中国科学: 信息科学 2021 年 第 51 卷 第 8 期: 1345–1359
SCIENTIA SINICA Informationis
论文

基于区块链和去中心属性密码的访问控制身份方案

陈泽宁[1,2], 张亮[1,2], 张双俊[1,2], 阚海斌[1,2,3*]

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复旦 – 众安区块链与信息安全联合实验室, 上海 200433
3. 上海先进通信与数据科学研究院, 上海 200433
* 通信作者. E-mail: hbkan@fudan.edu.cn

收稿日期: 2020–03–08; 修回日期: 2020–05–07; 接受日期: 2020–06–17; 网络出版日期: 2021–07–29

国家重点研发计划 (批准号: 2019YFB2101703)、国家自然科学基金 (批准号: 61672166, U19A2066) 和上海科技创新行动计划 (批准号: 20222420800, 20511102200) 资助项目

摘要 区块链上的身份体制是不健全的. 如何在区块链上认证某个用户的身份、如何确保身份的背书机构是真实的一直是一个挑战. 运行在区块链上的分布式公钥基础设施可以在某种程度上解决上述问题, 但是属性密码结合区块链可以提供更贴近于真实社会的身份模型. 提出一种基于区块链和去中心属性密码的访问控制身份方案, 利用用户和组织之间相互授权、背书身份属性实现信任成本的链接, 利用属性密码对链上数据进行访问控制和共享达到细粒度的访问控制和隐私保护. 设计了可多用户协同的属性密码, 为身份模型中的机构提供背书能力. 通过实验仿真和对比分析, 该方案在安全性和性能上都满足当前通用区块链的需求, 为其提供了一种通用基础的身份模型.

关键词 区块链, 属性密码, 访问控制, 身份认证, 隐私保护

1 引言

区块链作为近些年来新兴的集成技术, 因其 “去中心化” 和 “不可篡改” 特性, 多中心相互协作的组织和普通用户都可以在弱信任或无信任的情况下进行信息的交互. 但由于区块链的 “不可篡改” 和与其共生的公开透明的特性, 在区块链上的数据可以被所有人获取并分析, 因此区块链上隐私数据的保护是一个不能忽视的问题. 许多文献 [1~5] 也针对区块链上的隐私保护和数据访问控制提出新的架构模型或引入新的技术方案, 基于属性的加密算法 (attribute-based encryption, 简称为 ABE 或属性密码) 也是其中的一种有效的解决方案. 由于属性密码的良好特性, 许多人将其应用在基于区块链的各种方案中, 例如 IoT 网络 [6,7]、公有云存储 [8]、医疗数据共享 [9] 和数据溯源 [10] 等.

追根溯源, ABE 起源于模糊身份的加密方案 [11](fuzzy identity-based encryption, 简称 Fuzzy-IBE), Fuzzy-IBE 解决了传统加密系统中一对一加解密的问题, 身份可以由一组属性组成, 只要解密者与要求

引用格式: 陈泽宁, 张亮, 张双俊, 等. 基于区块链和去中心属性密码的访问控制身份方案. 中国科学: 信息科学, 2021, 51: 1345–1359, doi: 10.1360/SSI-2020-0048
Chen Z N, Zhang L, Zhang S J, et al. Access control scheme on blockchain and decentralized attributed-based algorithm with identity (in Chinese). Sci Sin Inform, 2021, 51: 1345–1359, doi: 10.1360/SSI-2020-0048

属性集的误差在一定范围内都可以解密, 实现了一对多的加解密. 进一步, Goyal 等 [12] 在 2006 年提出了基于属性的加密算法, 并将其分为基于密钥策略的加密算法 (key-policy attribute-based encryption, KPABE) 和基于密文策略的加密算法 (ciphertext-policy attribute-based encryption, CPABE), CPABE 和 KPABE 的区别仅在于访问控制策略和属性分别属于密文还是密钥. ABE 的提出细化了原先 IBE 针对属性阈值的加密, 根据访问控制策略 (access policy) 和属性实现对数据细粒度的访问控制. 由于 CPABE 中密文和访问控制策略一一对应, 因此更适合于公有存储上对加密数据的访问权限控制.

区块链上的身份体制一直以来也是一个不容忽视的话题. 自互联网起, 身份层就是一块缺失的领土, 2005 年 Kim Cameron 就提出了数字身份七法则 [13], 定义了一套以用户为中心的数字身份元系统, 但在互联网上, 身份管理仍然掌握在中心化机构手中. 标榜去中心化的区块链 "遗传" 了这一缺陷, 仅仅存在由非对称加密算法保障的账户体系, 而业务上的身份认证只能依赖引入可信第三方做身份认证, 因此在区块链上身份体制是十分脆弱的. 许多人通过分布式的密钥基础设施 (public key infrastructure, PKI) 为区块链提供身份支持 [14~16] 1)2), 门罗币和零币 [17,18] 等也另辟蹊径, 将身份匿名贯彻到底, 致力于通过隐藏交易中的身份信息保护隐私数据.

本方案的贡献在于, 提出了一种新型的基于区块链和去中心化属性密码的访问控制身份方案, 对现有的去中心化属性密码进行适当改进, 结合门限秘密分享算法构建了一种基于区块链的身份授权认证模型, 通过身份属性的授权与认证, 建立区块链上身份体制所需的信任背书链, 通过属性密码一对多的加解密机制, 为区块链提供加密数据的访问权限管理, 继而可为区块链上的其他应用提供身份支撑和数据分享的细粒度访问控制.

2 预备知识

本节首先简单介绍区块链和双线性映射的背景知识, 之后是单调访问控制结构和本方案中使用的 (t,n) 门限秘密分享算法, 最后介绍基本的 CPABE 和多授权中心的 CPABE.

2.1 区块链

区块链的概念源于 2008 年中本聪 (Satoshi nakamoto) 提出的比特币3), 原意只是构建一个去中心化的电子支付交易平台, 后续因区块链自身结构中所带来的不可篡改、公开透明等特性被推广到非金融领域的应用中, 在供应链、信息溯源、存证等领域大放异彩. 简单来说, 区块链可以视为根据时间顺序将数据组成的区块连接在一起, 从而藉由后续区块的添加而保证之前的区块固定不变的一种结构. 后续以太坊4)提出为区块链提供了一个图灵完备的虚拟机, 自此区块链可以作为一个 "世界状态机" 执行运行在其上的智能合约 (smart contract), 并通过分布式的共识算法确保区块链上数据的一致性. 后续 Linux 基金会开源了 Hyperledger Fabric 项目5), 支持以标准通用编程语言编写智能合约, 而无需依赖于加密货币, 本文的实验也是基于 Fabric 搭建的.

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

2) Lewison K, Corella F. Backing rich credentials with a blockchain PKI. 2016. http://pomcor.com.

3) Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. 2009. https://bitcoin.org/bitcoin.pdf.

4) Buterin V. A next-generation smart con-tract and decentralized application platform. 2013. https://github.com/ethereum/wiki/wiki/White-Paper.

5) Cachin C. Architecture of the hyperledger blockchain fabric. Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016. https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf.

2.2 合数阶双线性映射

本文使用的密码算法工作在合数阶双线性映射上, 由算法输入安全参数 λ 输出一个双线性群 G 以及 (p_1,p_2,p_3,G_T,G,e), 其中 p_1,p_2,p_3 均为不同的素数, 则 G,G_T 是阶 $N=p_1p_2p_3$ 下的循环群, 定义 $e:G^2\to G_T$ 为双线性映射且满足

- (双线性) $\forall g,h\in G,a,b\in\mathbb{Z}_N,e(g^a,h^b)=e(g,h)^{ab}$;
- (非退化) 存在 $g\in G$ 使得 $e(g,g)$ 在 G_T 中的阶为 n;
- (可计算) 对于所有的 $g,h\in G$, 存在一个有效的算法计算 $e(g,h)$.

合数阶双线性映射满足: 不同子群下元素组合求双线性映射后结果为 1.

2.3 单调访问控制结构

定义1 ([19]) 定义一组实体记为 $\{P_1,\dots,P_n\}$. 集合 $\mathbb{A}\subseteq 2^{\{P_1,\dots,P_n\}}$ 是单调的当且仅当对于所有的 B, C, 如果 $B\in\mathbb{A}$ 且 $B\subseteq C$, 则 $C\in\mathbb{A}$. 一个单调访问控制结构 $\mathbb{A}$ 是 $\{P_1,\dots,P_n\}$ 的非空子集构成的集合, 例如 $\mathbb{A}\subseteq 2^{\{P_1,\dots,P_n\}}\backslash\{\}$. $\mathbb{A}$ 中的集合被称为授权集, 不在其中的集合称为非授权集.

在本文中, 实体 P_i 就是一个属性, 也就是说访问控制结构 $\mathbb{A}$ 包含授权属性集.

2.4 (t,n) 门限秘密分享算法

常规的秘密分享可以视为 (n,n) 秘密分享, 独立的部分秘密是无用的. 在 Shamir [20] 提出 (t,n) 门限方案后, 后续也有一系列研究者针对不同问题改进门限方案. 本方案采用了 Pedersen [21] 的一种无需可信第三方的门限秘密分享算法, 以此适配区块链和所采用的属性密码算法的去中心化特性.

假定有一个群组需要分享秘密, 其中的参与者可以表示为 $P_i,i=1,2,\dots,n$, 他们分别具有唯一标识 id_i. 假定最终要协作分享的秘密为 $S=\sum_{i=1}^n S_i$, 其中 S_i 为 P_i 随机生成的秘密. 每个参与者 P_i 分别随机生成一个 $t-1$ 阶的多项式 $f_i(x)$, 使得 $f_i(0)=S_i$. P_i 计算 $\mathrm{share}_{ij}=f_i(\mathrm{id}_j),\forall j$ 并将其秘密发送给 P_j, 当参与者收到其他 $n-1$ 个秘密分享 share_{ji} 时, 结合自己生成的 share_{ii}, 则可以计算自己的部分秘密 $\mathrm{secret}_i=\sum_{j=1}^n\mathrm{share}_{ji}=\sum_{j=1}^n f_j(\mathrm{id}_i)$. 当需要 t 个参与者协作获得最终秘密 S 时, 假设 $F(x)=\sum_{j=1}^n f_j(x)$, 则 $\mathrm{secret}_i=\sum_{j=1}^n f_j(\mathrm{id}_i)=F(\mathrm{id}_i)$. 因为 $F(x)$ 是 $t-1$ 阶多项式, 所以最终我们可以通过拉格朗日 (Lagrange) 插值法求出 $F(0)=S$, 即最终秘密.

2.5 CPABE

ABE 的概念 [11] 最早在 2005 年由 Sahai 和 Waters 提出. 在该方案中, 密文由公钥以及一个属性集所加密, 每个用户独自拥有一个属性集, 当用户属性集中的元素与密文属性集中的元素重叠超过 t 个时, 用户可以进行解密. 2006 年 Goyal 等将 ABE 分成了两个部分: KPABE 和 CPABE [12]. 在 CPABE 方案中密钥和属性集绑定, 密文和访问控制策略绑定, 访问控制策略可以转化为访问控制矩阵或访问控制树. 因为基础的 CPABE 存在中心化解密和分发密钥的局限, Sahai 和 Waters 提出了在多中心场景下构建 ABE 系统的问题, 随后 Chase 在文献 [22] 中第一次得到可能的解决方案, 该方案允许任意数量的独立授权机构监听属性并分发属性密钥, 加密者可以针对多个授权机构 d 指定一个阈值 d_k 和一个属性集, 只有当解密者至少拥有每个授权机构给定属性集中的 d_k 个属性时才能解密.

因为当时的多中心的 CPABE 方案仍存在可信中心机构参与到方案运行中, 因此 Lewko 等 [23] 在 2011 年提出了多授权中心的 CPABE 算法, 通过加入授权中心和用户的角色适配分布式体系下的属性密码结构, 授权中心和用户地位相同, 无需可信第三方介入即可进行申请和授权属性. 通常由以

下 5 个步骤组成. 如果对于任意的全局参数, 按以下步骤得到密文和单一用户的属性私钥集合后都可以解密得到正确的明文, 则证明这个多授权中心的 CPABE 是正确的.

Global Setup $(\lambda) \to \mathrm{GP}$ 此步骤通过传入一个安全参数 λ 计算输出供系统使用全局参数 GP.

Authority Setup $(\mathrm{GP}) \to \mathrm{SK}, \mathrm{PK}$ 此步骤使用全局参数 GP 来初始化一个授权机构实体, 生成与其相对应的公私钥对.

Encrypt $(M, (A, \rho), \mathrm{GP}, \{\mathrm{PK}\}) \to \mathrm{CT}$ 此步骤需要传入明文 M、访问控制矩阵 (A, ρ)、全局参数 GP、访问控制矩阵中用到的属性相关公钥集合 {PK}. 最后输出密文 CT.

KeyGen $(\mathrm{GID}, \mathrm{GP}, i, \mathrm{SK}) \to \mathrm{K}_{i,\mathrm{GID}}$ 此步骤通过传入用户唯一的身份标识 GID、全局参数 GP、属于某个授权中心的属性 i 和该授权中心的私钥 SK, 通过计算输出对应于该用户的特定属性私钥.

Decrypt $(\mathrm{CT}, \mathrm{GP}, \{\mathrm{K}_{i,\mathrm{GID}}\}) \to M$ 此步骤通过传入密文 CT、全局参数 GP、用户的属性私钥集合 $\{\mathrm{K}_{i,\mathrm{GID}}\}$. 如果用户所拥有属性 i 的集合满足加密时使用的访问控制矩阵, 则最终算法可以输出正确的明文 M, 否则解密失败.

3 系统模型

本节给出系统模型的定义.

3.1 符号说明

访问控制身份方案中系统架构和属性密码相关的符号如下所示.

- User: 用户, 构成身份模型的基本单位, 可进行授权、认证、加解密数据等功能.
- Org: 组织, 由多个用户组成, 其余功能与用户相同.
- G, g: 所使用的双线性群及其生成元.
- USK, UPK: 用户自身所持有的私钥与公钥.
- {OSK}, OPK: 组织的公钥、私钥集合由多用户持有其部分.
- ASK, APK: 属性相关的公钥和私钥.
- GID: 所有组织和用户都有唯一对应的全局 ID 标识.
- Attr, CT, M: 身份属性、密文结构和明文消息.

3.2 通用流程介绍

以本方案在社交媒体区块链上的使用为例, 通用流程如图 1 所示. 图中用户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, 同一个用户可以身兼数职, 例如内容分享者或认证授权者. 组织作为多用户的集合体, 具备授权身份属性的权力. 区块链和智能合约作为载体, 具备存储公共参数、密文, 执行基本的查询和用户注册操作等能力. 具体的流程如下.

(1) 前提: 所有的用户或组织可以通过区块链和智能合约获取链上数据 (包括但不限于公共参数), 并将初始化产生的公钥、加密数据产生的密文等公开上链. 用户与用户之间、用户与组织之间的所有通信皆通过智能合约发起.

(2) 用户注册到区块链上: 自身选择 GID 并执行用户初始化或声明新属性操作, 将产生的用户公钥或用户属性上链.

(3) 多用户协作注册组织到区块链上: 协同进行组织初始化或组织声明新属性操作, 该过程产生中间参数. 其中总用户数量为 n, 门限秘密分享的阈值为 t, $t > n/2$, 恶意用户的占比不超过 $n - t$. 组

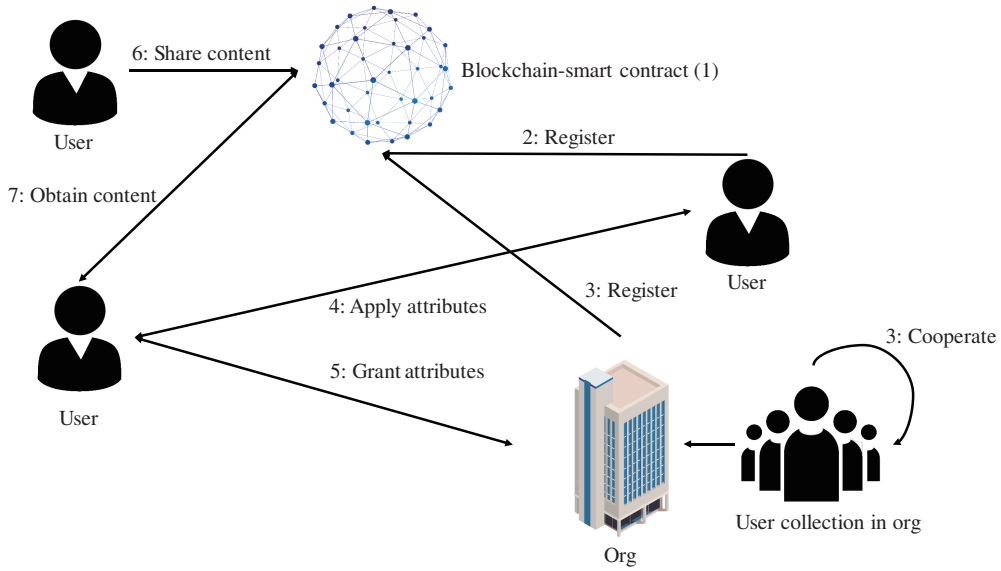


图 1 (网络版彩图) 通用流程

Figure 1 (Color online) Universal processes

织通过智能合约收集协作用户产生的中间参数, 将其组合计算组织公钥或组织属性, 并公开上链.

(4) 用户申请属性: 用户可以向其他任意用户或组织申请其拥有的属性作为自身身份认证、密码找回、内容推广等用途, 生成相应请求并上链.

(5) 用户/组织授予属性: 被申请者自行判断是否通过申请者的属性认证申请, 若通过则返回相应的秘密参数. 申请者根据秘密参数自行计算属性私钥并保存.

(6) 内容分享: 用户设定特殊的访问控制策略, 通过属性加密算法将数据加密得到密文后上链.

(7) 内容获得: 用户从区块链中获得其他用户分享的密文, 如果满足密文的访问控制策略, 则可使用自身属性私钥将其解密获得明文数据.

4 基于区块链和去中心属性密码的访问控制身份方案

本节分两个部分介绍, 第 1 部分概述基于区块链和去中心属性密码的访问控制身份方案区别于其他方案的特点. 第 2 部分则是关于访问控制身份方案的具体步骤介绍, 包括: 系统初始化、用户初始化及属性生成、组织初始化及属性生成、申请用户属性、申请组织属性、加密以及解密, 与通用流程中的步骤基本一致.

4.1 访问控制身份方案的特点

为了解决区块链上身份认证困难, 所引发的数据确权难、丢失私钥找回难等问题, 本方案引入了基于属性的密码, 藉其对属性细粒度的访问控制策略设计实现了一套访问控制身份方案, 与当前基于区块链的身份方案相比具备一些特点.

- 身份立体. 区块链的身份机制一直是较为薄弱的, 一对公私钥就作为区块链上所有身份认证的基础. 以太坊上交易的时候公钥输错了, 或者是误输入某个合约的地址; 溯源区块链上, 传输环节中某个非官方公钥混入其中; 比特币的公钥忘记了无法找回等等, 上述问题均是源自公私钥并不满足一个身

份所需要具备的特点, 仅仅是一个账户而已, 相对而言太过单薄. 因此本方案引入了属性密码作为身份和数据的访问控制密码体系, 利用属性所涵盖的广度 (可以涵盖多个领域, 例如基本身份标识、社会关系、学习经历等) 和访问控制策略所带来的深度 (通过设计特定的策略针对具备特定属性的用户), 可以使区块链的身份机制更加立体.

• 信任体系去中心化. 为了适配区块链自身具有的去中心特性, 本方案采用了去中心化的属性密码方案, 摒弃了其他中心化/弱中心化属性密码方案需要 CA 中心提供证书认证或是第三方认证机构支持的缺点, 不存在任何中心化的机构介入, 将所有的信任成本交给用户属性授权与认证实现, 最大程度模拟现实社会的意识形态. 并在原先的基础上设计增加了门限控制体系, 将原先单一的用户机制扩展到可由多用户共同决策的机构机制, 使得身份属性授权认证模型更加完整多元.

• 门限密码实用. 在本方案中, 通过合约调用, 所有的用户或是组织都可以在区块链中动态地增加自身可控制的身份属性, 并根据自身意愿将身份属性授权给他人. 在多用户组成的组织中, 不需要所有用户协作才能对组织授权决策, 但少于预设门限的用户即使合谋也无法得出组织及其所属的全部属性的秘密. 并且不同于 Shamir 秘密分享方案中, 重组秘密只能进行一次, 之后秘密暴露需要重新生成. 每个用户所掌控的秘密参数都可以多次使用来授权组织下属的不同属性给其他用户.

• 属性无需预设. 区别于很多其他的属性密码方案, 本方案为了身份属性的灵活定义和实时扩展, 采用的属性密码不需要在密码初始化时定义所有属性与双线性域上随机值的映射对, 用户或组织可以藉由区块链去中心化、完全透明的特性自由生成特定的属性. 进一步地, 可以通过智能合约限制新发布密文所使用的属性, 以此来实现特定属性的全局撤销功能.

综上所述的特点, 结合 3.2 小节举例使用的社交媒体区块链: 相比于其他区块链社交媒体, 本方案下的用户认证更加多元和可信, 也带来更为真实可靠的内容; 用户隐私和内容共享完全由自己控制想要分享的受众, 数据溯源可以融合身份因素; 用户密钥可以脱离公私钥的限制, 并且可以自由设定重置密码规则以防遗失或被盗.

4.2 访问控制身份方案的具体细节

4.2.1 系统初始化

Global Setup(λ) $\rightarrow$ GP: 在系统初始化中, 选择一个双线性群 G, 其阶数 $N = p_1p_2p_3$. 则公共参数为 GP, 包含 N 和 G_{p_1} 的生成元 g. 除此之外还需要选择一个哈希散列函数 $H : \{0,1\}^* \rightarrow G$ 来将全局的 ID 标识 GID 映射到群 G 中的元素. 在通用流程中由预设合约实例化时实现.

4.2.2 用户初始化及属性生成

User Setup(GP) $\rightarrow$ UPK, USK: 具备全局唯一 ID 标识 GID 的用户随机选择随机数 $\alpha \in \mathbb{Z}_N$, 然后公开 UPK $= \{e(g,g)^{\alpha}, \text{GID}\}$, 将 USK $= \{\alpha\}$ 作为其私钥.

User's Attributes Setup(GP) $\rightarrow$ APK, ASK: 针对用户想指定的新属性 Attr_x, 用户随机选择随机数 $y_x \in \mathbb{Z}_N$, 然后公开 APK $= \{g^{y_x}\ \forall x\}$ 作为属性集公钥, 将 ASK $= \{y_x\ \forall x\}$ 作为其属性集私钥.

4.2.3 组织初始化及属性生成

组织的初始化可以分为两个子过程: Org Setup 和 Org Mix, 在 Org Setup 时, 欲成立组织的成员用户相互协作, 最终形成自己的部分组织公钥 pk_i, 最终在 Org Mix 中交由智能合约整合形成组织公钥 OPK. 组织的属性生成与初始化过程类似, 由 Org's Attributes Setup 和 Org's Attributes Mix 组成.

Org Setup (GP): 假定具备全局唯一 ID 标识 GID 的组织由 n 个用户组成, 用户的 ID 标识可表示为 GID_i, 并设定 (t,n) 门限秘密分享算法的 t 值, 继而这 n 个用户需要进行如下的步骤进行协作:

- 每个 $\mathrm{User}_i, i=1,2,\ldots,n$ (如无特殊标识默认皆为全部 n 个用户) 需要选择一个随机数 $\alpha_i \in \mathbb{Z}_N$ 作为其组织秘密部分, 最终的组织秘密可以表示为 $\alpha = \sum_{i=1}^{n} \alpha_i$. 然后每个 User_i 需要随机选取一个 $t-1$ 阶多项式 $f_i(x) = a_0 + a_1 x + \ldots + a_{t-1}x^{t-1}$, 并使得 $f_i(0) = a_0 = \alpha_i$. 接着每个 User_i 针对全部 n 个用户生成 $\mathrm{share}_{ij} = f_i(\mathrm{GID}_j)$, 并将其通过可信信道秘密分享给 $\mathrm{User}_j, j=1,2,\ldots,n, j\neq i$, 自己留存 share_{ii}.
- 当 User_i 接收到其他 $n-1$ 个用户发送给自己的秘密分享后, 计算 $\mathrm{osk}_i = \sum_{j=1}^{n} \mathrm{share}_{ji}$ 和 $\mathrm{opk}_i = e(g,g)^{\mathrm{osk}_i}$. 并将 opk_i 通过智能合约公开上链.

Org Mix(GP) $\to$ OPK, {OSK}: 智能合约接收到具备 GID 的组织下所有成员的 opk_i 之后, 从中随机选取 t 个成员, 由秘密分享原理可计算

$$\begin{aligned} e(g,g)^{\alpha} &= e(g,g)^{\sum_{i=1}^{n}\alpha_i} \\ &= e(g,g)^{\sum_{i=1}^{t}(\mathrm{osk}_i \prod_{j=1,j\neq i}^{t} \frac{\mathrm{GID}_j}{\mathrm{GID}_j - \mathrm{GID}_i})} \\ &= \prod_{i=1}^{t} \mathrm{opk}_i^{\prod_{j=1,j\neq i}^{t} \frac{\mathrm{GID}_j}{\mathrm{GID}_j - \mathrm{GID}_i}}. \end{aligned} \tag{1}$$

并最终将 $\mathrm{OPK} = \{e(g,g)^{\alpha}, \mathrm{GID}\}$ 公开上链.

Org's Attributes Setup(GP): 组织属性的生成与初始化类似, n 个用户需要进行如下的步骤进行协作.

- 全部 User_i 需要选择一个随机数 $y_i \in \mathbb{Z}_N$ 作为其组织属性秘密部分, 最终的组织属性秘密可以表示为 $y = \sum_{i=1}^{n} y_i$. 然后 User_i 随机选取 $t-1$ 阶多项式 $f_i(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}$, 并使得 $f_i(0) = a_0 = y_i$. 最后生成 $\mathrm{share}_{ij} = f_i(\mathrm{GID}_j)$, 并将其秘密传输给其他用户.
- User_i 汇集秘密分享后, 计算 $\mathrm{ask}_i = \sum_{j=1}^{n} \mathrm{share}_{ji}$ 和 $\mathrm{apk}_i = g^{\mathrm{ask}_i}$. 并将 apk_i 通过智能合约公开上链.

Org's Attributes Mix(GP) $\to$ APK, {ASK}: 智能合约接收到所有成员的 apk_i 之后, 从中随机选取 t 个并计算

$$\begin{aligned} g^{y} &= g^{\sum_{i=1}^{n} y_i} \\ &= g^{\sum_{i=1}^{t}(\mathrm{ask}_i \prod_{j=1,j\neq i}^{t} \frac{\mathrm{GID}_j}{\mathrm{GID}_j - \mathrm{GID}_i})} \\ &= \prod_{i=1}^{t} \mathrm{apk}_i^{\prod_{j=1,j\neq i}^{t} \frac{\mathrm{GID}_j}{\mathrm{GID}_j - \mathrm{GID}_i}}. \end{aligned} \tag{2}$$

4.2.4 申请用户属性

User's Attributes Auth$(\mathrm{GID}_i, \mathrm{Attr}_x, \mathrm{USK}, \mathrm{ASK}, \mathrm{GP}) \to K_{x,\mathrm{GID}_i}$: 用户 User_i 可向用户 User_j 申请其专属属性 Attr_x. User_j 通过申请用户的 GID_i 计算 $\mathrm{auth}_{\mathrm{Attr}_x,\mathrm{GID}_i} = g^{\alpha} H(\mathrm{GID})^{y_x}$ 返回给申请用户, 用户将其作为自己的部分属性私钥 K_{x,GID_i}.

4.2.5 申请组织属性

Org's Attributes Auth$(\mathrm{GID}_i, \mathrm{Attr}_x, \{\mathrm{OSK}\}, \{\mathrm{ASK}\}, \mathrm{GP}) \to K_{x,\mathrm{GID}_i}$: 用户 User_i 可向组织 Org 申请其专属属性 Attr_x, 申请属性需经过 Org 下 t 个参与成员的同意. 成员用户 $\mathrm{User}_j, j=1,2,\ldots,t$ 通

过申请用户的 GID_i 计算 $\mathrm{auth}_{\mathrm{Attr}_x,\mathrm{GID}_i j}=g^{\mathrm{osk}_j}H(\mathrm{GID})^{\mathrm{ask}_{xj}}$ 返回给申请用户. 用户获得 t 个 auth 之后计算对应的属性私钥为

$$\begin{aligned}K_{x,\mathrm{GID}_i}&=g^{\alpha}H(\mathrm{GID}_i)^{y_x}\\&=g^{\sum_{j=1}^{n}\alpha_j}H(\mathrm{GID}_i)^{\sum_{j=1}^{n}y_{xj}}\\&=g^{\sum_{j=1}^{t}(\mathrm{osk}_j\prod_{k=1,k\neq j}^{t}\frac{\mathrm{GID}_k}{\mathrm{GID}_k-\mathrm{GID}_j})}H(\mathrm{GID}_i)^{\sum_{j=1}^{t}(\mathrm{ask}_{xj}\prod_{k=1,k\neq j}^{t}\frac{\mathrm{GID}_k}{\mathrm{GID}_k-\mathrm{GID}_j})}\\&=\prod_{j=1}^{t}\mathrm{auth}_{\mathrm{Attr}_x,\mathrm{GID}_i j}^{\prod_{k=1,k\neq j}^{t}\frac{\mathrm{GID}_k}{\mathrm{GID}_k-\mathrm{GID}_j}}.\end{aligned}\tag{3}$$

4.2.6 加密

$\mathrm{Encrypt}(\mathrm{M},(\mathrm{A},\rho),\{\mathrm{UPK},\mathrm{OPK},\mathrm{APK}\},\mathrm{GP})\to\mathrm{CT}$: 在加密算法中, 输入参数分别为明文消息 M, 一个 $n\times l$ 的访问控制矩阵 A 和将矩阵行映射到属性集的映射关系 ρ, 涉及到的所有用户、组织、属性的公钥以及全局公开参数. 首先选择一个随机数 $s\in\mathbb{Z}_N$ 和一个随机向量 $v\in\mathbb{Z}_N^{\ell}$, 其中 s 是向量 v 的第一个元素. 再选择随机向量 $w\in\mathbb{Z}_N^{\ell}$, 其中第一个元素为 0. 最后对于矩阵 A 中的某一行 A_k, 选择一个随机数 $r_k\in\mathbb{Z}_N$, 则密文结构可以表示为 $\mathrm{CT}=\{C_0,\{C_{1,k},C_{2,k},C_{3,k}\ \forall k\}\}$, 其中

$$C_0=Me\left(g,g\right)^{s},\quad C_{1,k}=e\left(g,g\right)^{A_k\cdot v}e\left(g,g\right)^{\alpha_{\rho(k)}r_k},\quad C_{2,k}=g^{r_k},\quad C_{3,k}=g^{y_{\rho(k)}r_k}g^{A_k\cdot w},\quad\forall k.$$

4.2.7 解密

$\mathrm{Decrypt}(\mathrm{CT},\{K_{x,\mathrm{GID}_i}\},\mathrm{GP})\to M$: 在解密算法中, 输入参数分别为密文 CT, 用户匹配访问控制策略所使用的属性 $\{K_{x,\mathrm{GID}_i}\}$ 以及全局公开参数. 解密者首先需要计算 $\omega_k\in\mathbb{Z}_N$, $\forall k$, 使得 $\sum_k\omega_k A_k=(1,0,\ldots,0)$. 之后针对每一行 k, 计算

$$C_{1,k}\cdot e\left(H(\mathrm{GID}_i),C_{3,k}\right)/e\left(K_{\rho(k),\mathrm{GID}_i},C_{2,k}\right)=e\left(g,g\right)^{A_k\cdot v}e\left(H(\mathrm{GID}_i),g\right)^{A_k\cdot w}.$$

接下来针对每行获得的值进行乘方和累乘, 借助公式 $\sum_k(A_k\cdot v)\omega_k=v\cdot(1,0,\ldots,0)=s$ 和 $\sum_k(A_k\cdot w)\omega_k=w\cdot(1,0,\ldots,0)=0$ 可以得到

$$\prod_k\left(e\left(g,g\right)^{A_k\cdot v}e\left(H(\mathrm{GID}_i),g\right)^{A_k\cdot w}\right)^{\omega_k}=e\left(g,g\right)^{s}.$$

最终可以求得明文消息 $M=C_0/e\left(g,g\right)^{s}$.

5 安全性分析

本节将介绍常见的攻击模型及本方案如何抵抗这些典型的攻击, 常见攻击模型如下.

- 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. 在分布式体系中常常体现在攻击者通过多个秘密集合可以反向计算出授权方的主密钥或计算出其他用户的被授权密钥.
- 中间人攻击: 中间人攻击指的是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.
- 链接攻击: 链接攻击指的是攻击者通过链接同一个地址的多笔交易查找用户的隐私数据. 在比特币中就可以通过分析交易流水定位用户信息.

本方案从几个方面可以抵抗上述攻击: 首先本方案使用的去中心的属性密码 [23] 有所不同, 用户的密钥结构并非由中心机构一次生成, 而是由属性对应一个个密钥最终组成的简单集合. 这种特殊的密钥结构包含了被授权用户所对应 GID 的哈希值, 因此即便有多个用户合谋, 也无法共享属性或是在解密算法中使用不同 GID 的属性密钥进行解密, 因此可以抗共谋攻击; 另外因为所有的用户和组织都有唯一的 GID, 且该 GID 可以直接作为公钥并公布于区块链上, 所以本方案中所有的通信都可以加上相关的公钥签名, 返回的秘密数据也可以通过对方的公钥进行加密. 中间人无法通过篡改公钥地址或伪造合法签名来通过验证, 继而实现抗中间人攻击; 最后身份属性相关的授权过程是否公开由用户自主决定, 用户可以选择公开被授权身份属性对应的交易来表明身份, 也可以隐藏身份信息以保护隐私. 而加密数据在链上都是密文, 其安全性通过非对称加密算法和基于属性的加密算法所保证, 攻击者无法获得更多的用户隐私信息, 做到抗链接攻击.

6 方案分析

本节将会从数字身份管理、功能特性对比、计算开销对比及实验仿真 4 个方面分析本方案作为基于区块链的访问控制身份方案的优劣.

6.1 数字身份管理

Kim Cameron 在 2005 年就提出了数字身份七法则, Dunphy 等也在文献 [24] 中对比分析了 4 种其他方案, 类似地, 本方案在该法则框架下的解释如下.

(1) User Control and Consent: 本方案下, 与用户相关的隐私信息均使用属性密码进行加密上链. 除了用户需要公示的自身下属属性, 即便是向其他用户和组织申请属性的交易也可以通过非对称密钥算法进行加密, 用户完全控制其身份属性, 并且属性是否公开取决于其自身意愿.

(2) Minimal Disclosure for a Constrained Use: 属性密码提供的细粒度访问控制权限很好地解决了因身份载体绑定而导致的信息泄露, 例如网吧要求提供身份证证实顾客成年, 却同时可以收集顾客的姓名和家庭住址等信息.

(3) Justifiable Parties: 本方案中, 用户向他们披露相关身份信息来证实自己的身份以获得新的身份属性或其他业务需求, 该操作是用户自由选择相关的身份进行组合并请求业务当事人的, 因此不存在不相干的人员获取用户身份信息.

(4) Directed Identity: 本方案中用户可以选择是否披露自身相关属性资料, 也可以直接搜索其他用户公开的属性资料.

(5) Pluralism of Operators and Technologies: 多个场景下, 用户可支持提供不同类型的身份信息, 这些不同类型的身份信息也可以方便地结合.

(6) Human Integration: 在本方案中, 用户的身份信息是否真实与其所获得的其他身份信息息息相关, 所使用的有效身份信息越多, 其所获得的信任背书就越多. 不再依仗 CA 证书的认证, 而是将其作为分布式的信任根据属性的授权关联起来, 最终抵抗身份造假.

(7) Consistent Experience Across Contexts: 无论是何种身份信息, 都是由身份属性所组合构成的.

6.2 功能特性对比

通过对比本方案与文献 [6, 8] 的功能特性, 分析本方案在数据分享和访问控制上的优缺点. 结果如表 1 所示.

表 1　功能特性对比

Table 1　Functional characteristics comparison

Features	Ref. [6]	Ref. [8]	Our scheme
Central manager's effect	Mastering main secret key and generate private keys for other devices.	Combine main public key at System Setup.	None. Nodes negotiate the choice of GP by Blockchain.
Blockchain's effect	Provide consensus mechanism like PBFT.	–	Save encrypted data.
Public cloud storage's effect	Save encrypted data.	Save encrypted data.	–
Smart contract's effect	Save, update, verify the access control policy; save encrypted symmetric keys.	–	Execute business logic; initialize system, organization and authorize attributes.
Role definition	Not obvious, all devices except central manager are users.	One certificate authority, many attribute authorities and users.	Include organizations and users, which have similar features.
Users' identity authentication	Identities are supported by Blockchain and its consensus mechanism, or divided by business id.	Certificate authority is responsible for identity verification.	In different domains, users can combine specific identities by authorized attributes.
Access control on data	Smart contract saves and updates the access control policy.	ABE controls the access control policy.	ABE controls the access control policy.
Decentralization's support	PBFT and smart contract.	Multi-attribute Authorities.	User communicates with others by the chain of trust.
ABE's features	Centralized ABE, attributes are confirmed in system setup.	Decentralized ABE, attributes are confirmed in system setup.	Decentralized ABE, attributes are subsequently generated by organizations or users.

文献 [6] 需要中心管理者掌握系统中的主密钥, 在区块链中该模式是有悖去中心化的理念的. 文献 [8] 引入了多中心的 AA 来解决该问题, 系统中没有主密钥存在在任一组织中, 仅需要中心 CA 认证 AA 和用户的身份. 本方案中进一步取缔了中心 CA 的存在, 依据用户身份属性的背书授权构建基于区块链的身份体系.

在文献 [6] 中, 智能合约保存和更新访问控制策略, 设备用户向智能合约发起请求后, 通过多数节点共识之后, 合约返回相关的加密密钥密文给设备用户, 但区块链上、智能合约掌控的数据都是公开透明的, 这一步其实是不必要的, 包括数据所有者 "更新" 访问控制策略其实也只能是在策略中新增某用户, 并根据新的访问控制策略加密原有密钥并上传, 实际上不能算是真正意义上的 "更新". 而本方案中, 智能合约负责的是所有的业务逻辑, 包括但不限于用户之间的各种交互, 通过身份属性的授权和认证确保用户的可信, 而数据安全交由属性密码保障.

不同于文献 [8], 本方案中的组织角色并不需要在系统初始化时确定下来, 并且组织是由多用户组成的联合体, 即便是组织成员需要更新, 也可以通过再协商成立新的组织, 不会对系统中其他组织和用户的属性造成任何影响. 另外, 文献 [8] 中的 AA 节点的属性授权是全量的, 这意味着 AA 节点需要对授权用户所申请的所有属性具备背书的权力, 并能够通过某种 "渠道" 达成一致. 这点限制了业务的广

表 2 计算开销对比

Table 2 Calculation cost comparison

Steps	Ref. [6]	Ref. [8]	Our scheme
Global setup	$E+P$	$E+P+tE_T$	P
User setup	–	–	E_T+E
Org setup	–	$n(t-2)N+E_T$	User: $n(t-2)N+E_T$ Smart contract: tE_T
User generate attributes	–	–	E
Org generate attributes	–	–	User: $n(t-2)N+E$ Smart contract: tE
User KeyGen	$(3+\|S\|)E+P$	–	E
Org KeyGen	–	Org's member: $(3+\|S\|)E$ Request user: $(2+\|S\|)tE$	Org's member: E Request user: tE
Encryption	$E_T+(3\|C\|+1)E$	$E_T+(3\|C\|+1)E$	$E_T+\|C\|(2E_T+3E)$
Decryption	$P+\|S\|(2P+E_T)$	$P+\|S\|(2P+E_T)$	$\|S\|(2P+E_T)$

度和属性授权的实时性. 本方案的属性授权只关乎自身所控制的属性, 在属性涉及范围广的场景下更加灵活.

6.3 计算开销对比

通过理论数值分析对比本方案与文献 [6,8]. 假定只考虑耗时较长的指数运算与双线性配对运算, 忽略哈希、签名、智能合约调用、网络传播等影响. 令 $|U|$ 表示全局属性个数, $|C|$ 表示加密使用属性个数, $|S|$ 表示用户解密实际用到属性个数, t 和 n 为门限秘密分享算法里的参数, N 表示整数域上的指数运算, E 表示群 G 上的指数运算, E_T 表示群 G_T 上的指数运算, P 表示双线性配对运算. 结果如表 2 所示. 从中可以看到, 相比于文献 [6] 中传统的中心化属性密码, 文献 [8] 在系统初始化和用户申请属性密钥上因引入了门限秘密分享而多了一些计算步骤. 本方案在基础属性密码的选择上略有不同, 采用的是去中心的属性密码进行改进, 可以看到加解密复杂度大同小异, 但是申请组织属性时因为属性密钥构造的不同而简单许多. 较为复杂的组织初始化和生成属性的操作虽然与组织内用户数量相关, 但在方案中频率较低.

6.4 实验仿真

本方案主要通过测试仿真所提出的新的属性密码方案在区块链中各个步骤的运行时间, 进行对比分析. 仿真运行在 Hyperledger Fabric v1.4 官方合约容器环境下, 通过合约命令行调用. 宿主机内存为 8 G, CPU 型号为 i5-8400, 使用 Nik-U 开源的 Golang Pbc 库. 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. 测试基准的各种指数运算和配对运算所需的时间如表 3 所示.

表 4 以计算开销对比中相同的结构展示了本方案与文献 [8] 在简单环境下的运行时间对比. 其中门限秘密分享算法的参数 $t=2$, $n=3$, 加解密均只涉及两个属性. 可以看到本方案下耗时较长的加解密步骤也只需要毫秒级的计算时间, 可以满足实际区块链中的使用. 对比计算开销中的复杂度与实际运算时间可以发现与理论结果基本一致. 观察到文献 [8] 在申请组织属性时耗时较长, 这是因为文献 [8] 使用的基础属性密码的属性密钥结构中需要 3 个子部分, 且与用户授权申请的属性数量相关,

表 3　基准运算运行时间参考

Table 3　Benchmark operation time reference

Operation	Time (ms)
N	0.024
E_T	0.157
P	1.464
E	2.019

表 4　算法运行时间概览 (ms)

Table 4　Algorithm running time (ms)

Steps	Ref. [8]	Our scheme
Global setup	$4.665[E+P+tE_T]$	$3.440[P]$
User setup	–	$2.195[E_T+E]$
Org setup	$2.152[n(t-2)N+E_T]$	User: $2.234[n(t-2)N+E_T]$ Smart contract: $0.369[tE_T]$
User generate attributes	–	$2.032[E]$
Org generate attributes	–	User: $2.069[n(t-2)N+E]$ Smart contract: $4.089[tE]$
User KeyGen	–	$2.140[E]$
Org KeyGen	Org's member: $12.492[(3+\|S\|)E]$ Request user: $23.884[(2+\|S\|)tE]$	Org's member: $1.693[E]$ Request user: $4.085[tE]$
Encryption	$14.652[E_T+(3\|C\|+1)E]$	$13.090[E_T+\|C\|(2E_T+3E)]$
Decryption	$9.365[P+\|S\|(2P+E_T)]$	$6.431[\|S\|(2P+E_T)]$

表 5　组织初始化在不同 (t,n) 的运行时间 (ms)

Table 5　Org initialization's time with different (t,n) (ms)

(t,n)	User generate share	User generate opk_i	Smart contract calculate OPK
$(2,3)$	0.049	2.185	0.369
$(10,30)$	0.342	2.162	4.117
$(20,30)$	0.657	2.179	13.788
$(20,60)$	0.868	2.178	13.778
$(200,300)$	22.877	2.218	1151.472

而本方案中属性申请是可以分离的. 另外加解密中可以使用预先生成的一些固定参数简化处理, 从而稍微减少运算时间.

表 5 通过调整门限秘密分享中的参数 (t,n) 仿真用户初始化的过程验证理论结果. 值得一提的是, 在用户生成 share 的过程中, 可以使用加法和乘法替换原先 $n(t-2)$ 次对 GID 的指数运算, 算法复杂度约为 $O(nt)$. 从表中可以看到用户生成 share 因为只涉及到整数域上的加法和乘法运算, 耗时较短, 对用户的算力负担较轻; 用户生成 opk_i 的时间稳定, 约等于一次 E 运算的时间; 从第 3 和 4 行的数据纵向对比可以看到, 合约计算 OPK 的时间只与 t 有关, 但是当 t 逐渐增大后, 影响运行的时间因素不仅仅与 tE_T 线性相关, 合约组装 opk_i 时还涉及到 $O(t^2)$ 的减法和除法运算. 图 2 展示了在 $n=30$ 的时候, t 从 10 逐渐增加到 20 时合约计算 OPK 的仿真结果和其拟合曲线. 可以看到运行时间和 t 符

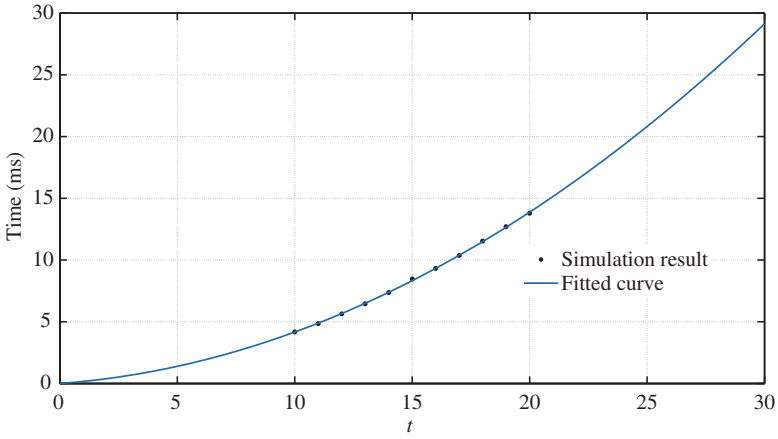


图 2 (网络版彩图) 不同 t 下计算 OPK 的时间拟合曲线

Figure 2 (Color online) Time curve fitting of calculating OPK with different t

表 6 不同属性个数下加解密步骤的运行时间 (ms)

Table 6 Encryption and decryption time with different attributes (ms)

Number of attributes used	Encryption with all "AND"	Encryption with all "OR"	Decryption with all "AND"	Decryption with all "OR"
2	13.079	9.071	6.382	6.462
4	25.838	17.822	12.640	12.618
8	51.526	35.306	25.248	25.170
16	102.436	70.267	50.829	50.058
32	204.648	140.809	99.709	100.947

合二次多项式关系, 拟合后可得二次项系数约为 0.02772 ms, 一次项系数约为 0.1394 ms.

表 6 通过调整加解密中实际使用的属性个数仿真加密过程的运行时间验证理论结果, 可以看到加密步骤运行时间与使用属性个数呈线性关系, 根据预估加密算法涉及 128 个属性的时候耗时约为 0.56~0.8 s, 符合实际使用需求. 并且运行时间还与访问控制策略中逻辑控制符有关, "与" 逻辑控制符比 "或" 逻辑控制符要复杂约 50%. 解密步骤与用户实际使用属性个数也呈线性关系, 符合理论结果.

7 结束语

区块链的身份体制近些年来一直是区块链能否实实在在用之于民的重要考虑因素, 本方案基于去中心化的属性密码提出了一种具体的访问控制身份方案. 通过用户和组织之间的身份属性授权和认证形成了区块链上的信任背书, 以此实现身份信任链上的信任成本转移. 方案中考虑到实际的应用可行性, 不需要任何中心化机构介入区块链或属性密码的控制和管理, 仅依靠在其上的智能合约进行业务逻辑的管理, 由去中心的属性密码实现用户隐私数据的加密储存和分享. 最后, 通过功能特性和实验仿真等分析表明, 本方案可以在现有具备智能合约功能的区块链上引入一种新型的身份机制, 并以此对用户或数据进行细粒度的访问控制. 因此本方案具备广阔的应用前景, 不仅可以为区块链上的身份或隐私数据的访问控制提供具体的思路和方法, 还可以为区块链的监管提供基于身份的解决方案. 未

来的研究方向是如何在不引入第三方可信机构的前提下优化链上密文的存储, 以提高在实际环境中的实用性.

参考文献

1 Huang S, Chen L W, Fan B B. Data security sharing method based on CP-ABE and blockchain. Comput Syst Appl, 2019, 28: 79–86 [黄穗, 陈丽炜, 范冰冰. 基于 CP-ABE 和区块链的数据安全共享方法. 计算机系统应用, 2019, 28: 79–86]

2 Wang X L, Jiang X Z, Li Y, et al. Model for data access control and sharing based on blockchain. J Softw, 2019, 30: 1661–1669 [王秀利, 江晓舟, 李洋, 等. 应用区块链的数据访问控制与共享模型. 软件学报, 2019, 30: 1661–1669]

3 Yang Y T, Cai J L, Zhang Y W, et al. Privacy preserving scheme in block chain with provably secure based on SM9 algorithm. J Softw, 2019, 30: 1692–1704 [杨亚涛, 蔡居良, 张筱薇, 等. 基于 SM9 算法可证明安全的区块链隐私保护方案. 软件学报, 2019, 30: 1692–1704]

4 Zyskind G, Nathan O. Decentralizing privacy: using blockchain to protect personal data. In: Proceedings of 2015 IEEE Security and Privacy Workshops, 2015. 180–184

5 Zhang Q H. Research on identification and access control in blockchain. Dissertation for Master's Degree. Beijing: Beijing Jiaotong University, 2018 [张青禾. 区块链中的身份识别和访问控制技术研究. 硕士学位论文. 北京: 北京交通大学, 2018]

6 Zhang Y, He D, Choo K K R. BaDS: blockchain-based architecture for data sharing with ABS and CP-ABE in IoT. Wirel Commun Mobile Comput, 2018, 2018: 1–9

7 Ding S, Cao J, Li C, et al. A novel attribute-based access control scheme using blockchain for IoT. IEEE Access, 2019, 7: 38431–38441

8 Li W, Xue K, Xue Y, et al. TMACS: a robust and verifiable threshold multi-authority access control system in public cloud storage. IEEE Trans Parallel Distrib Syst, 2016, 27: 1484–1496

9 Wang H, Song Y. Secure cloud-based EHR system using attribute-based cryptosystem and blockchain. J Med Syst, 2018, 42: 152

10 Tian Y L, Yang K D, Wang Z, et al. Algorithm of blockchain data provenance based on ABE. J Commun, 2019, 40: 101–111 [田有亮, 杨科迪, 王缵, 等. 基于属性加密的区块链数据溯源算法. 通信学报, 2019, 40: 101–111]

11 Sahai A, Waters B. Fuzzy identity-based encryption. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2005. 457–473

12 Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, 2006. 89–98

13 Cameron K. The laws of identity. Microsoft Corp, 2005, 12: 8–11

14 Fromknecht C, Velicanu D, Yakoubov S. A decentralized public key infrastructure with identity retention. IACR Cryptol ePrint Archive, 2014, 2014: 803

15 Axon L. Privacy-awareness in blockchain-based PKI. Cdt Technical Paper Series, 2015. https://ora.ox.ac.uk/objects/uuid:f8377b69-599b-4cae-8df0-f0cded53e63b/datastreams/ATTACHMENT01

16 Axon L, Glodsmith M. PB-PKI: a privacy-aware blockchain-based PKI. In: Proceedings of the 14th International Conference on Security and Cryptography, 2017

17 Miers I, Garman C, Green M, et al. Zerocoin: anonymous distributed e-cash from bitcoin. In: Proceedings of 2013 IEEE Symposium on Security and Privacy, 2013. 397–411

18 Sasson E B, Chiesa A, Garman C, et al. Zerocash: decentralized anonymous payments from bitcoin. In: Proceedings of 2014 IEEE Symposium on Security and Privacy, 2014. 459–474

19 Beimel A. Secure schemes for secret sharing and key distribution. Ph.D. Thesis, Israel Institute of Technology, 1996

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

21 Pedersen T P. A threshold cryptosystem without a trusted party. In: Proceedings of Workshop on the Theory and Application of of Cryptographic Techniques. Berlin: Springer, 1991. 522–526

22 Chase M. Multi-authority attribute based encryption. In: Proceedings of Theory of Cryptography Conference, 2007. 515–534

23 Lewko A, Waters B. Decentralizing attribute-based encryption. In: Proceedings of Annual International Conference

on the Theory and Applications of Cryptographic Techniques, 2011. 568–588
24 Dunphy P, Petitcolas F A P. A first look at identity management schemes on the blockchain. IEEE Secur Privacy, 2018, 16: 20–29

Access control scheme on blockchain and decentralized attributed-based algorithm with identity

Zening CHEN[1,2], Liang ZHANG[1,2], Shuangjun ZHANG[1,2] & Haibin KAN[1,2,3*]

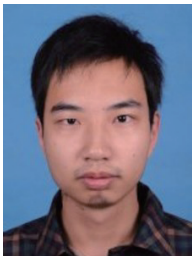
1. *Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai* 200433*, China*;
2. *Fudan-Zhongan Joint Laboratory of Blockchain and Information Security, Shanghai Engineering Research Center of Blockchain, Shanghai* 200433*, China*;
3. *Shanghai Institute for Advanced Communication and Data Science, Shanghai* 200433*, China*
* Corresponding author. E-mail: hbkan@fudan.edu.cn

Abstract The identity system on Blockchain is defective. There is a challenge of how to verify users' identities and ensure the endorsements' realities on Blockchain. The decentralized public key infrastructure running on Blockchain can solve the above problems to some extent, but we can provide an identity-model closer to the real society with the combination of the attribute-based algorithm and Blockchain. This paper proposes an access control scheme based on Blockchain and decentralized the attribute-based algorithm with the identity. It uses mutual authorization between users and organizations to get endorsements of identity's attributes for trust-cost links, and uses the attribute-based algorithm to control and share data on the chain for fine-grained access control and privacy protection. By designing a multi-user collaborative attribute-based algorithm, it offers endorsement capabilities for organizations in the identity-model. Through experimental simulation and comparative analysis, the solution meets the requirements of the current universal Blockchain in terms of security and performance, and provides a universal identity-model.

Keywords blockchain, attributed-based encryption, access control, identity authentication, privacy protection

Zening CHEN was born in 1997. He received a bachelor's degree in communication engineering from Fudan University, Shanghai, in 2018. Currently, he is pursuing his master degree in cyberspace security in Fudan University, Shanghai, China. His research interests include attribute-based encryption, blockchain, and security engineering.

Liang ZHANG was born in 1989. He received a bachelor's degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 2012. Currently, he is pursuing his Ph.D. in cyberspace security in Fudan University, Shanghai, China. His research interests include the industrial internet, blockchain, security engineering and cryptography.



Shuangjun ZHANG was born in 1994. He is a Ph.D. candidate in cyberspace security in Fudan University. His research interests include attribute-based encryption, zero-knowledge proof, verifiable computation, and computational complexity.

Haibin KAN was born in 1971. He received a Ph.D. from Fudan University, Shanghai, China, 1999. From June 2002 to February 2006, he was with the Japan Advanced Institute of Science and Technology as an assistant professor. He went back Fudan University in February 2006, where he is currently a full professor. His research interests include coding theory, cryptography, and computation complexity.