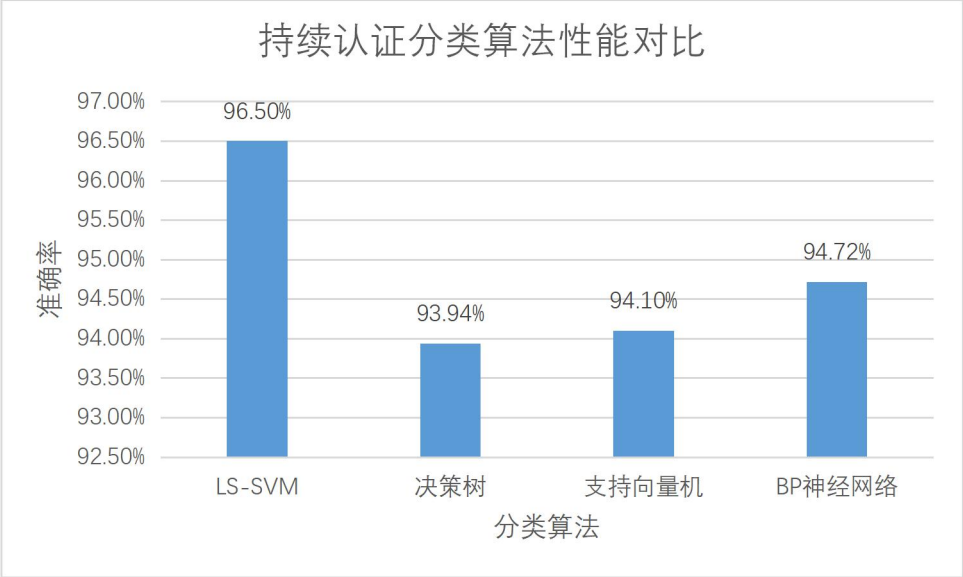

图4-4 不同算法准确率

综合对比其他算法，在都针对本作品所收集的数据集上，本作品选择的最小二乘支持向量机算法（LS-SVM）认证准确率最高，较好地通过了测试。

4.4 功能测试

4.4.1 PC客户端测试

4.4.1.1 数据收集功能测试

数据收集功能主要是用来收集用户的键盘与鼠标的击键数据，具体就是在系统启动开始，什么时间什么键有什么动作，比如（keydown,a,40598）表示在程序运行后第40598ms,a键被按下了。

具体测试用例如表4-8所示

表4-8 数据收集功能测试用例

用例编号	NO.3	模块名称	数据收集功能
测试方法	黑盒测试	测试日期	2020.5.22
测试说明	用户随机敲击鼠标键盘，观察信息是否被记录		
预置条件	布置好了钩子程序		
判断标准	是否收集到了预期的击键信息		
测试输出	用户击键信息的记录（txt）		
测试评价	成功收集到了数据，测试通过		

首先先观察数据收集功能，随意点击鼠标或键盘，发现信息都被记录，结果如图4-5所示：

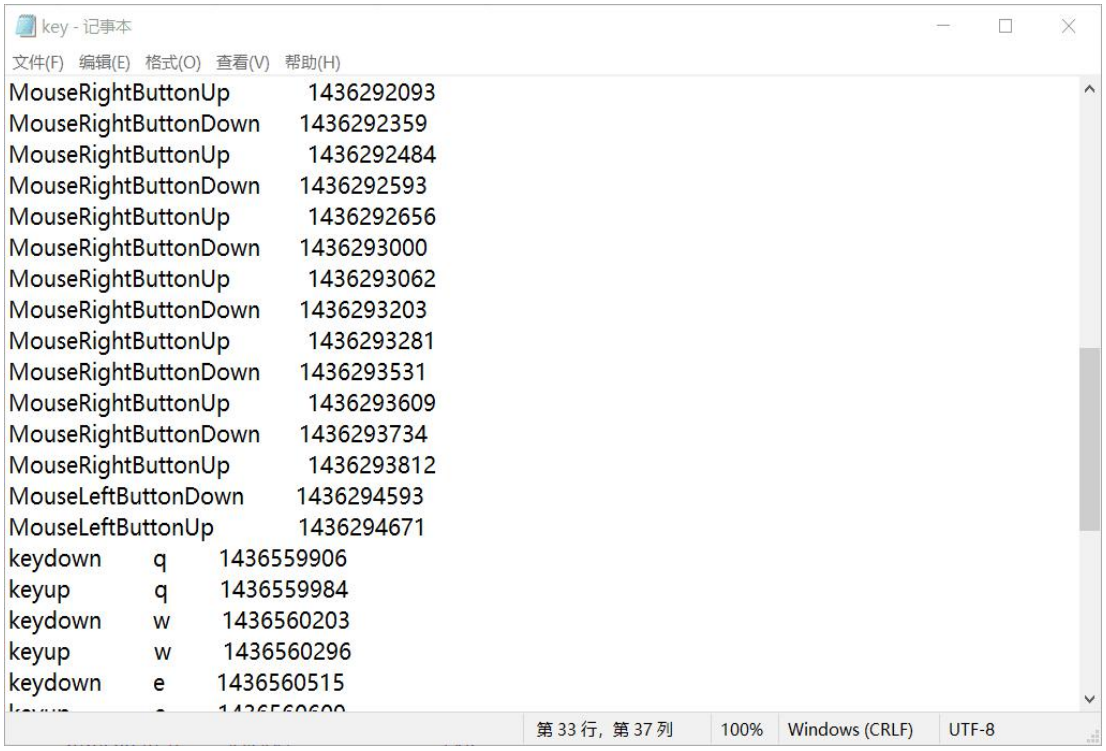

图4-5 数据收集结果

4.4.1.2 注册功能测试

用户进行注册，完成对应注册功能的测试。

用户需依次输入对应的用户名、口令、微信号、QQ号（非必须），完成一次成功的注册。

测试用例如表4-9所示。

表4-9 注册功能测试用例

用例编号	NO.4	模块名称	注册功能
测试方法	黑盒测试	测试日期	2020.5.22
测试说明	进行一次合法的注册。		
预置条件	无		
判断标准	是否产生了预期的效果		
测试输出	是否成功注册		
测试评价	功能与预期一致，测试通过		

结果如图4-6所示：

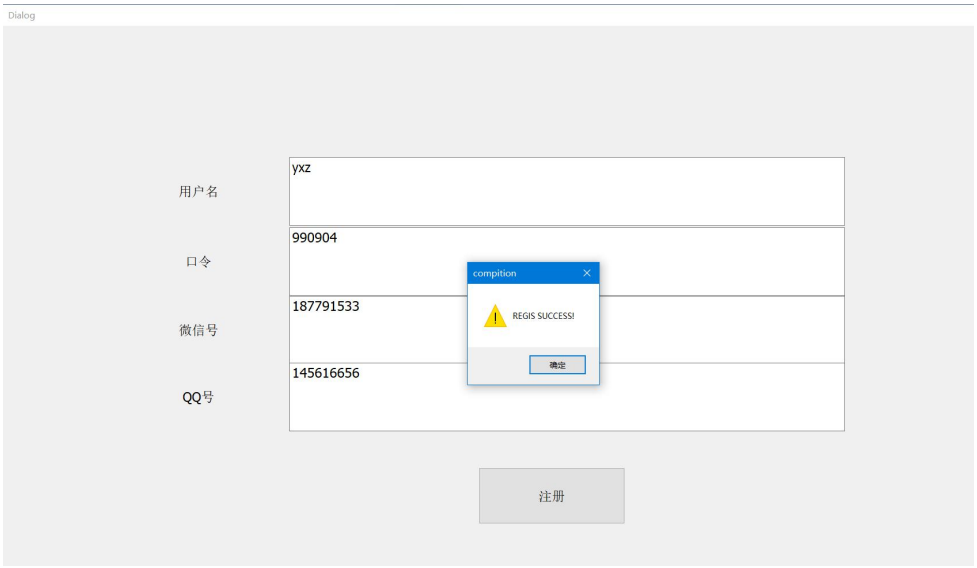

图4-6 注册结果

4.4.1.3 静态认证功能测试

静态认证测试，主要是针对用户登录系统时的，主要看是否实现了静态认证的功能，包括注册、用户名密码与击键特征双因素认证等。前提是需要布置好钩子程序，能收集到用户的击键数据。

具体的测试用例如表4-10所示。

表4-10 静态认证功能测试用例

用例编号	NO.5	模块名称	静态认证功能
测试方法	黑盒测试	测试日期	2020.5.22
测试说明	对用户进入系统时（注册登录）的认证，通过进行各种情况，判		

	断功能是否达到我们的需求
预置条件	布置好了钩子程序
判断标准	是否产生了预期的效果
测试输出	是否合理通过验证，进入系统
测试评价	功能与预期一致，测试通过

对于已经注册了的用户，再点击注册，结果如图4-7所示。

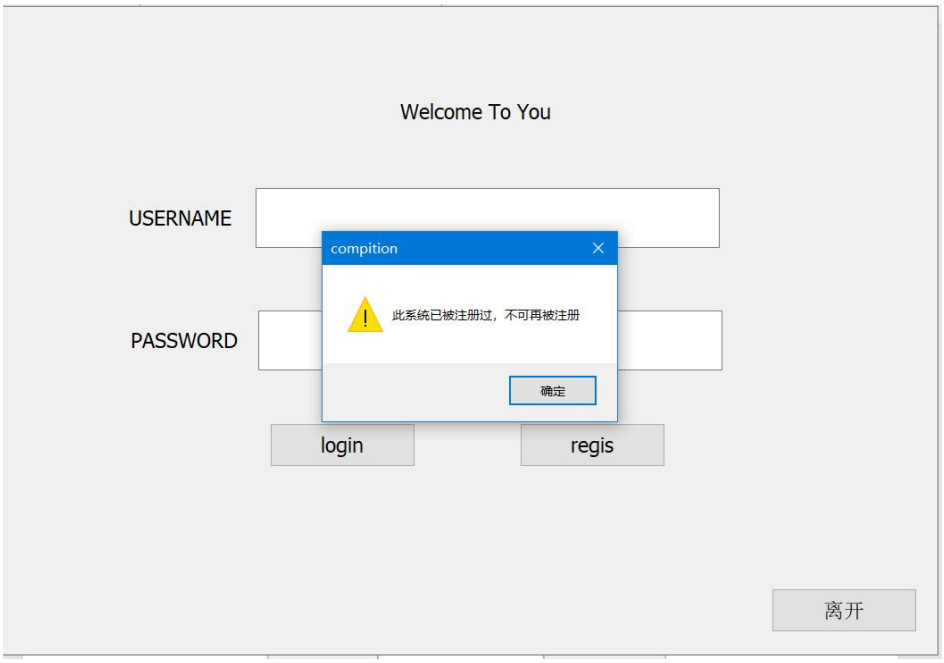

图4-7 再次注册结果

输入错误的密码（正确的为：yxz,990904），结果如图4-8所示。

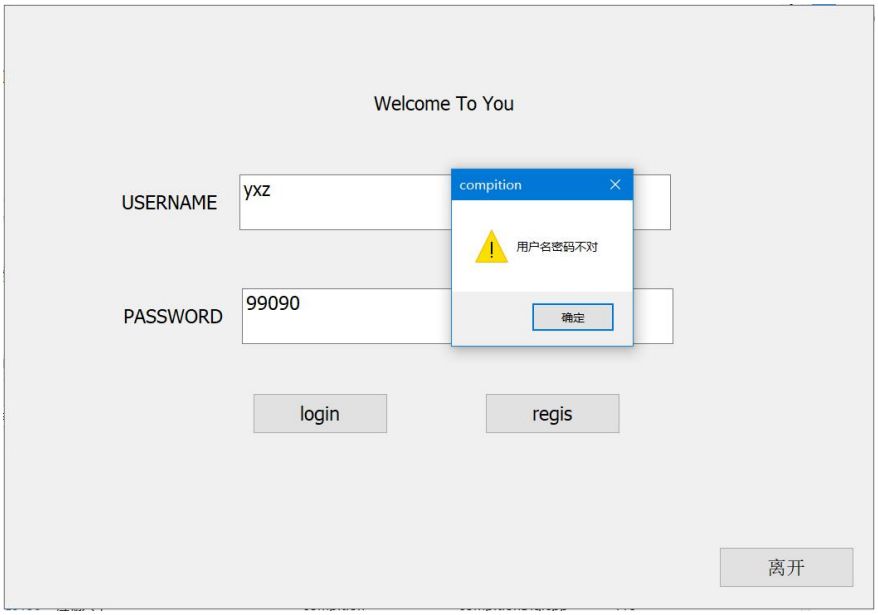

图4-8 密码错误结果

由非法用户输入正确密码，结果如图4-9所示。

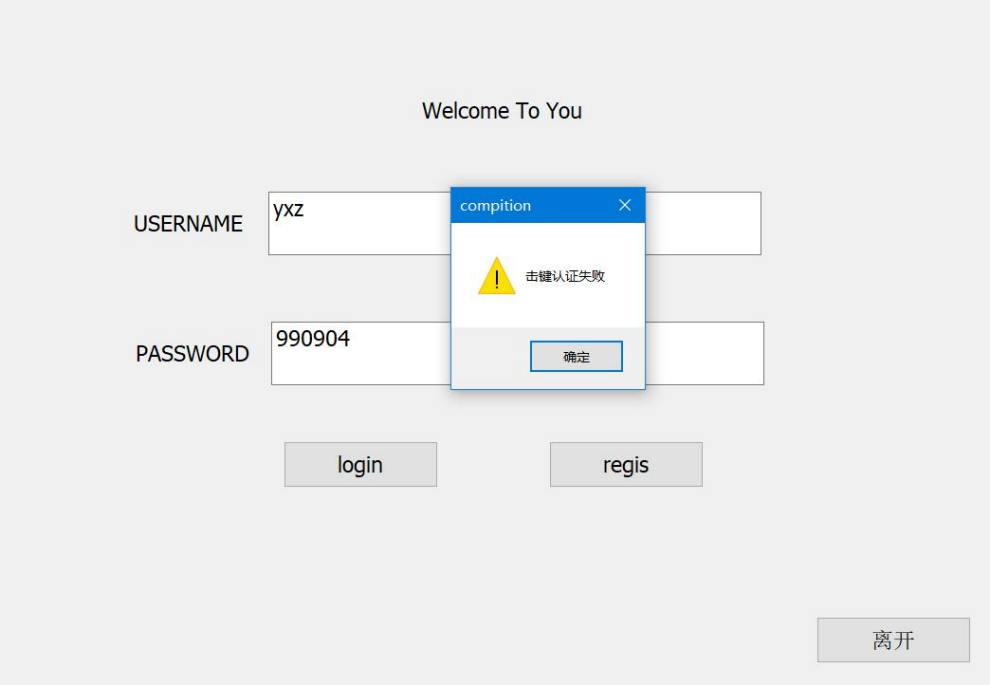

图4-9 输入正确密码，但击键习惯不同

4.4.1.4 持续认证功能测试

持续认证模块，主要是在用户通过登录后使用系统时，观察是否是合法用户正在使用系统，如果不是，最终会给出持续认证失败的提示，强制退出系统，

测试用例如表4-11所示。

表4-11 持续认证功能测试用例

用例编号	NO.6	模块名称	持续认证功能
测试方法	黑盒测试	测试日期	2020.5.22
测试说明	非法用户进入系统，依次鼠标双击、键盘输入nihaozhongguo		
预置条件	布置好了钩子程序，开启了后台监听程序		
判断标准	合法用户继续使用系统，非法用户被强制退出系统		
测试输出	非法用户被强制退出系统		
测试评价	功能与预期一致，测试通过		

如果用户成功登录了系统，分类模型建立后，如果用户的鼠标与击键行为不符合平常习惯，则会强制退出系统。如今我们以非法用户进入系统，依次鼠标双击、键盘输入nihaozhongguo。测试结果如图4-10,4-11,4-12所示。

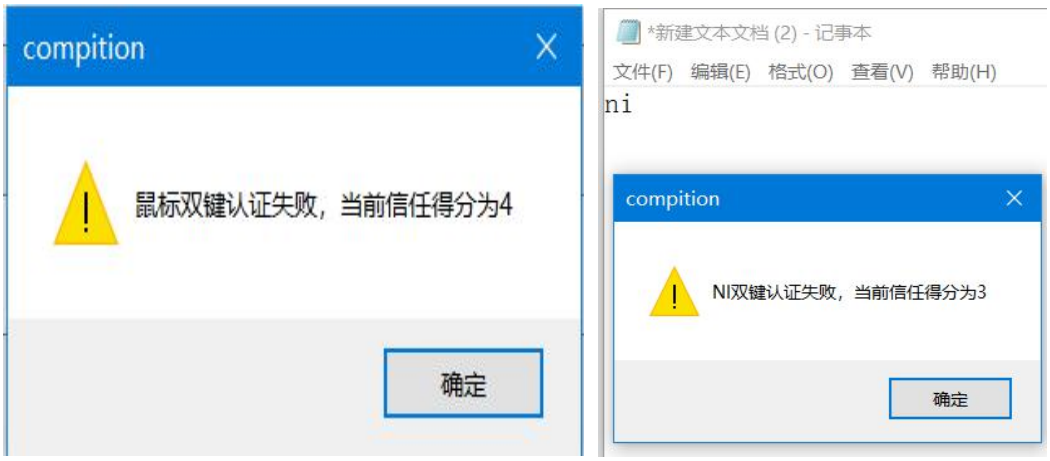

图4-10 持续认证失败界面1

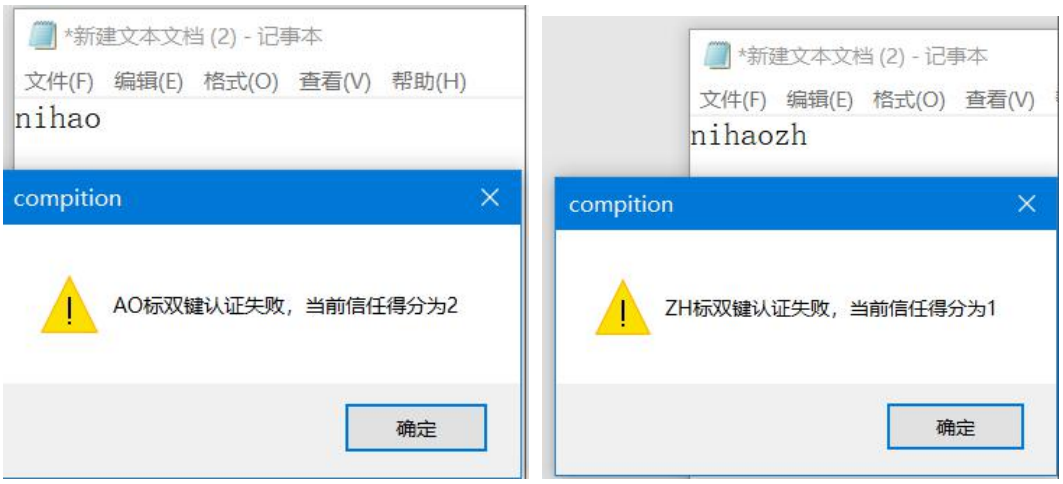

图4-11 持续认证失败界面2

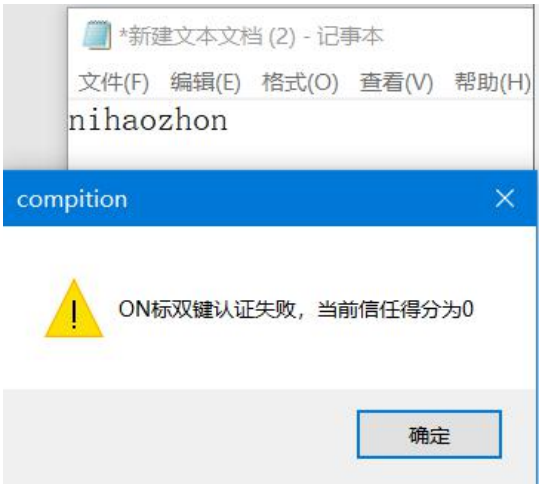

图4-12 持续认证失败界面3

如上图4-10,4-11所示，非法用户在使用电脑的过程中依次命中了鼠标双键,NI,AO,ZH,ON这五个分类器，被分类为非法用户，信任得分逐渐降低，低于0时，

被强制退出系统，如图4-13所示。

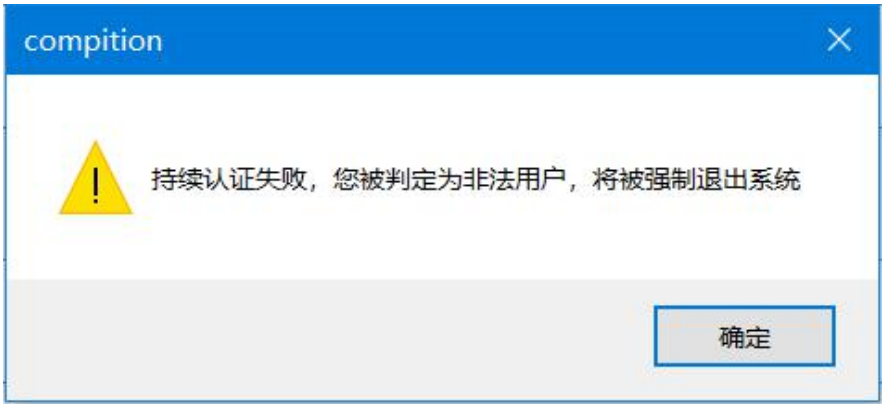

图4-13 被强制退出系统

综上所述，持续认证功能通过了我们的测试。

4.4.2 击键认证小助手功能测试

4.4.2.1 登录功能测试

是对小程序登录功能的测试，阻止非法用户登录。

测试用例如表4-12所示。

表4-12 登录功能测试用例

用例编号	NO.7	模块名称	小程序登录功能
测试方法	黑盒测试	测试日期	2020.8.8
测试说明	用户输入错误的密码或不在注册时所在的微信登录		
预置条件	服务器开启		
判断标准	是否成功登录		
测试输出	没有成功登录		
测试评价	功能与预期一致，测试通过		

密码错误时，结果如图4-14所示。

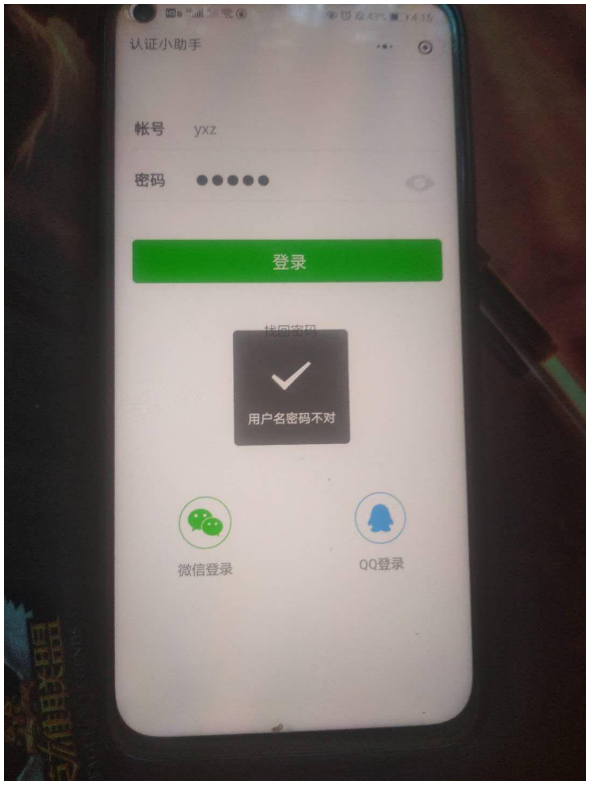

图4-14 密码错误

密码错误时，结果如图4-15所示。

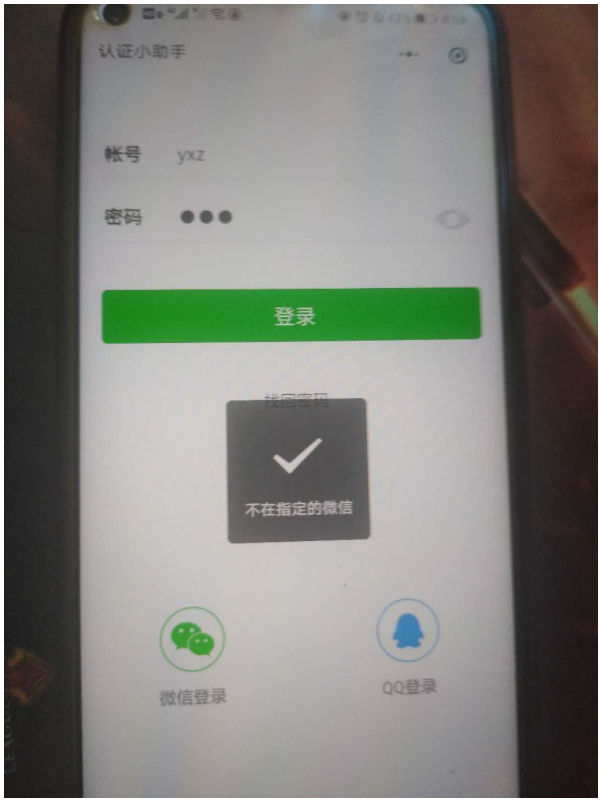

图4-15 不在指定微信号登录

4.4.2.2 修改密码功能测试

测试用例如表4-13所示。

表4-13 修改密码功能测试用例

用例编号	NO.8	模块名称	小程序修改密码功能
测试方法	黑盒测试	测试日期	2020.8.8
测试说明	修改登录密码		
预置条件	服务器开启		
判断标准	是否修改成功		
测试输出	修改成功		
测试评价	功能与预期一致，测试通过		

结果如图4-16所示：

图4-16 修改密码

4.4.2.3 强制进入/退出系统功能测试

测试用例如表4-14所示。

表4-14 强制操作功能测试用例

用例编号	NO.9	模块名称	小程序强制进

			入退出系统功能
测试方法	黑盒测试	测试日期	2020.8.8
测试说明	对目标系统进行操作		
预置条件	服务器开启		
判断标准	是否操作成功		
测试输出	操作成功		
测试评价	功能与预期一致，测试通过		

点击如图4-17所示的对应的按钮，完成了对应的功能。

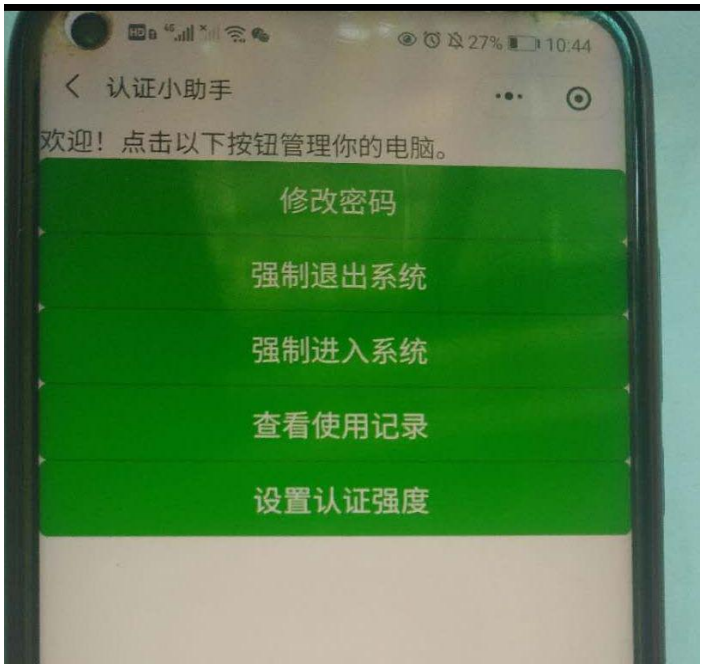

图4-17 强制操作系统

4.4.2.4 查看系统使用记录功能测试

测试用例如表4-15所示。

表4-15 查看记录功能测试用例

用例编号	NO.10	模块名称	小程序查看系统使用记录功能
测试方法	黑盒测试	测试日期	2020.8.8
测试说明	查看操作记录		

预置条件	服务器开启
判断标准	是否操作成功
测试输出	操作成功
测试评价	功能与预期一致，测试通过

结果如图4-18所示。

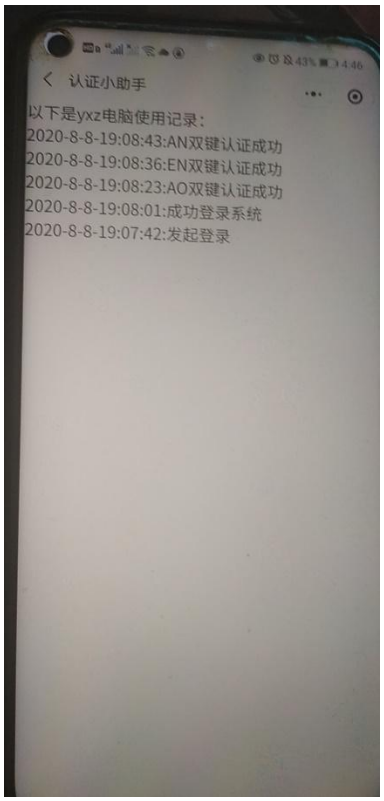

图4-18 查看记录

4.4.2.5 设置认证强度功能测试

测试用例如表4-16所示。

表4-16 认证强度功能测试用例

用例编号	NO.11	模块名称	小程序改变认证强度功能
测试方法	黑盒测试	测试日期	2020.8.8
测试说明	改变认证强度		
预置条件	服务器开启		

判断标准	是否操作成功
测试输出	操作成功
测试评价	功能与预期一致，测试通过

结果如图4-19所示。

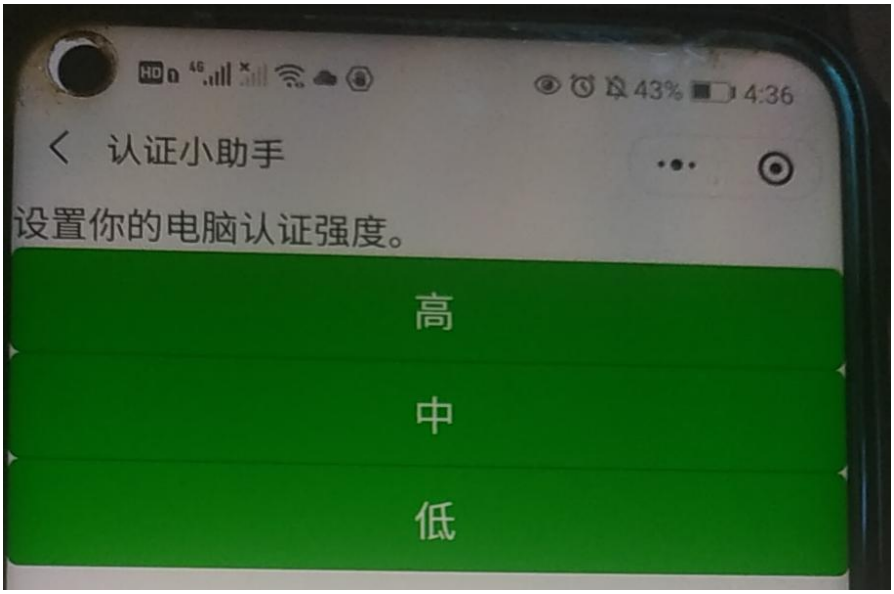

图4-19 改变认证强度

4.4.3 管理员后台网站功能测试

4.4.3.1 登录功能测试

测试用例如表4-17所示。

表4-17 登录功能测试用例

用例编号	NO.12	模块名称	网站登录功能
测试方法	黑盒测试	测试日期	2020.8.8
测试说明	管理员进行登录，输入正确密码		
预置条件	服务器开启		
判断标准	是否操作成功		
测试输出	操作成功		
测试评价	功能与预期一致，测试通过		

结果如图4-20所示。

图4-20 登录结果

4.4.3 查看记录功能测试

测试用例如表4-18所示。

表4-18 查看记录功能测试用例

用例编号	NO.13	模块名称	网站查看记录功能
测试方法	黑盒测试	测试日期	2020.8.8
测试说明	管理员查看有关记录		
预置条件	服务器开启		
判断标准	是否操作成功		
测试输出	操作成功		
测试评价	功能与预期一致，测试通过		

结果如图4-21，4-22所示。

图4-21 认证强度记录

图4-22 用户

4.4.3 删除用户功能测试

测试用例如表4-19所示。

表4-19 删除用户功能测试用例

用例编号	NO.14	模块名称	网站删除用户功能
测试方法	黑盒测试	测试日期	2020.8.8
测试说明	删除Jack用户		
预置条件	服务器开启		
判断标准	是否操作成功		
测试输出	操作成功		
测试评价	功能与预期一致，测试通过		

结果如图4-23所示。

图4-23 删除结果

4.5 性能测试

本节通过分别模拟50，100，150，200，250，300，350，400个用户同时登录或命中双键分类模型，观察服务器平均响应时间，结果如图4-24所示：

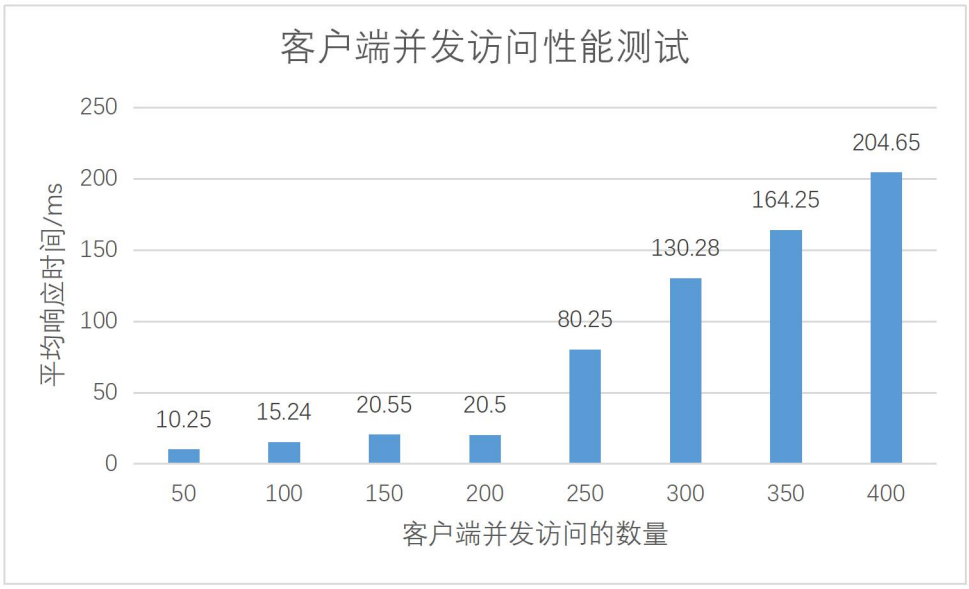

图4-24 性能测试

如图4-10所示，随着客户端并发访问数量的增多，平均响应时间也在变长，当400个客户端并发访问服务器时，服务器的平均响应时间为204.65ms，整体可以接受。

4.6 小结

本章主要对本作品的核心算法通过选用公开数据集或收集数据等手段进行了算法测试，同时对本作品的功能、性能进行了测试，最终均得到了比较满意的结果。

第五章 安全性分析

基于鼠标与键盘击键行为的身份认证系统，其安全性包括系统的稳定性和承受恶意攻击的能力。安全性分析是有关验证应用程序的安全服务和识别潜在安全性缺陷的过程。本章从非法访问控制、暴力破解控制、信息伪造控制、数据库注入攻击与 DDOS 攻击五个方面对本系统进行相应的安全性分析。

5.1 非法访问控制

非法访问是指通过扫描仪、黑客程序、隐蔽通道、远端操纵、密码攻击等方式窃取或截获用户名、口令，窃取超级用户权限，破解密码等[12]，这种攻击的示意图如图 5-1 所示。针对非法访问的密码攻击手段，相较于传统文本密码的身份认证或其他生物认证方式，本系统采取的是击键识别方式，不仅能有效抵御攻击，而且部署成本低，无需额外的辅助设备。如第四章作品测试所述，我们将击键识别的安全性指标采用量化的方法来表示，我们引入误识别率作为描述指标。在进行大量测试之后，找到最合适的误识别率阈值，既能确保大多数合法用户通过认证，也满足软件对于安全性的要求。非法访问如图 5-1 所示。

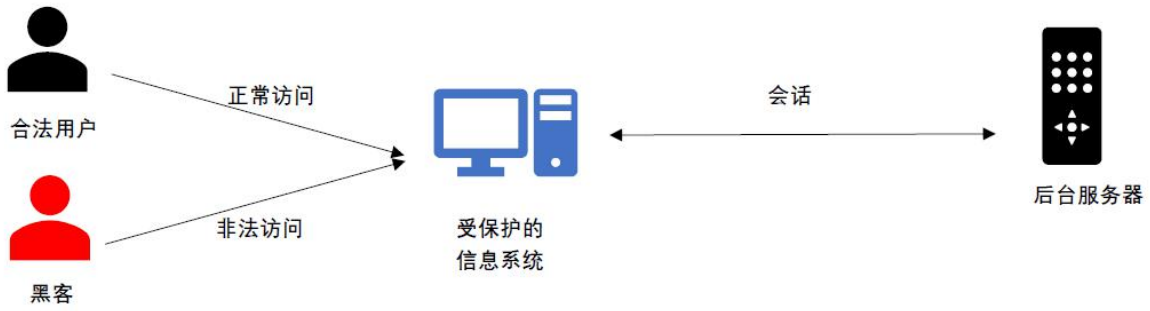

图 5-1 非法访问示意图

对于登录时的非法访问，我们采用的是用户名密码与击键习惯进行双因素认证，即使攻击者知道了密码，也难以模仿用户的击键习惯。

而同时，就算攻击者进入了系统（比如登录时知道了密码且静态认证模型还未建立或者用户使用系统时中途离开，攻击者乘虚而入），后台依然运行着持续认证模型，一旦攻击者操作鼠标或键盘被判定为非法用户，使得认证得分低于阈值，则会强制退出系统。

使用系统时，防止非法访问的过程如图 5-2 所示。

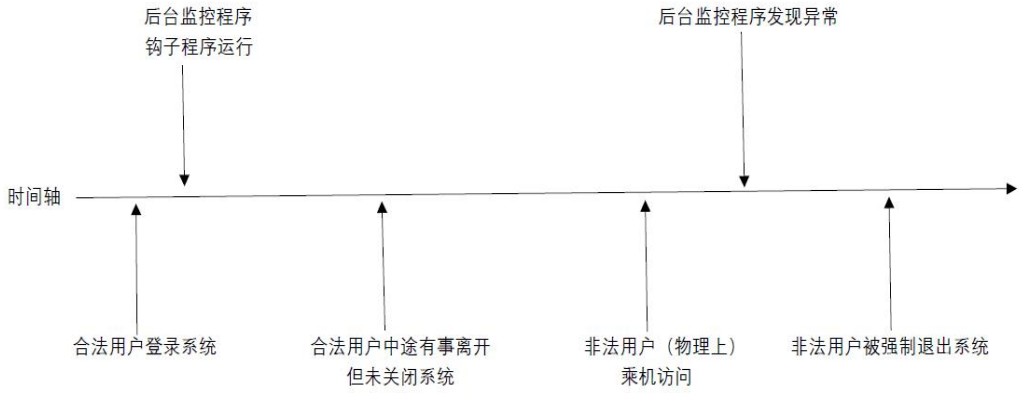

图 5-2 使用系统时，攻击者的入侵以及抵御方式

针对非法访问攻击网络通信以及数据库操作的方式，基于鼠标与键盘击键行为的身份认证系统采用国密 SM9 算法对网络传输数据进行安全有效的传输。由于采取了 SM9 加密，攻击者也无法进行报文数据解密，用户数据仍然是安全的，这样就会使得针对信息网络传输和数据库的非法攻击的方式对本系统而言基本无效。

5.2 暴力破解控制

暴力破解的核心思想就是穷举法。穷举法对有限集合模型的系统具有很大的威胁。攻击者通过系统地列举所有可能性（比如登录时用到的用户名、密码），尝试所有的可能组合破解用户的用户名、密码等敏感信息。

对于这种攻击，我们可以考虑使用更加复杂的密钥，比如增大密码位数、使用数字字母与其他符号多元组合的方式来抵御攻击，目的就是让攻击者即使可以采用穷举攻击，也要花费足够长的时间（可能数十年甚至更多）来破解。

对于本作品，由于登录或使用系统时主要采用击键特征作为认证方式，即使攻击者破解了登录密码，也复制不了登录习惯。

而攻击者可能采取的一种攻击方式是监听网络数据传输过程，从而获取一些个人信息。达到一些非法目的。本作品在通信时采用 SM9 密钥加密，SM9 加密算法采用 256b 素域上的椭圆曲线，离散对数的复杂性约为2^{128}次基本运算，故具有很好的稳定性。椭圆曲线-256 求解时间表如下，系统破解花费时间巨大，目前为止，还没有找到有效的方法破解 256 位椭圆曲线的密码[7]，无论是个人、组织，还是国家，求解时间都足够长。

椭圆曲线-256 求解时间具体如表 5-1 所示。

表 5-1 椭圆曲线-256 求解时间表

破解主体	个人	组织	国家
时间	$2^{64.1}$	$2^{57.4}$	$2^{54.1}$

综上所述，对于暴力破解攻击，本作品能够很好地抵御。

5.3 信息伪造控制

信息伪造指攻击者使用虚假的身份与信息与服务器通信，从而进行数据破坏或数据窃取等攻击。信息伪造攻击如图 5-3 所示。

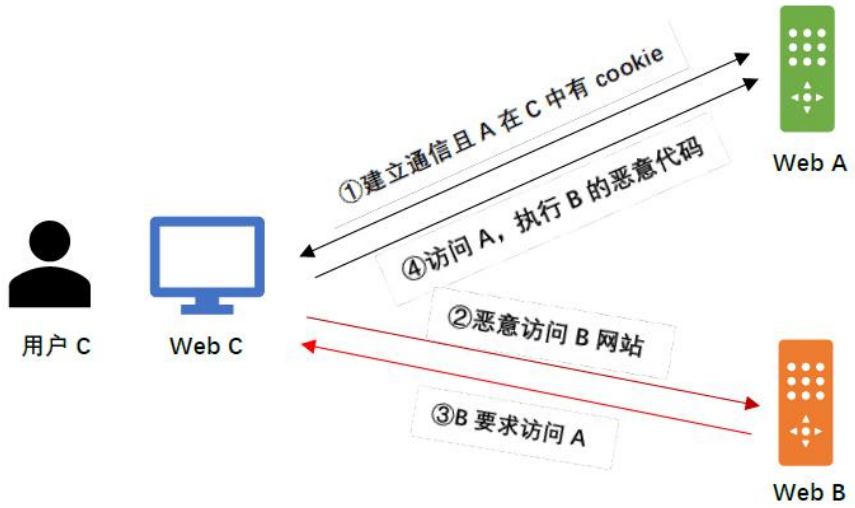

图 5-3 信息伪造攻击示意图

如图 5-3 所示，当前用户端在接收到这些攻击性代码后，根据网站 B 的请求，由于在合法用户不知情的情况下携带了 Cookie 信息，向网站 A 发出请求。网站 A 并不知道该请求实际是由 B 发起的，所以会根据合法用户 C 的 Cookie 信息以合法用户请求的方式处理，导致来自网站 B 的恶意代码被执行。

本认证系统主体是采用击键特征为认证标准的，采用加权贝叶斯分类与欧式距离结合的算法或最小二乘支持向量机算法在用户使用系统的全过程进行认证，利用底层钩子程序收集击键数据，具有不易伪造、击键数据不易被窃取等特点。甚至，即使掌握了击键特征的具体数据，在操作时，由于人与人之间击键时延等特征不同（差距往往是几百毫秒），所以几乎不可能通过模仿另一个人的击键特征方式进入信息系统。所以，人工伪造的方式基本不可行。

5.4 数据库注入攻击

数据库注入攻击，就是通过构建特殊的数据库命令，并且将它插入到 Web 表单提交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意代码，进行一些非法活动，非法获取信息。具体来说，它是利用当前现有应用程序，将恶意构造的数据库命令注入到后台数据库引擎执行，违背设计者的意图，改变程序本来流程，获取一些不该被获取的信息或进行一些不该进行的活动。常见的数据库注入攻击如图 5-4 所示。

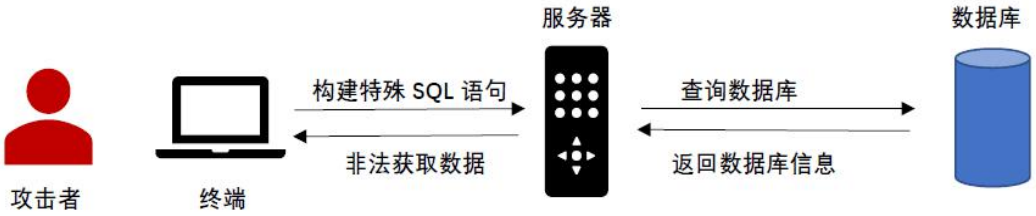

图 5-4 数据库注入攻击示意图

对于数据库注入攻击，本作品做的的防范方式有：（1）不使用管理员权限的数据库连接，为每个用户使用单独的权限有限的数据库连接。（2）不把机密信息以明文的方式直接存放，使用 hash 加密密码，并在数据传输过程中使用 SM9 加密算法。（3）通过正则表达式和限制长度对用户输入进行限制。

5.5 小结

在本章，我们对本作品的安全性进行了阐述。分析了击键认证方式以及 SM9 加密算法的安全性与稳定性，并评估了本系统对于非法访问控制、信息伪造控制、暴力破解控制以及数据库注入攻击等可能面临的安全问题的防护能力，确保本系统的稳定性和有效性。

第六章 创新性说明

本章通过对比现有的其他有关击键认证的作品或有关研究，对本作品提出的基于鼠标与键盘击键行为的身份认证系统进行创新性说明，主要体现在如下几个方面。

1、首个基于击键行为的面向用户使用全过程的身份认证系统

目前，有许多种认证方式，无论是传统的基于用户名密码、人脸识别、指纹识别等认证方式，还是其他的基于鼠标与键盘击键认证方式，本质上都是单次认证，即只在用户登录系统时进行认证，而在用户进入系统之后就不再进行认证。显然，如果攻击者盗取了用户的认证信息或者用户在使用系统时短暂离开，那么攻击者便可乘机进入系统，查看一些个人信息。

而本作品提出的基于鼠标与键盘击键行为的身份认证系统，不仅在用户登录时，根据用户的输入的用户密码与击键特征进行认证，而且还在用户进入系统之后，依据用户产生的鼠标或键盘的击键行为进行认证，从而确保用户使用系统整个过程都是安全的，不会被攻击者使用，做到了对个人信息系统的全方位的守护。这是当前诸多认证系统都未实现的功能。

2、基于加权贝叶斯分类与欧式距离结合算法的静态认证模块

对于用户登录系统时，我们采用的是用户名密码与用户击键特征双因素认证的方式。一些其他的击键静态认证方案，使用了卷积神经网络、BP 神经网络、决策树、朴素贝叶斯分类等分类算法，它们存在着认证精确率不高、认证时间过长、选取的击键特征过多等问题。而本作品采用的是以单键持续时间和双键间隔时间作为主要分类特征，使用基于加权贝叶斯分类与欧式距离结合的分类算法，考虑到了每个特征的重要程度不同与系统可用性和安全性平衡的需要。用户输入密码时产生的击键特征，不仅要通过加权贝叶斯分类而且欧氏距离必须要低于阈值，才可被判定为合法用户。最终静态认证模块认证准确率达到了 97.27%，高于当前绝大多数作品，且认证时间可以接受。

3、基于信任得分机制的持续认证模块

在持续认证模块，很多研究采用的是滑动窗口的机制，即必须要用户产生所有需要作为模型输入的特征，才开始认证，如果用户永远不输入某个特征，那么系统永远

无法进行认证。除此之外，只要用户在一次认证中被判定为非法用户，那么就必须强制退出系统，显然，这带有不可忽视的偶然性。

而本作品采用的是基于信任得分机制的持续认证方案，我们选取了用户最常使用的十一对鼠标或键盘双键，只要用户产生了十一对双键中任一对的数据，基于最小二乘支持向量机的分类模型便开始判断，其中分类模型准确率达到了 96.5%，高于当前绝大多数击键持续认证作品。整体的流程是，用户登录系统时，根据用户的击键特征吻合程度会产生一个初始信任得分，如果用户产生的击键行为命中了某个双键模型的话，会在用户的初始信任得分的基础上加上或减去一个分数，如果用户的得分低于某个阈值，则强制退出系统。

4、首个基于国密算法 SM9 与套接字的安全分布式认证系统网络

当前，网络形势非常严峻。我们在网络传输数据时需要考虑安全。我们在服务器与客户端通信时，使用了套接字作为基本通信方式，同时，使用了国密算法 SM9 对要传输的数据加密，完成了信息的安全可靠传输，建立了首个基于国密算法 SM9 与套接字的安全分布式认证系统网络。

第七章 总结与展望

当前信息安全局势严峻，身份认证作为安全系统的第一道防线，发挥着重要的作用。我们提出的基于鼠标与键盘击键行为的身份认证系统，编写底层HOOK程序收集击键数据，分类模型采用了加权贝叶斯分类与欧式距离结合的算法、最小二乘支持向量机算法，使得我们作品的认证精确率超过了几乎所有的同类产品。而采用的基于信任得分机制的持续认证方案，采用更加贴近实际应用的方式，保证了用户在使用系统时的安全。最终，我们的作品可以在无需其他任何额外辅助设备的情况下，仅依靠用户的鼠标与键盘击键行为便可完成对用户的认证，保障系统被使用的全过程的安全。

但本作品仍然存在着巨大的发展空间，比如:

（1） 可以将本作品迁移到移动设备上，不仅仅是局限在PC端。

（2） 可以根据用户的击键数据，判断用户的心理或生理情况，从而做出一些提示。

（3） 将本产品与一些具体的应用（比如QQ,WeChat）结合起来，可以防止一些盗号等现象的发生，而不是局限在物理系统的安全使用。

我们将对本作品进行持续的研究和完善，尽快推动产品落地，并继续攻克技术上的难题。同时，本作品也为身份认证研究提供了一种新的解决方案，能依靠击键行为完成对系统整个使用过程的认证。

参考文献

[1]陈功.基于击键行为与鼠标行为的动态认证[D].上海交通大学，2018.

[2]刘梦昕.基于用户击键行为的认证技术研究[D].北京邮电大学，2019.

[3]郑航，廖闻剑，唐楚俏.基于键盘与鼠标击键行为的用户身份识别[J].计算机与数字工程，2019(2)：476-480.

[4]Mondal S,Bours P.A study on continuous authentication using a combination of keystroke and mouse biometrics[J].Neurocomputing,2017:1-22 .

[5]E安全.身份认证新技术：美国国防部采用击键生物跟踪识别技术验证身.[EB/OL]. https://mp.weixin.qq.com/s?src=11&timestamp=1592105444&ver=2399&signature=jPXt*cQOTAykjIDTOmhhnCJ*UbMd2*0iG36u8xBAO0thRptwDyrROabfkXPgiYyzjcStQAtih0p0EDjjNoLqO7Xn92IchQvBGGQjcho4FjARgbUYA2XiOlFfIRCOMcEh&new=1.

[6]36氪.与现有生物认证体系相互补充,“集赢智能”提供击键识别认证方案.[EB/OL]. https://m.sohu.com/a/385312228_114778.

[7]岗位安——基于SM9与眼电图的行为识别系统.

[EB/OL]. https://www.ryjiaoyu.com/search?q=%E5%B2%97%E4%BD%8D%E5%AE%89

[8]施琛.基于钩子函数的应用程序快捷键设置[D].计算机与数字工程，2013(01):92-94.

[9]沈超等.基于鼠标行为特征的用户身份认证与监控[J].通信学报，2010(07):68-75.

[10]程然.最小二乘支持向量机的研究和应用[D] .哈尔滨工业大学，2019.

[11]袁峰,程朝辉.SM9标识密码算法综述[J].信息安全研究,2016,2(11):1008-1027.

[12]王瑞锦，周世杰等.信息安全系统研发——全国大学生信息安全竞赛（作品赛）指导教程[J].人民邮电出版社，2019，12(1):181-184.

[13]钩子技术介绍.

[EB/OL]. https://blog.csdn.net/hellokandy/article/details/72758809

[14]贺鸣，孙建军等. 基于朴素贝叶斯的文本分类研究综述[J].情报科学,2015,03.

[15]袁峰，程朝辉.SM9标识密码算法综述[J].信息安全研究,2016,11:1008-1027

[16]王凯等.基于MFC的航空电子系统综合检测设备客户端软件设计与实现[J].计算机测量与控制,2020,02:126-130

[17] Ting-Yi Chang, Cheng-Jung Tsai.New soft biometrics for limited resource in keystroke dynamics authentication[J]. Multimedia Tools and Applications, 2020(prepublish), pp.1-30

[18]王凯，宋礼鹏，郑家杰.融合击键内容与击键行为的持续身份认证[J].计算机工程与设计，2020,06:1562-1567

[19]钟意.基于用户交互行为特征的持续身份认证研究[D].重庆邮电大学，2019.

[20]王雨华.基于击键动力学的智能手机身份认证技术的研究与实现[D].北京邮电大学，2019.