

软件工程实训文档

北京交通大学

2017年7月

动态令牌认证系统

概要设计说明书

赵洲浩小组

成员：赵洲浩 刘晓宇 蒋小雨 孙林 王士昂

|  |  |
| --- | --- |
| 项目 | 动态令牌认证系统 |
| 文档 | 概要设计说明书 |
| 版本 | V 1.0 |
| 作者 | 赵洲浩组全体成员 |
| 最后更新时间 | 2017-07-08 |

版本更新概要

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 时间 | 更新人 | 更新摘要 |
| V1.0 | 2017-07-07 | 刘晓宇 | 初稿 |
| V1.1 |  |  |  |
| V1.2 |  |  |  |

目录

[1．引言 2](#_Toc506972944)

[1.1编写目的 2](#_Toc506972945)

[1.2项目背景 2](#_Toc506972946)

[1.3定义 2](#_Toc506972947)

[1.4参考资料 2](#_Toc506972948)

[2．总体设计 3](#_Toc506972949)

[2.1需求规定 3](#_Toc506972950)

[2.2运行环境 3](#_Toc506972951)

[2.3基本设计概念和处理流程 3](#_Toc506972952)

[3．系统架构设计 3](#_Toc506972954)

[3.1层次设计 3](#_Toc506972955)

[3.2层次结构和模块设计 3](#_Toc506972956)

[3.3 3](#_Toc506972957)

[4．功能模块设计 3](#_Toc506972958)

[4.1功能概要 3](#_Toc506972959)

[4.2设计 3](#_Toc506972960)

[5．系统接口设计 3](#_Toc506972961)

[5.1外部接口设计 3](#_Toc506972962)

[5.2内部接口设计 3](#_Toc506972963)

[6．数据库设计 3](#_Toc506972965)

[6.1数据库总体设计 3](#_Toc506972966)

[6.2数据库逻辑设计 3](#_Toc506972967)

[7．出错处理设计 3](#_Toc506972969)

[7.1出错输出信息 3](#_Toc506972970)

[7.2出错处理对策 3](#_Toc506972971)

[8.安全保密设计 3](#_Toc506972972)

[9.维护设计 3](#_Toc506972973)

# 1．引言

## 1.1编写目的

概要设计说明书根据动态令牌认证系统的特征、性能要求、可靠性要求、成本等方面的内容，针对产品使用的技术平台和软硬件架构，从总体上明确了系统的架构设计，以及系统各个功能模块设计、系统接口设计、数据库设计。

本概要设计说明书的目的就是进一步细化软件设计阶段得出的软件原貌，把它加工成在程序细节上非常接近与源程序开发的软件表示。

概要设计说明书预期读者：软件测试员、程序开发员、软件分析员

## 1.2项目背景

项目的委托单位：北京交通大学软件学院

项目开发单位： 北京交通大学软件学院赵洲浩组

项目主管部门： 北京交通大学软件学院赵洲浩组

动态令牌认证系统是根据专门的算法生成一个不可预测的随机数字组合，每个密码只能使用一次，成为最安全的[身份认证技术](http://baike.baidu.com/item/%E8%BA%AB%E4%BB%BD%E8%AE%A4%E8%AF%81%E6%8A%80%E6%9C%AF" \t "_blank)之一。动态令牌认证使用便捷，且与平台无关性，随着移动互联网的发展，[动态令牌认证技术](http://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E5%8F%A3%E4%BB%A4%E6%8A%80%E6%9C%AF" \t "_blank)已成为[身份认证技术](http://baike.baidu.com/item/%E8%BA%AB%E4%BB%BD%E8%AE%A4%E8%AF%81%E6%8A%80%E6%9C%AF" \t "_blank)的主流，被广泛应用于企业、网游、金融等领域。动态令牌认证系统是一种包含动态令牌和动态令牌认证综合系统，可以为应用系统提供动态令牌认证服务。

动态令牌认证系统由认证系统和密钥管理系统组成。动态令牌认证的设计思路是在登陆过程中加入不确定因素，使每次登陆过程中传递的信息都不相同, 从而提高信息系统的安全性。

## 1.3定义

下列术语和定义适用于本概要设计说明书。

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **描述** |
| **1.1** | 令牌  Dynamic Code Token | 即动态令牌，是生成并显示动态口令的载体 |
| **1.2** | 动态口令  Dynamic Password | 由令牌种子与其他数据，通过特定算法，运算生成的一次性口令。 |
| **1.3** | 静态口令  Static Password | 用户设置的，除非用户主动修改，否则不会发生变化的密码。该口令通常存储在认证系统中，用于配合动态口令使用。该口令也可由认证双方临时协商确定。 |
| **1.4** | SM3 | 国家密码管理局批准的杂凑算法。 |
| **1.5** | SM5 |  |
| **1.6** | UTC时间  Universal Time Coordinated | 协调世界时(Universal Time Coordinated)英文缩写，是由国际无线电咨询委员会规定和推荐,并由国际时间局(BIH)负责保持的以秒为基础的时间标度，是距1970年1月1日00:00时(格林尼治标准时间)的秒数。 |
| **1.7** | 令牌种子  Token Seed | 即令牌密钥，用于与其他数据组装，通过特定算法运算获得动态口令，同时存储于令牌和认证系统中. |
| **1.8** | 认证系统  Authentication System | 能够为应用系统提供动态口令身份认证服务的系统。 |
| **1.9** | 密钥管理  Key Management | 对种子密钥的生成、传输和存储的安全管理，种子密钥是否安全直接影响到整个认证系统是否安全。 |
| **2.0** | 密钥  Key | 一种参数，它是在明文转换为密文或将密文转换为明文的算法中输入的数据。 |
| **2.1** | 接口  Interface | 两个不同系统(或子程序)交接并通过它彼此作用的部分。 |

## 

## 1.4参考资料

动态令牌认证系统概要设计说明书参考《软件配置管理计划说明书》 、 《软件质量保证计划指导书》 、 《北京交通大学开发流程和标准》 、 《北京交通大学编码规范》 、 《动态令牌认证系统需求规格说明书》

# 2．总体设计

## 2.1需求规定

1. 基于现在静态密码认证技术的的不安全性，例如：容易被他人所盗取密码、密码根据用户的水平不同而难易程度不一。
2. 基于现在最发达的生物特征识别技术（指纹、虹膜等的识别技术）的局限性，例如：指纹被破坏等不可抗力。
3. 动态令牌认证技术采用一次一密（One Time Password）的技术，使得动态令牌认证系统成为现在世界上最安全的身份认证方式之一。
4. 由于本系统将会应用在个人隐私或者一些机密领域，因此可靠性要强，安全性要高，要有定期备份数据的功能，还要在数据的通信过程中进行严格的加密措施。

## 2.2运行环境

操作系统： WindowsXP以上

浏览器程序：浏览器IE 8.0 及以上、Firefox、Chrome

## 2.3基本概念设计和处理流程

# （需要补充！）

# 3．系统架构设计

## 3.1层次设计

动态令牌认证系统涉及到用户、服务器、浏览器验证，以及涉及到服务器与客户端之间的通信，因此在系统的总体设计上采取多层的架构设计。

客户端分为四层：View层、Logical层、Transport层、Storage层

Web端分为两层：View层、Transport层

服务器分为三层：Logical层、Transport层、Storage层

## 3.2层次结构和模块设计

|  |  |  |
| --- | --- | --- |
| 客  户  端 | View层 | 登录/注册、令牌生成/注销、动态口令显示、返回信息 |
| Logical层 | 用户信息判断、动态口令生成、用户功能/动态令牌/注销 |
| Transport层 | Socket连接、信息处理、加密 |
| Storage层 | 日志管理、种子存储 |
| Web端 | View层 | 令牌/动态验证码、返回信息 |
| Transport层 | HTTP连接、加密 |
| 服  务  器 | Logical层 | 动态口令生成/加密、Client处理、Web处理 |
| Transport层 | Socket连接、加密、线程池 |
| Storage层 | 日志管理 |

## 3.3功能分配

【表明各项功能与程序结构的关系。】

4．功能模块设计

## 4.1

## 4.2

【模块、层次之间的接口。】

# 5．系统接口设计

## 5.1

## 5.2

## 5.3

# 6．数据库设计

## 6.1数据库总体设计

数据库的数据主要来源于客户端的用户以及在认证过程中产生的数据。数据库要接收来自于服务器的访问，为了提高系统的在连接方面的健壮性，因此在数据库端设置线程池。

## 6.2数据库逻辑设计

## 7．出错处理设计

## 7.1出错输出信息

客户端：

|  |  |  |
| --- | --- | --- |
| 服务器返回信息 | 1001: | 登录成功 |
| 1002: | 用户不存在 |
| 1003: | 密码错误 |
| 1004: | 用户名重复 |
| 1005: | 注册成功 |
| 1006: | 令牌生成失败 |
| 1007: | 令牌生成成功 |
| 1008: | 未知错误 |
| 服务器接受信息 | 2001: | 请求登录 |
| 2002: | 请求注册 |
| 2003: | 请求令牌 |
| 2004: | 未知请求 |

## 

## Web端：

|  |  |  |
| --- | --- | --- |
| 服务器返回信息 | 3001: | 验证成功 |
| 3002: | 验证失败 |
| 3003: | 未知错误 |
| 服务器接受信息 | 4001: | 验证令牌、口令 |
| 4002: | 未知请求 |

## 7.2出错处理对策

（需要补充！）

# 安全保密设计

（需要补充！）

# 维护设计

（需要补充！）

1）在程序内部通过建立模块，不同窗体调用一个模块的形式进行设计，减少了代码的操作量，提高了运行的速度。

【说明为方便维护工作的设施，如维护模块等。】