**密钥管理系统**

**软件设计说明书**

航天二院爱威公司

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修订人** | **修订日期** | **修订内容** |
| 0.1 | 汤敬浩 | 2016/8/30 | Debug版本 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 文档概述

## 文档约定

文档的描述内容是密钥管理系统的软件设计说明书。

文档采用基于UML建模语言的面向对象建模方式对功能需求进行描述。

系统的结构设计由动态结构和静态结构组成，且静态结构通过动态结构汇总得到。

系统的动态结构组成：

1. 每个基本用例及其子用例，对应有一张系统顺序图（在需求规格说明说中已展现）。
2. 用例实现的设计说明系统顺序图中的每个系统事件，进入软件系统后由哪个软件对象接收，并由哪些其他的软件对象协同工作，按照需求分析规格说明的要求返回规定的结果给使用者。
3. 每个系统事件的交互图使用UML的顺序图表示某一时刻软件对象的交互示例。
4. 系统顺序图中有多少个系统事件，该用例实现的交互图就有多少个。
5. 每个系统事件有明确的方法和参数定义。

系统的静态结构组成：

系统的静态结构主要由软件类及其之间的关系表示，及类图；软件类之间的关系主要使用定向关联（表示一个软件类的方法调用另一个软件类的方法），在明确两个类之间由相互调用的情况下使用双向关联。

系统结构过于复杂的情况下，通过包图的方式描述系统的宏观静态结构；包图主要使用依赖关系表示包和包之间的调用关系。

文档中所有的UML图形均以visio 2016版本的建模模具进行绘制。

## 预期的读者和阅读建议

1. 预期的读者和阅读建议

|  |  |  |
| --- | --- | --- |
| 编号 | 预期读者 | 阅读建议 |
| 1 | 客户 | 确认文档中给出的功能需求描述 |
| 2 | 开发方 | 熟悉并掌握项目的各项功能需求 |

## 产品的范围

本文档涉及的项目范围是视频安防监控系统中的密钥管理系统，密钥管理系统包括TCP服务模块、事务处理模块、数据库模块、加密硬件模块。

# 项目背景描述

密码体制是信息安全的基础，密钥管理则负责密码体制中密钥的管理，包括密钥生成、密钥分发、密钥更新、密钥存储、密钥备份、密钥销毁等功能。借助密钥管理服务，用户无需花费大量成本来保护密钥的保密性、完整性和可用性，可以安全、便捷的使用密钥，专注于开发加解密功能场景。

本系统属于视频安防监控系统的子系统，每路视频在传输或保存时都需要使用对称密钥加密。传统的对称密钥管理方法是基于共同保存对称密钥实现的，要保证彼此密钥的交换是安全可靠的，同时还要设定防止密钥泄密和更改密钥的程序。这样，对称密钥的管理和分发工作将变成一件潜在危险的和繁琐的过程。使用公钥加密的数字信封技术是一种解决方案。例如在视频安防监控系统中，摄像头生成一次性对称密钥，用于加密视频流，并用公开密钥对该会话密钥进行加密，加密后与视频流封装在一起，转发给相应的实体，这种模式也有一定隐患，尽管密钥已经被加密，但由于与视频流封装在一起，在非安全的网络中传输或者在非安全的设备中存储（为了方便视频记录与回放，视频信息需要存储很久），仍然有一定风险。因此在这种应用场景下，需要更加安全的密钥管理方法以及系统。

## 运行环境

1. 运行环境

|  |  |  |
| --- | --- | --- |
| 编号 | 名称 | 运行环境 |
| 1 | 应用服务器 | Linux （kernel≥2.6） |
| 2 | 编译器版本 | gcc 6.1, cmake 3.6.1 |
| 3 | 数据库 | Mysql 5.7.13 |
| 4 | 依赖库 | Boost 1.61, Libevent 2.0.22 |
| 5 | 硬件设备 | SJK1238密码卡（调试暂用） |

## 设计和实现上的限制

1. 设计和实现上的限制

|  |  |  |
| --- | --- | --- |
| 限制因素 | 限制说明 | 备注 |
| 必须采用的技术、工具、编程语言、数据库等 | C++11标准、Cmake工具、数据库采用Mysql数据库其他无特殊限制 |  |
| 不能使用的技术、工具、编程语言、数据库等 | 无特殊限制 |  |
| 企业策略、政策法规、业界标准 | 必须遵守中华人民共和国的相关法律法规 |  |
| 硬件限制 | 无特殊限制 |  |

## 假定和依赖

1. 假设因素

|  |  |  |
| --- | --- | --- |
| 编号 | 假设 | 备注 |
| 1 | 视频安防监控系统具有CA认证中心、其中每个实体均具有数字证书。 |  |
| 2 | 对现有事务、功能描述与实际情况基本相符合，对需求变更不影响框架大调整。 |  |

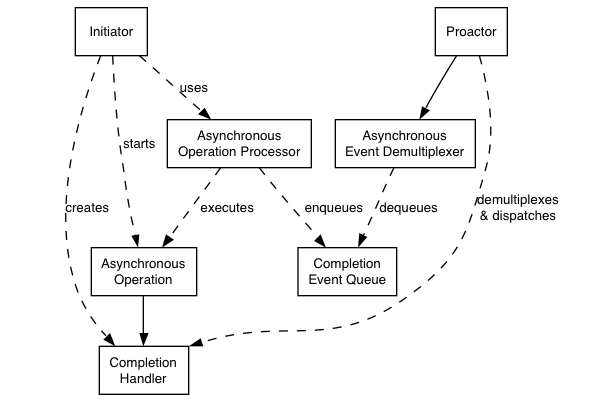
1. 依赖因素

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 依赖 | 依赖说明 | 备注 |
| 1 | Mysql 5.7.13 | 事务数据均存储在Mysql中 |  |
| 2 | Boost 1.61 | 调试日志使用了Boost log库  使用Boost asio异步IO网络库  使用Boost thread线程库 |  |

# 系统的技术架构说明

## 服务器并发模型

服务器并发模型采用Boost.Asio库+线程池来完成，Boost.Asio库采用了Proactor模式，其原理图如下：



1. Proactor原理图

具体技术细节详见[Proactor](http://www.boost.org/doc/libs/1_61_0/doc/html/boost_asio/overview/core/async.html)模式，简而言之，Proactor模式基于异步IO，Proactor框架中用户定义的操作是在实际操作之后调用的。比如你定义了一个处理函数要处理从SOCKET中读入的数据，那么当读操作完成以后，你的异步函数会被异步触发。

## 数据库和安全硬件接口

为了增强程序的扩展性，例如可以方便的扩展为支持多种数据库、安全硬件设备，本系统使用了“工厂模式”，定义了数据库和安全硬件的统一接口，工厂模式的实现方式详见类图。

## 数据库

数据库采用Mysql 5.7.13，定义了两个数据表

* tb\_key 存放被主密钥加密后的对称密钥，其字段如下：

1. tb\_key

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 长度(byte) | 描述 |
| key\_id | binary | 16 | 密钥id |
| key\_value | binary | 16 | 密钥（被主密钥加密）值 |
| generated\_time | int |  | unix time格式 |

* tb\_master 存放加密后的主密钥，其字段如下：

1. tb\_master

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 长度(byte) | 描述 |
| master\_key\_value | binary | 16 | 主密钥（被“主密钥加密密钥”加密）值 |

## 安全硬件设备

安全硬件负责生成主密钥，生成对称密钥，加密/解密的计算，“主密钥加密密钥”的存储等操作。

本系统使用SJK1238密码卡作为安全硬件设备。