|  |
| --- |
| [公司名称] |
| 镜像签名认证软件研制技术方案 |
| [文档副标题] |

|  |
| --- |
| [作者姓名]  **2025-4-20** |

**目录**

[一 研制任务要求 2](#_Toc456665631)

[1.1 项目概述 2](#_Toc72705839)

[1.2 技术要求 3](#_Toc48840930)

[二 技术方案 7](#_Toc530757356)

[2.1 签名认证系统客户端技术方案 7](#_Toc1939296301)

[2.1.1 架构设计 7](#_Toc1393620388)

[2.1.2 功能模块设计 8](#_Toc2121206934)

[2.1.3 基本实现方法 9](#_Toc748915891)

[2.1.4 技术选型 11](#_Toc627724970)

[2.1.5 部署方案 11](#_Toc1733896726)

[2.1.6 安全性与容错设计 11](#_Toc249184092)

[2.2 签名认证系统服务端技术方案 12](#_Toc443922594)

[2.2.1 架构设计 12](#_Toc648847680)

[2.2.2 功能模块设计 12](#_Toc260998294)

[2.2.3 基本实现方法 13](#_Toc1436720084)

[2.2.4 技术选型 14](#_Toc648324920)

[2.2.5 部署方案 15](#_Toc64905562)

[2.2.6 性能与扩展性 15](#_Toc2093571505)

[2.2.7 安全性与容错 15](#_Toc136728440)

[三 研制计划 16](#_Toc187388790)

# 一 研制任务要求

## 1.1 项目概述

本项目研制内容为镜像签名认证软件，其核心目标是实现镜像的签名认证功能。该软件的设计和实现基于 TUF (The Update Framework) 安全框架。TUF 是一个专注于安全软件更新的框架，通过采用 TUF 的多角色职责分离、多签名机制和密钥轮换机制。

本项目能够提供强大的内容信任能力，确保所使用的镜像未被篡改软件主要实现以下功能:

* 支持本地信任库的初始化和管理，包括生成和管理 TUF 元数据文件（如 root.json, targets.json, snapshot.json）。
* 提供全面的角色与密钥管理功能，支持生成和管理不同角色的密钥（Root, Targets, Timestamp, Snapshot），并实现密钥的加密存储、轮换和吊销检查。
* 实现与 Docker 环境的集成，支持在镜像推送时自动创建信任链并签名，在镜像拉取时进行信任验证和签名校验。
* 提供镜像信任信息的查看，能够显示签名者、密钥、信任链和密钥层级。
* 支持多种加密算法（如 ECDSA 和 RSA-PSS）和签名算法的可扩展性。
* 提供安全的通信通道、关键数据备份恢复 和详细的日志记录。

通过这些功能，镜像签名认证软件旨在确保在线升级系统中使用的镜像来源可信、内容完整，简化镜像的信任管理过程，提高软件供应链的抗攻击能力和韧性。

## 1.2 技术要求

1. 镜像签名认证软件客户端软件功能要求
   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. 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；
      3. 支持自定义配置 TUF 相关元数据文件存储路径功能；
      4. 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。
      5. 提供镜像签名认证软件信任库内容列举功能；
      6. 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；
      7. 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；
      3. 提供密钥轮换功能；
         1. 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；
         2. 提供密钥临期、逾期使用时告警，启动轮换流程；
         3. 定期自动轮换：支持按时间周期自动触发密钥轮换；
         4. 支持时间可配置轮换策略，按角色（Root、Targets、Snapshot）分层轮换。
      4. 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；
      3. 提供镜像元数据签名发布功能；
      4. 提供查看镜像信任信息功能；
      5. 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；
      6. 支持显示签名者公钥及其对应的密钥 ID；
      7. 支持验证信任链完整性功能；
      8. 提供撤销指定镜像信任功能；
      9. 支持元数据修改和版本更新；
      10. 支持向服务器端同步信任状态。
      11. 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；
      12. 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；
   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。
2. 镜像签名认证软件服务器软件功能要求
   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；
   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：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。
   5. 镜像签名认证软件客户端与服务端通用功能
      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；
      3. 支持加密算法兼容性：支持 ECDSA、RSA-PSS 算法；
      4. 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。
      5. 提供JWS 格式的签名封装；
      6. 提供关键数据的备份恢复功能；
      7. 提供日志记录的功能；

**性能要求**

* + 镜像签名认证软件服务器可支持不少于500个的软仓库公证维护；
  + 元数据文件生成及签名延时在1秒内；

**兼容性要求**

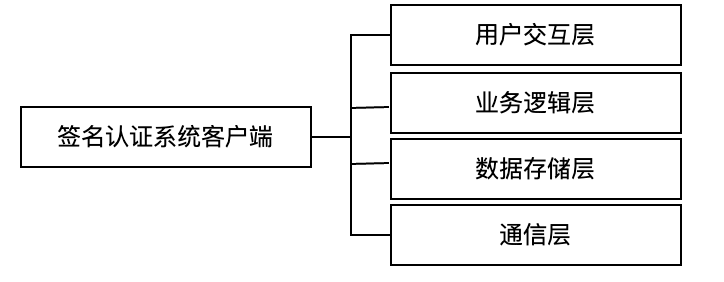
* + 支持与天脉操作系统的集成；
  + 客户端支持与Docker容器的集成；

# 二 技术方案

## 2.1 签名认证系统客户端技术方案

### 2.1.1 架构设计

分层架构：采用分层设计，分为用户交互层、业务逻辑层、数据存储层和通信层。



1. 用户交互层：

支持命令行工具（CLI）与Docker插件集成，提供操作指令输入与结果反馈。

1. 业务逻辑层：

实现镜像签名、信任库管理、密钥轮换等核心功能。

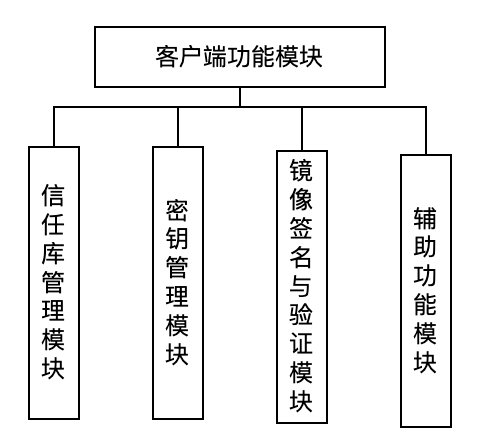
1. 数据存储层：

本地存储信任库元数据文件、密钥文件（私钥加密存储）。

1. 通信层：

与服务端通过HTTPS/TLS安全通道交互，支持JWS格式签名封装。

### 2.1.2 功能模块设计



（1）信任库管理模块

* 初始化本地信任库，生成TUF元数据文件（根密钥、目标密钥）。
* 支持信任库内容的增删改查、快照元数据生成与签名。

（2）密钥管理模块

* 密钥生成：支持根密钥（Root Key）、目标密钥（Target Key）的生成与加密存储。
* 密钥轮换：支持按策略（时间周期、角色分层）自动触发轮换流程。

（3）镜像签名与验证模块

* 镜像签名：推送时自动签名，生成签名元数据并上传至服务端。
* 镜像验证：拉取时验证信任链，检查签名有效性及密钥吊销状态。

（4）辅助功能模块

* 日志记录：记录关键操作（如密钥生成、签名失败）。
* 备份恢复：支持信任库元数据及密钥的备份与恢复。

### 2.1.3 基本实现方法

（1）信任库管理模块

* 使用C/C++调用OpenSSL库生成根密钥（Root Key）和目标密钥（Target Key），密钥对生成后私钥通过AES-256加密存储。
* 调用TUF框架API生成初始元数据文件（root.json、targets.json），存储路径通过配置文件动态指定（默认路径：/var/lib/mirror-sign/trusted）。
* 支持交互式命令行输入密码或通过环境变量（MIRROR\_SIGN\_KEY\_PASS）获取私钥解密密码。

1. 密钥管理模块

* 密钥生成与存储：
  + 根密钥与目标密钥的生成采用ECDSA算法，公钥明文存储于元数据文件，私钥加密后存储于本地安全目录（权限：600）。
  + 新增委托角色密钥时，通过libsodium生成临时密钥对，私钥加密后附加到目标元数据中。
* 密钥轮换：
  + 轮换策略基于定时任务（Cron）触发，检查密钥有效期（默认根密钥1年、目标密钥6个月）。
  + 轮换流程：生成新密钥→更新元数据版本→签名新元数据→同步至服务端。若失败，则回滚至上一版本元数据。

（3）镜像签名与验证模块

* 镜像签名：
  + 集成Docker Content Trust插件，拦截docker push命令，调用客户端工具生成镜像哈希值，使用目标密钥签名并封装为JWS格式。
  + 签名元数据（含时间戳、签名者ID）上传至服务端，服务端返回快照签名后更新本地缓存。
* 镜像验证：
  + 拉取镜像时，客户端从服务端获取最新元数据，验证信任链（根→目标→快照→时间戳）。
  + 若发现密钥吊销或签名不匹配，终止拉取并记录告警日志

（4）辅助功能模块

* 日志记录：
  + 采用syslog协议记录关键操作（如密钥生成、签名失败），日志文件加密存储于/var/log/mirror-sign。
* 备份恢复：
  + 每日定时备份信任库元数据及密钥文件至指定路径，支持通过命令行工具一键恢复。

### 2.1.4 技术选型

核心语言：

C/C++（遵循GJB 8114-2013安全编码规范）。

加密库：

OpenSSL（支持ECDSA、RSA-PSS算法）、Libsodium（临时密钥生成）。

Docker集成：

Docker Content Trust插件扩展，通过Docker Engine API拦截镜像操作。

TUF框架：

基于Notary参考实现移植，核心逻辑用C++封装。

安全性设计：

强制HTTPS/TLS 1.3，客户端证书双向认证（mTLS）。

SElinux策略限制客户端进程权限，禁止未授权文件访问

容错机制：

服务端不可用时，启用本地元数据缓存模式（最长有效期24小时）。

关键操作（如密钥轮换）支持原子性事务，失败自动回滚。

### 2.1.5 部署方案

* 环境依赖：
  + 操作系统：天脉OS（兼容性已验证）。
  + 运行时：Docker Engine 20.10+，OpenSSL 3.0。
* 安装方式：
  + 提供二进制包，支持一键部署脚本
  + 配置文件路径：/etc/mirror-sign/conf（密钥存储路径可自定义）。

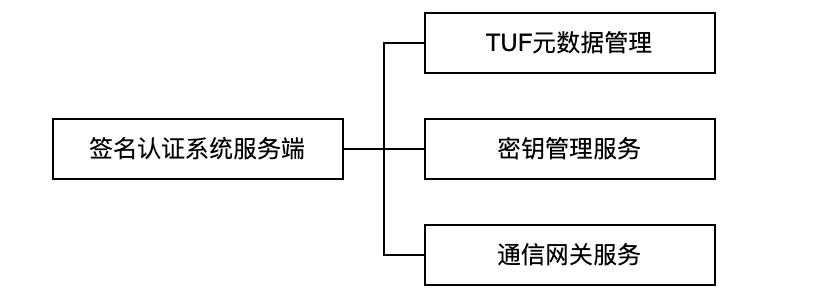
### 2.1.6 安全性与容错设计

* 安全性：
  + 私钥加密存储，密码通过环境变量或交互式输入。
  + 通信通道使用TLS 1.3加密，证书双向验证。
  + 日志文件权限限制，防止未授权访问。
* 容错设计：
  + 操作事务化：关键流程（如密钥轮换）支持回滚机制。
  + 心跳检测：定期检查服务端可用性，失败时触发本地缓存模式。

## 2.2 签名认证系统服务端技术方案

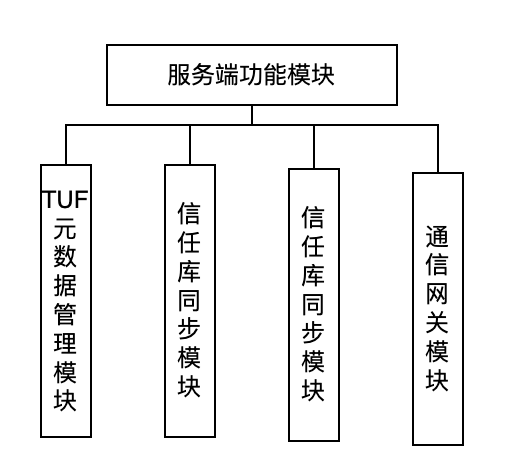
### 2.2.1 架构设计

采用微服务架构：



* 元数据管理服务：处理TUF元数据存储、版本更新与冲突检测。
* 密钥管理服务：管理时间戳密钥、快照密钥的生成与轮换。
* 通信网关服务：接收客户端请求，进行身份认证与请求路由。

### 2.2.2 功能模块设计



* TUF元数据管理模块
  + 存储与更新元数据文件（根、目标、快照、时间戳）。
  + 验证客户端提交的元数据格式、签名及版本一致性。
* 密钥管理模块
  + 生成时间戳密钥（Timestamp Key）、快照密钥（Snapshot Key）
  + 支持按角色分层轮换策略（如时间戳密钥每30天自动轮换）
* 信任库同步模块
  + 接收客户端的信任状态同步请求，更新全局信任库。
  + 提供吊销列表（CRL）的发布与校验功能。
* 网关通信模块
  + 确保通信稳定
  + 保证信道安全

### 2.2.3 基本实现方法

（1）TUF元数据管理模块

* 使用分布式文件系统（MinIO）存储元数据文件，每个仓库独立目录（如/repos/<repo\_id>/metadata），支持版本快照。
* 客户端提交元数据时，服务端校验签名有效性及版本连续性，冲突检测通过乐观锁（ETag）实现。

（2）密钥管理模块

* 时间戳密钥（Timestamp Key）采用RSA-PSS算法，快照密钥（Snapshot Key）采用ECDSA算法，私钥存储于HSM（硬件安全模块）。
* 密钥轮换由服务端定时任务触发，轮换后更新元数据并广播至所有客户端（通过长连接通知）。

（3）信任库同步模块

* 客户端发起吊销请求后，服务端更新全局CRL（证书吊销列表），并生成签名元数据增量包（delta metadata）。
* 客户端定期拉取CRL，本地缓存有效期24小时，超时后强制同步。

（4）通信网关模块

* 客户端需携带JWT令牌（通过双向TLS握手生成），网关服务校验令牌权限后路由至对应微服务。
* 高并发场景下，采用线程池（200线程）处理请求，超时阈值设为3秒。

（5）通用模块设计

* 安全通信
  + 客户端与服务端交互使用HTTPS双向认证，证书由内部CA签发，TLS 1.3协议强制启用。
  + 数据封装为JWS格式，签名算法标识符（如ES256、PS384）嵌入HTTP请求头。
* 算法扩展性
  + 提供动态链接库（.so文件）接口，支持用户替换默认签名算法（需实现sign()、verify()函数）。
* 监控与告警
  + 服务端性能指标（QPS、签名延时）通过Prometheus采集，Grafana可视化看板实时展示。
  + 异常事件（如密钥临期、服务不可用）触发企业微信/邮件告警。

### 2.2.4 技术选型

核心语言：

C/C++（高性能多线程模型，Boost.Asio处理网络IO）。

分布式存储：

MinIO（兼容S3协议）存储TUF元数据文件，支持版本快照。

密钥安全：

HSM（如YubiHSM 2）存储时间戳密钥、快照密钥私钥。

强制HTTPS/TLS 1.3，客户端证书双向认证（mTLS）。

元数据存储启用MinIO服务器端加密（SSE-S3）。

RBAC权限模型，管理员操作需二次认证（如OTP）。

服务架构：

gRPC协议实现元数据管理、密钥管理服务分离。

Prometheus采集QPS、签名延时等指标，Grafana可视化看板实时展示。

ELK栈（Elasticsearch+Logstash+Kibana）聚合分析操作日志，检测异常模式（如高频吊销请求）

### 2.2.5 部署方案

* 集群部署：支持多节点横向扩展，负载均衡（Nginx）。
* 容器化部署：提供Docker镜像，支持Kubernetes编排。
* 配置管理：通过Consul实现配置中心化，动态调整轮换策略。

### 2.2.6 性能与扩展性

性能优化：通过资源分配和调度优化，提高虚拟机和容器的整体运行效率。

扩展性设计：支持模块化设计，便于功能的扩展和系统升级。

### 2.2.7 安全性与容错

* 安全性：
  + 服务端密钥存储于HSM，禁止明文导出。
  + 客户端请求需携带JWT令牌，权限分级（如管理员、普通用户）。
* 容错设计：
  + 高可用：采用主从热备模式，故障时自动切换。
  + 数据冗余：元数据文件多副本存储，定期增量备份。

# 三 研制计划

项目交付计划以下具体如下表所示。

|  |  |
| --- | --- |
| 交付物 | 说明 |
| 技术方案 | 2025年4月 |
| 需求文档 | 2025年6月 |
| 设计文档 | 2025年6月 |
| 源码 | 2025年8月 |
| 测试文档 | 2025年9月 |
| 验收文档：  软件用户手册  软件研制总结报告  软件溯源分析报告  软件知识产权说明 | 2025年10月前完成 |