开题报告

基于B/S模式的文件数字签名管理器的设计与实现

1. 毕业论文（设计）题目的来源

数字签名有助于提高网络生活中的安全性，预防潜在的恶意攻击，但数字签名的实现比较专业，普通用户缺乏在知识背景的情况下难以直接生成、管理签名，因此一个可靠的管理系统成为了较好的解决方案。本题目是指导教师根据教学以及科研的需要自行拟定，师生双方进行了沟通与交流，确定以本题目开展毕业论文（设计）工作。

1. 毕业论文（设计）选题的目的和意义

本题目拟实现一个易于操作、对用户友好的数字签名管理器。在完成以后，成品将以一个完备的网站的形式存在。网站的使用方法简单，页面对用户友好，鲁棒性强，可以直接部署在服务器上。同时项目会很注重安全性，保证程序能够可靠的达成安全地管理数字签名的既定目标。

1. 国内外关于该选题的研究现状和发展趋势

一般用户目前在传输文件的时候并不注重文件的完整性（integrity），例如在网站上下载文件的时候，虽然很多资源提供者同时提供了文件的散列码或数字签名等信息，但在实际使用的过程中很少有用户会在传输完成后主动验证，这就给数据的安全、用户设备的安全造成了一定的隐患。一个可靠的管理系统可以极大程度的帮助用户完成数字签名的验证程序，确保安全性。

1. 毕业论文（设计）的主要内容、研究方法、技术路线及可行性分析
   1. 概况

文件数字签名管理器是讲是一个B/S架构的Web应用程序，主要分为前端网页和后端业务逻辑两部分组成。我将采用SpringBoot作为主体开发框架完成设计以及实现。密码学相关部分会调用JavaCrypto库中的相关代码。

* 1. 密码学设计

整个项目将首先有一个一级密钥作为主密钥，用于加密所有二级密钥。主密钥在项目部署的时候又程序随机生成，暂定存储在内存中。每个用户将拥有一个二级密钥。二级密钥在用户注册的时候由后台随机生成，经一级密钥加密后存储在数据库中。每个用户还将拥有一组三级密钥，三级密钥由一对DSA签名密钥和一个加密密钥组成。三级密钥也是在用户创建的时候由后台随机生成，其中DSA签名私钥和加密密钥被二级密钥加密后存储在数据库中。DSA签名私钥用于给该用户上传的所有文件生成数字签名。加密密钥则用于给文件进行加密。生成的数字签名会被存储在数据库中，文件则存储在服务器的文件系统里。

* 1. 安全性设计
     1. 为了确保密钥安全，所有的密钥都将是随机生成的，并且一个密钥只有一个用途，不会重复使用。
     2. 用户注册的时候会自己定义一个密码用于以后的登录，该密码有强度上要求，暂定需要符合以下几个条件
        1. 由 6-18个字符组成，每个字符都是可打印的ASCII字符
        2. 同时含有大写字母、小写字母、数字和特殊符号
     3. 用户的用户名和密码等相关信息会存储到服务器中，但密码不是明文存数的。密码会在加盐（salt）后计算哈希（hash），数据库中只存储哈希过后的密码。这样即便整个数据库泄露，用户的密码依然会一定程度的保护，加盐会让依靠常用密码库的暴力破解更难以实现。在这里盐是随机生成的16字节随机数。具体实现上，我会采用PBKDF2作为这一步哈希的算法，该算法的有点是可以通过增加迭代次数，增加完成哈希所耗费的时间，进一步提高被暴力破解的难度。
     4. 用户的所有输入，包括用户名、密码等等信息都会首先在后端进行验证，确保拒绝所有恶意输入，避免XSS攻击等
     5. 在整体实现完成后，将考虑把网站从http升级到https，加密通讯。
     6. 网站会自定义错误界面，将所有5xx服务器内部错误隐藏，保证stack trace不会泄露
     7. 用户登录后的session会在30分钟后过期，过期后需要重新登录
     8. 会通过给用户发送email的方式核实用户拥有其邮箱这件事，后期可以通过该邮箱找回密码、修改密码。
     9. 用户未登录的时候只能访问index，login，register，error这四个界面，其他界面都需要登录后才能访问
  2. 技术路线

主体采用SpringBoot框架并使用Java语言进行开发。SpringBoot的生态系统中有很多组件（Component），各组件之间能够很好地互动，极大地提高了开发的效率。就典型的MVC结构来说：

* + 1. 所有响应请求的业务逻辑由Controller组件完成。
    2. Model用Entity组件实现，并且利用Repository 组件实现了与数据库的自动连接，所有增删改查操作都可以自动被hibernate框架实现，极大地提高了编程效率。在实现中我打算采用H2 database，这是SpringBoot框架的默认数据库，是一种in-memory database。
    3. View则由thymeleaf负责生成。Thymeleaf是一个可以用来生成动态页面的框架。

在本次项目中，我打算使用gradle管理项目的各项dependency。

在项目开发的过程中，我会使用Postman作为测试工具，向网站发送各式请求，验证程序的正确性。同时我会使用EditThisCookie这个Chrome浏览器插件用于检测Cookie生成的正确性。

1. 毕业论文（设计）的进度安排

我将尽可能的在未来一个月内完成主体功能的实现，并在实现的过程中不断解决涌现的问题，并进一步细化设计，查漏补缺。

项目会部署在一台服务器上，可以通过以下链接访问

http://35.225.173.19:8080/

在主体实现完成后，我将开始整理资料，撰写文档、图标，并最终提交论文。

1. 毕业论文（设计）的主要参考文献

[1]Springboot guides: https://spring.io/guides

[2]Java Cryptography Extension: http://tutorials.jenkov.com/java-cryptography/index.html

1. 其他要求

方案设计合理，代码逻辑清晰、可读性强，论文内容详实。