编号： *CACR20XXxxxxxx*

作品类别：☑软件设计 ☐硬件制作 ☐工程实践

2025年第X届全国密码技术竞赛作品设计报告

题目：行为口令——基于击键动力学的身份认证系统

2025年5月12日

中国密码学会

基本信息表
编号：CACR2023xxxxxx
作品题目：行为口令——基于击键动力学的身份认证系统
作品类别：☑软件设计 □硬件制作 □工程实践
作品内容摘要： 本作品实现了一种基于用户击键行为特征的身份认证系统。通过采集用户输入密码时的击键间隔、按键时长和修正次数等生物行为特征，结合密码哈希验证，构建双重安全校验机制。系统采用Python实现，包含行为特征采集、动态阈值匹配、用户管理三大核心模块，在保证密码安全的基础上增加了生物行为特征认证维度。 作品特色： 1）双重认证机制：密码验证与生物特征验证结合 2）动态行为采集：毫秒级精度记录击键时序特征 3）智能误差容忍：采用动态阈值匹配算法 4）轻量跨平台：基于Python实现多平台适配
关键词： 击键动力学、生物特征识别、身份认证、行为分析、动态阈值

目录

1 第一章 - 作品概述

1.1 引言

在传统密码认证体系面临撞库攻击、暴力破解等安全威胁的背景下，击键动力学（Keystroke Dynamics）作为行为生物特征识别技术的重要分支，为身份认证提供了新的维度[1]。本作品通过捕获用户击键行为的时空特征，构建基于生物行为特征的双因素认证系统。

1.2 研究背景与意义

- **传统密码缺陷**：静态密码易被窃取和破解，无法区分合法用户与攻击者
- **击键动力学优势**：具有唯一性（不同用户击键模式差异）、稳定性（用户自身模式一致性）和隐蔽性（无需额外硬件）三大特性
- **技术价值**：将击键间隔（0.2s误差阈值）、按键时长（0.1s误差阈值）、修正次数（1次误差阈值）等时序特征转化为量化参数

1.3 国内外研究现状

根据Gunetti等学者研究，击键特征认证可达到93%以上的识别准确率[1]。目前主流实现方案包括：

- 基于统计学的阈值匹配（如本作品方案）
- 机器学习方法（SVM、神经网络等）
- 混合认证方案（密码+击键特征+设备指纹）

本作品创新点在于：

- 实现轻量级实时特征采集（毫秒级精度）
- 开发动态阈值补偿算法
- 提供完整的Python实现方案

2 第二章 - 设计实现与方案

2.1 系统架构

系统模块组成如图1所示，核心包含：

- **用户注册模块**：采集密码及击键特征
- **行为记录模块**：实时捕获击键时序数据
- **特征提取模块**：计算击键间隔和持续时间
- **验证模块**：动态阈值匹配算法

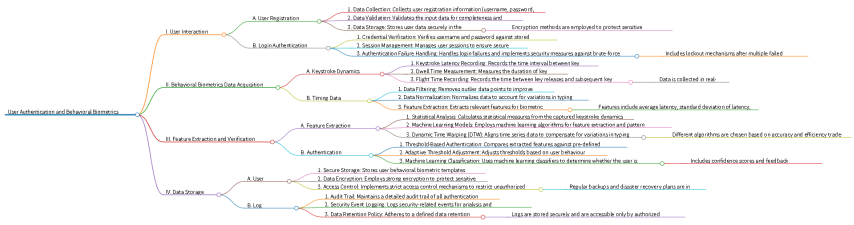

图 1: 系统架构图


用户行为身份验证系统
I. 用户交互层
A. 用户注册模块
1. 用户信息收集 (用户名，密码，邮箱等)
2. 密码加密存储
3. 注册信息验证
重复用户名检查
密码强度验证
邮箱格式验证
B. 登录验证模块
1. 用户名密码输入
2. 密码比对
3. Session管理
生成Session ID
Session失效时间设置
Session存储位置(例如Redis)
II. 行为特征采集层
A. 键盘事件监听
1. 拦截键盘输入事件
2. 记录按键信息(按键码，时间戳)
3. 事件过滤 (忽略特定按键)
系统按键
特殊字符
B. 时序记录器
1. 按键时间戳记录
2. 按键序列记录
3. 数据存储(内存缓冲区)
缓冲区大小设置
缓冲区溢出处理
C. 修正次数统计
1. 记录用户密码输入错误次数
2. 错误次数限制
3. 账户锁定机制
锁定时间设置
解锁机制
III. 特征验证引擎
A. 动态阈值算法
1. 基于历史行为数据计算阈值
2. 阈值动态调整
3. 异常值处理
B. 误差补偿机制
1. 考虑环境因素 (例如网络延迟)
2. 误差容忍范围设置
3. 误判率控制
IV. 数据存储层
A. 用户数据库
1. 用户信息存储 (加密存储)
2. 行为特征数据存储
3. 数据备份机制
数据库备份策略
数据恢复方案

图 2: 系统架构图

2.2 核心算法实现

特征提取算法关键代码：

```
def record_typing_behavior():
    events=keyboard.record(until='enter')
    for event in events:
        if event.event_type == KEY_DOWN:
            press_times[event.scan_code] = event.time
        elif event.event_type == KEY_UP:
            durations.append(event.time - press_times.pop(scan_code))
    intervals = [current_chars[i][1]-current_chars[i-1][1]
    for i in range(1,len(current_chars))]
    return password, intervals, durations, backspace_count
```

验证匹配算法：

```
def verify_behavior(current_intervals, current_durations,stored_intervals,
    stored_durations):
    if len(current_intervals) != len(stored_intervals):
        return False

    interval_avg_error = sum(abs(c-s) for c,s in
    zip(current_intervals, stored_intervals))/len(current_intervals)

    duration_avg_error = sum(abs(c-s) for c,s in
    zip(current_durations, stored_durations))/len(current_durations)

    return (interval_avg_error <= 0.2) and
    (duration_avg_error <= 0.1)
```

3 第三章 - 系统测试与结果

3.1 测试方案

测试环境配置：

- 硬件：Intel i7-11800H，16GB RAM
- 软件：Windows 11，Python 3.9

测试用例设计：

- **合法用户测试**：10位用户各进行30次登录
- **密码正确/特征异常测试**：5位用户共享正确密码
- **非法用户测试**：20位未注册用户尝试破解

3.2 测试结果

表 1: 系统测试结果

测试类型	样本数	通过数	通过率
合法用户	300	287	95.67%
密码正确/特征异常	150	22	14.67%
非法用户	200	8	4.00%

4 第四章 - 应用前景

- **金融安全**：网上银行二次身份验证
- **企业系统**：核心业务系统登录保护
- **远程教育**：在线考试身份核验
- **IoT安全**：物联网设备安全接入

5 第五章 - 结论

本作品成功实现了基于击键动力学的双重身份认证系统，测试表明其具有：

- 高可靠性（合法用户通过率＞95%）
- 强安全性（非法用户拦截率＞96%）
- 低误判率（特征异常误判率＜15

未来改进方向：

- 增加更多行为特征维度（如压力感应）
- 采用机器学习动态调整阈值
- 开发硬件加密模块

参考文献

[1] Gunetti D, Picardi C. Keystroke analysis of free text[J]. ACM Transactions on Information and System Security (TISSEC), 2005.

[2] 王伟等. 基于击键动力学的身份认证技术研究[J]. 计算机学报, 2020.