OTA安全管理系统

详细设计说明书

**天津三源电力智能科技有限公司**

文档修订记录

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **版本号** | **修改时间** | **修改人** | **审核人** | **批准人** | **备注** |
| 1 | V1.0 | 2022/1/10 | 何恩超 | 孙剑 | 孙剑 | 创建 |
| 2 | V1.1 | 2022/1/14 | 何恩超 | 孙剑 | 孙剑 | 根据评审内容调整。 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

[1 引言 4](#_Toc4355)

[1.1 目的和范围 4](#_Toc12937)

[1.2 术语和缩略语 4](#_Toc3107)

[1.3 参考文档 4](#_Toc6470)

[2 总体设计 4](#_Toc26299)

[2.1 系统总体架构确认 4](#_Toc6068)

[2.2 应用间接口设计 5](#_Toc4335)

[3 详细功能设计 6](#_Toc14392)

[3.1 OTA安全管理系统应用功能设计 6](#_Toc8990)

[3.1.1 OTA安全管理系统应用总体结构 6](#_Toc13085)

[3.1.2 模块及功能点划分 6](#_Toc30782)

[3.1.3 数据库设计 48](#_Toc21484)

[3.1.4 关键类设计 48](#_Toc24673)

[3.1.5 UI/UE设计 48](#_Toc14966)

[4 接口详细设计 48](#_Toc21515)

[5 数据迁移 49](#_Toc23871)

[6 非功能性设计 49](#_Toc6012)

[6.1 系统性能 49](#_Toc8234)

[6.2 资源使用率 49](#_Toc29989)

[6.3 信息安全 49](#_Toc10642)

[6.4 可靠性 50](#_Toc12507)

[6.5 易用性 50](#_Toc12607)

[6.6 互用性 50](#_Toc8581)

[6.7 扩展性 50](#_Toc8142)

[6.8 维护性 51](#_Toc3415)

[6.9 合法/合规性 51](#_Toc19073)

[7 附录 51](#_Toc1247)

[7.1 密钥算法清单 51](#_Toc13988)

# 引言

## 目的和范围

本文档编写的目的是针对OTA安全系统的各个子系统或模块的设计实现进行考虑，为开发人员编码提供基础，设计内容包括总体架构确认，各子系统的详细功能设计、数据库设计和接口设计等。

本文预期的读者包括：开发经理、架构师、项目经理、开发工程师、UI/UE设计师、QA等

## 术语和缩略语

| **术语/缩略语** | **描述** |
| --- | --- |
| 应用服务  （子系统） | 可独立部署的单元，也可称为子系统 |
| 模块 | 模块是从业务维度上对功能点集合的划分，一个模块对应于代码里的一个module目录 |
| 功能点 | 功能点是指通过用户交互触发、外部系统触发或后台程序触发来完成的一个完整的业务场景 |

## 参考文档

| **文档名** | **内容简介** | **类别** |
| --- | --- | --- |
| OTA安全管理系统系统需求规格说明书 |  | 依据 |
| OTA安全管理系统系统概要设计说明书 |  | 依据 |
| 编码规范-数据库设计规范 |  | 参考 |
| RFC5257 |  | 参考 |
|  |  |  |
|  |  |  |

# 总体设计

## 系统总体架构确认

|  |  |  |  |
| --- | --- | --- | --- |
| **逻辑分层（水平分层/垂直分层）** | **应用服务名称** | **应用服务的功能描述** | **开发方式** |
| 应用 | OTA安全管理系统 | 固件加密、签名处理 | 自主开发 |
| 硬件 | 加密机 | 数据加密、数据签名 | 采购 |
|  |  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |

## 应用间接口设计

*无*

# 详细功能设计

## OTA安全管理系统应用功能设计

### OTA安全管理系统应用总体结构

*{*

OTA安全管理系统模块依赖关系

### 模块及功能点划分

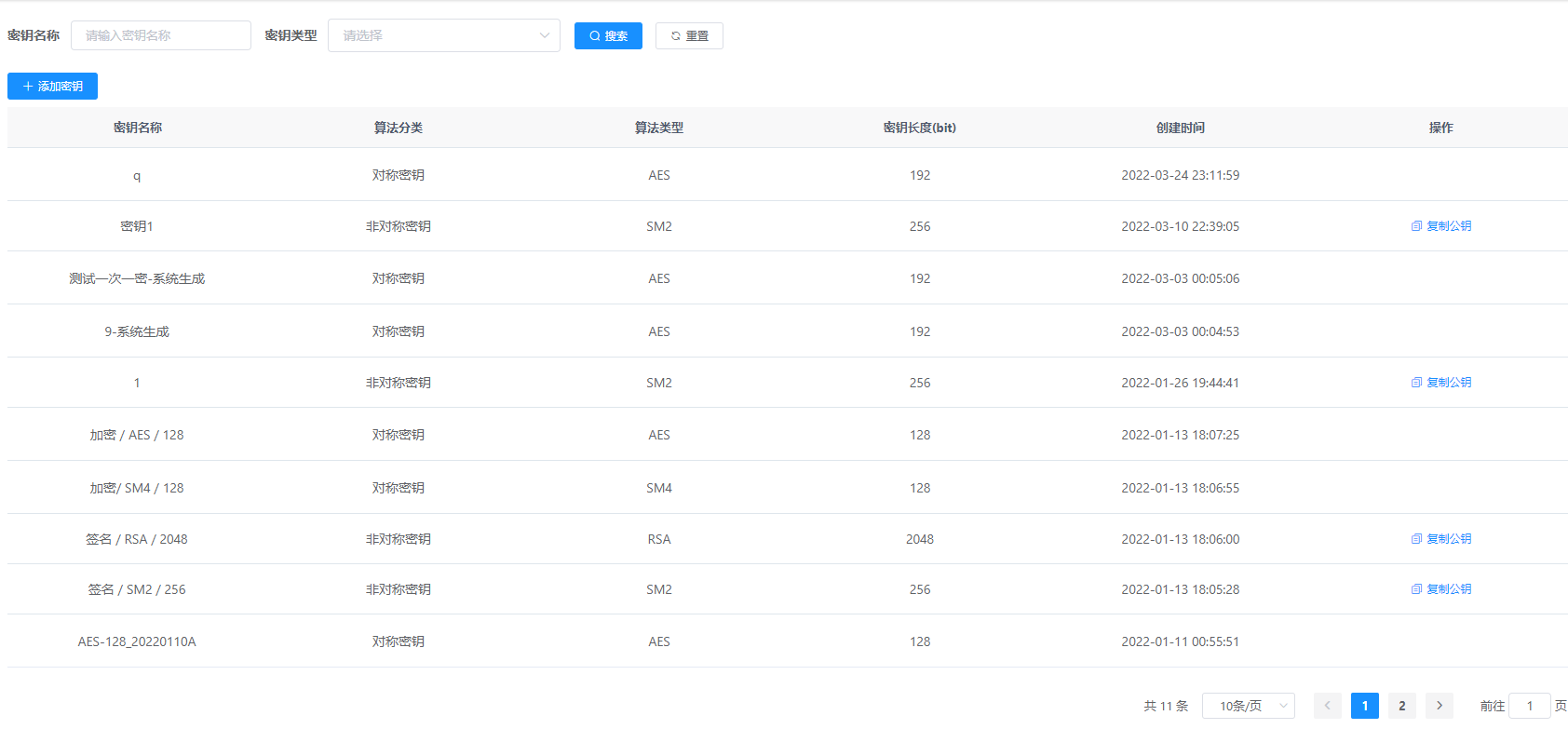
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块代码** | **模块名称** | **功能点编号**  **（系统需求ID）** | **功能点名称**  **（系统需求名称）** | **功能点简要描述** |
| **OTA-SRS-KM** | **密钥管理** | **OTA-SRS-KM-01** | **密钥列表** | **展示系统中已存在的密钥信息（包含对称密钥和非对称密钥）** |
| **OTA-SRS-KM-02** | **密钥查询** | **根据条件查询已存在的密钥** |
| **OTA-SRS-KM-03** | **密钥生成** | **根据条件生成指定的密钥数据** |
| **OTA-SRS-KM-07** | **公钥复制** | **复制非对称密钥的公钥** |
| **OTA-SRS-FW-** | **固件管理** | **OTA-SRS-FW-01** | **固件列表** | **展示所有固件信息** |
| **OTA-SRS-FW-02** | **固件查询** | **查询所有固件信息** |
| **OTA-SRS-FW-03** | **查看固件详情** | **查看固件处理的策略，加密、签名耗时等信息** |
| **OTA-SRS-FW-04** | **固件导入** | **导入待处理的固件** |
| **OTA-SRS-FW-05** | **固件解析** | **对导入的固件格式校验** |
| **OTA-SRS-FW-06** | **固件加密** | **对固件加密处理** |
| **OTA-SRS-FW-07** | **固件存储** | **对加密后的固件进行存储** |
| **OTA-SRS-FW-08** | **固件导出** | **对加密后的固件密文以及信封进行导出** |
| **OTA-SRS-FW-09** | **固件删除** | **删除历史的固件** |
| **OTA-SRS-TP** | **策略管理** | **OTA-SRS-TP-01** | **策略模板列表** | **展示所有固件模板的处理策略** |
| **OTA-SRS-TP-02** | **查询策略模版** | **根据条件查询固件的处理模板** |
| **OTA-SRS-TP-03** | **新建策略模版** | **定义、创建固件的处理模板，方便固件加密使用** |
| **OTA-SRS-TP-04** | **修改策略模版** | **固件的处理模板，方便固件加密使用** |
| **OTA-SRS-TP-05** | **默认策略模版** | **将某一模板设定为系统默认的处理模板** |
| **OTA-SRS-TP-06** | **删除策略模版** | **删除不使用的模板** |
| **OTA-SRS-LOG** | **日志管理** | **OTA-SRS-LOG-01** | **系统登录日志** | **记录系统的登录信息** |
| **OTA-SRS-LOG-02** | **用户操作日志** | **记录系统中用户的操作行为** |
| **OTA-SRS-LOG-03** | **固件日志** | **查看固件的处理日志信息** |
| **OTA-SRS-LOG-04** | **查看日志** | **查看日志处理的详情** |
| **OTA-SRS-LOG-05** | **查询日志** | **根据 条件查询日志** |
| **OTA-SRS-SM** | **系统管理模块** | **OTA-SRS-SM-01** | **用户管理** | **对系统的用户进行添加、删除等管理操作** |
|  | **OTA-SRS-SM-02** | **机构管理** | **对部门、岗位等管理** |
|  | **OTA-SRS-SM-03** | **权限管理** | **对操作用户的权限进行设置** |

***功能点的描述、业务规则、前置条件、处理流程、输入描述、输出描述等信息详见SRS文档。***

#### **密钥管理模块**

##### **密钥列表**

1. 页面设计:



1. 涉及表和字段信息：

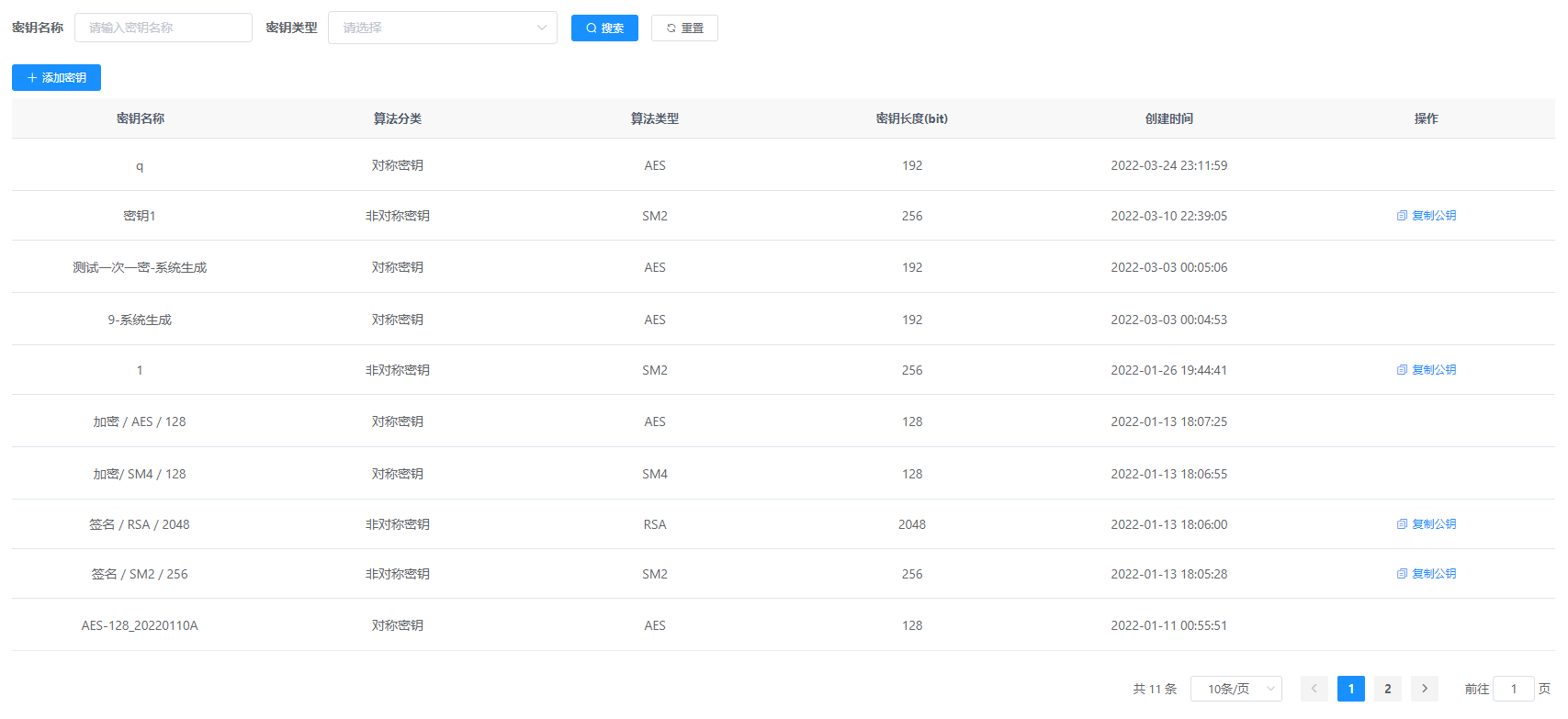


1. 实现思路

展示管理员添加的对称密钥和非对称密钥。其中非对称密钥可以复制公钥。

##### **密钥查询**

1. 页面设计：

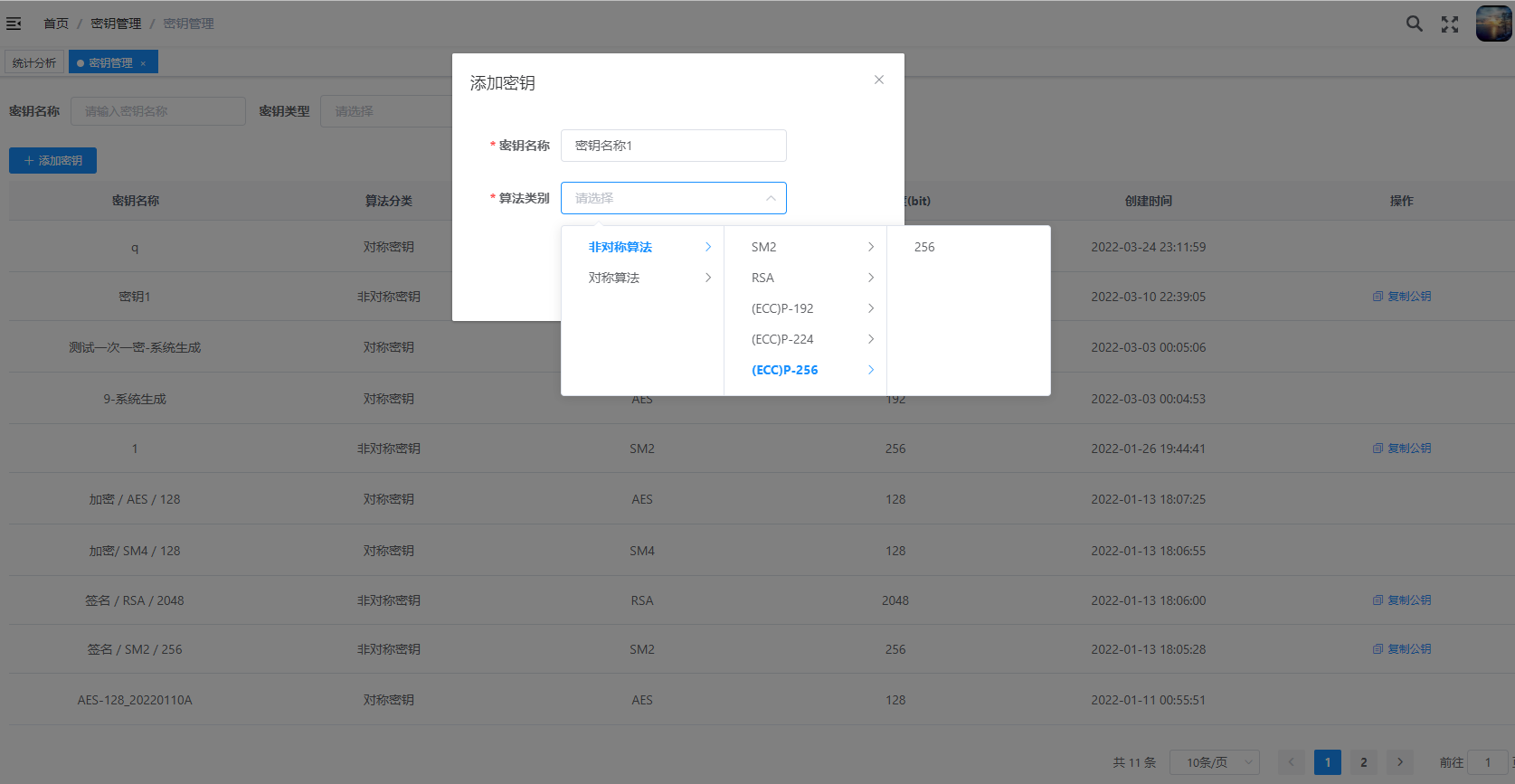


1. 涉及表和字段信息： 
2. 实现思路

根据密钥名称、密钥类型、进行搜索，搜索的结果在密钥列表中展示。

##### **密钥生成**

1. 页面设计



1. 流程图：



1. 涉及表和字段信息：



1. 实现思路：

页面选择对应的密钥分类【对称密钥、非对称密钥】，和密钥分类【对称密钥SM4(128bit)、AES(128、192、256bit)】对应的算法以及密钥长度，调用密钥生成模块产生密钥，并将密钥安全存储到数据库。

对称密钥生成: 软算法生成的对称密钥使用本地lmk加密存储到ota\_key\_info表的privatekey字段；加密机生成的对称密钥使用加密机lmk加密导出，存储到ota\_key\_info表的privatekey字段。

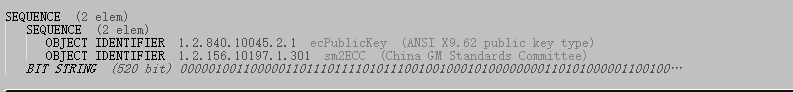
非对称密钥生成：软算法生成的非对称密钥，私钥使用本地lmk加密存储到ota\_key\_info表的privatekey字段，公钥以明文的形式存储到ota\_key\_info表的publickey字段；加密机生成的非对称密钥，私钥使用加密机lmk加密存储到ota\_key\_info表的privatekey字段，公钥以明文的形式存储到ota\_key\_info表的publickey字段。

1. 密钥格式

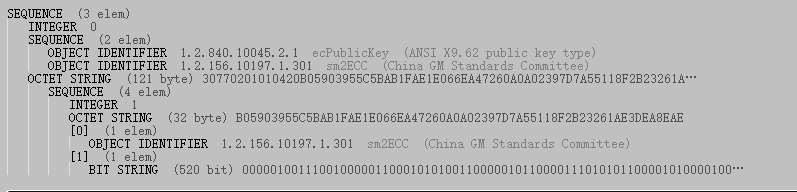
①对称密钥，对应密钥长度的二进制数据

②非对称密钥

公钥x509格式

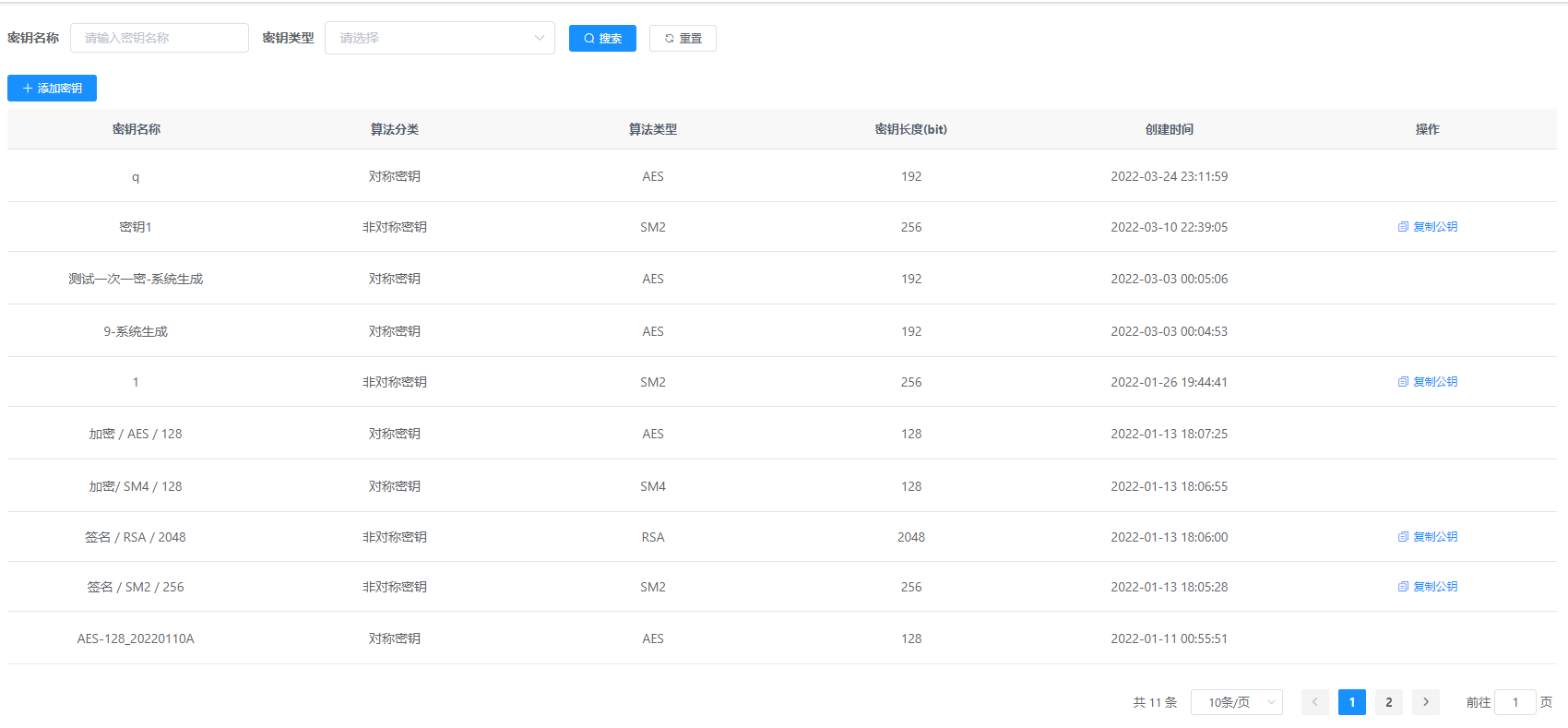


私钥PKCS8格式



##### **复制公钥**

1. 页面设计



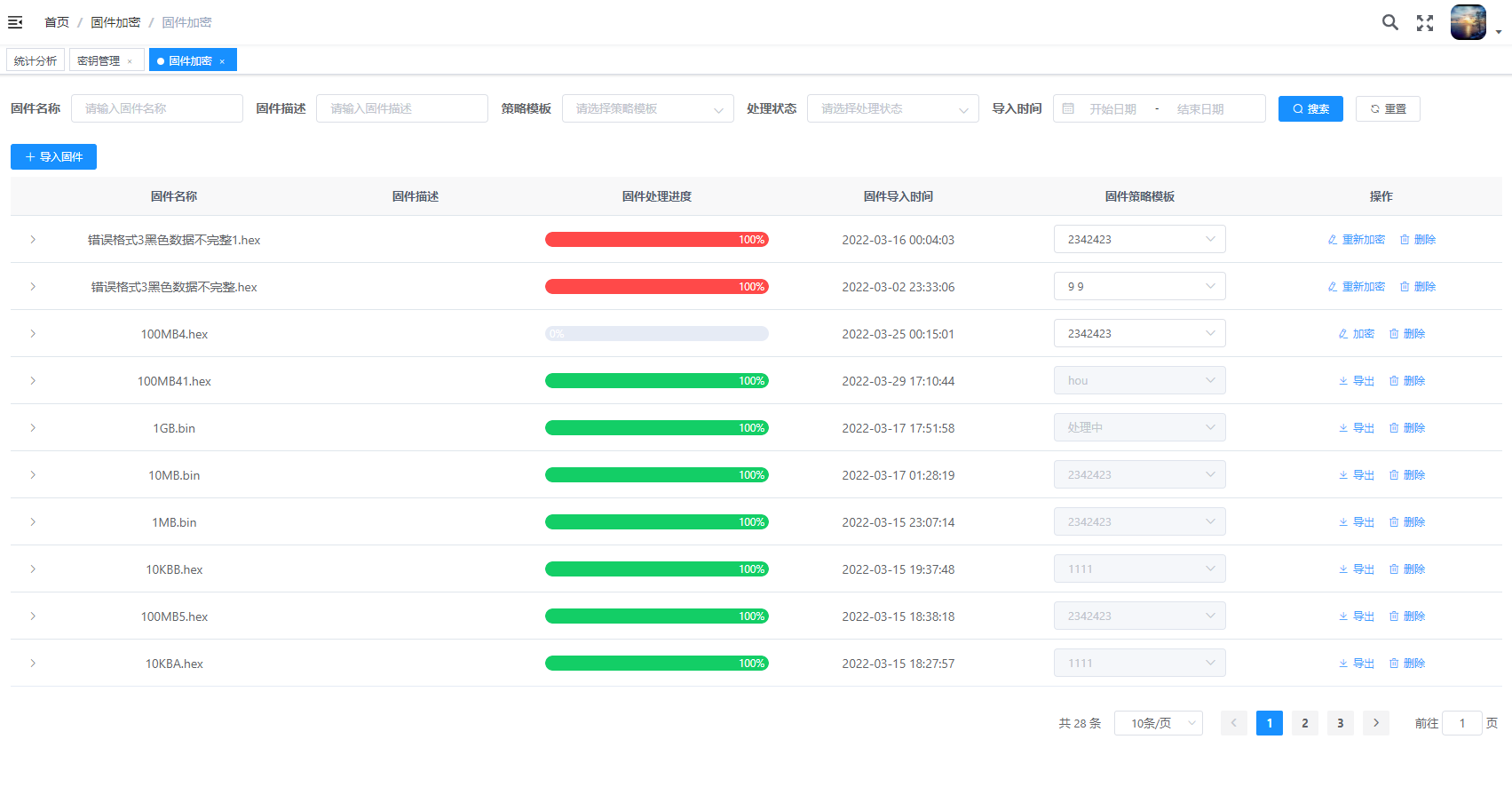
1. 实现思路

对非对称密钥的公钥进行复制，将复制的结果存入内存中，便于使用。

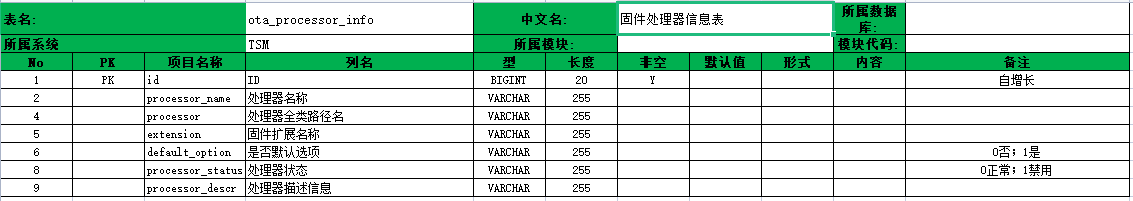
#### **固件管理模块**

##### **固件列表**

1. 页面设计



1. 涉及数据库





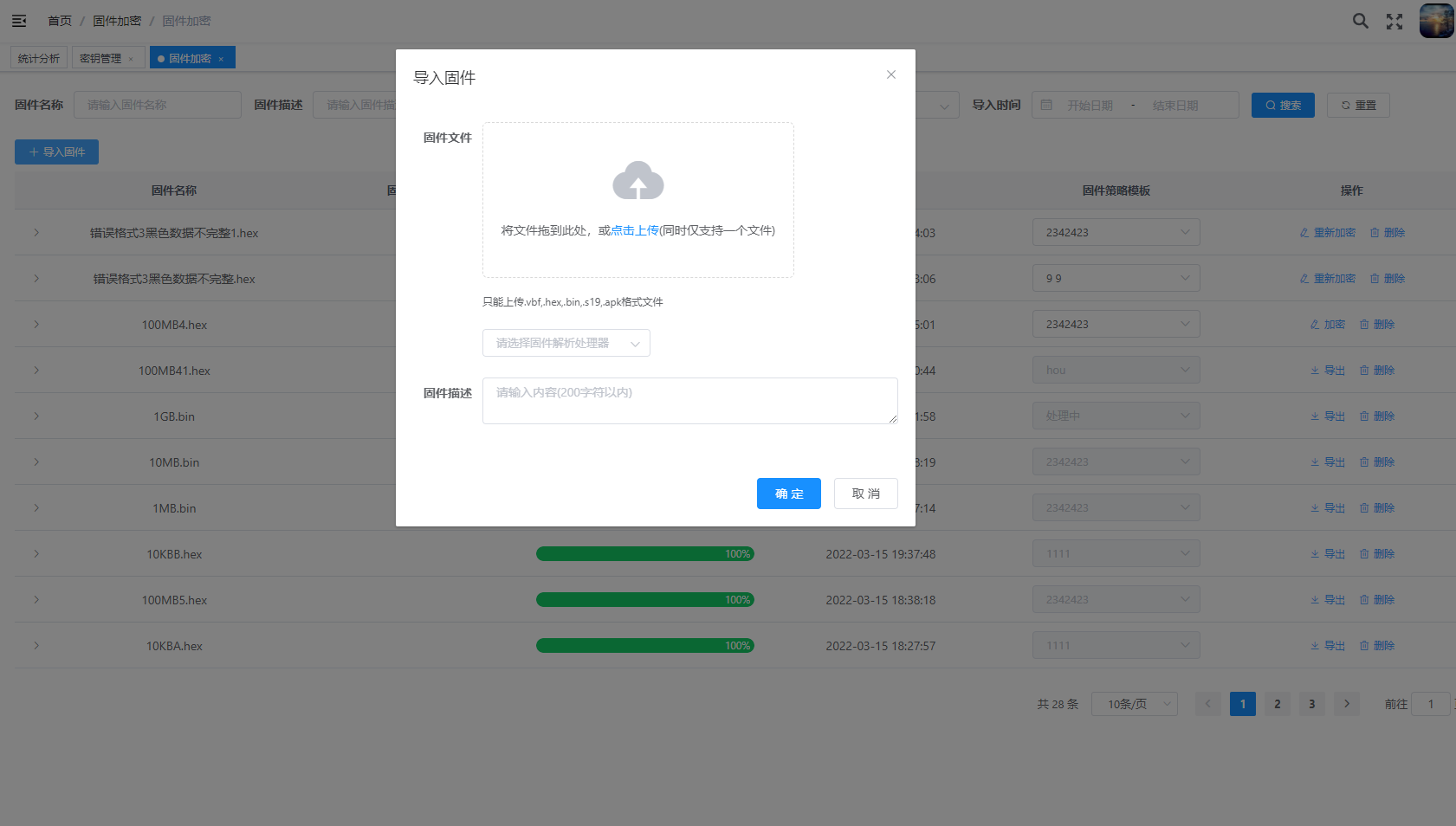


1. 实现思路

展示所有已导入的固件。其中未做处理的固件，进度条为灰色，可以进行固件加密处理和删除（物理删除固件和数据记录，并存储删除日志）。固件加密处理中的固件显示为蓝色，不可删除不可再加密；处理完成的固件进度条显示为绿色，并可以删除（物理删除固件、处理后的文件和数据库记录，并存储操作日志）；处理失败的固件，显示为红色，处理失败后日志记录失败原因，并可以再次进行加密处理。

##### 固件导入

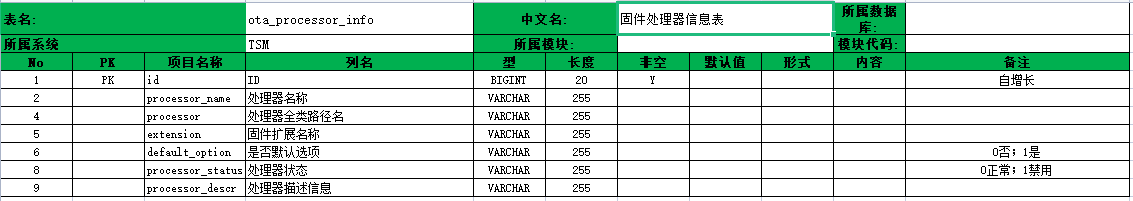
1. 页面设计



1. 流程图



1. 涉及表和字段信息：









1. 实现思路

固件上传过程包含以下步骤。

1. 在固件加密页面点击导入固件，从本地选择要导入到系统的固件文件，系统识并自动上传文件并返回文件存储路径，固件名称、已上传文件存储位置、文件大小（单位字节），如果固件名称重复则提示相关信息，终止文件上传。
2. 前端根据已上传的固件文件扩展名获取解析处理器下拉选择（数据来源于ota\_processor\_info固件处理器信息表中数据）。
3. 用户选择对应的处理解析器、描述性信息、点击确定将步骤①中返回的信息一并提交给服务器程序。
4. 服务器程序获取到前端相关传入信息、并获取默认策略模板（ota\_strategy\_template\_info中default\_template=0的模板记录id和name），一并将以上信息存储到数据库。涉及到插入的数据表和字段项有 ota\_firmware\_info （待处理固件信息表）【id自增加、file\_name固件名称、file\_desc固件描述、file\_type固件扩展名称类型、source\_file\_location固件文件存储路径、source\_file\_input\_type原始固件传入方式0为页面、source\_file\_size原始文件大小】ota\_firmware\_process\_task\_info （固件处理过程表）【biz\_no业务编号，uuid 32字节、file\_id固件文件表id，percentage处理进度0.0、task\_status任务状态 0未开始、file\_process\_status 0默认为成功、strategyId策略表id、strategy\_detail策略记录json数据、processorId固件处理器信息表id、processor\_name固件处理器信息表processor\_name处理器名称、processor固件处理器信息表processor处理器全类路径名】

##### **固件解析**

1. 页面设计

无

1. 涉及表





1. 实现思路

系统根据文件扩展名，对固件进行解析，固件解析格式，依据不同的固件类型，使用不同的固件解析器处理。解析器应包含格式检查器，固件处理引擎核心接口。

##### 固件加密

1. 页面设计
2. 流程图



1. 涉及表和字段信息：









1. 实现思路：

选择待处理的固件文件（ota\_firmware\_info数据）、固件处理策略（ota\_firmware\_process\_task\_info 中包含处理策略id 关联ota\_strategy\_template\_info 中数据数据）、固件处理器（ota\_firmware\_process\_task\_info 中包含 固件处理器id 关联ota\_processor\_info 固件处理器信息数据）程序根据以上条件对固件数据加密和（或）固件签名处理。

**固件格式校验**：依据不同文件类型，使用对应的处理器校验格式。

①bin文件格式：内容为大于0字节的任意内容。

②hex文件格式：每行是否以为英文：冒号开始；描述数据长度的字节和本行实际的数据长度是否一致；校验和的值与独立计算的校验和是否一致；

③S19文件格式：每行是否以:S开始；描述数据长度的字节和本行实际的数据长度是否一致；校验和的值与独立计算的校验和是否一致；

**固件加密密钥的获取**：如果策略不含固件加密处理，则略过此步骤。如果处理略为一次一密，则系统根据整体的密钥生成方式（软算法或加密机）、策略中选定的密钥算法、长度信息即时生成对称密钥，并将对称密钥以安全的形式存储到数据表（ota\_key\_info）；如果处理策略为非一次一密，则根据策略中选定的密钥id获取对应的密钥信息。

**固件签名密钥的获取**：如果策略不含固件签名处理，则略过此步骤。根据策略选定的固件签名密钥id获取对应的签名密钥信息。

**固件加密操作的处理**：系统依据固件扩展名称不同，采用不同的固件处理器；依据不同的策略，使用策略选定的固件加密密钥、加密模式、填充模式对固件进行加密处理，处理完成后的文件存储到系统的磁盘中，在数据库中记录（ota\_firmware\_process\_task\_info target\_file\_location）具体的磁盘路径。

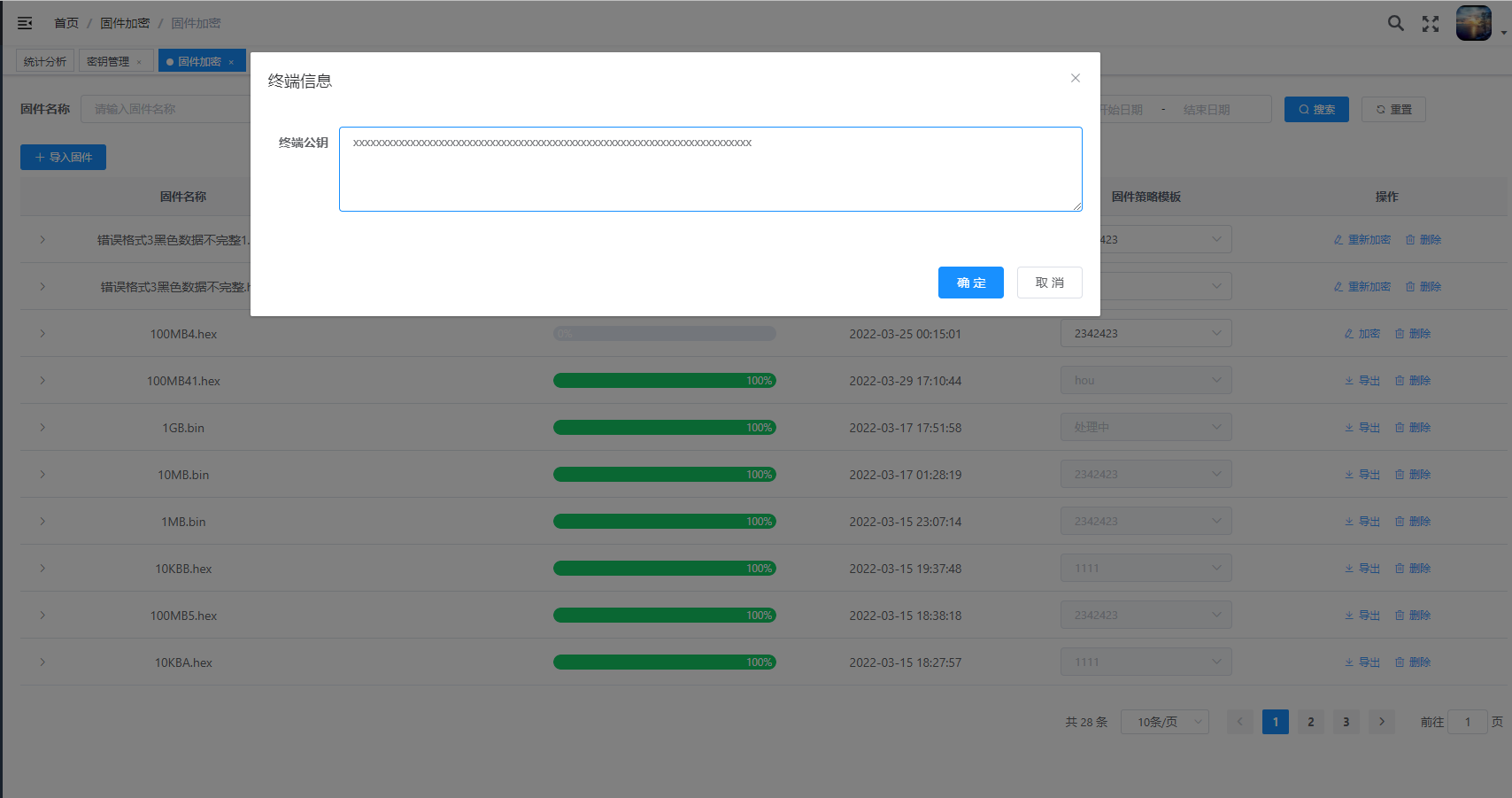
①bin文件整文件进行加密处理。

②hex文件只对纯数据部分加密，加密后将密文转成hex文件格式存储。

③S19 TODO

##### 固件导出

1. 页面设计：



1. 流程图：



1. 涉及表和字段信息：





1. 实现思路

信封结构依据文档(https://www.rfc-editor.org/rfc/rfc5652)，以下简称5652文档。以下描述的内容为内部约定的内容，标准的内容参考5652文档。

**仅签名信封**：仅签名信封采用5652文档(5.1节)SignedData结构。其中SubjectKeyIdentifier 的计算方法 为《0015 基于SM2密码算法的数字证书格式规范.pdf>> 5.2.4.2.2.2小结 a)。待签名数据位于SignedData 中 encapContentInfo的eContent内容部分，类型eContentType 5652文档中描述的 Data Content Type 类型，待签名的数据为固件原始文件的hash值，hash值的计算依据采用的签名算法的摘要算法，如SHA256withRSA，其hash算法为SHA256。

样例：

**仅加密信封**：仅加密信封采用5652文档(6.1节)EnvelopedData结构。其中待加密的数据结构为私有约定asn.1 编码的结构 ExchangedInfo，其结构如下：

|  |
| --- |
| ExchangedInfo ::= SEQUENCE {  encAlg AlgorithmIdentifier //包含了算法名称、密钥长度、加密模式 paddingType Integer //填充模式  key OCTET STRING //固件文件加密密钥  iv OCTET STRING OPTIONAL //固件文件加密所使用的iv  hashAlg AlgorithmIdentifier OPTIONAL //hash算法标识符  hashValue OCTET STRING OPTIONAL //hash值  } |

结构字段描述

(1)encAlg AlgorithmIdentifier 可参考 http://gmssl.org/docs/oid.html 中定义。来源为策略中选择的密钥对应的算法名称，密钥长度和加密模式。

AES/128/ECB 2.16.840.1.101.3.4.1.1

AES/128/CBC 2.16.840.1.101.3.4.1.2

AES/128/OFB 2.16.840.1.101.3.4.1.3

AES/128/CFB 2.16.840.1.101.3.4.1.4

AES/128/WRAP 2.16.840.1.101.3.4.1.5

AES/128/GCM 2.16.840.1.101.3.4.1.6

AES/128/WRAP\_PAD 2.16.840.1.101.3.4.1.8

AES/192/ECB 2.16.840.1.101.3.4.1.21

AES/192/CBC 2.16.840.1.101.3.4.1.22

AES/192/OFB 2.16.840.1.101.3.4.1.23

AES/192/CFB 2.16.840.1.101.3.4.1.24

AES/192/WRAP 2.16.840.1.101.3.4.1.25

AES/192/GCM 2.16.840.1.101.3.4.1.26

AES/192/WRAP\_PAD 2.16.840.1.101.3.4.1.28

AES/256/ECB 2.16.840.1.101.3.4.1.41

AES/256/CBC 2.16.840.1.101.3.4.1.42

AES/256/OFB 2.16.840.1.101.3.4.1.43

AES/256/CFB 2.16.840.1.101.3.4.1.44

AES/256/WRAP 2.16.840.1.101.3.4.1.45

AES/256/GCM 2.16.840.1.101.3.4.1.46

AES/256/WRAP\_PAD 2.16.840.1.101.3.4.1.48

SM4/128/ECB 1.2.156.10197.1.104.1

SM4/128/CBC 1.2.156.10197.1.104.2

SM4/128/OFB 1.2.156.10197.1.104.3

SM4/128/CFB 1.2.156.10197.1.104.4

SM4/128/GCM 1.2.156.10197.1.104.8

SM4/128/WRAP 1.2.156.10197.1.104.11

SM4/128/WRAP\_PAD 1.2.156.10197.1.104.12

(2)paddingType 采用数字表示 。来源为策略中配置的填充模式。

1: nopadding

2: Zero Byte Padding

3: PKCS#5/PKCS#7 Padding

4：TBC Padding

5：ISO7816-4 Padding

6: ISO101026-2 Padding

7: X9.23 Padding

(3)hashAlg、hashValue在仅加密结构中不填充值。

SHA1 1.3.14.3.2.26

SHA224 2.16.840.1.101.3.4.2.4

SHA256 2.16.840.1.101.3.4.2.1

SHA384 2.16.840.1.101.3.4.2.2

SHA512 2.16.840.1.101.3.4.2.3

SM3 1.2.156.10197.1.401

EnvelopedData中RecipientInfo根据用户公钥类型不同，采用不同的RecipientInfo。

其中RSA算法采用KeyTransRecipientInfo，密钥加密算法为AES128\_CBC；ECC、SM2 算法采用 KeyAgreeRecipientInfo类型，密钥协商算法为ECCDH\_SHA256KDF，密钥加密算法AES128\_WRAP，结构加密算法为AES128\_CBC。

样例：

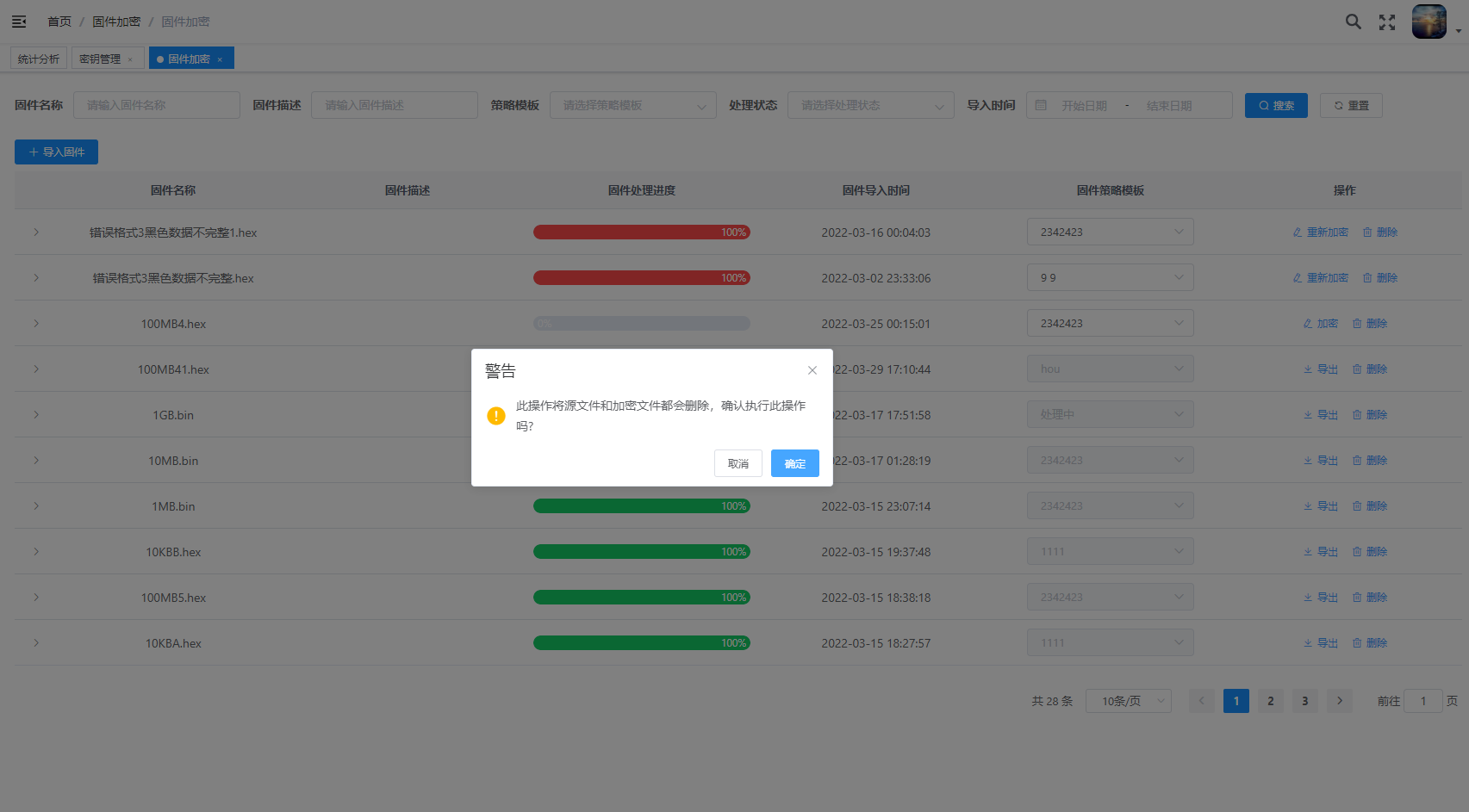
**加密签名信封**：加密签名信封采用5652文档中5.1和6.1结合，即将EnvelopedData作为SignedData待签名数据放到EnvelopedData的encapContentInfo中。 签名算法为系统策略中选取的签名算法，签名值为固件策略管理中选择的签名密钥和签名算法对EnvelopedData的签名。

在加密签名信封这种结构中，ExchangedInfo 的 hashAlg 和hashValue有值，hashAlg为签名算法对应的hash算法，如SHA256withRSA，hash算法为SHA256；hashVal为使用此hash算法对固件原文做摘要得到的值。

样例：

##### **固件删除**

1. 页面设计：



1. 涉及到的表



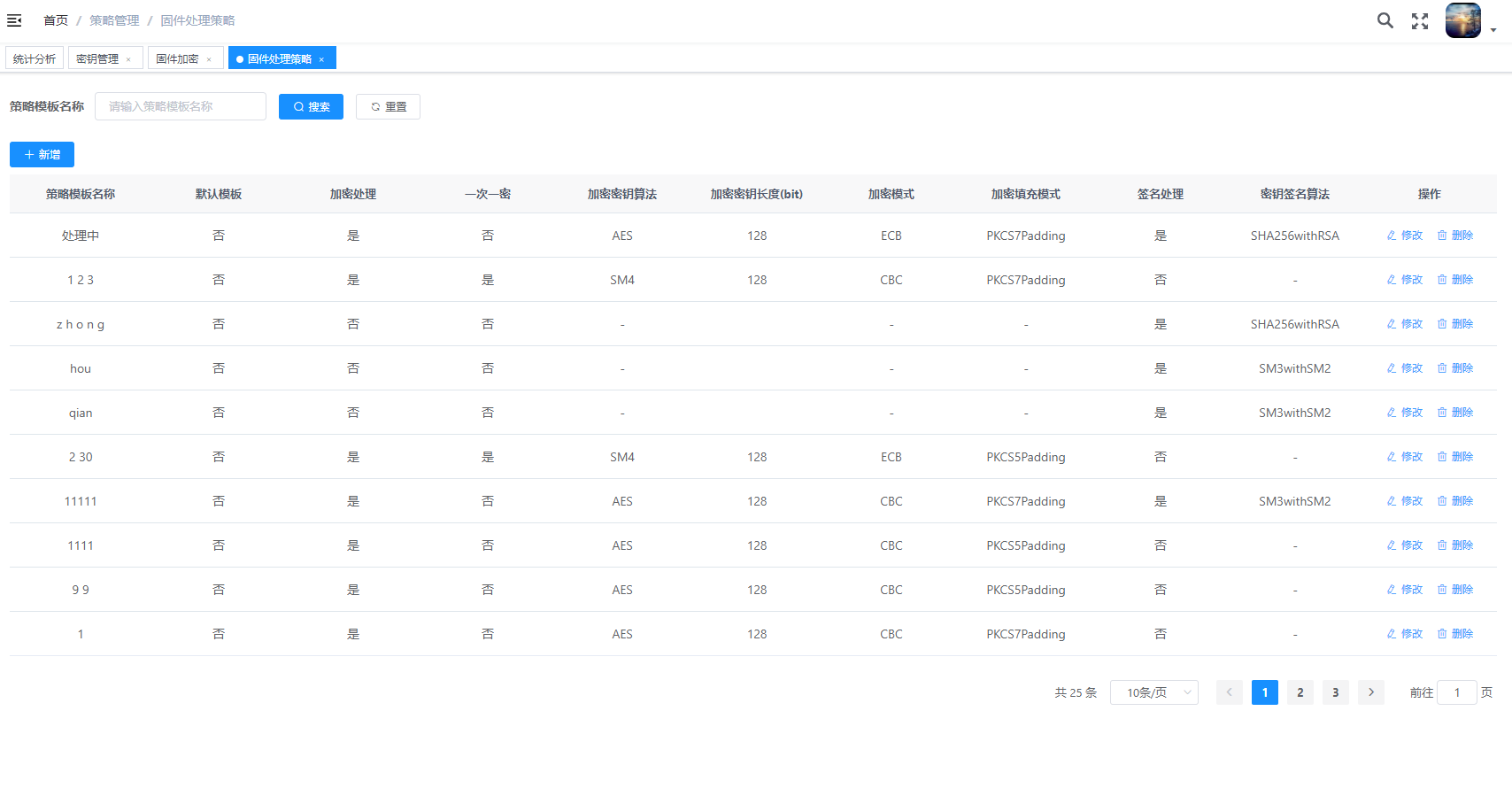


1. 实现思路。固件文件在固件导入后、固件加密处理完成后、固件处理失败后可进行删除。删除步骤是①将存储在服务器端的原始固件文件删除；②将存储在服务器端处理产生的文件删除（加密处理成功的前提下）；③将固件任务处理表数据删除；④将固件数据表数据删除；⑤记录相关的日志。

#### **策略管理模块**

##### **策略模板列表**

1. 页面设计



1. 涉及到的表和字段



1. 实现思路

通过列表查询所有的策略模板，并在数据列表展示。

##### **查询策略模板**

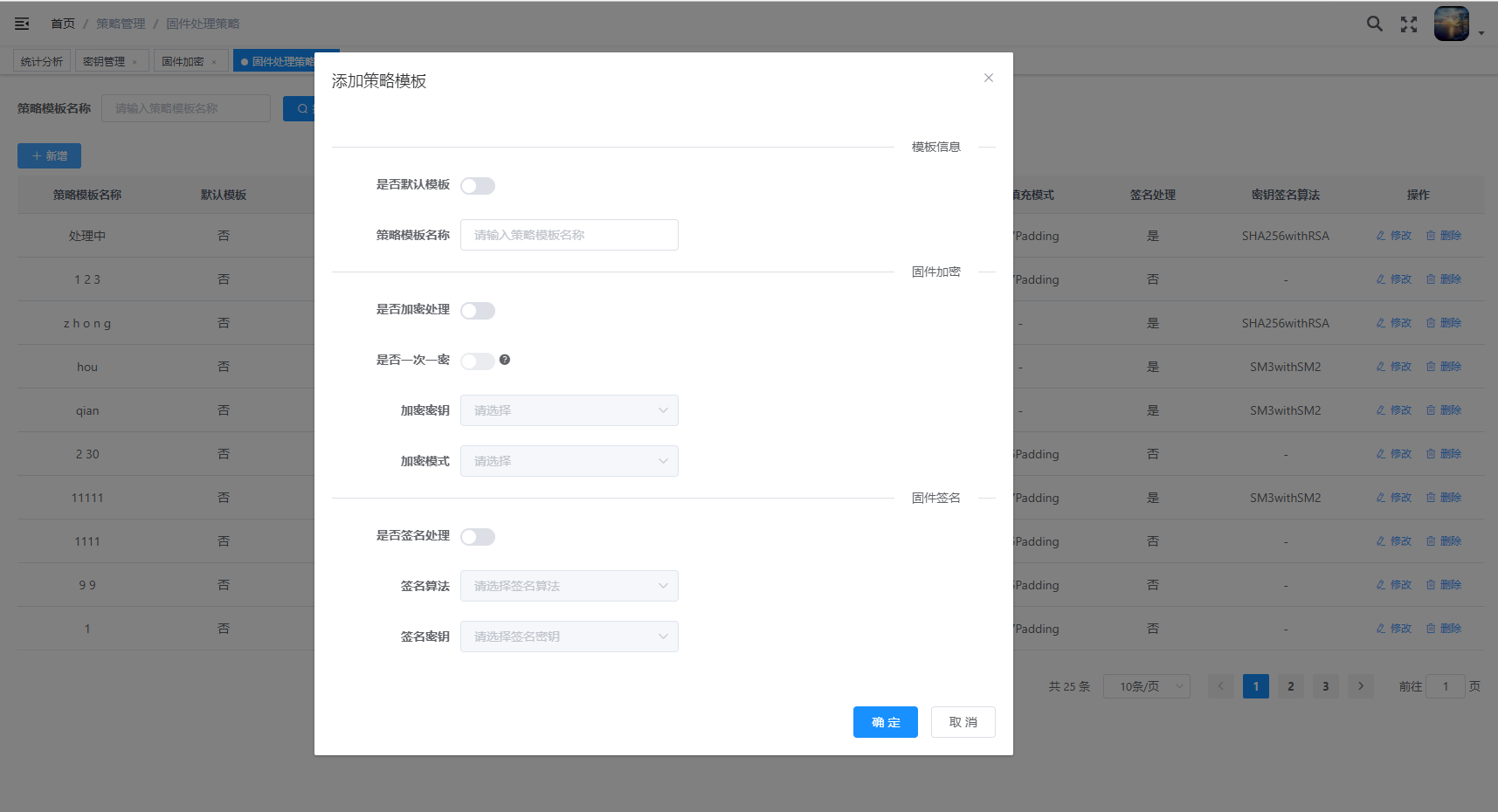
1. 页面设计
2. 涉及到的表



1. 实现思路。通过页面输入策略模板名称，点击搜索触发异步查询，根据条件查询数据库数据，并返回结果。

##### **新建策略模板**

1. 页面设计



1. 流程图：



1. 涉及表和字段信息：



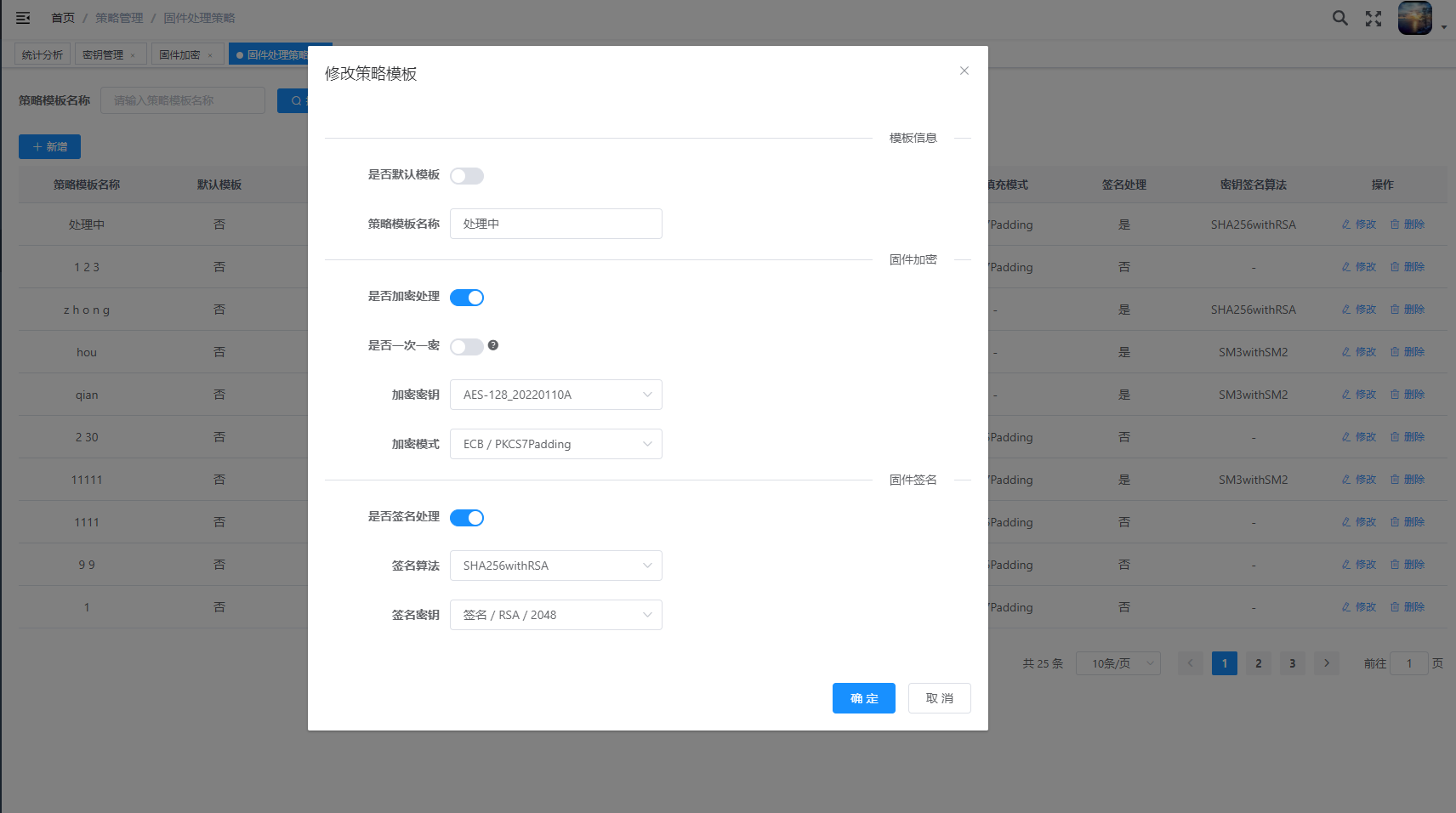
1. 实现思路：通过策略模板信息表，灵活定义系统在对固件处理时的策略。

①默认模板：系统第一次添加模板时，将该模板置为默认模板，如想将其他模板更改为默认模板，只需要将目标模板【是否默认模板】按钮勾选，保存成功后，新的模板即为默认模板。

②模板策略信息保存了本模板是否为默认模板（default\_template=0 默认，1非默认）、策略是否支持加密处理（do\_encryption=0 否，1是）、是否一次一密（once\_symmetry\_key=0否 ，1是）、一次一密时定义了加密的密钥算法（encryption\_key\_alg），长度（encryption\_key\_size），加密模式（encryption\_key\_mode）和填充模式（encryption\_padding\_mode）； 非一次一密时定义了选择的密钥id（encryption\_key\_id）密钥算法（encryption\_key\_alg）、密钥算法（encryption\_key\_alg），长度（encryption\_key\_size），加密模式（encryption\_key\_mode）和填充模式（encryption\_padding\_mode）；是否签名（do\_signature=0否1是），选择的签名密钥（signature\_key\_id）、签名算法（signature\_key\_alg）。

##### **修改策略模板**

1. 页面设计



1. 流程图：



1. 涉及表和字段信息：



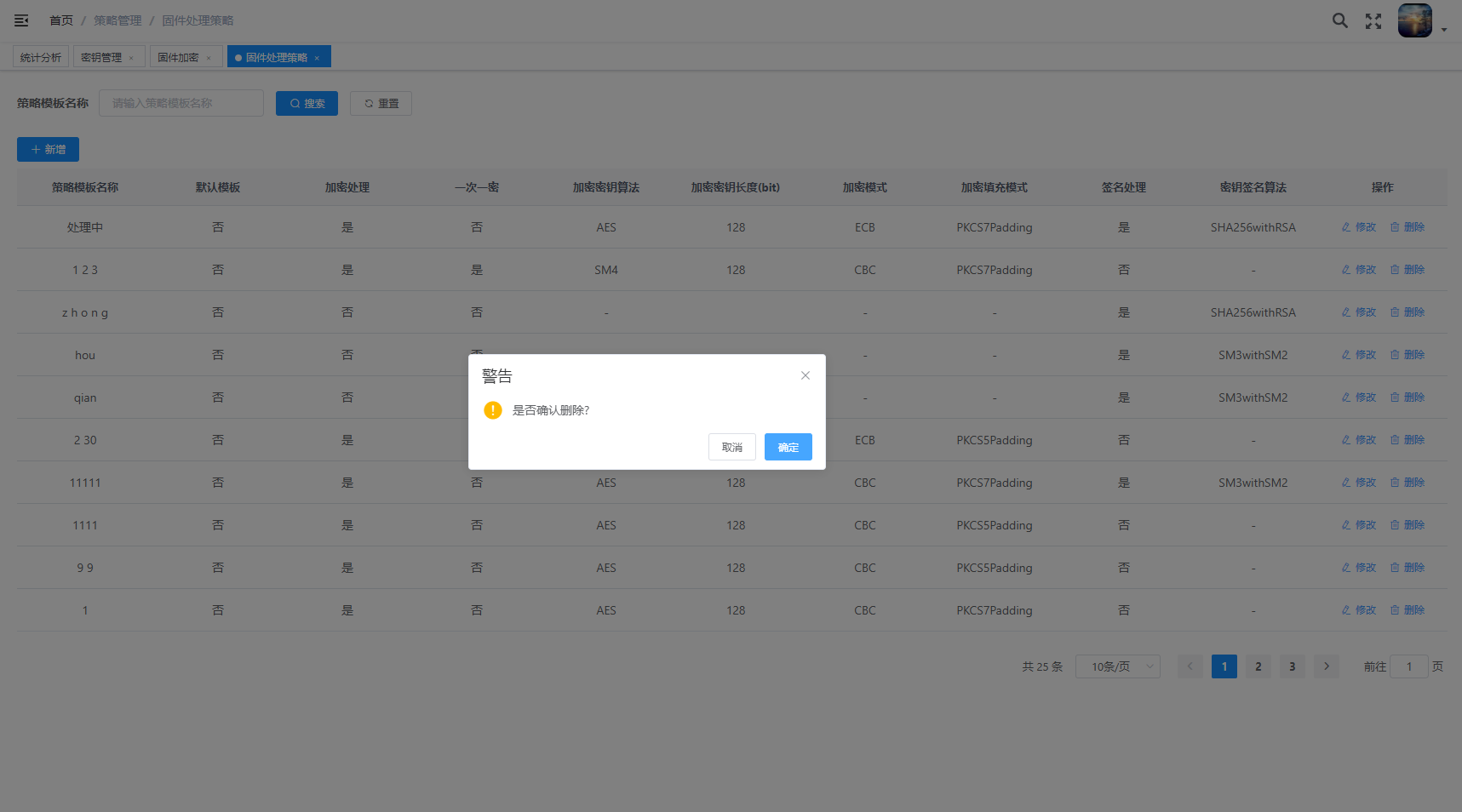
1. 实现思路：通过策略模板信息表，灵活定义系统在对固件处理时的策略。

①默认模板：系统第一次添加模板时，将该模板置为默认模板，如想将其他模板更改为默认模板，只需要将目标模板【是否默认模板】按钮勾选，保存成功后，新的模板即为默认模板。

②模板策略信息保存了本模板是否为默认模板（default\_template=0 默认，1非默认）、策略是否支持加密处理（do\_encryption=0 否，1是）、是否一次一密（once\_symmetry\_key=0否 ，1是）、一次一密时定义了加密的密钥算法（encryption\_key\_alg），长度（encryption\_key\_size），加密模式（encryption\_key\_mode）和填充模式（encryption\_padding\_mode）； 非一次一密时定义了选择的密钥id（encryption\_key\_id）密钥算法（encryption\_key\_alg）、密钥算法（encryption\_key\_alg），长度（encryption\_key\_size），加密模式（encryption\_key\_mode）和填充模式（encryption\_padding\_mode）；是否签名（do\_signature=0否1是），选择的签名密钥（signature\_key\_id）、签名算法（signature\_key\_alg）。

##### **删除固件模板**

1. 页面设计



1. 涉及到的表

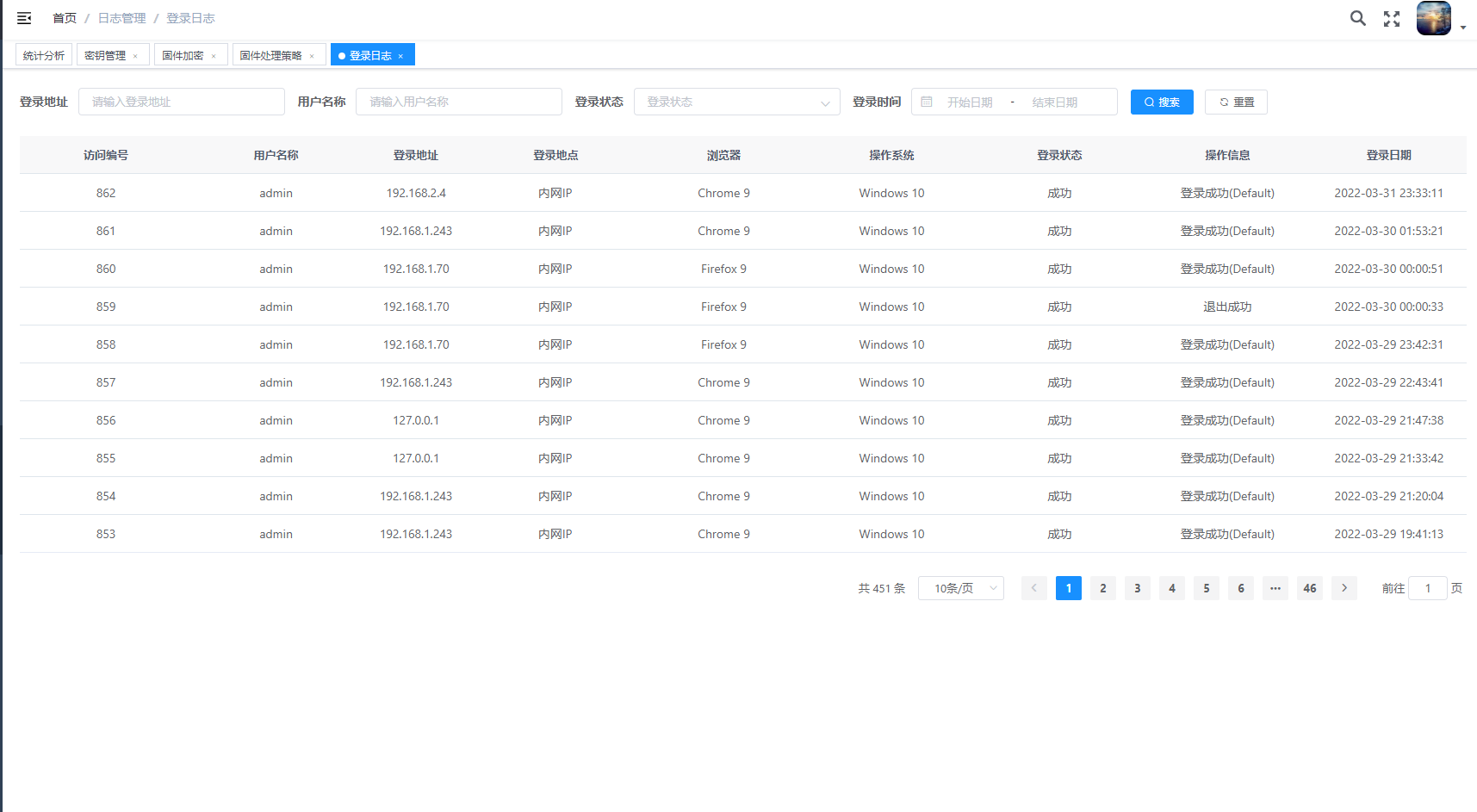


1. 实现思路。通过点击页面指定固件模板数据行的删除按钮，提醒用户确认删除，点击确定后，触发后台服务器请求，如果引用此模板的固件在处理中败则提示不允许删除。否则将该条记录从数据库中物理删除。

#### **安全审计模块**

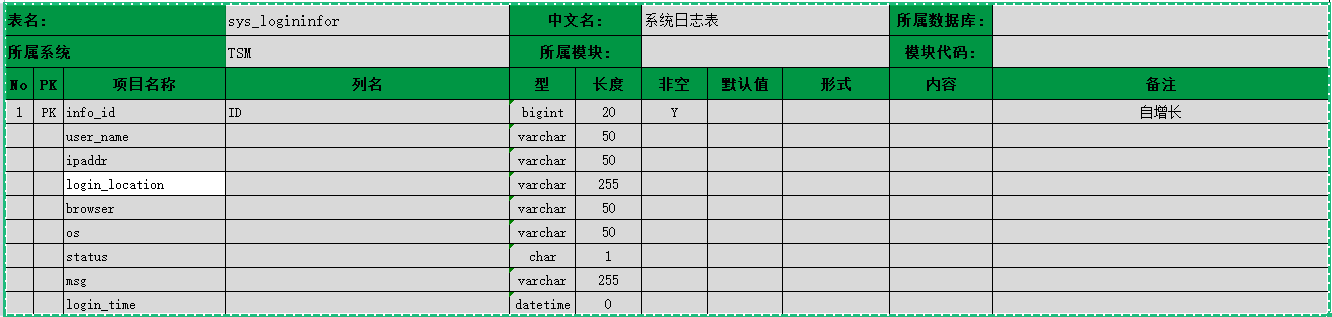
##### 系统登录日志

1. 页面设计



系统登录日志，记录系统用户登录的信息。方便后续追溯审计。系统登录日志记录管理员登陆的ip、管理员的信息、操作的成功与否、登录的时间等信息。

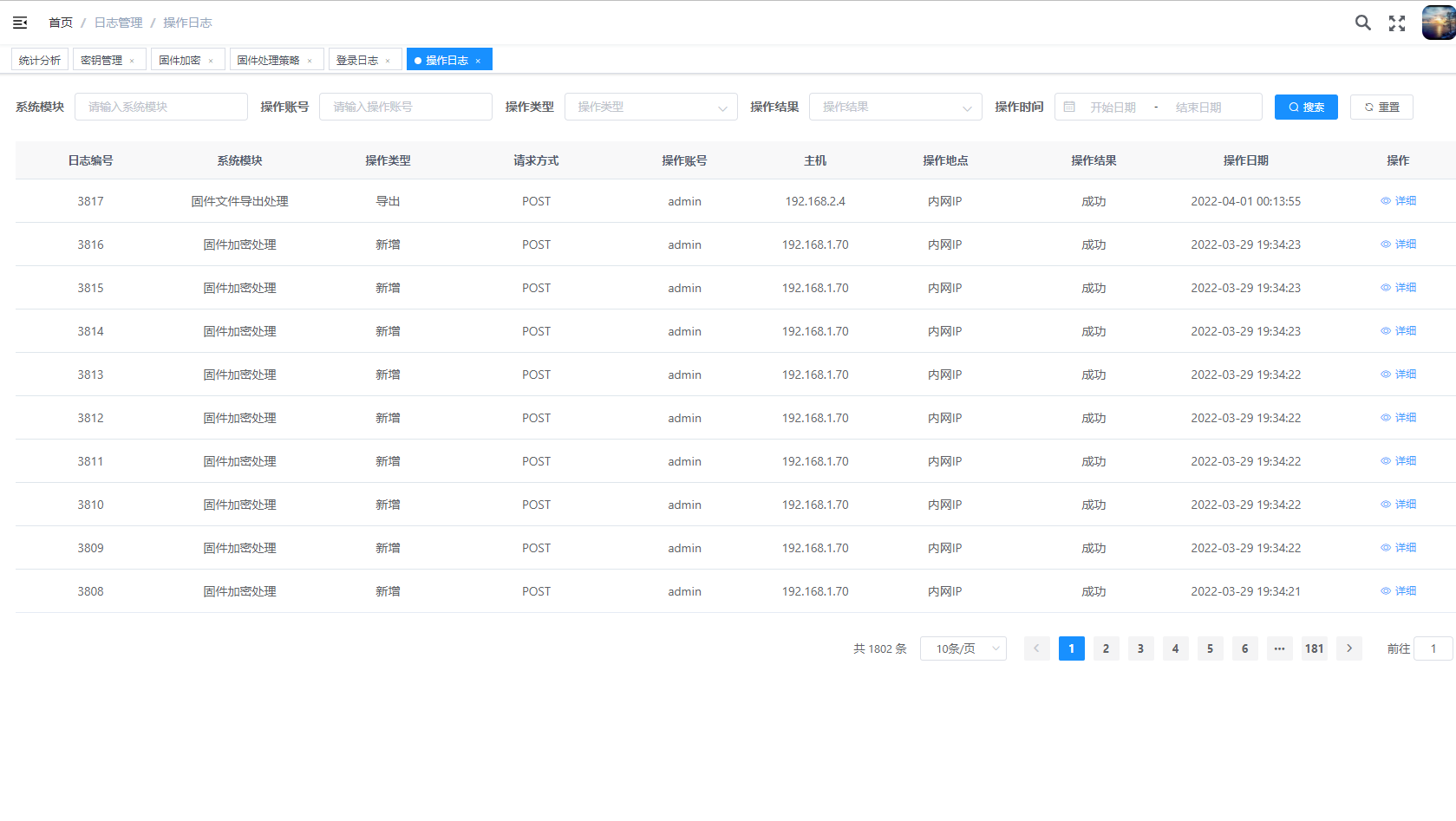
1. 涉及表和字段信息：



1. 实现思路。在登录功能逻辑处理完成后，根据登录功的返回值，判断本次是否登录成功，记录的信息通过http请求头中获取，将格式清洗转换后，存储到数据库中。

##### 用户操作日志

1. 页面设计



1. 涉及表和字段信息：



1. 实现思路 。用户操作日志记录管理员用户的日常操作行为，记录请求的数据和本次处理的结果。用于后续审计追溯。对每个涉及到数据变化的操作 ，记录请求的内容和响应的内容以及本次操作记录的结果。

##### 固件处理日志

1. 页面设计



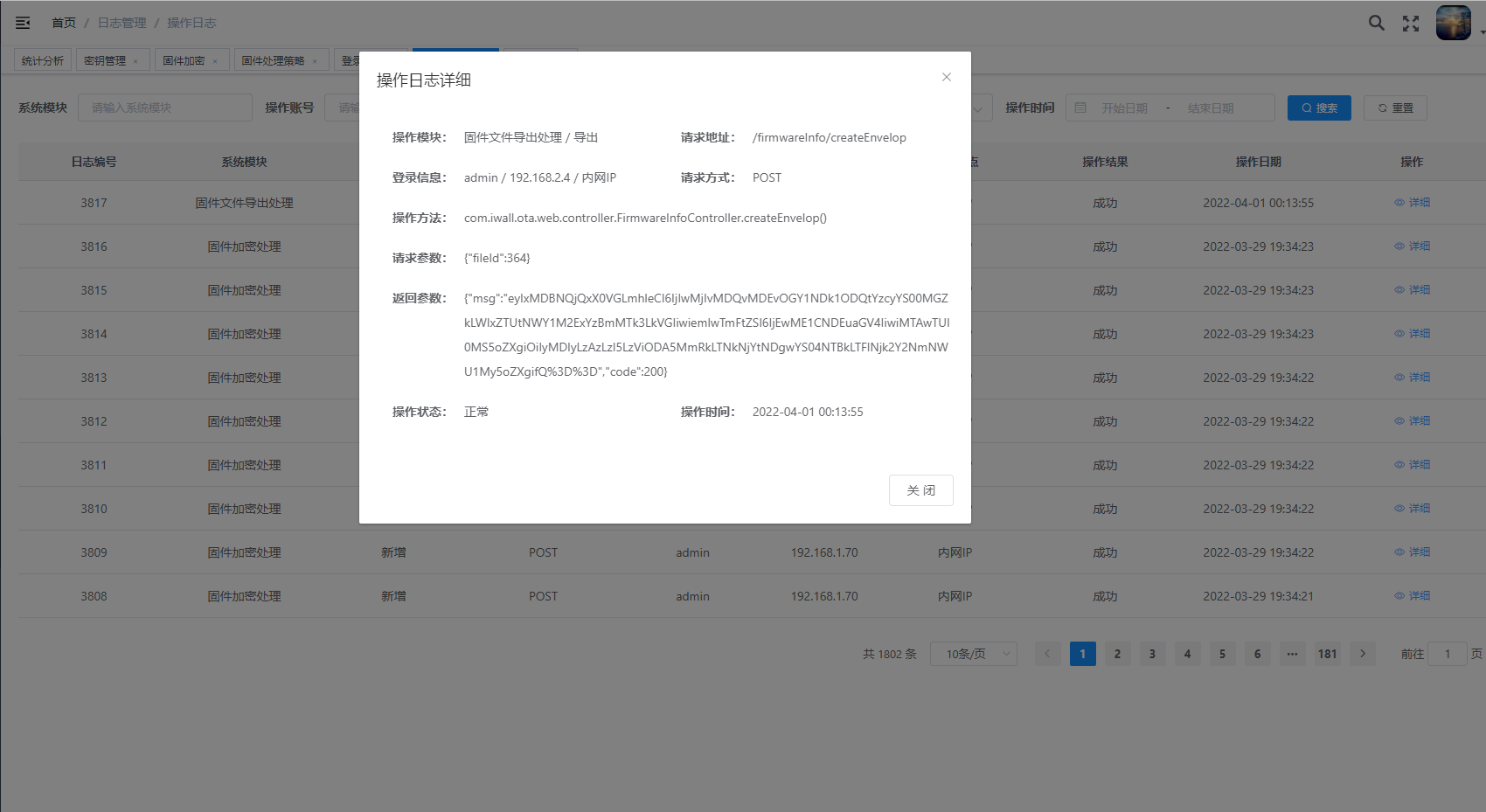
1. 涉及表和字段信息：



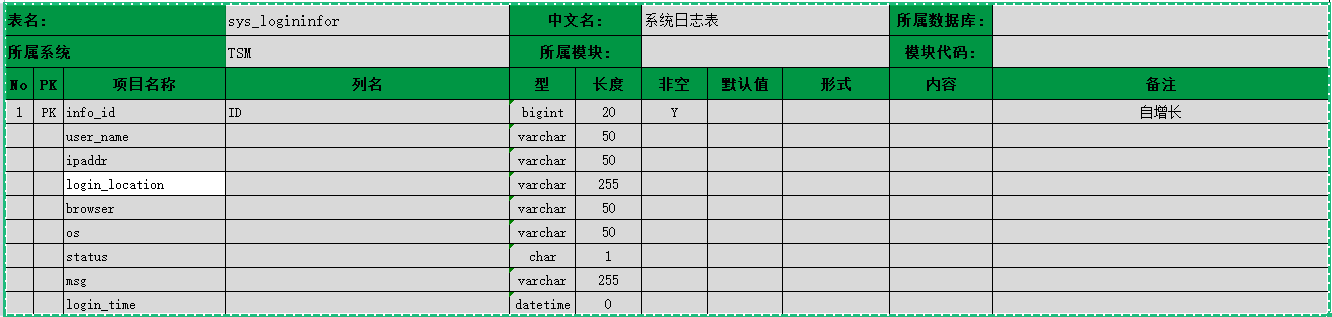
1. 实现思路 。对固件文件每一种操作（ 固件导入、固件加密[固件签名、固件加密]、固件导出、固件删除）均需要记录其操作痕迹。通过对各个文件操作步骤中的数据进行记录。涉及到固件加密【固件签名、固件加密】的，也存储固件的处理时间。

##### 查看日志

1. 页面设计



1. 涉及表和字段信息：



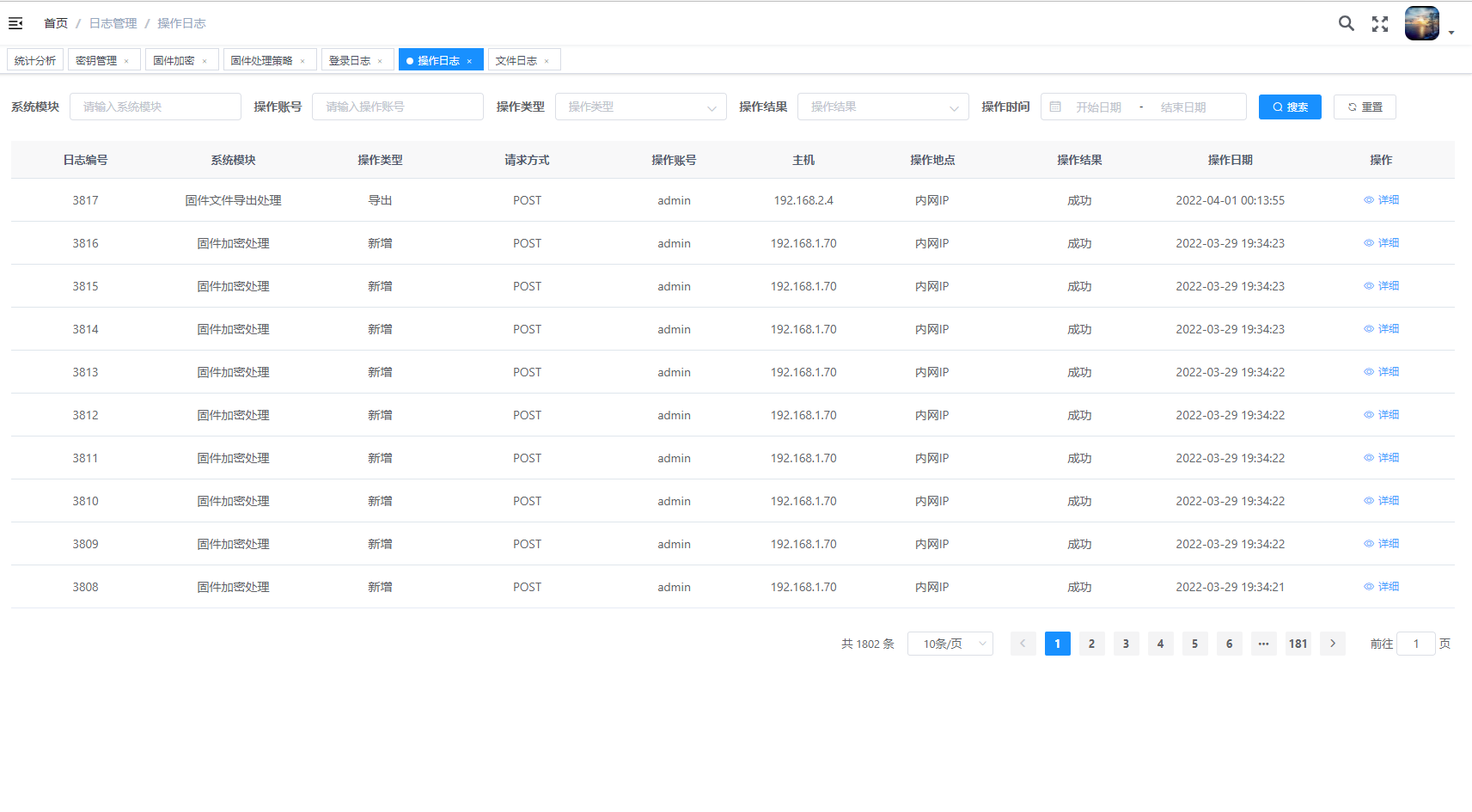




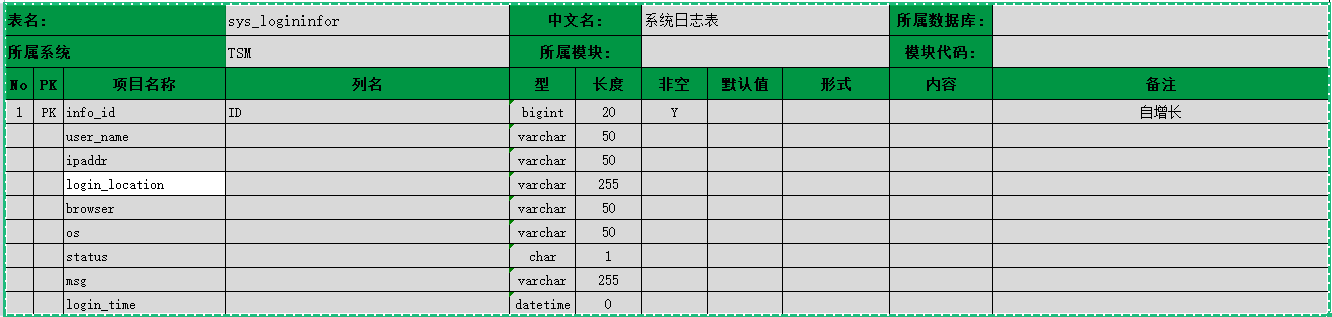
1. 实现思路。查看日志对已记录的日志内容进行详情查看，可看到的信息为日志处理涉及到的业务数据。通过点击某条日志记录，根据日志id触发后台服务请求， 服务器根据参数返回相应的日志存储记录，并通过前端组件渲染。

##### 查询日志

1. 页面设计



1. 涉及表和字段信息：





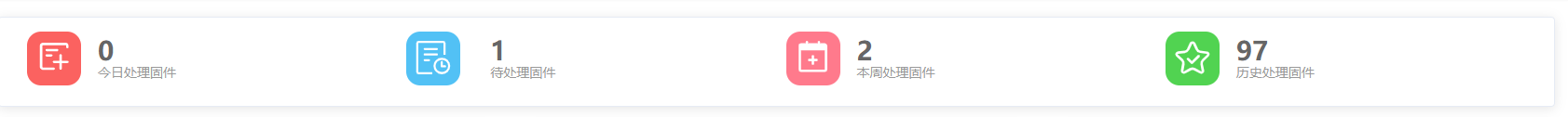


1. 实现思路。根据查询条件，展示指定条件下的日志内容。用户输入和选择对应的下拉选择框，点击查询按钮，前端组件触发调用服务器程序，后端服务器程序根据传入的条件，调用数据库，进行条件过滤，返回给前端组件并渲染呈现给用户。

#### **统计分析模块**

##### 固件处理数量统计

1. 页面设计



1. 涉及表和字段字段





1. 实现思路

统计系统今日处理固件数量、本周处理固件数量、历史处理固件数量、待处理固件数量。

实现思路：

今日处理固件数:按照操作历史中，处理时间当天的记录，在进入此系统页面后请求数据库服务，聚合统计后前端进行渲染。

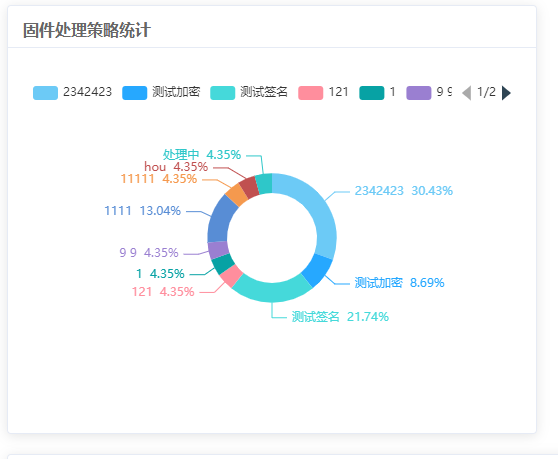
本周处理固件数量:按照操作历史中，处理时间为本周内的的记录，在进入此系统页面后请求数据库服务，聚合统计后前端进行渲染。

历史处理固件数量:按照操作历史中，处理时间为本周内的的记录，在进入此系统页面后请求数据库服务，聚合统计后前端进行渲染。

待处理固件数量：查询待处理固件数据表，为未处理的数据。

##### **固件处理策略统计**

1. 页面设计



1. 涉及到表

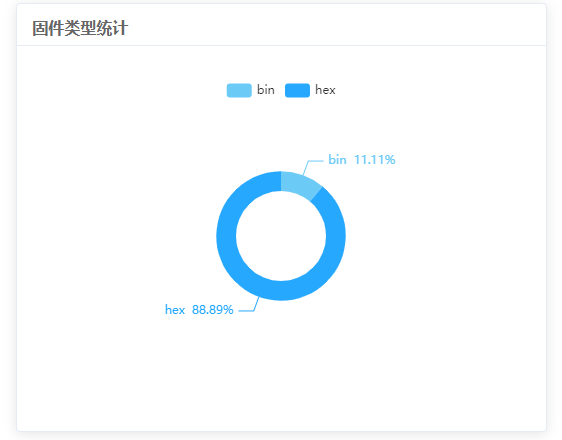




1. 实现思路：统计对比系统中各策略模版使用次数

##### **固件类型统计**

1. 页面设计



1. 涉及到表



1. 实现思路：统计对比系统处理的各种类型固件的数量

##### **业务成功率统计**

1. 页面设计
2. 涉及到的表和字段

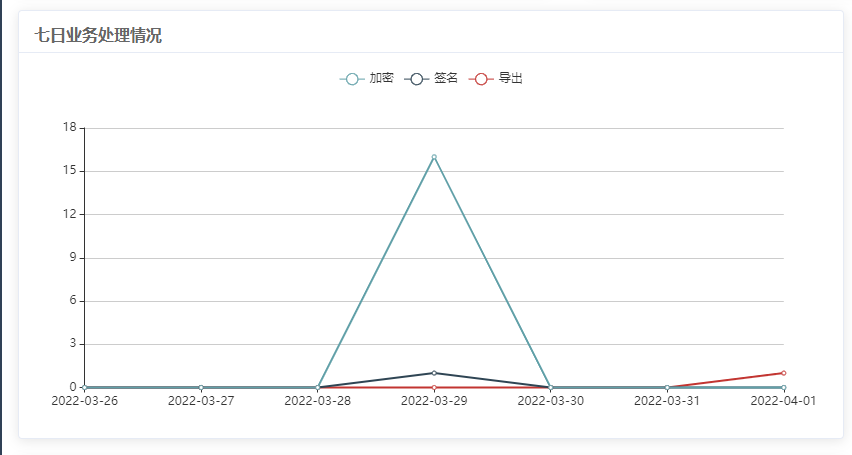


1. 实现思路：统计系统处理签名、加密、导出的成功及失败次数。对固件处理历史数据分类聚合搜索，计算处理成功数和总数的比率。

涉及到数据表

##### **七日业务处理情况**

1. 页面设计



1. 涉及到的表和字段



1. 实现思路：统计系统七日内处理。按照当前时间往前推算7日内数据，对固件处理历史数据分类聚合搜索。

##### **固件处理用时（单固件）**

1. 页面设计



1. 涉及到表:



1. 实现思路：统计系统中各个固件的处理耗时，对已经处理的固件数据记录，展示每个固件的用时消耗。

##### **固件处理用时（平均值）**

1. 页面设计



1. 涉及到数据表



1. 实现思路：统计系统处理加密、签名的平均耗时，对固件加密、固件签名分别取平均处理时间，然后通过ui界面展示。

##### **系统使用情况**

1. 页面设计



1. 涉及表和字段

无

1. 实现思路：查看系统磁盘、CPU、内存当前使用情况，用户在进入系统统计页面，页面发起远程请求，服务端程序调用系统能力获取磁盘、cpu、内存使用的数据，返回页面，页面渲染展示给操作用户。

##### **操作日志**

1. 页面设计



1. 涉及到数据表：



1. 实现思路

显示系统操作内容，前端页面加载时，调用后端服务器程序，服务器程序调用数据库引擎， 返回数据，在页面渲染展示。

### 数据库设计

#### 数据模型

**

#### 数据库结构设计

*详见OTA文件加密系统\_核心数据库设计\_基础版本.xlsx*

### 关键类设计

*无*

### UI/UE设计

*<https://app.mockplus.cn/run/prototype/mjGKNDW_SrYg/W35bUvlwF45u/MiVIC_HUj297Q?cps=collapse&ha=1>*

# 接口详细设计

详见《接口设计说明书》

# 数据迁移

无

# 非功能性设计

## 系统性能

系统应提供快速响应的能力，针对固件包导入、固件包加密、固件包导出等业务，并发性能需达到2000tps，响应时间小于1000ms。平均加密速率不小于100MB/s，平均签名速率不小于200MB/s。

## 资源使用率

在2000用户并发压力下，系统的CPU使用率不应超过70%，内存使用率不应超过80%。

## 信息安全

|  |  |  |
| --- | --- | --- |
| **需求属性** | **需求描述** | **备注** |
| 1强认证 | 系统使用本身的认证机制，需满足强密码策略和锁定策略，并在保证基础用户体验的情况下，选择合适的图形验证码和短信验证码； |  |
| 2 访问控制最小化 | 2.1 系统合理划分各种类型的权限管理，包括管理员、审计员、操作员，保证三权分立； | 必选 |
| 2.2 系统权限控制的管理粒度达到表单的角色级别（或单个用户的增删改查） | 明确系统权限的控制粒度，选择需求中一种或两种。 |
| 3 会话的安全性 | 3.1 系统会话标识随机并唯一，登录后重新分配，并对会话的失效周期进行定义 | 必选 |
| 4 日志安全审计 | 4.1 系统日志需以功能模块方式提供，日志记录覆盖每个用户，无论是内部用户或外部用户，记录应对重要的操作行为（登入登出、权限变更、数据导入导出等其他必要的操作）进行记录，并具有筛选功能；日志记录的内容应包含日期、时间、用户标识、行为的过程和结果等 | 必选，其他必要的操作行为记录可根据应用系统的不同业务定义，对业务审计有需求的内容，各编制人员可自行补充； |
| 4.2 日志记录的管理权限不应对基础用户提供，对开放给审计员，并禁止提供删除和修改的功能。 | 必选 |
| 5 软件容错 | 5.1 系统具备容错设计，提供可用性，对异常或错误的页面或功能进行跳转； | 必选 |
| 6 数据安全 | 6.1 应考虑系统数据的保密性和完整性，包括数据的传输加密https，敏感数据存储加密 | 必须 |
| 7 输入输出安全性 | 5.4.7.1 应用系统应对所有类型的用户输入做校验，尤其是那些用于数据库操作、系统命令执行、程序命令执行、显示到页面的重要参数，并记录拒绝后的错误日志。 | 必须 |
| 5.4.7.2 应用系统应对输出参数做严格的限制，保证非功能必须的参数不在输出的参数中，防止多余信息引起的数据泄露。 | 必须 |

## 可靠性

系统应保证可靠运行，可支持全天候24小时处理事务能力。系统可提供可靠的长时间工作，年故障停止服务次数少于 1次，故障停机时间小于120 分钟。通过系统级别定时任务监控唤醒应用服务。

## 易用性

提供良好的用户界面以方便用户操作。进行系统界面设计时，遵循KISS（keep it simple and stupid，简单易用）原则。

捕捉系统级和应用级的错误，包括用户的输入错误提供相关友好的出错提示页面。

## 互用性

系统采用java作为后端服务器程序编程语言，可以灵活的部署在LINUX、Windows等主流操作系统上。

## 扩展性

采用模块化的设计方法，每个模块之间相互独立性，功能模块的修改以及新增新的业务模块。

## 维护性

系统提供系统运行日志和业务日志，方便维护人员及时处理系统的问题。同时，应提供配置信息的页面修改功能，以方便维护人员对经常需要修正的相关配置信息进行修改。

## 合法/合规性

|  |  |  |
| --- | --- | --- |
| 法律法规名称 | 选择情况 | 备注 |
| WP29 R156 | 可选 |  |

# 附录

## 密钥算法清单

|  |  |  |
| --- | --- | --- |
| 算法名称 | 算法模式或者类型 | 密钥长度&IV长度 |
| SM4算法  （国密算法） | ECB模式  CBC模式  OFB模式  CFB模式  CTR模式  GCM模式 | 密钥：128Bit  IV：128Bit |
| AES128算法 | ECB模式  CBC模式  OFB模式  CFB模式  CTR模式  GCM模式 | 密钥：128Bit  IV：128Bit |
| AES192算法 | ECB模式  CBC模式  OFB模式  CFB模式  CTR模式  GCM模式 | 密钥：192Bit  IV：128Bit |
| AES256算法 | ECB模式  CBC模式  OFB模式  CFB模式  CTR模式  GCM模式 | 密钥：256Bit  IV：128Bit |
| SM3算法  （国密算法） | - | 结果：256Bit |
| SHA192算法 | - | 结果：192Bit |
| SHA224算法 | - | 结果：224Bit |
| SHA256算法 | - | 结果：256Bit |
| SHA384算法 | - | 结果：384Bit |
| SHA512算法 | - | 结果：512Bit |
| SM2算法  （国密算法） | Prime256 | 密钥：公钥512Bit 私钥256Bit |
| RSA算法 | \*RSA1024  RSA2048  RSA3072  RSA4096 | 密钥：公钥1024Bit 私钥1024Bit  密钥：公钥2048Bit 私钥2048Bit  密钥：公钥3072Bit 私钥3072Bit  密钥：公钥4096Bit 私钥4096Bit |
| ECC算法 | Prime192V1  Prime224V1  Prime256V1  Prime384V1  Prime521V1 | 密钥：公钥384Bit 私钥192Bit  密钥：公钥448Bit 私钥224Bit  密钥：公钥512Bit 私钥256Bit  密钥：公钥768Bit 私钥384Bit  密钥：公钥1042Bit 私钥521Bit |