# PDX SurePKI设计说明书

版本：V1.0.0

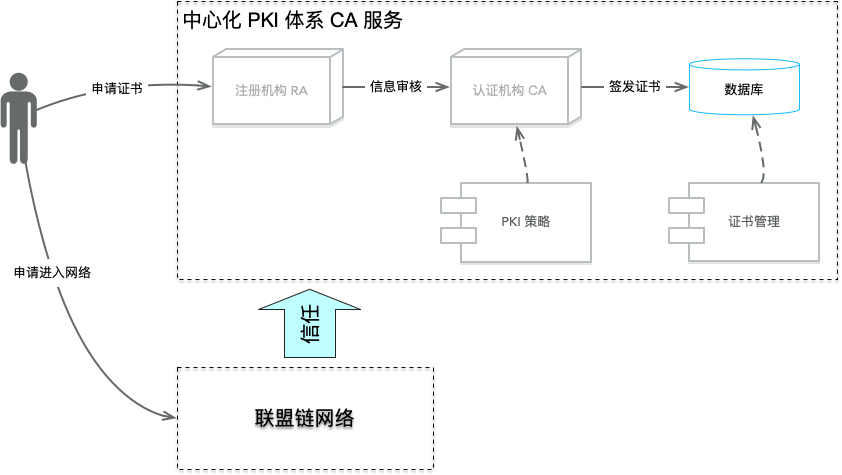
## 前言

在联盟链中需要的并非传统 CA 公共服务，是否允许谁进入联盟链网络通常是由联盟的相关管理机构决定的，传统 CA 公共服务的授权并不具备此信任基础，因此我们看到很多联盟链的方案中，CA 被设计成一个私有的独立的中心化服务，区块链节点使用数字证书鉴权并与 PKI 体系兼容，但需要依赖外部的 CA 服务；

为了最大程度的简化 CA 的开发、部署与维护，以及最大程度保留系统的去中心化自制和安全与稳定性，在 PDX Utopia 协议栈上将会提供去中心化 CA 服务。

本章将概述PDX Utopia 网络层为联盟链提供的更加优雅的 PDX SurePKI 解决方案。

## 现状

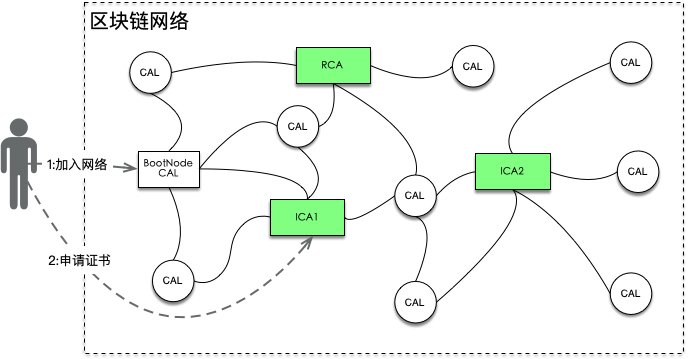


**图1**

图1 是目前在联盟链中流行的 CA 服务架构，搭建联盟链时需要额外部署一套 中心化CA服务，用户想加入联盟链网络需要先将证书请求发送到认证中心，如果联盟比较大，有多级认证体系时，可能需要搭建多个这样的中心服务，增加了部署和维护成本，而且需要联盟链节点与链外数据源进行交互，增加了不确定性和安全性问题；

搭建和使用一套系统是十分复杂的工作，需要专门的技术人员进行操作和维护，为了确保服务的稳定与可靠需要额外付出很多代价；

## 目标



**图2**

PDX SurePKI的目标是将提供一套更加简单和稳定的解决方案，将 CA 服务设计为标准模块嵌入在节点中，提供此服务的 RCA/ICA 节点与联盟中的其他节点属于对等关系，即所有节点都可以申请成为 CA 节点，为其组织内的普通节点提供证书的授权服务，上图中可以看到绿色的节点是符合规则的可以提供 CA 服务的角色，其信息会被放入 CAL 被同步到每一个对等节点中，此时的 CAL = [RCA, ICA1, ICA2]，CAL的准确性和一致性由区块链来保证；

当一个新的节点想要加入网络时，他可以任意找到一个节点作为 Bootnode 来进入网络并且获取 CAL ，遍历 CAL 信息找到其组织对应的 CA 并提出入网申请，在得到证书后即可与网络中的节点进行握手并升级为 peer 关系，之后便可同步账本使用区块链服务；

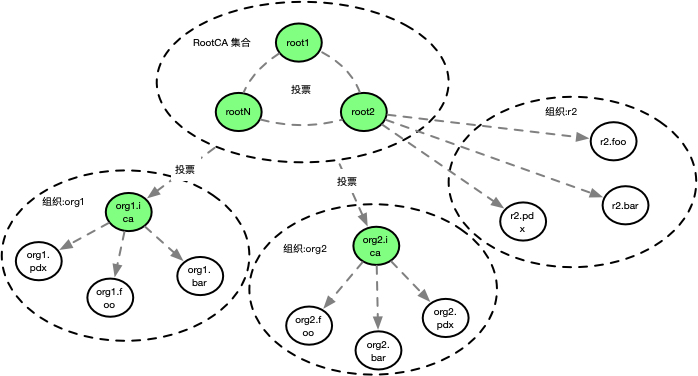
## 约定

**为完成 “目标” 我们需遵循如下约定，来使用数字证书和**PDX SurePKI**服务：**

* 联盟链只接受两级证书信任链，即一级RCA和第二级ICA；
* 二级中间证书不能再签发下一级中间证书，只能用于签发用户证书；
* 启动联盟链的第一个节点之前，要先生成根证书，并将根证书放入 genesis 区块中
* 用来生成证书的密钥使用 ECC 算法生成，并使用 P256 曲线
* 证书的 Subject.CommonName 需要填写正确的 ECC S256K1 公钥信息
* 证书模块会维护两个列表，CAL / CRL 分别代表合法的 CA 证书和已撤销的证书列表

## 治理规则

在网络中一些持有 CA 私钥的节点可以提供 CA 服务，在创世节点中可以指定初始的 RootCA 集合，如果想在 RootCA 集合中增加 RCA 角色节点，则需要全部 RCA 节点进行投票，每一个 ICA 节点负责签发其组织内的 CSR，在网络中增加 ICA 角色时，也需要全部的 RCA 节点进行投票；



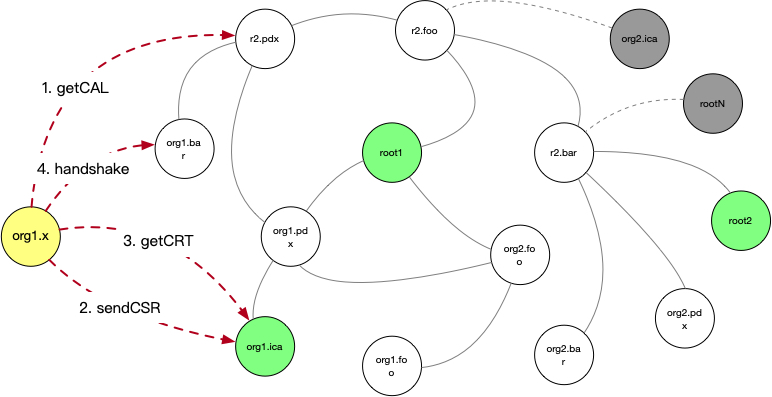
**图3**

图3 描述了上述规则，其中 RootCA 集合成员 root2 分管 “组织r2” ，org1.ica 和 org2.ica 是被 RootCA 投票产生的两个二级 CA 角色，分别管理 “组织org1” 和 “组织 org2”。

如果联盟初始状态没有那么复杂的组织关系，可以弃用投票规则，只需要在 genesis 中指定单根证书即可，当组织演化为有必要使用多个 RCA 时，只需要当前的 RCA 向 CAL 中添加新的 RCA 成员即可，再次委任新 CA 时因为 RCA 数量大于 1 ，即会激活投票规则。

***注：****图中的连线表示的是节点证书与 CA 之间的关系，并非连接关系；*

## 入网用例



**图4**

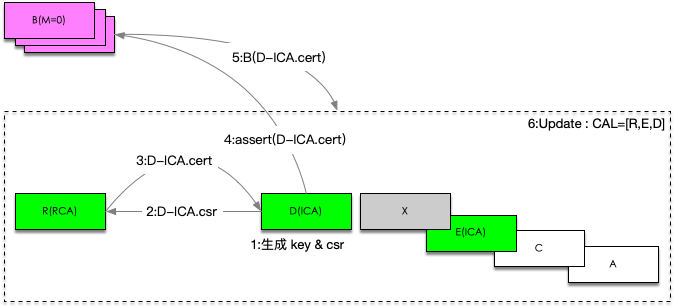
图4 中概述了新节点 org1.x（黄色） 节点想要加入联盟链网络的基本交互过程，我们假设此过程开始前节点已经完成了 “图2” 中描述的 “加入网络” 的步骤

1. *getCAL : 随机选择对等节点 r2.pdx 获取 CAL ，此时 CAL = [root1, root2, rootN, org1.ica, org2.ica]，当前节点属于组织 org1, 所以此处选择节点 org1.ica 提供服务；*
2. *sendCSR : 向目标 CA (org1.ica) 节点发送 CSR 申请证书；*
3. *getCRT : 收到审核通过的通知后，即可获取 CRT 证书；*
4. *Handshake : 得到合法的证书后去和其他节点进行握手，正式加入联盟。*

## 功能描述

### 授权CA证书

根证书的持有人可以签发下一级的CA证书，可以主动制作证书也可以由节点提出申请，下图的交互描述了节点 D 主动申请成为 ICA 的过程

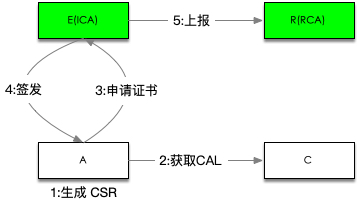


**图5**

1. 节点 D 使用 Utopia 节点提供的工具生成私钥和相应的 CSR;
2. 将 ICA.csr 发送给R (RCA) 进行申请，节点 R 的地址是通过 RCA 证书获取的;
3. 节点 R 收到 ICA.csr 后生成一个待处理的任务，需要人工参与审核，如果审核通过则签发 ICA.crt ，如果节点D在线则发送给节点D，否则等待节点D拉取证书；
4. 节点 D 得到正确的 ICA 证书后，需要向全网广播自己的 ICA 身份，通过 Assert 块进行广播，可以重复广播，直到同步到的 CAL 列表中包含自己为止；
5. 节点 B 此时为 M0 打块节点，将节点 D 广播过来的身份升级信息随 Commit 块广播到全网；
6. 每个收到新的 Commit 块的节点，都向自己的 CAL 列表中增加节点 D.

### 签发用户证书

节点启动时，只要使用了正确的 genesis 和 bootnode 就可以加入到联盟链的网络中，但是不能和 peer 进行握手，因为还没有得到正确的证书，需要完成下图描述的签发过程，才可以正式以 peer 身份加入网络



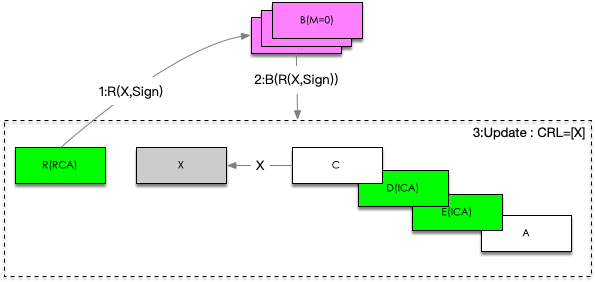
**图6**

1. 节点 A 启动时，节点发现 A 没有入网证书即生成 CSR 备用；
2. 通过 bootnode 接入网络，并随机找到了节点 C 去获取网络中最新的 CAL 列表;
3. 操作节点 A 的用户根据 CAL 列表的信息选择了向 ICA 节点 E 申请证书；
4. 节点 E 收到 A 的 CSR 后生成一个待处理的任务，需人工参与审核，审核通过则将 crt 发送给节点 A，或等待A来拉取;
5. 节点 E 将新签发的证书信息上报给 RootCA 节点 R.

### 撤销证书

撤销证书的约定是根证书有权撤销所有证书，包括中间证书，中间证书只能撤销自己签发的用户证书；更常见的撤销请求是由用户主动发起，例如将节点私钥丢失或者需要变更时，旧的证书则可进行撤销操作

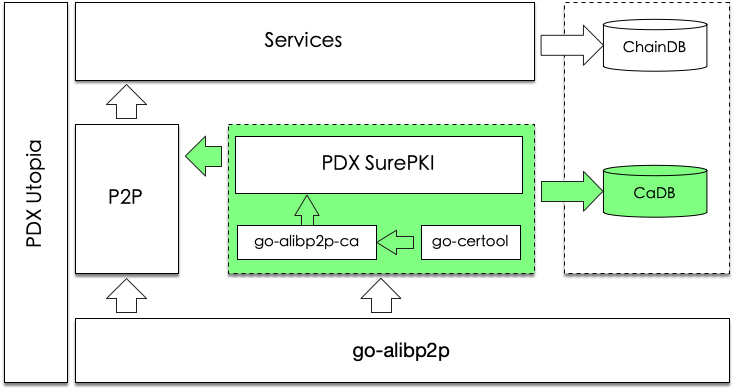
无论如何撤销的结果都是由CA节点以Assert的方式通知给打块节点，并随commit块同步到每个节点；下图演示了RCA节点R撤销节点X的用户证书的过程：



**图7**

1. 节点 R 可管理全网的证书，假设 X 节点用户证书需要撤回，则 R 节点签发一个 CRL 请求，并以 Assert 方式发送给节点 B;
2. 节点 B 此时为 M0 打块节点，将节点 R 广播过来的撤销信息随 Commit 块广播到全网；
3. 每个收到新的 Commit 块的节点，都向自己的 CRL 列表中增加节点 X.
4. 每个节点更新完自己的 CRL 后都重新检查一遍自己的连接中是否有节点存在于 CRL 中，此时节点 C 发现 X 和自己处于 peer 关系，则主动解除 peer 关系.

## 架构说明



**图8**

架构图摘取了与 PDX SurePKI 相关的模块进行描述，该模块被设计成可插拔式的独立模块，依赖于底层的 alibp2p 网络，同时为 p2p 模块提供服务，数据层使用了独立的 CaDB 数据库进行必要的持久化操作.

## 名词解释

|  |  |
| --- | --- |
| **名词** | **解释** |
| PKI | Public Key Infrastructure 公钥基础设施，一个典型的PKI系统包括PKI策略、软硬件系统、证书机构CA、注册机构RA、证书发布系统和PKI应用等。 |
| CA | Certificate Authority证书颁发机构,即颁发数字证书的机构。是负责发放和管理数字证书的机构。 |
| RootCA / RCA | Root Certificate Authority持有根证书的证书颁发机构 |
| ICA | Intermediate Certificate Authority 持有中间证书的证书办法机构，中间证书由根证书签发 |
| CSR | Certificate Signing Request 证书签发请求 |
| CAL | Certificate Authority Lists 证书办法机构列表，包括根证书和中间证书 |
| CRL | Certificate revocation Lists 证书废除列表，又称证书黑名单 |
| Bootnode | 网络中任何一个节点都可以充当 Bootnode 角色，相当于一个介绍人，介绍一个新的节点进入网络 |
| genesis | 创世区块，每个链都有其独一无二的创世区块 |
| peer | 在PDX Utipia 的网络中节点间的关系分两种，一种是连接关系，一种是 peer 关系，当节点提供正确的 networkid 后即可确定连接关系，想升级为peer关系需要提供额外的握手条件 |
| assert | assert消息定义在 PDX Utopia 共识协议中，用来传递共识相关的消息给委员会中的打块节点 |
| commit | Commit 消息定义在 PDX Utopia 共识协议中，主要用来确认两个 Commit 区块之间的 Normal 区块，并提供有关共识的证明信息 |