# 文件加密存储系统设计文档

版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V0.1/创建 |  |  |  |  |
| V0.2/更新 |  |  |  |  |
| V0.3/发布 |  |  |  |  |

目录

[文件加密存储系统设计文档 1](#_Toc77946634)

[1 引言 3](#_Toc77946635)

[1.1 编写目的 3](#_Toc77946636)

[1.2 背景 3](#_Toc77946637)

[1.3 基线 3](#_Toc77946638)

[1.4 特殊名词定义 3](#_Toc77946639)

[2 总体设计 4](#_Toc77946640)

[2.1 系统框图 4](#_Toc77946642)

[2.2 功能模块 5](#_Toc77946643)

[2.2.1 用户模块 5](#_Toc77946644)

[2.2.2 客户端分析模块 5](#_Toc77946645)

[2.2.3 数据分析模块 6](#_Toc77946646)

[2.2.4 文件AES加解密模块 6](#_Toc77946647)

[2.2.5 密钥RSA加密模块 6](#_Toc77946648)

[2.2.6 密钥RSA解密模块 6](#_Toc77946649)

[3 数据的组织结构 7](#_Toc77946650)

[4.1 数据加密流程 8](#_Toc77946653)

[4.2 数据解密流程 9](#_Toc77946654)

# 1 引言

## 编写目的

认识加密存储系统，指导开发人员进行编码。本文档预期读者为：系统设计者，系统开发者。

## 背景

IPFS是用于存储和访问文件、网站、应用程序和数据的分布式文件系统。但是并不关心数据内容的隐蔽性，为了让其安全性得到更高效的保证，需要一套更加完善的系统来加密文件的信息，该系统可以应用于多种储存场景。

## 基线

用户文件数据始终在去中心化的互联网上流通，是完全隐蔽且唯一的。

## 特殊名词定义

**KMS(Key Management Server)：**密钥管理服务，身份认证中心。

**MDMS(Metadata Management Server)**：元数据管理服务，记录文件CID、名称、指纹、文件密钥等。

**Encrypted file**：加密文件。

**File fingerprint**：文件指纹，文件内容的唯一标识。

**Raw file**：原始文件，即未经过加密处理的文件。

**文件密钥**：文件通过AES加密，文件密钥即提供给AES加密的字符串。

# 2 总体设计



## 系统框图

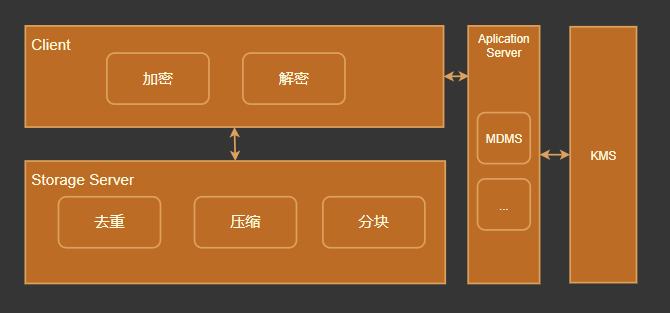


图1 系统框架图

系统遵循用户隐私性，加解密的过程都放在客户端中完成，也就是说只有用户有权知道他们的文件内容。

其中MDMS提供存储在StorageServer端的文件信息，一些关键信息如指纹、文件密钥等。KMS则负责管理所有的用户的公钥，以及服务自己的公钥和私钥。

数据会在Client端中计算出指纹信息，它是识别文件内容的唯一标识，然后请求到MDMS中辨别是否存在相同的文件，如果不存在则由MDMS请求到KMS拿到KMS的公钥返回给Client，Client端采用对称加密算法，得到一个文件密钥，同时将文件加密。将文件密钥二次加密，随后，文件发给存储端，文件信息以及文件密钥则由MDMS保管。

通过对文件密钥的二次加密我们可以将文件安全等级分成两级，由用户自己控制（去中心化），即将上传的文件安全性质级别：



图2 加密等级图

一级：使用对称加密算法AES密钥加密文件后，用KMS公钥对该密钥二次加密，其他用户访问时，只需要向MDMS申请访问即可。

二级：使用对称加密算法AES密钥加密文件后，用户的公钥对该密钥加密，再用KMS的公钥再次加密，该文件是非公开的，每次有其他用户想要对它解密，都需要经过上传者的同意。同样，如果丢失或修改的密钥，那么该文件的访问权就会永久丢失。

注：零级的情况即为不加密，公开的文件，也无需解密，是否保留该功能根据实际应用场景决定。

## 功能模块

### 用户模块

* 通过ssh-keygen生成RSA公钥与私钥
* 公钥通过User Server注册到KMS中
* 验证密钥有效性

### 客户端分析模块

* 鉴别用户客户端
* 指定客户端限制文件上传量级

### 数据分析模块

* 对文件计算指纹
* 在现有库中比对是否已存在

### 文件AES加解密模块

* 对上传的文件进行AES加解密

### 密钥RSA（ECC）加密模块

* 对AES加密的密钥多次RSA加密
* 分析用户加密等级
* 通过加密等级多次RSA加密AES密钥
* 上传到MDMS中

### 密钥RSA（ECC）解密模块

* 分析文件加密等级
* 1、通过加密等级MDMS请求KMS解密文件密钥2、等待Rawfile所有者确认访问权限

对加密二级以上文件记录用户访问痕迹

# 数据的组织结构

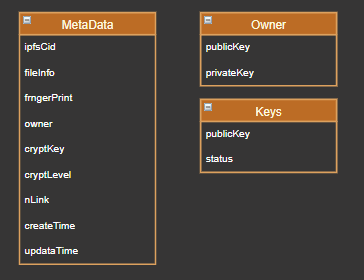


图3 数据组织结构图

**MetaData（元数据）：元**数据信息，保存文件CID、文件描述信息、指纹、所有者（公钥）、文件密钥、文件加密等级等，根据用户密钥有效性来判断是否需要在ipfs中pin。

**Owner（KMS自身密钥）：**存放KMS的公钥与私钥。

**Keys（公钥列表）：**存放所有使用系统的用户的公钥，以及该密钥状态（是否失效等）。



## 数据加密流程

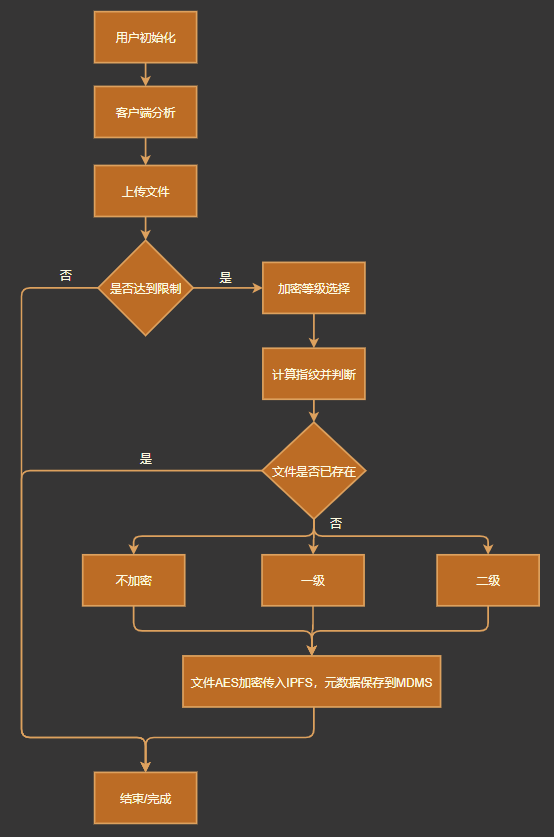


图4 加密流程图

数据加密基本流程如下：

1. 用户生成出一对RSA公钥与私钥。
2. 判断用户所使用的客户端，根据部分客户端对上传的文件应该要有所限制（比如浏览器最大只能4G的文件）。
3. 上传文件时候选择加密等级，然后计算指纹给予MDMS比对。
4. 如果不存在相同文件，返回KMS的公钥。
5. 生成随机字符串对文件采用AES加密，对该字符串二次加密KMS公钥，根据密钥等级对该字符串再次加密。
6. 文件发送给ipfs，元数据保存到MDMS中

## 数据解密流程

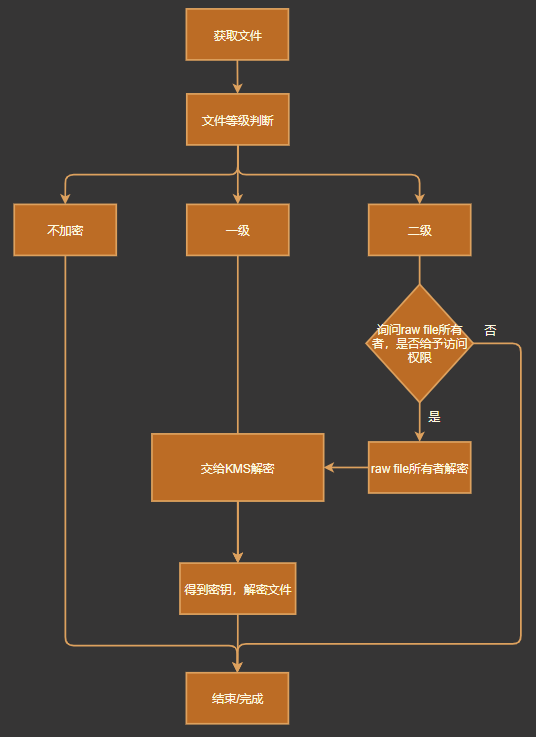


图5 解密流程图

数据解密流程如下：

1. 通过文件CID去MDMS中找到文件元信息
2. 根据加密等级，MDMS向KMS申请解密，KMS使用所有者的公钥对文件密钥解密，再用自身私钥对文件密钥解密，最后用访问者的公钥对文件密钥加密
3. 如果文件是二级加密，那么MDMS会先向文件所有者发起询问，等待所有者同意，即所有者使用自己的私钥对文件密钥解密，后回传给MDMS，MDMS再发给KMS执行后续流程