中国科学: 信息科学 2022 年 第 52 卷 第 6 期: 1135–1148
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
论文

基于区块链和去中心不可否认属性签名的分布式公钥基础设施方案

袁和昕[1,2], 刘百祥[1,2], 阚海斌[1,2,3*], 陈泽宁[1,2]

1. 复旦大学计算机科学技术学院, 上海市智能信息处理重点实验室, 上海 200433
2. 上海市区块链工程技术研究中心, 上海 200433
3. 电磁波信息科学教育部重点实验室, 上海 200433
* 通信作者. E-mail: hbkan@fudan.edu.cn

收稿日期: 2021–05–20; 修回日期: 2021–06–29; 接受日期: 2021–07–08; 网络出版日期: 2022–06–01

国家重点研发计划 (批准号: 2019YFB2101702)、国家自然科学联合基金重点项目 (批准号: U19A2066)、上海市科技创新行动计划 (批准号: 20222420800, 20511102200) 和广东省重点领域研发计划 (批准号: 2020B0101090001) 资助项目

摘要 灵活有效的身份体制方案一直是信息时代的核心需求之一. 传统的中心化公钥基础设施存在诸多缺陷, 而已有的运行在区块链上的分布式的公钥基础设施同样存在诸如性能、鲁棒性、不可否认性、身份灵活性等方面的问题. 本文创新地将区块链与去中心不可否认属性签名结合, 提出一种新型分布式公钥基础设施方案, 方案利用门限算法和属性签名对身份进行细粒度的管理, 并引入非交互式零知识证明使得证书具有不可否认的性质, 利用区块链的共识机制进行证书库的同步以实现分布式的身份认证. 本文通过实验仿真和分析并结合具体场景, 证明了该方案在安全性和可用性上都满足实际落地的需求.

关键词 区块链, 公钥基础设施, 属性签名, 门限算法, 零知识证明, 身份认证

1 引言

灵活有效的身份认证/管理方案一直是信息时代的核心需求之一, 通过灵活有效的身份认证/管理方案, 我们可以唯一确定互联网中每个实体的身份. 公钥基础设施 (public key infrastructure, PKI) 是典型代表之一, PKI 通过管理数字证书, 从而能够解决不同实体之间的信任问题, 是当前互联网的重要基石之一. 然而传统的中心化 PKI 存在诸多问题, 其中最大的缺陷是 CA (certificate authority) 必须完全可信, 当 CA 被攻击或者 CA 自己就是作恶节点时, 其颁发证书对应实体的身份要么无法认证、要么不可信, 这样会对互联网的身份认证体制造成冲击.

区块链的去中心化、用于同步的共识机制、防篡改等性质, 对 PKI 的发展提供了新思路, 针对中心化的身份认证体制存在的问题, 许多方案 [1~5] 将传统的 CA 布置在区块链的多个节点上以实现分

引用格式: 袁和昕, 刘百祥, 阚海斌, 等. 基于区块链和去中心不可否认属性签名的分布式公钥基础设施方案. 中国科学: 信息科学, 2022, 52: 1135–1148, doi: 10.1360/SSI-2021-0177
Yuan H X, Liu B X, Kan H B, et al. Distributed public key infrastructure scheme based on blockchain and decentralized undeniable attribute-based signature (in Chinese). Sci Sin Inform, 2022, 52: 1135–1148, doi: 10.1360/SSI-2021-0177

布式的认证. 区块链的引入带来了很多好处: 第一, 区块链的共识机制原生支持多节点的数据同步, 可以利用运行在区块链上的高级编程语言智能合约将相关数据存储于区块链, 这样用户可以在多节点下进行证书申请与查询; 第二, 区块链因其去中心化与不可篡改的性质, 其下的通信具有信任基础, 不同用户可以进行安全的信息交互.

但是已有的方案仍然存在一些问题. 文献 [2] 引入了密码累加器来实现对公钥的快速验证, 这样提升了查找性能, 却大量增加了存储开销. 文献 [5] 针对现有 PKI 存在的中间人攻击、CA 抗攻击不足、恶意证书处理问题展开研究, 却忽略了计算开销. 文献 [1] 通过门限算法实现了多 CA 协同处理与验证, 避免了传统 PKI 单点 CA 作恶的风险, 但 CA 数量是固定的, 忽略了 CA 宕机的问题, 具有较低的鲁棒性. 文献 [3] 保障了用户之间的信任关系, 却忽略了证书的不可否认性, 存在伪造证书的可能. 文献 [6] 引入了不可否认签名的概念, 其最本质的性质是无签名者的合作下不可能验证签名, 这样一是可以防止 CA 随意签发假证书, 二是可以防止证书申请者对证书进行抵赖 (例如证书内容在某个场景对自己不利, 他否认这个证书并咬定是 CA 颁发的假证书), 这一概念最新的解决方案是零知识证明.

基于属性的密码学, 由于可以提供细粒度的灵活的访问控制, 亦为 PKI 提供了新的发展方向 [7~10]. 身份可以由一组属性组成, 只要用户的属性集与所要求的属性集的误差在一定范围都可以认为是认证成功. 属性密码学的引入使得认证实体身份更为立体, 实体的身份可以由多种属性构成 (例如标识信息、组织关系等), 更符合真实世界的身份机制 [7]. 在分布式公钥基础设施与属性签名配合使用的研究领域, 笔者暂时未找到同类文献.

本文创新性地提出了一种新型的基于区块链和去中心化不可否认属性签名的分布式公钥基础设施方案, 对现有的方案进行了改进, 具有以下特点:

• 通用. 本文设计的是通用的方案, 适用于各种场景, 本文以智慧城市下实体间的身份认证场景为例, 结合实验, 具体说明了本方案的通用性.

• 性能良好. 本文借鉴文献 [7] 所构造的高效属性签名方案, 并使用了效率更高的非交互式零知识证明 (zero-knowledge succinct non-interactive arguments of knowledge, zk-SNARKs). 经过实验测试, 本方案在计算开销、通信开销、并发方面都满足实际落地需求.

• 身份机制灵活且细粒度. 本方案利用属性所涵盖的广度使得身份更为立体, 同时使得 CA 数量可以灵活动态的增加, 并结合门限签名算法, 使整个方案具有细粒度的身份认证以及一定的容错性.

• 不可否认性. 本文通过在证书申请者侧引入全局唯一的 ID, 并结合非交互式零知识证明实现了证书的不可否认性, 在提升性能的同时, 还使得验证双方无需进行交互式的验证, 扩展了本方案的应用广度.

2 预备知识

2.1 分布式一致性

分布式一致性, 即在分布式系统中, 如何保证系统内的各个节点之间数据的一致性或能够就某个提案达成一致, 是所有分布式系统都必须面对的问题. 在分布式公钥基础设施的场景中, 证书库需要在多个节点进行同步.

传统的分布式一致性算法主要是 Paxos [11] 以及其变种 (如 Raft [12] 等), 传统的分布式一致性算法往往只考虑节点宕机的情况而没有考虑节点作恶, 区块链引入的共识机制大多基于拜占庭 (Byzantine)

容错, 例如比特币的 PoW, Hyperledger Fabric 的改版 Raft. 我们可以利用区块链来存储并同步证书库与证书吊销列表 (certificate revocation list, CRL).

2.2 门限签名方案

本文的 (t,n) 门限签名方案借鉴了文献 [7] 所构造的 BTS (basic threshold signature) 方案. 具体的, 选择文献 [13] 的 (t,n) 门限方案作为其秘密分享方案, 使用文献 [14] 的 Schnorr 协议作为其基础的 Σ 协议, 再使用文献 [15] 的 Fiat-Shamir 转换即可得到此签名方案, 具体如下所示:

(1) Setup. 选择一个循环群 G, 其素数阶 $N=p$, 则公共参数为 GP, 包含 $N=p$ 和 G 的生成元 g. 另外还需要选择 1 个哈希 (Hash) 函数 $H_2:\{0,1\}^*\times G^{n+1}\to Z_p^*$.

(2) Gen. 随机生成的秘密值 $s_i\in_R Z_p^*$ 作为私钥, 并计算公钥:

$$Y_i=g^{s_i},\ i=1,\ldots,n.$$

(3) Sign. 不妨假设签名人拥有前 t 个私钥, 对消息 $m\in\{0,1\}^*$ 进行签名.

(i) 对于 $i=1,\ldots,t$, 随机选取 $t_i\in Z_p^*$, 对于 $i=t+1,\ldots,n$, 则随机选取 $c_i,d_i\in Z_p^*$, 计算

$$R_i=\begin{cases}g^{t_i}, & i=1,\ldots,t,\\ g^{d_i}Y_i^{c_i}, & i=t+1,\ldots,n.\end{cases}$$

(ii) 计算

$$c=H_2(m,R_1,\ldots,R_n).$$

(iii) 用 $n-t+1$ 个点 $(0,c),(t+1,c_{t+1}),\ldots,(n,c_n)$ 构造 $n-t$ 次拉格朗日 (Lagrange) 插值多项式 $P_{n-t}(x)$:

$$P_{n-t}(x)=\sum_{i=1}^{n-t+1}P_{n-t}(x_i)\prod_{j=1,j\neq i}^{t}\frac{x-x_j}{x_i-x_j}.$$

(iv) 计算

$$c_i=P_{n-t}(i),\ d_i=t_i-c_is_i,\ i=1,\ldots,t.$$

(v) 输出多项式 $P_{n-t}(x)$ 和签名:

$$\sigma=\langle c_i,d_i,R_i\rangle,\ i=1,2,\ldots,n,$$

其中多项式也可不传, 可根据签名中的参数重构, 但会增加验证者的计算量.

(4) Verify. 验证

$$\begin{cases}P_{n-t}(x)\text{ is a polynomial of degree }\ (n-t),\\ P_{n-t}(i)==c_i,\ i=1,\ldots,n,\\ R_i==g^{d_i}Y_i^{c_i},\ i=1,\ldots,n.\end{cases}$$

以上如果有一个不通过, 则签名为无效签名, 否则验证人接受此签名.

2.3 属性签名

基于属性的加密 (attribute-based encryption, ABE) 由 Sahai 和 Waters [16] 共同提出, 属性加密是基于身份加密的扩展, 在分布式环境下有巨大应用潜力 [9], 属性加密可以提供细粒度和灵活的访问控制, 因而广受关注.

随着 ABE 的发展, 基于属性的签名 (attribute-based signature, ABS) 也开始进入学界的视线. 2008 年, Maji 等 [17] 给出了 ABS 的形式化定义. 如同 ABE, 在 ABS 中每个用户拥有一系列属性, 用户可以从属性的权威机构获得属性, 随后用户可以用属性私钥与属性谓词 (属性集的结构) 进行签名, 只要用户的属性集合满足签名谓词, 签名即是有效的.

2.4 零知识证明与 zk-SNARKs

零知识证明 (zero-knowledge proof), 指的是证明者向验证者证明并使其相信自己知道或拥有某一消息, 但证明过程不能向验证者泄漏任何关于被证明消息的信息, 一个零知识证明系统应该保证正确性 (soundness)、完备性 (completeness)、零知识性 (zero-knowledge) 3 个要求, 系统可以分成交互式和非交互式, 交互式零知识证明系统需要验证者与证明者之间进行多次交互. 后者不需要进行多次交互, 证明者只需要将其生成的证明公布出来, 任何验证者都可以进行验证. 针对区块链下的证书认证场景, 非交互式零知识证明显然更有优势, 一是相关证明可以上传到链上, 这样可以提高认证的效率与安全性; 二是很多验证实体根本不具备交互认证所需的各种数学运算的条件.

在本文构造的场景下, 使用 Fiat-Shamir Heuristic [15] 的非交互零知识证明可以证明两个离散对数相等而不泄露离散对数具体值.

zk-SNARKs 在非交互式零知识证明的基础上做了一些优化, 减少了证明的大小与验证时间, 这样一是很大程度减少了计算开销, 二是因为证明大小的优化可以减少通信开销. zk-SNARKs 目前已经有很多开源的实现, 如 ZoKrates [18], 这些开源实现的性能已经满足了本方案的需求, 所以本文的系统构建也将使用这些开源的实现.

3 系统模型

本节会给出系统模型的定义, 包括符号说明、通用流程与证书格式.

3.1 符号说明

基于区块链和去中心不可否认属性签名的分布式公钥基础设施方案 (DPKI) 的参数描述如表 1 所示, 同时本方案包括如下几种角色:

• User. 为 DPKI 的普通节点, 相互通过证书进行身份验证.

• $\text{User}_{\text{Applicant}}$. 该节点可以向属性权威机构申请属性, 也可将证书所需的字段发送给 RA (registration authority) 节点以进行证书申请.

• $\text{User}_{\text{Verifier}}$. 该节点可以向属性权威机构申请属性, 也通过 DPKI 来验证证书所有者的信息, 以此断言证书所有者的身份.

• RA. 该节点是确保证书的有效性和证书正确注册的注册机构, RA 节点作为 DPKI 与外界交互的通道, 负责接受对证书的请求并对发出请求的 User 进行身份验证, 验证方式包括但不限于线下认证. RA 节点数量是不定的并且可动态扩展的.

表 1 符号及其描述

Table 1 Symbol and description

Symbol	Description	Symbol	Description
G	Cyclic group	p	Order of G
g	Generator of G	Attr_i	Attribute or CA
M, m	Certificate and required information	CSK, CPK	CA's private and public key
UID_u	User's global identity	S_u	Parameter of UID_u
zkParams	zk-SNARKs global parameters	Ω	Total attributes
ASK, APK	User's attribute private and public key	Params	DPKI global parameters
H_1, H_2	Hash functions	e_i, s_i, t_i, R_i, Y_i	Temporary variables
π_u	Proof of zk-SNARKs	w	Parameters of zk-SNARKs
x	Proposition of zk-SNARKs	M_L	Turing machine algorithm
c_i, d_i, T_i	Parameters of signature	σ	Signature

• CA. 该节点是颁发证书的实体之一, CA 节点代表某一个属性的权威机构, 具有某一属性的公私钥对, 可对 User 的属性申请进行审批. 多个 CA 节点协同对证书内容进行签名, 他们共同充当受信任的第三方, 由 $\mathrm{User}_{\mathrm{Verifier}}$ 与 $\mathrm{User}_{\mathrm{Applicant}}$ 进行信任. CA 节点数量是不定的并且可动态扩展的.

3.2 通用流程

本方案设计的是通用的基于属性签名的分布式公钥基础设施, User 可以通过公开的 API (application programming interface) 任意申请本系统的 CA 机构颁发的证书以及对证书进行验证. 本方案以区块链和智能合约作为载体, 实现了以去中心不可否认属性签名为基础的分布式公钥基础设施体系框架, 该框架结构如图 1 所示, 图中节点作为系统的基本单位, 根据其在流程中执行操作的不同而承担不同的角色.

本方案以区块链和智能合约作为载体, 由于区块链节点与节点公钥一一对应的特性, 能让节点间通过智能合约发起安全的秘密通信, 同时所有节点可以通过智能合约获取区块链上的数据 (包括但不限于属性签名与零知识证明的公共参数), 所有节点需将初始化产生的公钥、加密产生的密文等内容公开上链, 所有区块链上的数据 (包括证书库等) 会通过区块链的共识机制进行同步, 本方案的具体流程为:

(1) $\mathrm{GlobalSetup}(\lambda) \to \mathrm{Params}$. 系统初始化去中心不可否认属性签名的相关参数化并公开上链.

(2) $\mathrm{CASetup}(\mathrm{Params}) \to \mathrm{CSK}, \mathrm{CPK}$. 代表属性的权威机构的 CA 节点的初始化, CA 随机生成属性的私钥 CSK, 并由私钥计算出公钥 CPK, 将 CPK 等信息公开上链.

(3) $\mathrm{ZKSetup}(1^n) \to \mathrm{zkParams}$. 系统初始化 zk-SNARKs 的相关参数并公开上链.

(4) $\mathrm{USetup}(\lambda_u) \to S_u, \mathrm{UID}_u$. User 的初始化, 除了注册区块链所需的信息, $\mathrm{User}_{\mathrm{Applicant},u}$ 还需要随机生成秘密值 S_u, 使得所计算的 ID 标识 UID_u 是区块链上全局唯一的.

(5) $\mathrm{UAttrSetup}(\mathrm{UID}_u, \mathrm{Attr}_i) \to \mathrm{ASK}_{u,i}, \mathrm{APK}_{u,i}$. User 向若干个 CA 节点申请属性, 获得 User 的属性私钥与公钥 $\mathrm{ASK}_{u,i}, \mathrm{APK}_{u,i}$.

(6) $\mathrm{Sign}(m, S_u, \mathrm{UID}_u, \Omega, \{\mathrm{APK}_{u,i}, \mathrm{ASK}_{u,i}\}, \{\mathrm{CPK}_i, \mathrm{CSK}_i\}) \to P_{n-t}(x), \sigma$. 期望申请特定 n 个 CA 签发证书的 $\mathrm{User}_{\mathrm{Applicant},u}$ 拥有 n 个属性中的若干属性, 他通过区块链网络向 RA 发送入网请求证书服务, $\mathrm{User}_{\mathrm{Applicant},u}$ 提交证书所需的各种信息, RA 通过各种方式 (包括但不限于线下认证) 确认信息,

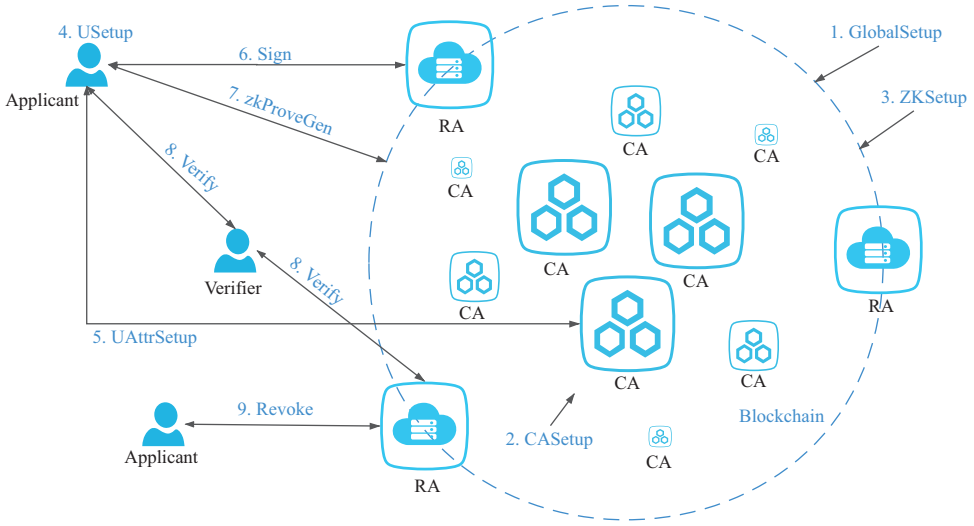

图 1　(网络版彩图) DPKI 框架

Figure 1　(Color online) DPKI framework

如果信息有误, 则拒绝该节点请求, 否则将信息发送给 n 个 CA, CA 协同对证书进行签名, RA 收集签名消息生成签名, 并存储于区块链的证书库中, 区块链会将证书库自动同步.

(7) $\text{zkProveGen}(\text{zkParams}, w, x, M_L) \to \pi_u$. $\text{User}_{\text{Applicant},u}$ 执行 zk-SNARKs 的生成证明算法, 生成证书的不可否认的证明.

(8) $\text{Verify}(m, \sigma, P_{n-t}(x), \pi_u) \to$ True/False. 期望验证 $\text{User}_{\text{Applicant},u}$ 身份的 $\text{User}_{\text{Verifier},s}$, 他对 $\text{User}_{\text{Applicant},u}$ 发起验证请求, $\text{User}_{\text{Applicant},u}$ 首先去 RA 获取自己的证书, 并将证书发送给 $\text{User}_{\text{Verifier},s}$, $\text{User}_{\text{Verifier},s}$ 对证书进行验证. 首先看证书是否是合法的 DPKI 的 CA 签发, 查看 $\text{User}_{\text{Applicant},u}$ 的证书是否被撤销, 查看证书的有效期限并查看是否是所要求的 n 个 CA 节点签发的; 如果证书已过期、被撤销、或者证书不是合法的节点签发, 则身份认证失败, 否则对证书进行认证, 验证包括签名验证与零知识证明 zk-SNARKs 验证, 验证通过即验证成功 $\text{User}_{\text{Applicant},u}$ 的身份.

(9) $\text{Revoke}(M, \pi_u, \text{UID}_u) \to$ True/False. 证书撤销分为过期撤销与 $\text{User}_{\text{Applicant}}$ 主动撤销: 证书吊销列表 (CRL) 会被 RA 周期性的更新; $\text{User}_{\text{Applicant},u}$ 如果向 RA 发送关于自身证书撤销的请求, RA 验证零知识证明以确认 $\text{User}_{\text{Applicant},u}$ 身份, 验证通过将证书添加入 CRL.

3.3　证书的结构

本方案设计的证书借鉴了 X.509 证书格式, X.509 是基于公钥加密和数字签名的证书标准, 该标准下用户的证书必须由专业的权威机构 CA 签发, 证书和公钥的管理集中在 CA. X.509 的证书结构无法满足本方案对证书的要求, 我们需要对一些字段进行修改: (1) 由于本方案对实时性、动态性的要求, 证书结构不能太过复杂; (2) 本方案基于去中心不可否认签名, 需要相关字段进行签名的支撑. 部分证书字段如表 2 所示.

4　分布式公钥基础设施方案

本节阐述一个基于去中心化不可否认属性签名的分布式公钥基础设施方案, 设计思路借鉴了文

表 2 证书结构

Table 2 Certificate structure

Field	Description	Field	Description
Version number	Certificate version	ABSUID	Unique identification of the owner
Serial number	Certificate serial number	ABSAttributes	Attributes set
Signature algorithm ID	Signature algorithm (attribute-based)	ABSSignature	Attribute-based signature
Issuer name	The issuer	...	...
Validity period	Not valid after this date		

献 [7] 的设计方案, 采用了 2.2 小节所描述的门限签名方案作为 BTS 方案, 并在其上融合了零知识证明 zk-SNARKs 进行非交互验证. 方案具体为以下几个阶段.

4.1 系统初始化阶段

(1) GlobalSetup(λ) $\rightarrow$ Params. 选择一个循环群 G, 其素数阶 $N=p$, 其生成元为 g. 对应分布式公钥基础设施的 n 个 CA 节点, 我们有属性总体 $\Omega=\{\mathrm{Attr}_1,\mathrm{Attr}_2,\ldots,\mathrm{Attr}_n\}$, 另外还需要选择 2 个哈希函数 $H_1:\{0,1\}^*\times G^3\rightarrow Z_p^*$, $H_2:\{0,1\}^*\times G^{n+1}\rightarrow Z_p^*$. 将公共参数 $\mathrm{Params}=\langle G,p,g,\Omega,H_1,H_2\rangle$ 打包上传至区块链.

(2) CASetup(Params) $\rightarrow \mathrm{CSK},\mathrm{CPK}$. n 个 CA 节点, 即属性的权威机构 Attr_i, 随机生成自己的私钥 $\mathrm{CSK}_i\in {}_RZ_p^*$, 并计算出公钥 $\mathrm{CPK}_i=g^{\mathrm{CSK}_i}$, 并将 CPK_i 公开上链.

(3) ZKSetup(1^n) $\rightarrow$ zkParams. 通过 zk-SNARKs 自带的 Setup(1^n) 算法进行公共参数 zkParams 的初始化, 并将 zkParams 公开上链.

4.2 用户初始化阶段

(1) USetup(λ_u) $\rightarrow S_u,\mathrm{UID}_u$. $\mathrm{User}_{\mathrm{Applicant},u}$ 随机生成秘密值 $S_u\in Z_p^*$, 使得所计算的 ID 标识 $\mathrm{UID}_u=g^{S_u}$ 是区块链上全局唯一的.

(2) UAttrSetup($\mathrm{UID}_u,\mathrm{Attr}_i$) $\rightarrow \mathrm{ASK}_{u,i},\mathrm{APK}_{u,i}$. $\mathrm{User}_{\mathrm{Applicant},u}$ 通过各种方式 (包括但不限于线下申请) 向属性权威机构 Attr_i 申请属性, 并由属性权威机构进行确认, Attr_i 随机选取 $\mathrm{ASK}_{u,i}\in {}_RZ_p^*$ 作为 $\mathrm{User}_{\mathrm{Applicant},u}$ 的属性 Attr_i 的私钥, 并计算 $\mathrm{APK}_{u,i}=g^{\mathrm{ASK}_{u,i}}$ 作为 $\mathrm{User}_{\mathrm{Applicant},u}$ 的属性公钥, 将 $\langle\mathrm{APK}_{u,i},\mathrm{ASK}_{u,i}\rangle$ 通过安全的秘密信道发送给 $\mathrm{User}_{\mathrm{Applicant},u}$ 并将 $\mathrm{APK}_{u,i}$ 公开上链.

4.3 签名颁发证书阶段

不妨假设有期望申请证书的 $\mathrm{User}_{\mathrm{Applicant},u}$, 其期望申请属性集 $\Omega=\{\mathrm{Attr}_1,\mathrm{Attr}_2,\ldots,\mathrm{Attr}_n\}$ 对应的证书, $\mathrm{User}_{\mathrm{Applicant},u}$ 拥有其中的 t 个属性. $\mathrm{User}_{\mathrm{Applicant},u}$ 可视为区块链中的普通节点, 其通过区块链网络向 RA 节点发送入网请求证书服务. $\mathrm{User}_{\mathrm{Applicant},u}$ 提交证书所需的各种信息 m (包括 UID_u), RA 通过各种方式 (包括但不限于线下认证) 确认信息, 如果信息有误, 则拒绝该节点请求, 否则将信息发送给 n 个 CA 节点:

(1) Sign($m,S_u,\mathrm{UID}_u,\Omega,\{\mathrm{APK}_{u,i},\mathrm{ASK}_{u,i}\},\{\mathrm{CPK}_i,\mathrm{CSK}_i\}$) $\rightarrow P_{n-t}(x),\sigma$. $\mathrm{User}_{\mathrm{Applicant},u}$ 具有 t 个属性, 此时 DPKI 的 n 个 CA 节点收到 m (包括 UID_u), 开始对证书消息 $m\in\{0,1\}^*$ 进行签名.

(i) 对于 $\mathrm{User}_{\mathrm{Applicant},u}$ 拥有属性的 CA 节点 Attr_i, 不妨令其为 $i=1,\ldots,t$, Attr_i 随机选取 $t_i\in Z_p^*$,

计算

$$\begin{aligned} e_i &= H_1(\mathrm{Attr}_i, \mathrm{APK}_{u,i}, \mathrm{UID}_u, \mathrm{CPK}_i), \\ s_i &= \mathrm{ASK}_{u,i} + e_i \mathrm{CSK}_i, \\ R_i &= g^{t_i}. \end{aligned}$$

Attr_i 将 R_i 通过安全的秘密信道发送给 RA 节点.

(ii) 对于 $\mathrm{User}_{\mathrm{Applicant},u}$ 不拥有属性的 CA 节点 Attr_i, 不妨令其为 $i = t+1, \ldots, n$, Attr_i 随机选取 $c_i, d_i \in Z_p^*$, 随机生成 $\mathrm{APK}_{u,i} \in G$, 计算

$$\begin{aligned} e_i &= H_1(\mathrm{Attr}_i, \mathrm{APK}_{u,i}, \mathrm{UID}_u, \mathrm{CPK}_i), \\ Y_i &= \mathrm{APK}_{u,i} \mathrm{CPK}_i^{e_i}, \\ R_i &= g^{d_i} Y_i^{c_i}. \end{aligned}$$

Attr_i 将 $\langle c_i, d_i, \mathrm{APK}_{u,i}, R_i \rangle$ 通过安全的秘密信道发送给 RA 节点.

(iii) RA 节点将 R_i, $i = 1, \ldots, n$ 发送给 $\mathrm{User}_{\mathrm{Applicant},u}$, $\mathrm{User}_{\mathrm{Applicant},u}$ 计算

$$T_i = {R_i}^{S_u}, \quad i = 1, \ldots, n.$$

(iv) 并返回给 RA 节点, RA 节点计算

$$c = H_2(m, T_1, \ldots, T_n, \mathrm{UID}_u).$$

(v) 随后用 $n-t+1$ 个点 $(0, c), (t+1, c_{t+1}), \ldots, (n, c_n)$ 构造 $n-t$ 次拉格朗日插值多项式 $P_{n-t}(x)$:

$$P_{n-t}(x) = \sum_{i=1}^{n-t+1} P_{n-t}(x_i) \prod_{j=1, j\neq i}^{t} \frac{x - x_j}{x_i - x_j}.$$

(vi) 将 $P_{n-t}(x)$ 发送给 $\mathrm{User}_{\mathrm{Applicant},u}$ 拥有属性的 CA 节点 Attr_i, $i = 1, \ldots, t$. CA 计算

$$c_i = P_{n-t}(i), \ d_i = t_i - c_i s_i, \ i = 1, \ldots, t.$$

Attr_i 将 $\langle c_i, d_i \rangle$ 通过安全的秘密信道发送给 RA 节点.

(vii) RA 输出多项式 $P_{n-t}(x)$ 和签名:

$$\sigma = \langle c_i, d_i, T_i, \mathrm{APK}_{u,i}, \mathrm{UID}_u \rangle, \ i = 1, 2, \ldots, n.$$

(viii) RA 将 $P_{n-t}(x)$ 和 σ 附在证书信息 m 后, 生成证书 M, 将其通过安全的秘密信道发送给 $\mathrm{User}_{\mathrm{Applicant},u}$, 并把 UID_u 与时间戳的拼接作为 key, 证书内容作为 value 保存在证书库中, 多个证书库进行自动同步备份.

(2) $\mathrm{zkProveGen}(\mathrm{zkParams}, w, x, M_L) \rightarrow \pi_u$. $\mathrm{User}_{\mathrm{Applicant},u}$ 执行 zk-SNARKs 的生成证明算法 $\mathrm{Prove}(\mathrm{zkParams}, w, x, M_L)$, 其中输入参数中, 证据 $w = \langle \mathrm{UID}_u = g^{S_u}, R_i = g^{d_i} Y_i^{c_i}, Y_i = \mathrm{APK}_{u,i} \mathrm{CPK}_i^{e_i}, e_i = H_1(\mathrm{Attr}_i, \mathrm{APK}_{u,i}, \mathrm{UID}_u, \mathrm{CPK}_i) \rangle$, 生成的证明 π_u 采用 Fiat-Shamir Heurisitc 的形式, 具体的, π_u 可证明 $\mathrm{User}_{\mathrm{Applicant},u}$ 知道离散对数 S_u 满足 $\mathrm{UID}_u = g^{S_u}$ 且 $T_i = R_i^{S_u}$, $i = 1, \ldots, n$. $\mathrm{User}_{\mathrm{Applicant},u}$ 将证明 π_u, 命题 x 与图灵机算法 M_L 传输上链方便查询.

4.4 证书验证阶段

不妨假设有期望验证 $\mathrm{User}_{\mathrm{Applicant},u}$ 身份的 $\mathrm{User}_{\mathrm{Verifier},s}$, 它期望验证 $\mathrm{User}_{\mathrm{Applicant},u}$ 的身份. $\mathrm{User}_{\mathrm{Applicant},u}$ 首先去 RA 节点获取自己的证书, 并将证书发送给 $\mathrm{User}_{\mathrm{Verifier},s}$, $\mathrm{User}_{\mathrm{Verifier},s}$ 对证书进行验证. 首先看证书是否是合法的 DPKI 的 CA 节点签发, 查看 $\mathrm{User}_{\mathrm{Applicant},u}$ 的证书是否被撤销, 查看证书的有效期限. 如果证书已过期、被撤销、或者证书不是合法的节点签发, 则身份认证失败, 否则对证书进行签名认证.

$\mathrm{Verify}(m,\sigma,P_{n-t}(x),\pi_u)\to \mathrm{True/False}$.

(i) 验证

$$\begin{cases} P_{n-t}(x) \text{ is a polynomial of degree } n-t, \\ P_{n-t}(i) == c_i, \quad i=1,\ldots,n, \\ P_{n-t}(0) == H_2(m,T_1,\ldots,T_n,\mathrm{UID}_u). \end{cases}$$

以上如果有一个不通过, 则签名为无效签名.

(ii) 以上各项验证通过后, $\mathrm{User}_{\mathrm{Verifier},s}$ 在区块链上获取证明 π_u 以及 zk-SNARKs 相关参数, 执行 zk-SNARKs 的验证算法 $\mathrm{Verify}(\mathrm{zkParams},x,M_L,\pi_u)$, 如果验证不通过, 则签名为无效签名; 如果验证通过, 则签名为有效签名.

4.5 证书的撤销阶段

$\mathrm{User}_{\mathrm{Applicant},u}$ 向 RA 节点发送某个特定证书 M 的撤销请求.

$\mathrm{Revoke}(M,\pi_u,\mathrm{UID}_u)\to \mathrm{True/False}$.

RA 在区块链上获取证明 π_u, 执行 zk-SNARKs 的验证算法 $\mathrm{Verify}(\mathrm{zkParams},x,M_L,\pi_u)$, 如果验证通过, 向 n 个 CA 节点发送证书撤销请求, 否则返回错误信息.

CA 节点销毁生成的相关中间参数的信息, 并向 RA 节点返回撤销成功的撤销证书 Cer_i. RA 收集 n 个撤销证书 Cer_i 合成吊销证书存入 CRL, 并将 $\mathrm{User}_{\mathrm{Applicant},u}$ 的证书从证书库撤销. 随后区块链中的多个证书库与 CRL 进行自动同步.

5 方案分析

本节将对本方案进行理论上的安全性分析, 并对可用性进行实验测试.

5.1 安全性分析

5.1.1 正确性分析

假定 $\sigma=\langle c_i,d_i,T_i,\mathrm{APK}_{u,i},\mathrm{UID}_u\rangle$, $i=1,2,\ldots,n$ 为 $\mathrm{User}_{\mathrm{Applicant},u}$ 申请证书的签名, π_u 为 zk-SNARKs 生成的对应的证明, $\mathrm{User}_{\mathrm{Applicant},u}$ 拥有属性集 $\Omega=\{\mathrm{Attr}_1,\mathrm{Attr}_2,\ldots,\mathrm{Attr}_n\}$ 中的 t 个属性, 根据本方案的验证算法, 可以得到:

(i) 根据拉格朗日插值多项式的构造方法

$$P_{n-t}(x)=\sum_{i=1}^{n-t+1}P_{n-t}(x_i)\prod_{j=1,j\neq i}^{t}\frac{x-x_j}{x_i-x_j},$$

表 3 实验配置

Table 3 Experimental configuration

Field	Configuration	Field	Configuration
OS	macOS Big Sur 11.3.1	Hyperledger Fabric	v2.2
CPU	Intel Core i7 2.6 GHz x6	ZoKrates	0.7.1
RAM	32 GB 2667 MHz DDR4		

可知由 $n-t+1$ 个点构造的 $P_{n-t}(x)$ 确实是 $n-t$ 次多项式, 且满足

$$P_{n-t}(i)=\begin{cases} c_i, & i=1,\ldots,n,\\ H_2(m,T_1,\ldots,T_n,\mathrm{UID}_u), & i=0. \end{cases}$$

(ii) 对于 zk-SNARKs 的生成证明算法 $\mathrm{Prove}(\mathrm{zkParams},w,x,M_L)$ 中的输入参数证据 $w=\langle \mathrm{UID}_u=g^{S_u},R_i=g^{d_i}Y_i^{c_i},Y_i=\mathrm{APK}_{u,i}\mathrm{CPK}_i^{e_i},e_i=H_1(\mathrm{Attr}_i,\mathrm{APK}_{u,i},\mathrm{UID}_u,\mathrm{CPK}_i),i=1,\ldots,n\rangle$, 可以计算

$$\begin{aligned}(g^{d_i}Y_i^{c_i})^{S_u}&=(g^{t_i-c_is_i}(\mathrm{APK}_{u,i}\mathrm{CPK}_i^{e_i})^{c_i})^{S_u}\\&=\left(g^{t_i}g^{-c_i(\mathrm{ASK}_{u,i}+e_i\mathrm{CSK}_i)}(g^{\mathrm{ASK}_{u,i}}g^{\mathrm{CSK}_ie_i})^{c_i}\right)^{S_u}\\&=(g^{t_i})^{S_u}=R_i^{S_u}=T_i,\quad i=1,\ldots,n.\end{aligned}$$

综上, 本方案是正确的.

5.1.2 不可否认性

本文构造的方案的不可否认性基于离散对数问题, 由于 BTS 方案满足文献 [19] 定理 8 的条件, 同时根据文献 [15] 的 Fiat-Shamir 转换定理可以证明:

如果离散对数问题是困难的, 则本文的签名方案是不可否认的, 同时签名所使用的私钥是隐藏的, 也是不可区分的.

基于不可否认性, 本方案同时可以抵御中间人攻击 (攻击者与通讯的两端分别建立独立的联系, 并交换其所收到的数据, 从而监听或篡改信息). 本文的的属性签名方案借鉴了文献 [7] 的思路, 文献 [7] 亦从属性隐私性 (attributes privacy) 和不可传递性 (non-transferability) 对去中心不可否认属性签名方案进行了详细的证明, 这里囿于篇幅, 不对证明进行复述.

5.1.3 抗合谋攻击

合谋攻击指的是多个秘密持有者合谋可以破解原先不属于自己的秘密. 本文的证书签名过程中. CA 生成的 $\mathrm{User}_{\mathrm{Applicant},u}$ 属性私钥 $\mathrm{ASK}_{u,i}$ 并非初始化的时候一次性生成的. 而是一一对应的. 并且最终输出的签名结构也包括了由 $\mathrm{User}_{\mathrm{Applicant},u}$ 独有的 UID_u 所生成的结构. 因此即便有多个 User 合谋, 也无法共享属性或者是伪造证书.

5.2 可用性分析

本文所提出的新的基于区块链和去中心不可否认属性签名的分布式公钥基础设施方案已在 GitHub 上开源, 作为对比的 RSA, ECDSA 算法采用 Golang 自带的 crypto 库, zk-SNARKs 采用开源实现的 ZoKrates. 仿真运行在 Hyperledger Fabric v2.2 官方合约容器环境下, 通过合约命令行调用. 相关实验配置如表 3 所示.

表 4 不同算法运行时间 (s)

Table 4 Running time of different algorithms (s)

Algorithm/(t,n)	Sign	Verify	Algorithm/(t,n)	Sign	Verify
RSA	0.16394	0.00454	(30, 40)	0.48761	0.65509
ECDSA	0.11224	0.00713	(40, 50)	0.52433	0.68716
(2, 3)	0.13933	0.27902	(50, 60)	0.56731	0.70149
(5, 7)	0.20518	0.31922	(60, 70)	0.62145	0.78003
(10, 15)	0.31059	0.46538	(70, 80)	0.74124	0.99875
(20, 30)	0.44818	0.59538	(90, 100)	0.97201	1.27450

5.2.1 计算开销

针对本文提出的方案, 作者目前没有在网上见到同类的方案设计, 因此只能与传统的中心化的公钥基础设施方案 (如基于 RSA, ECDSA 签名的 PKI) 进行效率上的比较. 由于去中心化的签名是并行进行的, 所以不能简单地在理论计算复杂度上与传统的 RSA, ECDSA 签名进理论数值分析, 我们可以直接进行实验测试.

我们将通过仿真来测试所提出的新的基于区块链和去中心不可否认属性签名的分布式公钥基础设施方案在不同 (t,n) 下颁发证书与验证的运行时间, 并与传统的基于 RSA, ECDSA 签名的中心化公钥基础设施进行对比分析, 每个步骤默认运行 1000 次, 为了减少性能抖动带来的影响, 在不同时间段运行 10 次并取最终的平均值. 测试基准的证书颁发与证书验证所需的时间如表 4 所示.

从表 4 中可以看到, 在证书颁发时, 尽管本方案采用了分布式公钥基础设施方案, 但在整体耗时上与传统的签名算法并没有很大差距. 事实上, 证书颁发阶段对性能要求不是很高, 并且流程中最耗时的步骤往往是线下验证. 在证书验证步骤, 本方案与传统的签名算法相差较大, 其差距主要来自于多节点的通信与收集、zk-SNARKs 的验证与区块链的同步机制等, 但耗时尚在实际应用可接受的范围内. 同时随着 (t,n) 的增加, 消耗的时间呈缓慢增加趋势, 证明多属性的情况下本方案仍然能够正常运行. 因此, 尽管本方案为了满足诸如分布式、属性等特性, 使得证书颁发/验证具有更高的计算开销, 但这个开销是可以接受的.

5.2.2 通信开销

在通信开销方面, 由于证书结构与传统的 X.509 证书高度相似, 所以本方案更多的在 zk-SNARKs 生成的参数与证据等方面有额外的通信开销. 开源实现的 zk-SNARKs 的公共参数 (包括验证 key (verification key) 和证明 key (proving key)) 与生成的证明 (proof π) 大小都与 (t,n) 有关, 我们对不同 (t,n) 进行了测试, 如图 2 与 3 所示, 图中标注了具体的 (t,n).

可见 verification key 与 proof 会随着 (t,n) 的增加而增加, 但是即使在 $(90,100)$ 的情况下大小仍然维持在数 KB, 这个开销在实际网络传输中可以忽略不计; 而 proving key 几乎不随 (t,n) 变化, 其大小一直维持在 28 MB 左右, 尽管这个通信开销略微偏大, 但一条链仅在初始化时存储一次公共参数, 而实际证书颁发/验证过程中的额外的通信开销的主要是证明 π. 综上, 尽管本方案为了满足不可否认性而引入非交互式零知识证明 zk-SNARKs, 但实验结果表明方案带来的额外通信开销在实际场景下是可以接受的.

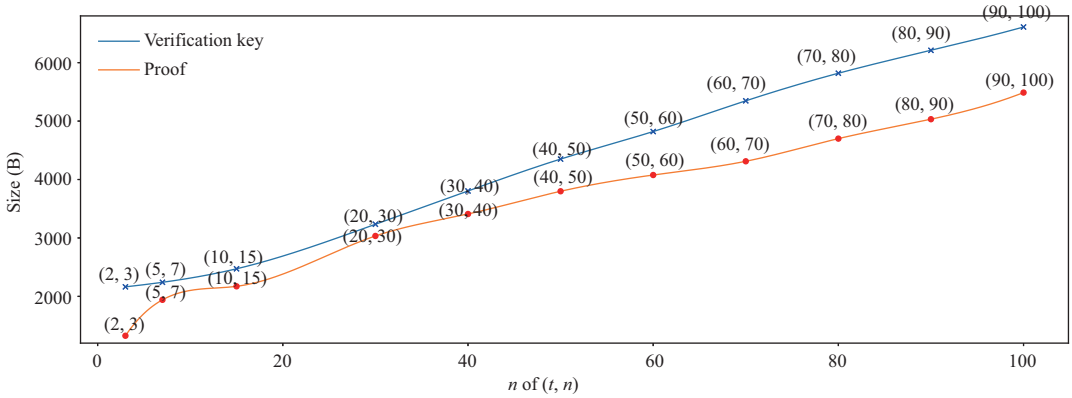

图 2 (网络版彩图) zk-SNARKs 验证 key 和证明的开销 (字节)

Figure 2 (Color online) Size of verification key and proof (B)

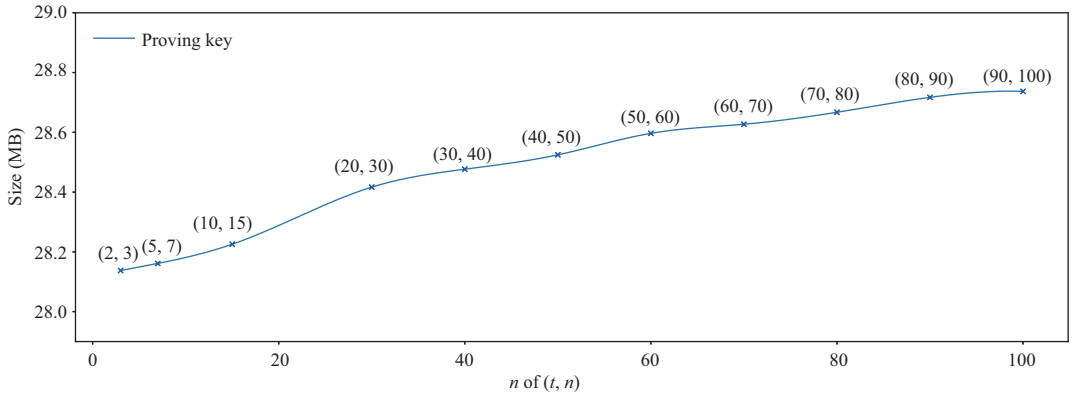

图 3 (网络版彩图) zk-SNARKs 证明 key 的开销 (兆字节)

Figure 3 (Color online) Size of proving key (MB)

5.2.3 并发测试

并发测试即测试本方案能够承载的 User 访问量, 囿于实验条件限制, 无法最大程度挖掘本方案的并发潜力, 在实际测试中, 能够支持分布式 CA $\geqslant$ 100 个, 身份认证平均时间 $\leqslant$ 1 s, 支持证书请求处理并发数 $\geqslant$ 100000, 可见本系统具有良好的并发能力.

综上, 在性能方面, 虽然我们在传统的中心化 PKI 基础上, 为了支持分布式、支持不可否认等特性引入了属性签名、零知识证明等多个算法与步骤, 但实际的性能表现还是令人满意的. 本文实验部分的 CA 节点都是随要求动态增加的, 同时之前所颁发的签名仍然能够得到正确的验证, 可见本系统在灵活性与细粒度身份认证上具有巨大的优势.

6 应用场景

本方案设计的是一种通用的基于区块链和去中心不可否认属性签名的分布式公钥基础设施, 适用于各种场景下的实体间的身份认证, 应用范围十分广泛. 本节将以本方案所属的国家重点研发计划中的智慧城市场景进行说明.

智慧城市由大量实体构成, 如数据采集设备、监控系统等, 它们在相互间的身份认证存在诸多问题: 实体间技术架构存在差异, 需要一种独立而通用的身份验证机制; 实体需要根据各种属性标签确定身份, 例如某些数据采集设备会被赋予 “所属区域”、“数据类型” 等属性标签, 跨域流转数据系统会根据这些属性标签将采集设备对接到不同的监控系统; 实体可能部署在不同的网络域, 该场景具有跨域交叉认证的需求; 实体可能为烧录了单一功能的单片机, 无法与验证实体进行交互性验证.

现有的方案无法很好地同时解决以上问题, 因此, 本文提出了基于区块链和去中心不可否认属性签名的分布式公钥基础设施方案, 突破多 CA 协同处理与验证、数字证书的分布式认证与管理、信任关系细粒度动态管理维护、分布式处理的性能等关键技术, 可以满足智慧城市中实体身份认证的各个需求, 系统模型如图 1 所示, 系统流程与 3.2 小节一致. 由 5.2 小节的测试可得本方案具有实用价值.

7 结束语

灵活有效的身份认证/管理方案一直是信息时代的核心需求之一, 本方案基于区块链和去中心不可否认属性签名提出了一种新型的的分布式公钥基础设施方案. 通过多 CA 协同处理与验证进行数字证书的智能生成与管理, 通过属性签名与门限算法实现信息关系细粒度动态管理维护与可信保持并引入了零知识证明确保了证书的不可否认性. 本方案在实际实验测试中也有良好的性能表现, 所以本方案具有实际的应用可行性. 未来的研究方向是如何优化以缩短证书验证时间, 从而进一步提高在实际环境中的实用性.

参考文献

1 Dai Z F, Wen Q Y, Li X B. The authentication technology of P2P network based on distributed PKI. ACTA Electron Sin, 2009, 37: 2561–2564 [代战锋, 温巧燕, 李小标. 基于分布式 PKI 的 P2P 网络认证技术. 电子学报, 2009, 37: 2561–2564]

2 Fromknecht C, Velicanu D, Yakoubov S. Certcoin: a namecoin based decentralized authentication system 6.857 class project. 2014. http://courses.csail.mit.edu/6.857/2014/files/19-fromknecht-velicann-yakoubov-certcoin.pdf

3 Al-Bassam M. SCPKI: a smart contract-based PKI and identity system. In: Proceedings of ACM Workshop on Blockchain, Cryptocurrencies and Contracts, 2017. 35–40

4 Han K H, Hwang S O. A PKI without TTP based on conditional trust in blockchain. Neural Comput Applic, 2020, 32: 13097–13106

5 Matsumoto S, Reischuk R M. IKP: turning a PKI around with blockchains. 2016. https://eprint.iacr.org/2016/1018.pdf

6 Chaum D. Undeniable signatures. In: Proceedings of Conference on the Theory and Application of Cryptology, New York, 1989. 212–216

7 Wei L, Huang Z J, Chen Q S, Decentralized atribute-based undeniable signature. Comput Eng Sci, 2020, 42: 56–65 [魏亮, 黄振杰, 陈群山. 去中心基于属性不可否认签名. 计算机工程与科学, 2020, 42: 56–64]

8 Ding S L, Zhao Y M, Liu Y Y. Efficient traceable attribute-based signature. In: Proceedings of the 13th International Conference on Trust, Security and Privacy in Computing and Communications, 2014. 582–589

9 Liu Y Y, Zhao Y M. A white-box traceable attribute-based signature scheme. Comput Eng, 2017, 43: 126–132 [刘雨阳, 赵一鸣. 白盒可追踪的属性签名方案. 计算机工程, 2017, 43: 126–132]

10 Fu X J, Zhang G Y, Ma C G. Dynamic threshold attributes-based signature scheme. Comput Sci, 2013, 40: 93–97

11 Lamport L. Paxos made simple. ACM Sigact New, 2001, 32: 18–25

12 Ongaro D, Ousterhout J. In search of an understandable consensus algorithm (extended version), 2013. https://courses.cs.duke.edu//spring18/compsci510/resources/papers/raft.pdf

13 Shamir A. How to share a secret. Commun ACM, 1979, 22: 612–613

14 Schnorr C P. Efficient signature generation by smart cards. J Cryptology, 1991, 4: 161–174

15 Fiat A, Shamir A. How to prove yourself: practical solutions to identification and signature problems. In: Proceedings of Advances in Cryptology, 1999. 186–194
16 Sahai A, Waters B R. Fuzzy identity-based encryption. In: Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques, 2005. 457–473
17 Maji H K, Prabhakaran M, Rosulek M. Attribute-based signatures. In: Proceedings of Cryptographers' Track at the RSA Conference, 2011. 376–392
18 Eberhardt J, Tai S. Zokrates-scalable privacy-preserving off-chain computations. In: Proceedings of IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018. 1084–1091
19 Cwi R C, Cwi B S. Proofs of partial knowledge and simplified design of witness hiding protocols. In: Proceedings of Annual International Cryptology Conference, 1994. 174–187

Distributed public key infrastructure scheme based on blockchain and decentralized undeniable attribute-based signature

Hexin YUAN[1,2], Baixiang LIU[1,2], Haibin KAN[1,2,3*] & Zening CHEN[1,2]

1. *Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai* 200433*, China*;
2. *Shanghai Engineering Research Center of Blockchain, Shanghai* 200433*, China*;
3. *Key Laboratory for Information Science of Electromagnetic Waves (MoE), Shanghai* 200433*, China*
* Corresponding author. E-mail: hbkan@fudan.edu.cn

Abstract A flexible and effective identity system scheme has always been one of the core needs of the information age. Traditional centralized public key infrastructure has a number of flaws, and the present distributed public key infrastructure based on blockchain has a number of issues with performance, resilience, non-repudiation, identity flexibility, and other factors. This paper innovatively combines blockchain with decentralized undeniable attribute-based signatures and proposes a novel distributed public key infrastructure, which uses threshold algorithms and attribute-based signatures for fine-grained management of identities; the paper also introduces non-interactive zero-knowledge proof to make the certificate undeniable and uses the blockchain consensus mechanism to synchronize the certificate library to achieve distributed identity authentication. Through experimental modeling and analysis combined with specific scenarios' actual landing demand, this article indicates that the solution is adequate in terms of security and usability.

Keywords blockchain, public key infrastructure, attribute-based signature, threshold algorithm, zero-knowledge proof, identity authentication