

**信息与软件工程学院**

**综合设计I中期报告**

课程名称： 综合设计I

课题名称： 基于音频的安全D2D通信系统

指导教师： 陈大江

所在系别： 互联网安全

执行学期： 3

学生信息：

|  |  |  |
| --- | --- | --- |
| 序号 | 学号 | 姓名 |
| 1（组长） | 2019091616012 | 冉江皓 |
| 2 | 2019091616018 | 邓亚辉 |
| 3 | 2019091616007 | 官宏 |
| 4 | 2019091603029 | 冯一凡 |
| 5 | 2019091603009 | 罗驰宇 |

目 录

第一章 综合设计的进展情况 1

1.1 需求分析与建模 1

1.2 复杂工程问题归纳 1

1.3 实施方案与可行性研究 1

第二章 存在问题与解决方案 2

2.1 存在的主要问题 2

2.2 解决方案 2

第三章 前期任务完成度与后续实施计划 3

参考文献 4

**说明:**

1. **报告要求2000字以上。**
2. **本模板仅为基本参考，请各位同学根据个人情况进行目录结构扩展。**
3. **报告正文必须双面打印。**

# 第一章 综合设计的进展情况

1.1 需求分析与建模

随着网络和通信技术的发展，移动终端身份认证的需求已经成为必然。基于音频的认证系统相比现有的认证手段，是一种解决了现存方法中的一部分弊端的手段。课题核心要求为：设计一套基于音频的安全D2D通信系统。

认证系统基于移动设备，得以具有良好的普适性，并要求系统本身具有足够的可靠性和稳定性。且因为移动设备几乎都具备麦克风和扬声器，且不同型号的设备之间存在差异，使得系统能够实现认证功能。

课题的主要内容为：

①设计基于音频硬件物理指纹设备认证应用软件

②设计基于音频硬件的安全信息传输的应用软件

③完成基于音频的安全D2D通信系统

其中，根据学期不同，本学期主要完成的目标为第一项。基于硬件和软件的需求，我们需要在移动设备上开发一款使用音频硬件物理指纹来实现设备认证的应用。

这款应用应当实现的功能，有：生成随机认证音频、学习认证音频并提取指纹、基于已有指纹验证新的认证音频。其中，生成随机认证音频部分考虑如何产生适当的音频，考虑功能性；学习认证音频需要考虑如何从收到的音频中排除干扰，提取出指纹，并且防止可能的攻击伪装；认证部分同样需要有相关考虑。

1.2 复杂工程问题归纳

在目前阶段的讨论与设计中，我们主要遇到了以下的工程问题：

对于现有算法的研究、认证精度和抗攻击能力的测试。

现有的算法可能难以满足该课题的需求。课题需要使用音频作为认证依据，而考虑到嘈杂环境中，需要对于噪声、环境音等干扰音源进行处理，且需要利用算法对于音频的特征进行分析，在满足相对较高的认证成功率时，降低被攻击的可能性，特别是可能出现的相同硬件型号攻击等成功可能性较高的攻击手段。如何改进音频生成、认证使用的相关频点，也是一大难点。

对于认证流程的实现和改进。

认证流程涉及到各种过程，其中包括用户直接接触的交互界面、认证的算法使用和速度优化、学习与认证的相互关联。其中的流程改进和用户引导，与交互体验直接相关。如果不能给用户提供一个清晰的逻辑，就会造成认证系统在使用上给用户出难题。认证算法应该包装为合适的函数，供软件内多种功能调用，同时进行相应的异常处理、风险控制和权限控制。学习与认证的转换也涉及到各种认证函数的复用，以及后期的加强学习等等，可以从每次认证中进行更多的学习，采集数据用于改进认证精度。

对于软件开发所需技能的实践。

在开发过程中，考虑到现在的手机主要操作系统为Android和iOS，若只开发适用于一侧的软件，将会使另一侧用户遇到无法使用的问题。并且，现在的跨平台开发框架正在蓬勃发展，并且日趋成熟，使用跨平台框架也可以构建出和原生应用体验相近的跨平台应用。我们考虑将使用跨平台框架开发作为首要目标，这其中需要使用一些公共库，以及部分特性可能需要平台相关的原生组件。如果在跨平台开发中遇到了难以解决的问题，则改为开发仅适用于Android平台的软件。原生平台已有众多开源库可供使用，可以保证开发顺利进行，且所需要的Java语言也是我们目前学习的内容，只需要再多了解与Android相关的部分。但使用跨平台开发，既涉及到Java/Kotlin，又涉及到iOS所使用的Objective-C/Swift，需要掌握多门语言，难度上升。

1.3 实施方案与可行性研究

对于算法的研究，我们通过对以往相关研究论文的学习，发现考虑使用“带状匹配算法”时可以有效匹配音频硬件物理指纹。这种匹配算法不仅具有一定的开创性，也较好地满足了本课题的认证需求。对于噪音的处理，通过对于环境噪音的采样分析，使用高通滤波滤除4kHz以下的频段即可。对于指纹的频点，可以考虑使用混频技术进行发送。抗攻击可以考虑使用足够多的设备进行充分的攻击测试，通过大量筛选，尝试确定匹配算法的最佳阈值。如果难以确定，则考虑从算法层面重新设计。

对于认证流程的实现，我们首先规划各个界面的草图，然后使用Material Design对UI进行绘制，再实现对应功能的活动。在不同的步骤操作时，会有一些相应的弹窗提示和toast提示，用于引导用户。对于各种函数，我们进行了对应的函数说明，在内部引用函数时可以得到详细的说明。

对于软件开发，我们计划兵分两路，一路主要推进跨平台开发，另外一路推进原生开发，作为备份。目前主要的跨平台框架是Flutter和React Native，两者皆能支持在iOS和Android上跨平台开发，以及利用平台私有特性和原生代码对对应平台实现不同功能。

# 第二章 存在问题与解决方案

2.1 存在的主要问题

目前存在的主要问题是：对于算法的设计完善和软件的开发过程中存在一些问题。

2.2 解决方案

算法方面，对于原有算法的优化点有点难以确定，并且我们目前比较缺乏通信的相关知识，所以对于算法的改进会比较难。同时，对于如何利用傅里叶变换等方法，我们仍然需要更多的学习。

软件开发中，目前Java基础学习已经告一段落，但是在Java转换到Android这方面给仍然有大量的知识和Android的平台特性需要去学习。关于跨平台框架，我们目前选择的是Flutter，而其使用的程序语言是Dart，需要对Dart进行全面的学习，以及是否有对应的数学库能用于音频认证。Dart中已有可用于处理音频信号的FFT库[1][2]，如果使用顺利，则应当可以在算法方面顺利实现开发目标。而对于软件界面和交互的开发已经基本完善，不过仍然存在诸多可以改进的地方，包括但不限于遵守新的开发规范、考虑实际使用情况等。

# 第三章 前期任务完成度与后续实施计划

前期任务包括设计草案、算法分析等，进展虽然总体偏慢但已基本推进完毕。后期需要加快语言学习以及功能开发，以期按时完成项目。

# 参考文献

1. fft 0.3.0, <https://pub.dev/packages/fft>
2. smart\_signal\_processing 2.1.1, https://pub.dev/packages/smart\_signal\_processing