**镜像签名认证软件研制技术方案**

承研单位 西安电子科技大学

负 责 人 李龙海

联系电话 18992864681

时 间 2025 年 4 月

**目 录**

[1. 范围 1](#_Toc195988446)

[1.1 项目概述 1](#_Toc195988447)

[1.2 需求分析 3](#_Toc195988448)

[1.2.1 功能需求 4](#_Toc195988449)

[1.2.2 非功能需求 8](#_Toc195988450)

[2 总体设计 8](#_Toc195988451)

[2.1 运行环境（没写） 8](#_Toc195988452)

[2.1.1 软件环境 8](#_Toc195988453)

[2.1.2 硬件环境 9](#_Toc195988454)

[2.2 架构设计 9](#_Toc195988455)

[2.3 外部接口 9](#_Toc195988456)

[2.4 软件部件 11](#_Toc195988457)

[2.5 技术方案 11](#_Toc195988458)

[2.5.1 通用模块技术方案 11](#_Toc195988459)

[2.6 内部接口设计 16](#_Toc195988460)

[2.6.1 通用模块接口 16](#_Toc195988461)

[3 测试与验证策略 18](#_Toc195988462)

[3.1 单元测试策略 18](#_Toc195988463)

[3.2 集成测试策略 18](#_Toc195988464)

[3.3 系统测试策略 19](#_Toc195988465)

[3.3.1 功能测试方法 19](#_Toc195988466)

[3.3.2 性能测试方法 19](#_Toc195988467)

[4 项目验收与交付 19](#_Toc195988468)

# 范围

## 项目概述

**（1）项目背景**

在容器化和虚拟化广泛应用的现代计算环境中，软件镜像成为应用部署和分发的核心载体。镜像的可信性和完整性直接关系到系统运行的安全性和稳定性。随着软件供应链攻击事件频发，保障镜像内容来源可验证、签名可信、内容未被篡改，已成为构建安全、可靠基础设施的重要技术需求。

镜像签名认证机制通过对镜像生成数字签名，并在分发与拉取过程中进行验证，可有效防止恶意篡改、伪造镜像等攻击行为。同时，结合软件更新元数据管理机制，可以实现对镜像版本的可追溯、可回滚管理，提升镜像生命周期的可控性与安全性。

The Update Framework（TUF）作为一种安全的软件更新框架，提出了角色划分、元数据签名、密钥轮换等机制，能够有效抵御元数据伪造、密钥泄露等复杂攻击。基于 TUF 的签名认证机制，不仅具有强安全性保障，还具备良好的可扩展性和跨平台适应能力，适用于多种镜像分发场景。

然而，当前实际应用中仍存在若干亟待解决的问题：

a. 缺乏一套独立、可控、适用于本地化部署或私有环境的镜像签名认证系统，难以满足对数据安全性、系统可控性要求较高的业务场景；

b. 现有方案对签名算法和封装格式的支持有限，难以灵活切换或扩展以满足特定算法或标准需求，且密钥轮换策略难以配置，不能满足高安全性场景的密钥生命周期管理需求；

c. 缺少灵活的信任库管理机制与签名元数据服务组件，难以满足大规模多用户环境下的权限控制、签名策略隔离等需求。

因此，研发一套基于 TUF 模型的通用镜像签名认证软件，具有重要意义。该软件将具备客户端与服务端功能模块，支持镜像签名与验证、元数据生成与管理、密钥管理与轮换、安全通信等核心能力，并支持多种签名算法和签名封装格式，适配多类型镜像对象。该系统能够为多样化部署环境下的镜像可信分发提供统一、可控、安全的技术支撑，提升镜像分发体系的整体可信性与安全性。

**（2）使用场景、功能简要介绍**

针对上述问题，本项目旨在针对嵌入式容器云平台研制一套镜像签名认证软件系统，面向容器镜像及虚拟机镜像等镜像分发对象，提供一整套符合 TUF模型设计思想的签名、验证、信任管理能力，适用于私有仓库、企业内部部署环境、安全隔离场景下的镜像可信分发体系构建。

本项目研制的镜像签名认证软件包含客户端与服务端两部分，它们的应用场景和主要功能如图1所示。



图1 应用场景和功能示意图

镜像签名认证软件主要用于容器镜像生命周期中的内容信任控制与签名认证管理，旨在提升容器镜像的安全性与可追溯性。该系统基于TUF（The Update Framework）构建，提供从镜像签名生成、信任验证到密钥管理的全流程能力，支持ECDSA、RSA-PSS等加密算法与JWS签名封装格式。

镜像签名认证软件包含客户端与服务器两部分，客户端工具支持开发者本地初始化信任库、管理密钥、生成镜像签名并将元数据发布至服务器端，客户端工具能够与Docker进行集成并兼容天脉操作系统；服务器端负责集中管理TUF元数据、执行签名校验、冲突检测与密钥轮换等任务，确保签名策略的统一性与数据的一致性。该软件还提供可配置的信任策略开关，可强制容器云平台在推送与拉取镜像时执行签名验证流程，保障仅已签名、经认证的镜像被使用。

对于**容器应用开发人员**，可使用客户端工具将构建完成的容器镜像进行签名，生成对应的信任元数据并上传至签名认证服务器端；对于**部署与运维人员**，系统提供了镜像拉取过程中的内容信任验证功能。

当开启镜像信任策略时，容器云平台客户端仅允许拉取通过签名认证服务验证的可信镜像，确保部署到生产环境中的每一个镜像都是经由开发人员签名、具备可溯源性的安全版本。

## 需求分析

镜像签名认证软件可在本地或远程环境中初始化镜像信任仓库，生成并维护基于 TUF（The Update Framework）标准的元数据，支持自定义元数据存储路径及安全密钥加密存储机制。软件支持自动生成、管理根密钥（Root Key）、目标密钥（Target Key）、时间戳密钥及快照密钥，满足从镜像构建、签名、上传到拉取、验证等各阶段的安全性要求。

镜像签名认证软件需要嵌入现有的容器镜像构建、发布与部署的系统架构中，实现对镜像操作行为的细粒度信任策略控制。支持设定镜像信任验证开关，允许用户强制启用内容信任策略，从而使容器云平台客户端仅可操作通过签名认证的软件镜像。

在复杂的镜像签名与信任策略场景中，软件需要支持灵活的密钥轮换策略及委托角色管理机制。支持基于时间周期、密钥状态或策略变更的自动密钥轮换，满足长期运行环境中的信任维护需求；同时支持委托角色的添加、删除与授权控制，便于组织内部按职责划分签名权限，实现分级安全防护。

镜像签名认证软件需要提供完整的服务端功能模块，包括信任库管理、元数据存储与验证、签名发布与吊销等能力。支持客户端与服务端间的安全通信、算法多样性扩展（支持 ECDSA 与 RSA-PSS）、JWS 格式签名封装、关键数据备份恢复与日志记录等辅助管理功能。

### 功能需求

镜像签名认证软件分为客户端软件与服务器软件。

客户端软件的主要功能包括：（1）信任库与目标文件管理功能，支持本地信任库的初始化与配置，生成符合 TUF标准的元数据文件，并对目标镜像文件进行管理，保障本地签名体系的完整性；（2）密钥与角色管理功能，提供本地密钥的生成、加密存储、列举与轮换功能，支持根密钥与目标密钥的创建与替换以及密钥的管理功能；（3）容器镜像签名与管理功能，提供对容器镜像进行签名、验证和撤销的功能，支持设置镜像信任验证开关，提供镜像元数据签名的发布、信任信息的查询、签名链的验证显示，确保镜像在传输和部署过程中的可信性。（4）辅助管理功能，支持配置签名认证服务器地址、本地密钥安全存储路径，提供客户端帮助信息的查看功能，提升系统的易用性和可维护性。

服务器软件的主要功能包括：（1）信任库与目标文件管理功能，支持服务端信任库的初始化与配置，保证服务端签名状态与客户端保持一致；（2）TUF元数据管理功能，提供元数据的存储、签名、更新以及客户端提交元数据的格式与签名验证，确保签名数据一致性与可信度；（3）密钥与角色管理功能，支持服务端密钥（如时间戳密钥、快照密钥）的生成、列举、签名与吊销操作，并按角色分层实施密钥管理策略，保障签名链的有效性和安全性；（4）辅助管理功能：支持通过配置文件定义服务器行为、安全存储设定及服务端帮助信息查看，增强服务端运行的灵活性和可管理性。

软件还包含一些客户端与服务端通用功能，主要包括：（1）安全通信功能：支持客户端与服务端之间的安全通信通道建立，保障签名与验证过程中的数据传输安全性；（2）加密与签名算法支持功能：支持多种签名算法（如ECDSA、RSA-PSS），具备算法可扩展性，以适应不同安全合规需求。（3）签名封装功能：支持使用 JWS（JSON Web Signature）格式封装签名内容，便于签名的标准化和跨平台验证。（4）数据管理与日志功能。

镜像签名认证软件客户端功能要求如下：

（1）信任（仓）库与目标文件管理相关功能：

1. 提供镜像签名认证软件信任（仓）库初始化功能；

——支持初始化本地信任库，并生成TUF（The Update Framework）相关元数据文件的功能；

——支持自定义配置TUF相关元数据文件存储路径功能；

——支持自动生成根密钥（Root Key）、目标密钥（Target Key）；

——支持密钥加密存储的功能；其中私钥需要加密存储，其密码可通过环境变量或交互方式输入。

1. 提供镜像签名认证软件信任库内容列举功能；
2. 提供镜像签名认证软件信任库对目标文件的增、删、更、查功能；
3. 提供快照元数据生成并签名功能。

（2）角色与密钥管理相关功能：

1. 提供本地存储所有密钥列举功能；
2. 提供新公私钥对生成功能；

——提供根密钥（Root Key）对，私钥需加密存储，公钥发布；

——提供目标密钥（Targets Key）对，私钥需加密存储，公钥发布。

1. 提供密钥轮换功能；

——支持策略变更：当信任策略更新（如新增/移除签名角色）时自动关联密钥轮换；

——提供密钥临期、逾期使用时告警，启动轮换流程；

——定期自动轮换：支持按时间周期自动触发密钥轮换；

——支持时间可配置轮换策略，按角色（Root、Targets、Snapshot）分层轮换；

1. 提供委托角色（密钥）添加、回收、列举功能。

（3）容器镜像签名与管理相关功能：

1. 提供为镜像创建签名功能；
2. 提供镜像信任验证开关设定功能；

——支持开启镜像信任开关时，强制启用镜像签名认证软件内容信任策略，Docker客户端仅允许推送和拉取其已签名的镜像，且必须通过镜像签名认证软件服务验证签名的有效性。

——支持关闭镜像信任开关时，禁用内容信任，Docker 客户端允许操作未签名镜像，绕过镜像签名认证软件的签名验证流程。

1. 提供镜像元数据签名发布功能；
2. 提供查看镜像信任信息功能；

——支持查看目标文件签名元数据的详细信息，包括签名者身份、使用的密钥、签名时间；

——支持显示签名者公钥及其对应的密钥ID；

——支持验证信任链完整性功能；

——支持密钥层级的显示功能。

1. 提供撤销指定镜像信任功能；

——支持元数据修改和版本更新；

——支持向服务器端同步信任状态。

1. 提供密钥吊销检查的功能，发起吊销请求、生成签名元数据更新；
2. 提供推送镜像时自动签名功能，支持信任链构建，完成签名生成与上传；
3. 提供对拉取的镜像信任验证功能，支持信任链和签名的验证。

（4）辅助管理功能：

1. 提供镜像签名认证软件服务器设定功能；
2. 提供本地密钥文件安全存储信息设定功能；
3. 提供客户端使用帮助信息查看功能。

镜像签名认证软件服务器软件功能需求如下：

（1）信任（仓）库与目标文件管理相关功能

1. 提供镜像签名认证软件信任（仓）库初始化功能；
2. 提供镜像签名认证软件信任库内容列举功能；
3. 提供镜像签名认证软件信任库对目标文件的增、删、更、查功能；
4. 提供服务端快照元数据生成并签名功能。

（2）TUF元数据管理相关功能

1. 提供元数据存储与更新的功能；
2. 提供目标文件元数据验证与冲突检测功能，支持对客户端提交的元数据进行格式、签名及版本的验证。

（3）角色与密钥管理相关功能

1. 提供本地存储所有密钥列举功能；
2. 提供新公私钥对（时间戳密钥、快照密钥）生成功能；
3. 提供时间戳和快照元数据签名的功能；
4. 提供密钥吊销检查的功能；
5. 提供密钥轮换功能；

——支持策略变更：当信任策略更新（如新增/移除签名角色）时自动关联密钥轮换；

——提供密钥临期、逾期使用时告警，启动轮换流程；

——定期自动轮换：支持按时间周期自动触发密钥轮换；

——支持可配置轮换策略，按角色（时间戳密钥、快照密钥）分层轮换。

（3）辅助管理功能

1. 提供镜像签名认证软件服务器设定功能：支持通过配置文件定义服务端行为；
2. 提供本地密钥文件安全存储信息设定功能；
3. 提供服务端使用帮助信息查看功能。

镜像签名认证软件客户端与服务端通用的功能：

1. 提供服务端与客户端交互安全通信通道功能；
2. 提供多种算法支持和扩展性的功能；

——支持加密算法兼容性：支持ECDSA、RSA-PSS算法；

——支持签名算法可扩展性，用户可替换使用私有的签名算法的功能。

1. 提供JWS格式的签名封装；
2. 提供关键数据的备份恢复功能；
3. 提供日志记录的功能。

### 非功能需求

#### 性能要求

1. 镜像签名认证软件服务器可支持不少于1000个的软仓库公证维护；
2. 镜像签名认证软件服务器至少可支持100个镜像签名认证软件客户端的并发访问；
3. 时间戳元数据文件生成及签名延时0.5秒；
4. 快照元数据文件生成及签名延时0.5秒；

#### 软件要求

1. 兼容天脉操作系统
2. 镜像签名认证客户端能够与docker完成集成验证
3. 采用C/C++语言开发
4. 编码标准遵循GJB 8114-2013《C/C++语言编程安全子集》。

# 总体设计

## 运行环境（没写）

### 软件环境

镜像签名认证软件系统的软件环境如表1所示。

表1 镜像签名认证软件系统软件环境

|  |  |
| --- | --- |
| 集群环境 |  |
| 目标机环境 |  |
| 宿主机环境 |  |
| 开发语言 | C++ |
| 开发工具 | GCC、CMake |
| 调试工具 | GDB |

### 硬件环境

## 架构设计

## 外部接口

镜像签名认证软件系统与外部的接口如图所示，主要包括与用户之间的人机交互接口以及集成Docker等容器引擎所提供的镜像安全相关接口。



图2 虚拟机编排调度系统外部接口

（1）与用户之间的人机交互接口

镜像签名认证软件提供命令行工具作为用户与系统交互的主要手段，用于对本地信任库、密钥、签名以及目标镜像进行管理。主要交互命令包括：

1. 信任库初始化与管理相关命令接口

* 信任库初始化，用于初始化信任仓库，并生成TUF相关元数据文件。
* 信任库内容管理，提供列举信任库内容以及对信任库目标文件的增、删、更、查等。
* 元数据生成与签名，提供对生成的相关元数据进行签名的功能。

1. 角色与密钥管理相关命令接口

* 密钥生成与存储，提供生成指定密钥对（如根密钥）的功能，并将私钥加密存储，公钥发布
* 密钥轮换，提供密钥的轮换功能，支持信任策略更新自动轮换、定期自动轮换、时间可配置轮换等多种轮换策略。
* 委托角色管理，提供对委托角色（密钥）的添加、回收、列举等功能。

1. 辅助管理相关命令接口

* 镜像签名认证软件服务器设定功能，支持通过配置文件定义服务端行为。
* 本地密钥文件安全存储信息设定功能。
* 客户端与服务端帮助信息查看功能。

（2）与应用容器引擎之间的接口

镜像签名认证软件系统与 Docker 等应用容器引擎之间通过标准的内容信任接口进行交互。其交互机制如下：

1. 容器镜像签名与管理接口

* 镜像签名创建接口，提供当用户推送镜像时，为镜像创建签名的功能。
* 镜像元数据发布接口，提供推送镜像时，将镜像相关的元数据进行发布的功能。
* 镜像信任信息查看功能，支持查看镜像信任信息，包括相关元数据的详细信息、签名者公钥及其对应密钥ID显示、信任链完整性验证以及密钥层级显示等功能。
* 镜像信任撤销接口，支持撤销指定镜像的信任功能，并支持元数据的修改与版本更新，以及同步服务器端的信任状态。
* 镜像信任验证接口，提供执行镜像拉取时，对拉取的镜像进行信任验证的功能，支持信任链和签名的验证。

1. 环境变量接口

* 镜像信任验证开关接口，提供对镜像信任机制进行开关的功能，开启时强制启用镜像签名认证软件内容信任策略，Docker客户端仅允许推送和拉取其已签名的镜像，且必须通过镜像签名认证软件服务验证签名的有效性；关闭时禁用内容信任，Docker 客户端允许操作未签名镜像，绕过镜像签名认证软件的签名验证流程。
* 内容信任验证所使用的签名认证服务器地址配置接口；
* 配置密钥操作时的密码配置接口。

## 软件部件

## 技术方案

### 通用模块技术方案

通用模块作为签名上传系统的基础支撑模块，负责系统的服务入口、加密算法实现、运行配置管理及日志审计与备份等核心通用功能，贯穿系统的各个子系统与服务模块。该模块以模块化、解耦化为设计原则，划分为安全信道通信子模块、签名认证算法子模块、配置管理子模块、日志记录子模块和数据备份子模块五个部分，分别提供外部通信接口、加密服务支持、全局配置管理以及系统运行和容灾保障。

通用模块的整体目标是为系统提供稳定、高可用、可审计、易维护的运行基础环境，支撑签名上传、验证、元数据处理、密钥管理等核心业务流程的顺利执行。



**图 3 通用模块功能划分**

#### 安全信道通信子模块

安全信道通信模块作为镜像签名认证软件客户端与服务器端的桥梁，主要用于客户端与服务器的安全通信以及元数据的安全传输，负责建立客户端与服务器端的安全通信通道，支持客户端与服务器端之间的认证与加密传输，并为系统提供统一的错误重试、连接管理机制，提高可靠性与容错能力。其在系统中的位置如下图所示。



**图 4 安全信道通信模块系统功能**

镜像签名认证软件启动时，系统会对安全信道通信模块进行初始化，进行握手与认证，客户端建立与服务端的HTTPS/TLS安全连接，执行身份认证，建立安全通信通道。

客户端将签名后的 TUF 元数据（如 root.json, targets.json, snapshot.json, timestamp.json）通过 POST/PUT 请求或RPC请求发送至服务器时，安全信道通信模块负责进行必要的签名封装（如 JWS）与 HTTP Header 加密设置，服务端安全信道通信模块接收请求并转发至 TUF 校验逻辑模块，服务端校验成功后写入后端存储并返回 200/ACK 响应，若请求失败，则返回错误码，由客户端安全信道通信模块进行错误处理或重试。

当用户拉取镜像时，客户端会通过安全信道通信模块向服务端安全信道通信模块发起元数据请求，服务端返回 TUF 元数据，安全信道通信模块负责校验数据完整性与签名有效性，客户端据此决定是否允许镜像拉取操作。

同时安全信道通信模块还支持网络异常重试、连接超时与断线重连、错误日志记录以及自定义错误响应类型等错误处理机制。

#### 签名认证算法子模块

签名认证算法模块是整个镜像签名与验证系统的核心安全基础，为签名生成、签名验证、公私钥管理、加密封装等多种功能提供基础支持，并提供加密算法多样性适配。其在系统中的功能如下图所示。



**图 5 签名认证算法模块系统功能**

用户推送镜像或元数据进行签名时，会接收元数据结构并提取签名角色及其私钥，之后调用对应算法（如ECDSA/RSA-PSS）进行签名，并根据用户配置封装为JWS格式，之后返回签名后的数据由安全信道通信模块进行上传。

用户因拉取镜像而需要对签名进行验证时，系统提取元数据 JSON 和签名字段，从本地或根元数据提取相应公钥，根据签名算法校验签名是否合法，返回验证结果，通知客户端是否允许拉取镜像。

签名认证算法模块提供多种签名认证算法，包括ECDSA、RSA-PSS算法，并支持签名算法的扩展，用户可使用私有的签名算法进行替换；该模块还提供JWS格式的签名封装算法支持。

#### 配置管理子模块

配置管理模块负责系统中可变参数和运行配置项的统一管理，确保各功能模块能够灵活、可控地获取运行所需的配置信息。提供统一配置读取与分发、配置校验与默认值处理、服务端行为定义配置以及敏感配置项保护（如私钥存储路径、访问令牌等）功能。模块整体工作流程如下图所示。



**图 6 配置管理模块工作流程**

系统启动时会对配置管理模块进行初始化，加载配置文件、读取环境变量并应用配置默认值，之后会对关键配置项进行合法性与完整性校验，包括密钥路径是否存在、密钥密码是否为空、用户指定的加密算法是否合法，并对配置格式进行验证，之后缓存配置项供其他模块使用。

用户修改配置项时，首先通过客户端输入修改参数或修改配置文件提交配置变更请求，配置管理模块会重新加载并校验变更项，对需要重启系统的配置项进行用户提示输出，最后将修改后的配置项广播到依赖模块（如加密模块与签名模块）实现配置项更新。

用户可以通过配置管理模块配置镜像信任验证开关，当该项开启时，强制启用镜像签名认证软件内容信任策略，Docker客户端仅允许推送和拉取其已签名的镜像，且必须通过镜像签名认证软件服务验证签名的有效性；该项关闭时禁用内容信任，Docker 客户端允许操作未签名镜像，绕过镜像签名认证软件的签名验证流程。同时还支持用户进行密钥密码配置、密钥轮换策略配置、加密算法配置以及私钥存储路径配置等多种配置项，并支持通过配置文件定义服务端行为。

#### 日志记录子模块

日志记录子模块负责记录系统运行、用户操作、签名请求、配置变更、安全事件等，便于审计与追踪问题。

日志记录子模块支持多种类型的日志记录，包括系统启动日志（系统启动、配置加载、模块初始化等），操作审计日志（用户登录、配置修改、签名请求、信任验证过程等），错误日志（签名失败、配置错误、通信失败、安全检查不通过等）以及安全事件日志（密钥轮换、认证失败、非法请求、访问敏感资源等），并支持DEBUG / INFO / WARN / ERROR多种级别日志输出。

#### 数据备份子模块

数据备份子模块负责对 TUF 元数据、密钥文件、重要配置等进行备份，保障系统安全性与可恢复能力。

数据备份子模块支持对关键数据的备份与恢复功能，提供本地压缩打包以及远程上传的备份方式，可通过定时任务或手动触发，备份过程中会提供元数据的一致性保障，避免文件写入中断，并对密钥与配置等关键数据的备份进行加密保护，系统还提供恢复脚本用于快速恢复的支持。

## 内部接口设计

### 通用模块接口

表2‑2通用模块内部接口

|  |  |  |
| --- | --- | --- |
| 功能 | 接口名称 | 接口功能 |
| 安全信道通信 | HandleClientRequest() | 统一接收客户端请求，如签名上传、镜像推送、验证等，进行解析与初步处理 |
| AuthenticateUser() | 对请求用户进行身份鉴权，判断其角色与权限 |
| AuthorizeKeyUsage() | 将解析后的请求分发至内部相关模块 |
| RespondToClient() | 将处理结果封装为响应信息返回客户端 |
| 签名认证算法 | SignContent() | 对给定内容进行数字签名，支持算法如 RSA-PSS、ECDSA |
| VerifySignature() | 验证数字签名是否合法，支持多种算法与安全参数 |
| EncryptData() | 对输入数据进行加密处理 |
| DecryptData() | 对输入数据进行解密处理 |
| ComputeDigest() | 计算输入数据的摘要值 |
| LoadCryptoConfig() | 从配置模块加载当前加密算法及参数策略 |
| RefreshKeyCache() | 密钥变更或轮换后刷新密钥缓存 |
| 配置管理 | LoadConfigFromFile() | 从配置文件中加载系统初始化配置信息 |
| LoadConfigFromEnv() | 从环境变量中加载配置信息 |
| GetConfigValue() | 获取指定配置项的当前值 |
| GetModuleConfig() | 获取指定模块的完整配置信息 |
| SetConfigValue() | 更新某项配置，并触发模块热更新 |
| ReloadModuleConfig() | 通知模块重新加载配置（如算法切换等场景） |
| GetKeyRotationPolicy() | 获取密钥轮换策略配置 |
| GetSignaturePolicy() | 获取签名算法策略与密钥参数配置 |
| GetStoragePolicy() | 获取密钥与元数据的存储路径与访问配置 |
| GetRepoSyncPolicy() | 获取信任仓库同步策略与相关配置 |
| 日志记录 | LogInfo() | 记录系统运行过程中的普通信息日志 |
| LogWarn() | 记录警告类日志，如配置异常、响应超时等 |
| LogError() | 记录错误日志，如模块故障、签名失败、非法访问等 |
| QueryLog() | 按条件查询日志记录，支持关键字、时间范围、日志级别筛选 |
| 数据备份 | BackupMetadata() | 备份当前元数据状态，防止系统故障时丢失 |
| BackupKeyStorage() | 备份密钥文件目录，支持周期性自动备份 |
| RestoreFromBackup() | 从指定备份恢复元数据或密钥文件 |
| ScheduleBackupJob() | 设置备份任务调度策略，如定时、触发备份等 |
| ValidateBackupFile() | 校验备份文件完整性，确保可用于恢复 |

# 测试与验证策略

## 单元测试策略

## 集成测试策略

## 系统测试策略

### 功能测试方法

### 性能测试方法

# 项目验收与交付