**加密机子系统类及策略设计**

• 加密机连接类HsmSocket（IoConnector）

1.1主要用途：

• 管理socket连接，实现socket监控并自动重连。

• 实现加密机长连接的心跳策略，超时未收到心跳回应则断开长连接。

• 实现加密机连接的状态变更策略。

• 定义加密机连接的状态，根据每次的请求的耗时统计，根据策略动态更新连接的状态。（有没有更好的方法）

• 加入消息队列

• 由时间来定义繁忙的状态？

1.2类设计：

Connect();//实现加密机的连接功能，包括添加心跳策略和断开自动重连

disconnect();//断开长连接

SendAndRecv (String buffer);//请求处理，发送加密机并收回请求

calculate(long cost);//实现状态变化策略

getConnStatus(); //获取连接状态

1.3加密机长连接状态变化策略：

• 连不上加密机状态设为ERROR；

• 请求处理时间连续5次超过阈值，则设状态为BUSY；（这5次请求照常处理）

• 请求处理时间连续5次未超过阈值，则设状态为FREE；

• BUSY状态下不再接受业务请求，但可以发送心跳请求。

• 由策略来分配请求（轮询，权值）

• 加密机连接管理类HsmSockMngr

• 主要用途：

• 连接管理（调度策略）

• 连接状态

• 对上层统一接口

• 加密机类Hsm

• 主要用途：

• 管理加密机的所有长连接；

• 加密机实例

• 定义加密机的繁忙状态；

• 每个加密机与一个appname绑定，参与这个加密机的加解密服务；

• 实现加密机状态的变更策略。

• 提供对上层接口

• 类设计：

start();//启动加密机的所有长连接

SendAndRecv (String buffer);//发送外部请求

getHsmStatus();//根据策略实时计算加密机的状态

String appName; //应用名称，用于加密机分组

HsmSockets;//连接管理

2.3加密机状态变更策略：

• 加密机过半连接状态为FREE时，加密机状态为FREE；

• 加密机过半连接状态为BUSY时，加密机状态为BUSY；

• 加密机有一个连接状态为ERR时，加密机状态为DOWN.

• 加密机池类HsmPool

3.1主要功能：

• 加密机管理；

• 选择低负载的加密机进行加解密运算；

• 对外提供统一接口

3.2类设计：

start();//启动所有加密机的连接

getHsmPoolSize();//获取加密机数量

getHSM(String name);//通过应用名称来获取加密机，如果无匹配的加密机，或者匹配的加密机繁忙，则可以选用其他组的闲置加密机

getHSM();//选择闲置的加密机

sendAndRecv(String buffer, String appname);//选择闲置的加密机进行运算

HSMs；

• 公共接口ICryptService（HsmWorkMngr）（异步 – 关于响应时间统计）

• 主要功能：

• 提供对外的加解密运算接口，由HsmPool、HsmClient实现接口

• 提供接口管理类，对不同应用划分

• 接口设计：

class CryptService {

GenCVC()

VerifyCVC()

VerifyTC()

VerifyARQC()

GenARPC()

GenScript()

BIN\_keys[]

LUKs[]

hsm\_pool

}

• 心跳处理类（过滤器）（实际业务指令作为请求）

6.1主要功能：

还需要添加断线重连功能

HsmKeepAliveFilterFactory定义心跳的发送和接收报文，并定义接受报文后的处理方式。

HsmKeepAliveRequestTimeoutHandler定义心跳报文超时处理方法。

• 统计监控类（monitor）

响应时间， overload监测

• 调度策略

• 轮询

寻找第一个符合条件的实例

• 权值（rsa运行时间长）

选取权值最低的符合条件的实例

• 通信协议

• 新报文格式

• 自定报文格式

• 数据库访问（内存存取）