软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software [doi: 10.13328/j.cnki.jos.007208]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于区块链和去中心化可问责属性认证的众包方案*

陶静怡[1,2], 张 亮[3], 阚海斌[1,2,4]

[1](复旦大学 计算机科学技术学院, 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](海南大学 网络空间安全学院 (密码学院), 海南 海口 570228)

[4](复旦大学 义乌研究院, 浙江 金华 322000)

通信作者: 阚海斌, E-mail: hbkan@fudan.edu.cn

摘 要: 众包是一种分布式解决问题的方式, 可以降低成本并有效利用资源. 区块链技术的引入解决了传统众包平台集中化程度过高的问题, 但它的透明性却带来了隐私泄露的风险. 传统的匿名认证虽然可以隐藏用户身份, 但存在匿名滥用的问题, 同时还增加了对工作者筛选的难度. 提出一种去中心化可问责属性认证方案, 并将其与区块链结合设计一种新型众包方案. 该方案利用去中心化属性加密与非交互式零知识证明技术, 在保护用户身份隐私的同时实现可链接性和可追踪性, 并且请求者可以制定访问策略来筛选工作者. 此外, 该方案通过门限秘密分享技术实现了属性授权机构和追踪组, 提高系统的安全性. 通过实验仿真和分析证明该方案在时间和存储开销上符合实际应用需求.

关键词: 区块链; 属性加密; 零知识证明; 众包; 匿名认证; 可问责制

中图法分类号: TP393

中文引用格式: 陶静怡, 张亮, 阚海斌. 基于区块链和去中心化可问责属性认证的众包方案. 软件学报. http://www.jos.org.cn/1000-9825/7208.htm

英文引用格式: Tao JY, Zhang L, Kan HB. Crowdsourcing Scheme Based on Blockchain and Decentralized Accountable Attribute-based Authentication. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7208.htm

Crowdsourcing Scheme Based on Blockchain and Decentralized Accountable Attribute-based Authentication

TAO Jing-Yi[1,2], ZHANG Liang[3], KAN Hai-Bin[1,2,4]

[1](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](School of Cyberspace Security (School of Cryptology), Hainan University, Haikou 570228, China)

[4](Yiwu Research Institute, Fudan University, Jinhua 322000, China)

Abstract: As a distributed approach to problem solving, crowdsourcing reduces costs and efficiently utilizes resources. While blockchain technology is introduced to solve the problem of over-centralization in traditional crowdsourcing platforms, its transparency brings the risk of privacy leakage. The traditional anonymous authentication can hide the user's identity, but the anonymity is abused, and the worker selection gets more difficult. In this study, a decentralized accountable attribute-based authentication scheme is proposed and combined with blockchain to design a novel crowdsourcing scheme. Using decentralized attribute-based encryption and non-interactive zero-knowledge proof, the scheme protects the privacy of users' identities with linkability and traceability, and the requester can devise access policies to select workers. In addition, the scheme improves the security of the system by implementing attribute authorization authority

* 基金项目: 国家重点研发计划 (2019YFB2101703); 国家自然科学基金 (62272107, 62302129); 上海市科技创新行动计划 (21511102200); 广东省重点研发计划 (2020B0101090001); 海南省重点研发项目 (ZDYF2024GXJS030)
收稿时间: 2023-11-07; 修改时间: 2024-02-07; 采用时间: 2024-04-12; jos 在线出版时间: 2024-07-03

and tracking groups through the threshold secret sharing technique. Through experimental simulation and analysis, it is demonstrated that the scheme meets the requirements of time and storage overhead in practical application.

Key words: blockchain; attributed-based encryption; zero-knowledge proof; crowdsourcing; anonymous authentication; accountability

众包提供了一种分布式完成任务的方式, 为众多领域如程序开发、图像标注和医疗数据收集等提供了高效、灵活的任务解决方案. 典型的众包系统由请求者、工作者和众包平台组成, 请求者通过平台发布任务, 工作者参与任务并提交结果. 传统的众包系统依赖于集中的服务器, 因此存在单点故障、数据丢失和隐私泄露等问题[1,2]. 区块链技术通过分散的信任管理缓解了传统众包系统集中化的问题[3,4], 但区块链的透明性为数据隐私保护带来了新挑战. 在众包系统中, 为减少用户使用伪造身份带来的风险, 身份验证通常是基本的要求. 但如果在区块链中使用传统认证方案, 所有请求和提交的历史记录都将被存储在区块链上, 这会导致信息泄露并危及用户的隐私.

匿名性被广泛认为是保护用户隐私的有效手段, 匿名认证的概念最早由 Chaum 在 1983 年提出[5], 并由 Camenisch 等人[6]形式化, 允许用户证明拥有凭据而避免被追踪. 群签名[7]、环签名[8]、基于属性的签名[9]和匿名证书[10]等都是常见的匿名认证机制.

匿名性通过隐藏用户身份来保护隐私, 但匿名性的滥用会增加众包平台的风险. 恶意工作者可能通过重复提交来骗取报酬, 或故意提交错误的工作结果. 通过可问责的匿名认证机制, 可以在保护用户隐私的同时对恶意用户进行问责.

可问责匿名认证中的问责机制主要包括可链接性和可追踪性两种. 其中可链接性是指同一个用户提交的多次认证能被链接, 而可追踪性则是指用户的身份在必要时可以被揭示. 在基于属性的方案中, Gu 等人[11]提出的可追踪属性签名方案和 Kaaniche 等人[12]提出的基于属性签名的匿名证书方案提供了细粒度控制, 同时通过引入可信第三方来追踪签名者的身份. 可追踪方案通常依赖可信方来实现可追踪性, 一些研究者通过门限机制减少风险, 将追踪权利分散到多个机构[13,14]. 大多数方案只实现了一种问责机制, 而可链接与可追溯的群签名[15,16]和环签名[17,18]则同时实现了两种问责机制. 然而, 这些方案存在缺乏细粒度访问控制或需要可信第三方的问题.

针对众包系统可问责匿名认证的需求, 研究人员提出了不同的解决方案. Rahaman 等人[19]提出基于群签名的匿名可问责的众包协议, 群管理员可撤销不端用户的匿名性. Lin 等人[20]利用可追踪群签名匿名认证, 使用属性密码保护任务机密性, 但难以检测工作者重复提交的行为, ZebraLancer[21]实现了公共前缀可链接性, 可以检测出工作者重复提交的行为, 但无法追踪工作者身份, 且需要可信注册机构. Li 等人[22]实现了细粒度访问控制、匿名性和公开问责性, 但只有被链接后才能追踪到工作者.

本文基于 ZebraLancer[21]的公共前缀可链接匿名认证方案, 在其匿名性和可链接性的基础上, 通过将去中心化属性加密与非交互式零知识证明技术结合, 提供了细粒度控制, 使得请求者可以对工作者进行筛选, 同时还进一步实现了可追踪性. 此外, 本文的方案不需要可信注册机构, 去中心化程度更高.

除了可问责匿名认证外, 众包方案还需要确保工作结果和报酬之间的公平交换. 同时, 由于区块确认通常不是实时完成的, 需要避免恶意工作者复制并提交他人的任务结果来获取奖励. 为实现公平交换, 本文基于文献 [23] 的不可否认数据交换协议设计了请求者和工作者之间的交易流程.

本文基于 ZebraLancer[21]提出了一种去中心化可问责属性认证方案, 并与区块链结合设计了一种众包方案. 通过使用去中心化属性加密与非交互式零知识证明技术, 该匿名认证方案满足匿名性、属性隐私性、不可伪造性、可链接性和可追踪性, 并通过属性权威组织的构建提高了属性权威的容错性. 基于该认证方案, 本文提出的众包方案为任务内容提供了细粒度访问控制, 只有符合访问策略的工作者才能获取任务内容并生成有效匿名认证, 提高了任务结果的质量. 本方案在保护用户身份隐私的同时, 可以检测出工作者重复提交的行为, 同时使用门限秘密分享技术构建了追踪组以追踪恶意工作者, 并允许追踪组成员动态加入和退出. 本方案在无需仲裁中心的情况下实现了请求者和工作者之间的公平交换, 经过实验测试, 本方案在时间和存储开销上符合实际应用需求.

本文的主要贡献如下.

(1) 基于 ZebraLancer[21]提出了一种去中心化可问责属性认证方案, 在其匿名性和可链接性的基础上实现了去

中心化属性认证和可追踪性.

(2) 通过使用门限秘密分享技术, 使得属性权威和追踪组均可以由多个用户构成, 增加了方案的容错性, 同时允许追踪组成员动态加入和退出.

(3) 将本文提出的去中心化可问责属性认证方案与区块链相结合, 设计了一种新型众包方案, 请求者能对工作者进行筛选, 在保护工作者身份隐私的同时可以防止工作者重复提交的行为, 并能够对恶意工作者进行追踪.

(4) 本文的众包方案在无需仲裁中心的情况下就能实现请求者和工作者之间的公平交换.

本文第 1 节介绍本文所需的基础知识. 第 2 节介绍本文提出的去中心化属性匿名认证方案. 第 3 节详细介绍本文基于第 2 节中提出的匿名认证与区块链结合设计的众包方案. 第 4 节对本文的方案进行分析, 包括正确性、安全性与公平性分析. 第 5 节通过实验与分析比较了本文方案与现有方案, 同时通过实验仿真验证了方案在时间和存储开销上符合实际应用需求. 最后总结全文.

1 基础知识

1.1 双线性映射

设 $\mathbb{G}_1,\mathbb{G}_T$ 为 p 阶循环群, p 为素数. 若存在映射 $e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_T$ 满足如下性质, 则称 e 为双线性对.

(1) 双线性: 对 $\forall g,h\in\mathbb{G}_1,\forall a,b\in\mathbb{Z}_p$, 都有 $e(g^a,h^b)=e(g,h)^{ab}$.

(2) 非退化性: 存在 $g\in\mathbb{G}_1$, 使得 $e(g,g)\neq 1$.

(3) 可计算性: 对 $\forall g,h\in\mathbb{G}_1$, 存在一个有效的算法来计算 $e(g,h)$.

1.2 门限秘密分享

门限秘密分享可以增加多方之间的容错, 使得恶意攻击者需要付出更大的代价攻击更多的节点来得到秘密, 同时还增加了诚实节点可能发生宕机的容错性. 1979 年, Shamir[24]和 Blakey[25]分别通过拉格朗日插值法和多维空间点的性质实现了秘密分享协议. Pedersen[26]根据 Shamir 的方案提出了一种无需可信第三方的门限秘密分享算法, 该算法的流程如下.

首先考虑一个要分享秘密的群组, 参与者为 P_i, $i=1,2,\ldots,n$, 每个参与者拥有唯一的标识 id_i. 假设最终要协作分享的秘密为 S, 这个秘密由各参与者随机生成的秘密 S_i 组成, 即 $S=\sum_{i=1}^{n}S_i$.

接下来, 每个参与者 P_i 随机生成一个 $t-1$ 阶多项式 $f_i(x)$, 其中 $f_i(0)=S_i$, 定义 $F(x)=\sum_{i=1}^{n}f_i(x)$. P_i 计算 $f_i(id_j)$, 并将这个值发送给参与者 P_j. 当参与者 P_i 接收到其他 $n-1$ 个参与者分享的值时, 可以结合自己生成的 $f_i(id_i)$ 计算出自己的部分秘密 $F(id_i)=\sum_{j=1}^{n}f_j(id_i)$. 由于 $F(x)$ 是一个 $t-1$ 阶多项式, 因此可以通过拉格朗日插值法最终计算出 $F(0)=S$.

本文将门限秘密分享算法与属性密码相结合实现了属性权威组织, 同时还将其应用于构造追踪组来分散信任、提供容错性, 并进一步实现了成员动态变化的算法.

1.3 属性密码

Goyal 等人[27]在 2006 年提出了基于属性的加密算法, 其中在基于密文策略的加密算法 (ciphertext-policy attribute-based encryption, CPABE) 中密钥和属性集绑定, 密文和访问控制策略绑定. 因为基础的 CPABE 存在中心化解密和分发密钥的局限性, 去中心属性密码被提出[28,29], 任何用户都可以声明属性成为授权机构, 适用于分布式体系. Rouselakis 等人提出的去中心属性密码[28]的基本构造和过程如下.

(1) $GlobalSetup(\lambda)\to GP$: 传入安全参数 λ 计算输出全局参数 GP.

(2) $AuthSetup(GP,aid)\to PK,SK$: 以 GP 和权威机构的标识符 aid 为输入生成公私钥对 $SK=\{OSK=\alpha,ASK=y\}$, $PK=\{OPK=e(g,g)^{\alpha},APK=g^{y}\}$.

(3) $KeyGen(GID,aid,x,SK,GP)\to K_{x,GID},K'_{x,GID}$: 属性密钥生成算法通过请求者的身份标识符 GID、全局参数 GP、属于权威机构 aid 的属性 x 和该权威机构的私钥 SK, 为该属性、身份对生成属性密钥 $K_{x,GID}$, $K'_{x,GID}$.

(4) $Encrypt(M,(A,\rho),\{PK\},GP)\rightarrow CT$: 加密算法通过消息 M 、访问结构、相关属性权威的公钥集和全局参数, 计算输出密文 CT .

(5) $Decrypt(CT,\{K_{x,GID},K'_{x,GID}\},GP)\rightarrow M$: 解密算法输入密文、用户 GID 的属性密钥集合和全局参数. 当用户所拥有的属性集合满足对应于密文的访问结构时, 输出消息 M, 否则解密失败.

1.4 零知识证明

零知识证明是指证明者在不泄露论据的情况下, 向验证者证明论题为正确的. 一个零知识证明系统具备完备性 (completeness)、可靠性 (soundness) 和零知识性 (zero-knowledge). 其中可靠性是指如果论题是假的, 任何证明者都不能说服验证者相信它是真的, 有时我们需要更强的可靠性, 即知识证明 (proof of knowledge). 知识证明是指对于任何证明者, 都存在一个与证明者交互的提取算法, 该算法可以输出论据.

对于 NP 完全语言 $L=\{x|w \text{ s.t. } C(x,w)=1\}$, 零知识证明算法可以用 (S,P,V) 表示.

(1) $S(C)\rightarrow crs$: 初始化算法为证明者和验证者输出公共参数.

(2) $P(crs,x,w)\rightarrow\pi$: 证明算法可以使用论据 w 生成证明 π , 证明 $x\in L$.

(3) $V(crs,x,\pi)\rightarrow 1/0$: 验证算法可以验证证明的正确性.

零知识证明系统分为交互式和非交互式两种类型, 而 zk-SNARKs[30]是非交互式零知识证明的一种优化形式, 它减小了证明大小和验证时间, 显著降低了计算和通信成本.

1.5 ZebraLancer[21]的公共前缀可链接匿名认证方案

本文所提出的去中心化可问责属性认证方案基于 ZebraLancer[21]中的公共前缀可链接匿名认证方案实现, 该方案的基本流程如下.

(1) $Setup(\lambda)$. 生成 ZK 的公共参数 PP , 同时为数字签名算法生成公私钥对 (msk,mpk) .

(2) $CertGen(msk,pk_i)$. 对 pk_i 执行签名算法, pk_i 是用户的公钥, 生成的签名为 σ_i , 输出证书 $cert_i=\sigma_i$.

(3) $Auth(p\|m,sk_i,pk_i,cert_i,PP)$. 输入前缀为 p 的消息 $p\|m$, 计算 $t_1=H(p,sk_i),t_2=H(p\|m,sk_i)$, H 是安全的哈希函数, 对如下语言执行 $ZK.Prove$: $L_T=\{t_1,t_2,\vec{x}=(p\|m,mpk):\exists\vec{w}=(sk_i,pk_i,cert_i)$ s.t. $CertVerify(cert_i,pk_i,mpk)=1\wedge pair(pk_i,sk_i)\wedge t_1=H(p,sk_i)\wedge t_2=H(p\|m,sk_i)\}$, 其中 $CertVerify$ 算法是签名验证算法, 用来验证证书的有效性, 而 $pair$ 则验证公私钥对. 得到证明 η 后, 算法输出 $\pi=(t_1,t_2,\eta)$.

(4) $Verify(p\|m,\pi,mpk,PP)$. 验证算法执行 $ZK.Verify$ 通过则输出 1, 否则为 0.

(5) $Link(m_1,\pi_1,m_2,\pi_2)$. 对 $\pi_1=(t_1^1,t_2^1,\eta_1)$, $\pi_2=(t_1^2,t_2^2,\eta_2)$ 算法验证是否有 $t_1^1=t_1^2$, 如相等则输出 1, 否则为 0.

2 去中心化可问责属性认证方案

本文提出了一种去中心化可问责属性认证方案, 该方案基于 ZebraLancer[21]提出的公共前缀可链接匿名认证方案实现, 在其基础上实现了属性认证和可追踪性, 并通过门限秘密分享实现了权威机构组织和追踪组. 本方案使用了 RW[28]去中心属性密码, 主要由以下几个算法组成.

(1) $GlobalSetup(\lambda)\rightarrow GP$. λ 为安全参数, 初始化算法生成全局参数 GP .

(2) $UserSetup(GP)\rightarrow GID,S_u$. 生成用户的私钥 S_u 和全局唯一 GID .

(3) $AASetup(n,t,aid,GP)\rightarrow PK,\{SK_i\}$. 属性权威机构 aid 的初始化, 由 n 个用户组成, 门限值为 t , 生成属性权威的公钥和部分私钥.

(4) $TracerSetup(n,t,GP)\rightarrow TPK,\{TSK_i\}$. 追踪组初始化, 由 n 个用户组成, 门限值为 t , 生成追踪组的公钥和部分私钥.

(5) $KeyGen(aid,x,GID,\{SK_i\},GP)\rightarrow K_{x,GID},K'_{x,GID}$. 通过 t 个属性权威 aid 的部分私钥 SK_i , 生成用户 GID 属性 x 的属性密钥 $K_{x,GID},K'_{x,GID}$.

(6) $Auth(M,\{K_{x,GID},K'_{x,GID}\},S_u,GID,(A,\rho),\{PK_j\}_{j=1}^{|\Omega|},TPK,GP)\rightarrow\eta$. 输入用户 GID 满足访问策略 (A,ρ) 的属性密钥及其秘密值 S_u , 结合策略相关的属性权威公钥 PK_j , 追踪组公钥 TPK 和 GP , 为 M 生成匿名认证 η .

(7) $Verify(M,\eta,(A,\rho),\{PK_j\}_{j=1}^{|\Omega|},TPK,GP)\rightarrow 1/0$. 验证匿名认证, 验证通过输出 1 否则为 0.

(8) $Link(M,M',\eta,\eta')\rightarrow 1/0$. 对两个匿名认证进行链接, 如结果为 1 则代表同一个用户对前缀相同的两个消息生成了认证, 否则为 0.

(9) $Trace(\eta,\{TSK_i\})\rightarrow GID$. 通过 t 个追踪组的部分私钥可以由认证获得生成该认证用户的身份.

3 基于区块链和去中心化可问责属性认证的众包方案

本文将所提出的去中心化可问责属性认证方案与区块链技术结合, 设计了众包场景下请求者和工作者交互的流程, 实现了一种新型众包系统. 本文提出的基于区块链和去中心化可问责属性认证的众包方案包含 4 类角色, 分别是属性权威、请求者、工作者和追踪者, 所有角色均加入区块链, 成为系统的用户. 其中属性权威是分布式的且每个权威可以由多个用户组成, 任意属性权威可以声明自己的属性, 并将属性私钥授权给有该属性的用户. 请求者制定访问策略并将加密后的任务通过平台发布, 只有属性满足访问策略的工作者可以执行任务并获得奖励. 追踪组由多个用户组成, 可以协作获得工作者的身份. 图 1 中的步骤为本方案的整体流程, 以医学图像标注的任务为例, 首先进行初始化与属性申请, 属性权威是去中心化的, 且属性权威和追踪组都可以由多个成员组成. 请求者发布一个医学图像标注的任务到基于区块链的众包平台, 请求者可以设置访问策略如要求工作者要有医学相关领域的属性, 从而提高标注结果的质量. 请求者将任务描述、访问策略、通过属性加密后的标注数据集、提交的截止时间和奖励信息等发布到众包平台, 同时传入押金. 有医学相关领域属性的工作者才能解密获取任务的完整数据进行标注并生成有效认证. 当标注完成后, 工作者将用密钥 S_2 对标注结果进行加密得到密文 S_1, 并计算 $Tag=H(S_1)\oplus H(S_2)$, 记 $m=H(S_1)$, 然后生成匿名认证, 将 Tag、m、匿名认证和押金一起提交到众包平台. 请求者通过智能合约验证工作者的匿名认证, 同时检查该认证是否能与之前收到的认证链接, 如果工作者重复提交则能被链接到. 验证通过后将提交的内容放入工作结果集合中, 否则不放入且不退还押金. 超过提交截止时间后, 将停止接收工作结果, 通过验证的工作者要在规定时间内通过链下安全通道发送 S_1. 请求者通过智能合约对收到的 S_1 计算 $H(S_1)$, 并与链上的 m 进行比较, 一致则确认该工作者的提交, 同时请求者的奖励也会放入合约. 被确认的工作者要在规定时间内发布 S_2, 合约重新计算 Tag 进行验证, 验证不通过的工作者不能得到奖励和押金, 同时会被追踪组追踪. 最后请求者可以用 S_2 解密 S_1 得到标注结果, 并分配奖励给工作者.

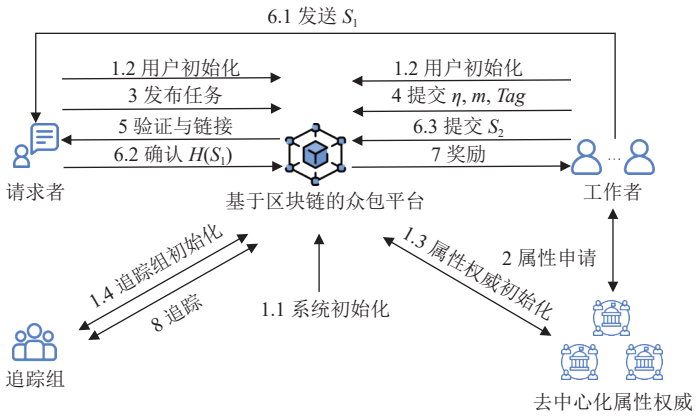

图 1 众包方案流程

具体算法可分为以下几个阶段.

3.1 初始设置阶段

(1) $GlobalSetup(\lambda)\rightarrow GP$. 在系统初始化阶段, 输入安全参数 λ, 随机选择一个阶为素数 p 的双线性群 $\mathbb{G}$, 生成元为 g. 哈希函数: $H^*:\{0,1\}^*\rightarrow\{0,1\}^*, H:\{0,1\}^*\rightarrow\mathbb{G}, H_1:\{0,1\}^*\times\{0,1\}^*\rightarrow\mathbb{G}, H_2:\{0,1\}^*\times\{0,1\}^*\rightarrow\mathbb{G}, F:\{0,1\}^*\rightarrow\mathbb{G}$,

同时执行 zk-SNARKs 的初始化算法生成参数 crs. 将公共参数 $GP=\{p,\mathbb{G},g,crs,H^*,H,H_1,H_2,F\}$ 公开上链.

(2) $UserSetup(GP)\rightarrow GID,S_u$. 用户注册时, 选择秘密值 $S_u\xleftarrow{R}\{0,1\}^*$, 计算用户的唯一标识 $GID=H^*(S_u)$, 由智能合约保证 GID 在区块链上全局唯一.

(3) $AASetup(n,t,aid,GP)\rightarrow PK,\{SK_i\}$. 属性授权机构 aid 由 n 个成员组成, 门限值为 t. 首先, 机构的每个成员 $Member_i$ 选择 $\alpha_i\xleftarrow{R}Z_p,y_i\xleftarrow{R}Z_p$ 和 $f_i\xleftarrow{R}Z_p[x],f_i'\xleftarrow{R}Z_p[x]$ 满足 $deg(f_i)=t-1,deg(f_i')=t-1$ 且 $f_i(0)=\alpha_i,f_i'(0)=y_i$. 则机构的私钥为 $SK=\{OSK=\sum_{i=1}^{n}\alpha_i,ASK=\sum_{i=1}^{n}y_i\}$. 接着针对全部 n 个成员生成秘密 $share_{ij}=\{f_i(GID_j),f_i'(GID_j)\}$, 并将其通过可信信道秘密分享给 $Member_j,j\in[1,n],j\neq i$, 自己保留 $share_{ii}$. 所有成员都给其他 $n-1$ 个成员发送了对应的秘密, 也拥有了 n 个不同成员生成的秘密. $Member_i$ 接收到 n 个不同成员生成的秘密 $share_{ji},j\in[1,n]$, 可以计算得到部分机构私钥 $OSK_i=\sum_{j=1}^{n}f_j(GID_i),ASK_i=\sum_{j=1}^{n}f_j'(GID_i)$ 和部分机构公钥 $OPK_i=e(g,g)^{OSK_i},APK_i=g^{ASK_i}$. 智能合约选取 t 个部分机构公钥生成属性机构公钥 $PK=\{OPK,APK\}$ 并公开上链.

$$OPK=\prod_{i=1}^{t}OPK_i^{\prod_{j=1,j\neq}^{t}\frac{GID_j}{GID_j-GID_i}},\ APK=\prod_{i=1}^{t}APK_i^{\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}}.$$

(4) $TracerSetup(n,t,GP)\rightarrow TPK,\{TSK_i\}$. 追踪者协作分享秘密, $Tracer_i$ 选择 $z_i\xleftarrow{R}Z_p$, $\theta_i\xleftarrow{R}Z_p[x]$, 满足 $deg(\theta_i)=t-1$ 且 $\theta_i(0)=z_i$. 将 $\theta_i(GID_j)$ 发送给 $Tracer_j$, $j\in[1,n],j\neq i$. $Tracer_i$ 计算部分公私钥为 $TSK_i=\sum_{j=1}^{n}\theta_j(GID_i),TPK_i=g^{TSK_i}$. 智能合约挑选 t 个追踪者生成的部分公钥 TPK_i 可以生成追踪组公钥 TPK 并公开上链.

$$TPK=\prod_{i=1}^{t}TPK_i^{\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}}.$$

3.2 属性申请阶段

$KeyGen(aid,x,GID,\{SK_i\},GP)\rightarrow K_{x,GID},K'_{x,GID}$. 用户的身份标识为 GID, 向属性权威申请属性 x, t 个属性权威的成员协作生成属性密钥, 成员 i 选择 $\beta_i\xleftarrow{R}Z_p$, 计算用户关于属性 x 的部分属性密钥:

$$partK_{x_i,GID}=g^{OSK_i}H(GID)^{APK_i}F(x)^{\beta_i},\ partK'_{x_i,GID}=g^{\beta_i}.$$

智能合约通过 t 个部分属性私钥计算用户 GID 的属性密钥:

$$K_{x,GID}=\prod_{i=1}^{t}partK_{x_i,GID}^{\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}},\ K'_{x,GID}=\prod_{i=1}^{t}partK'_{x_i,GID}{}^{\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}}.$$

用户可以通过 $e(g,K_{x,GID})=OPK\cdot e(APK,H(GID))\cdot e(K'_{x,GID},F(x))$ 验证属性密钥正确性.

3.3 任务发布阶段

$Encrypt(M,(A,\rho),\{PK_j\}_{j=1}^{|\Omega|},GP)\rightarrow C$. 请求者制定访问策略对任务内容进行加密, $|\Omega|$ 为访问策略中所涉及属性对应的属性机构公钥的数量. 首先选择一个随机数 $s\xleftarrow{R}Z_p$ 和随机向量 $v\xleftarrow{R}Z_p^l$, 其中 s 是向量 v 的第 1 个元素. 再选择随机向量 $w\xleftarrow{R}Z_p^l$, 其中第 1 个元素为 0. 最后对于矩阵 A 中的每一行 A_k, 选择一个随机数 $r_k\xleftarrow{R}Z_p$, 则可以使用如下公式加密任务信息 M 得到密文 $C=\{C_0,\{C_{1,k},C_{2,k},C_{3,k},C_{4,k}\ \forall k\}\}$, 其中,

$$\begin{cases}C_0=M\cdot e(g,g)^s\\ C_{1,k}=e(g,g)^{A_k\cdot v}e(g,g)^{\alpha_{\rho(k)}r_k},C_{2,k}=g^{-r_k},C_{3,k}=g^{y_{\rho(k)}r_k}g^{A_k\cdot w},C_{4,k}=F(\delta(k))^{r_k},\end{cases}$$

δ 将 A 的每一行映射为一个属性字符串. 请求者发布任务号 tid, 访问策略, 任务密文, 奖励信息, 签名和时间限制, 并把押金传入合约.

3.4 提交认证阶段

(1) $Decrypt(C,\{K_{x,GID},K'_{x,GID}\},GP)\rightarrow M$. 符合策略的工作者首先需要计算 $\omega_k\in Z_p,\forall k$, 使得 $\sum_k\omega_kA_k=(1,0,\ldots,0)$. 之后针对每一行 k, 计算:

$$C_{1,k}\cdot e(K_{\delta(k),GID},C_{2,k})\cdot e(H(GID),C_{3,k})\cdot e(K'_{\delta(k),GID},C_{4,k})=e(g,g)^{A_k\cdot v}\cdot e(H(GID),g)^{A_k\cdot w}.$$

接下来针对每行获得的值进行乘方和累乘, 可以得到: $\prod_k(e(g,g)^{A_k\cdot v}\cdot e(H(GID),g)^{A_k\cdot w})^{\omega_k}=e(g,g)^s$ 最终可以求得明文 $M=C_0/e(g,g)^s$.

(2) $Auth(tid\|m,\{K_{x,GID},K'_{x,GID}\},S_u,GID,(A,\rho),\{PK_j\}_{j=1}^{|\Omega|},TPK,GP)\to\eta$. 工作者在完成任务后, 随机选择对称密钥 S_2 对工作结果加密得到密文 S_1, 并计算 $Tag=H(S_1)\oplus H(S_2)$, 记 $m=H(S_1)$. 然后使用 TPK 对 GID 加密, 选择 $l\xleftarrow{R}Z_p$, 则 $Ex_1=GID\cdot TPK^l, Ex_2=g^l$, 记 $\sigma=Enc_{tr}(TPK,GID,l,GP)=\{Ex_1,Ex_2\}$. 对任务 tid 计算标记 $t_1=H_1(tid,S_u), t_2=H_2(tid\|m,S_u)$. 对如下语言执行 $ZK.Prove(\vec{x},\vec{w},crs)$ 生成证明 π:

$$L_1=\{\vec{x}=(tid\|m,(A,\rho),\{PK_j\}_{j=1}^{|\Omega|},t_1,t_2,\sigma,TPK,GP):\exists\vec{w}=(GID,\{K_{x,GID},K'_{x,GID}\},\omega,S_u,l)\ \text{s.t.}\ \omega A=(1,0,\ldots,0) \tag{1}$$

$$\wedge_{i=1}^{|\Omega|}(\omega_i=0\vee e(g,K_{\delta(i),GID})=OPK_{\rho(i)}\cdot e(APK_{\rho(i)},H(GID))\cdot e(K'_{\delta(i),GID},F(\delta(i)))) \tag{2}$$

$$\wedge\ t_1=H_1(tid,S_u)\wedge t_2=H_2(tid\|m,S_u) \tag{3}$$

$$\wedge GID_i=H^*(S_u)\wedge\sigma=Enc_{tr}(TPK,GID,l,GP) \tag{4}$$

工作者将 Tag,m 和匿名认证 $\eta=(t_1,t_2,\sigma,\pi)$ 一起提交, 同时发送押金.

3.5 验证认证阶段

(1) $Verify(tid\|m,\eta,(A,\rho),\{PK_j\}_{j=1}^{|\Omega|},TPK,GP)\to 1/0$. 请求者收到工作者的提交信息后, 通过智能合约执行 $ZK.Verify(\vec{x},\pi,crs)$ 验证证明, 如验证通过则输出 1, 否则输出 0.

(2) $Link(M,M',\eta,\eta')\to 1/0$. 对工作者的两个不同信息 $M=tid\|m,M'$ 和认证 η,η' 比较是否有 $t_1=t'_1$, 如相等则代表同一个工作者对这个任务提交了两次工作结果, 输出 1, 否则输出 0.

匿名验证不通过和提交两次的用户不会被放入该任务的工作者列表中, 且不会返还押金.

3.6 任务结果提交阶段

(1) 通过后验证后工作者在规定时间内通过链下安全通道发送加密后的工作结果 S_1.

(2) 在到达规定时间后, 请求者对收到的所有 S_1 计算 $H(S_1)$, 并与链上的 m 进行比较, 一致则确认该工作者的提交. 请求者将奖励放入合约, 未被确认的工作者的押金被退还.

(3) 被确认的工作者在规定时间内发布 S_2.

3.7 奖励阶段

(1) 智能合约验证 $Tag=H(S_1)\oplus H(S_2)$, 验证不通过的工作者不能得到奖励和押金, 同时通知追踪者追踪.

(2) 请求者用 S_2 解密 S_1 得到工作结果, 可以在规定时间内提交奖励分配的零知识证明, 对应的语言下:

$$L_2=\{\vec{x}=(\{S_{2,j}\},\{R_j\},\{m_j\}):$$

$$\exists\vec{w}=(\{S_{1,j}\},\{Ans_j\})\ \text{s.t.}\wedge_{j=1}^{n}Ans_j=Dec(S_{1,j},S_{2,j})\wedge_{j=1}^{n}H(S_{1,j})=m_j\wedge_{j=1}^{n}R_j=R(Ans_j;Ans_1,\ldots,Ans_n,\tau)\},$$

其中, n 为参与任务的工作者数量, R 为请求者在发布任务时制定的奖励规则, Ans_j 为工作结果的明文. 智能合约验证通过后会发送给工作者奖励 R_j, 从而避免工作者提交无效结果骗取奖励的情况. 如请求者未提交则合约将平均分配奖励, 并返还工作者和请求者的押金.

3.8 追踪阶段

$Trace(\eta,\{TSK_i\})\to GID$. 当工作者不能通过验证时, 将得不到奖励和押金, 同时会使用追踪机制. 追踪者生成追踪线索, 至少 t 个追踪者协作可以获得工作者身份. 通过认证 η 中的 σ, $Tracer_i$ 计算 $Ex_2^{TSK_i}$. 当系统收集到 t 份时, 可以解密出:

$$GID=Ex_1\Big/\prod_{i=1}^{t}(Ex_2^{TSK_i})^{\Pi_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}}.$$

3.9 追踪者动态变化

(1) 追踪者加入

假设 $Tracer_r$ 加入, 追踪者变为 $n+1$ 个, 原来的 n 个成员 $Tracer_i$ 选择 $h_i\xleftarrow{R}Z_p[x]$, 满足 $deg(h_i)=t-1$ 且 $h_i(GID_r)=0$. 接着针对原来的 n 个成员生成 $h_i(GID_j)$, 并将其通过可信信道秘密分享给 $Tracer_j, j\in[1,n], j\neq i$. $Tracer_j$ 计算 $TSK'_j=TSK_j+\sum_{j=1}^{n}h_i(GID_j)$ 并发给 $Tracer_r$, $Tracer_r$ 收到至少 t 个 TSK'_j 后, 计算 $TSK_r=\sum_{j=1}^{t}TSK'_j\prod_{k=1,k\neq j}^{t}\frac{GID_r-GID_k}{GID_j-GID_k}$,

$TPK_r = g^{TSK_r}$ 作为 $Tracer_r$ 的部分公私钥.

(2) 追踪者退出

假设 $Tracer_n$ 退出, 追踪者变为 $n-1$ 个, 剩下的 $n-1$ 个成员 $Tracer_i$ 选择 $q_i \overset{R}{\leftarrow} Z_p[x]$, 满足 $deg(q_i) = t-1$ 且 $q_i(0) = 0$. 接着针对剩下 $n-1$ 个成员生成 $q_i(GID_j)$, 并将其通过可信信道秘密分享给 $Tracer_j, j \in [1, n-1], j \neq i$. $Tracer_j$ 计算 $TSK'_j = TSK_j + \sum_{j=1}^{n-1} q_i(GID_j), TPK'_j = g^{TSK'_j}$ 作为新的部分公私钥.

4 方案分析

根据上述关于本方案的具体构造, 下面给出关于本方案的正确性及安全性分析.

4.1 正确性分析

1) 组织公钥与属性密钥生成正确性: 本方案的属性加解密算法基于 RW 属性密码[28]实现, 因此加解密算法正确性由 RW 算法保证. 同时本方案引入了门限秘密分享来构造属性权威组织和追踪组. 属性权威组织的 OPK 生成过程如下:

$$OPK = e(g,g)^{OSK} = e(g,g)^{\sum_{i=1}^{n}\alpha_i} = e(g,g)^{\sum_{i=1}^{n} f_i(0)} = e(g,g)^{\sum_{i=1}^{t}\left(OSK_i \prod_{j=1,j\neq i}^{t} \frac{GID_j}{GID_j - GID_i}\right)}$$
$$= \prod_{i=1}^{t} OPK_i^{\prod_{j=1,j\neq i}^{t} \frac{GID_j}{GID_j - GID_i}}.$$

授权组织公钥 APK 和追踪组公钥 TPK 的生成算法同理. 属性私钥的生成过程如下:

$$\begin{cases} K_{x,GID} = g^{OSK} H(GID)^{APK} F(x)^{\beta} = g^{\sum_{i=1}^{n}\alpha_i} H(GID)^{\sum_{i=1}^{n} y_i} F(x)^{\beta} = \\ g^{\sum_{i=1}^{t}(OSK_i \prod_{j=1,j\neq i}^{t} \frac{GID_j}{GID_j - GID_i})} H(GID)^{\sum_{i=1}^{t}(ASK_i \prod_{j=1,j\neq i}^{t} \frac{GID_j}{GID_j - GID_i})} F(x)^{\sum_{i=1}^{t}(\beta_i \prod_{j=1,j\neq i}^{t} \frac{GID_j}{GID_j - GID_i})} = \prod_{i=1}^{t} partK_{x_i,GID}^{\prod_{j=1,j\neq i}^{t} \frac{GID_j}{GID_j - GID_i}} \\ K'_{x,GID} = g^{\beta} = g^{\sum_{i=1}^{t}(\beta_i \prod_{j=1,j\neq i}^{t} \frac{GID_j}{GID_j - GID_i})} = \prod_{i=1}^{t} partK'_{x_i,GID}^{\prod_{j=1,j\neq i}^{t} \frac{GID_j}{GID_j - GID_i}}. \end{cases}$$

2) 验证正确性: 证明 π 通过 $ZK.Prove(\vec{x}, \vec{w}, crs)$ 生成, 在语言 L_1 中, 公式 (1)、公式 (2) 证明了工作者拥有正确的满足访问策略的属性密钥, 公式 (3) 证明 t_1, t_2 正确计算, 公式 (4) 证明了工作者用追踪公钥正确加密了自己的身份标识 GID, 由 ZK 的知识证明性确保工作者拥有满足策略的属性, 且正确计算了 t_1, t_2, σ.

3) 链接正确性: 当同一个用户对同一个 tid 提交了两次不同的信息 M, M' 时, 如提交的认证 η, η' 能通过验证, 则有 $t_1 = H_1(tid, S_u) = t'_1$, 因此 $Link(M, M', \eta, \eta') \to 1$.

4) 追踪正确性: 用户 GID 的认证为 $\eta = (t_1, t_2, \sigma, \pi)$, 其中 σ 为用 TPK 加密 GID 得到的, 由 ElGamal 算法[31]的正确性确保 $Trace(\eta, \{TSK_i\}) \to GID$. 由于引入了门限秘密分享, 追踪时计算 GID 的过程为:

$$GID = Ex_1 / Ex_2^{TSK} = Ex_1 / Ex_2^{\sum_{i=1}^{n} z_i} = Ex_1 / \prod_{i=1}^{t} (Ex_2^{TSK_i})^{\prod_{j=1,j\neq i}^{t} \frac{GID_j}{GID_j - GID_i}}.$$

5) 追踪者加入和退出正确性:

追踪者加入正确性: $Tracer_r$ 加入, 追踪者变为 $n+1$ 个, 原来的 n 个成员 $Tracer_i$ 选择的 h_i 满足 $h_i(GID_r) = 0$. 则由于 $TSK'_j = TSK_j + \sum_{j=1}^{n} h_i(GID_j)$, 计算 $TSK_r = \sum_{j=1}^{t} TSK'_j \prod_{k=1,k\neq j}^{t} \frac{GID_r - GID_k}{GID_j - GID_k} = \sum_{i=1}^{n} \theta_i(GID_r) + \sum_{i=1}^{n} h_i(GID_r) = \sum_{i=1}^{n} \theta_i(GID_r)$. 则 TSK_r 为函数 $\sum_{j=1}^{n} \theta_j(x)$ 在 GID_r 处的值, 且再有新追踪者加入仍能执行该加入算法为新成员生成追踪公私钥, 因此追踪者加入算法是正确的.

追踪者退出正确性: 假设 $Tracer_n$ 退出时, 原先的函数为 $\sum_{i=1}^{n} \theta_i(x)$, 现更新为 $\sum_{i=1}^{n} (\theta_i(x) + q_i(x))$. $Tracer_n$ 的部分私钥 $TSK_n = \sum_{i=1}^{n} \theta_i(GID_n)$ 不再是该函数上的点, 无法执行插值算法, 因此 $Tracer_n$ 的部分私钥失效. 而由于 $q_i(0) = 0$, 则 TSK 不变, 原先用 TPK 加密生成的认证仍能被追踪, 因此追踪者退出算法是正确的.

4.2 安全性分析

本方案的安全模型满足匿名性、属性隐私性、不可伪造性、可链接性和可追踪性. 在给出安全性的形式化定

义之前, 给出一个预言机服务.

假设有挑战者 C 与敌手 $\mathcal{A}$,C 初始化系统, 设定权威机构和追踪组的门限值为 (t,n), 按第 3.1 节所述步骤将 GP,PK,TPK 等信息公开. C 向 $\mathcal{A}$ 提供以下服务.

(1) 用户的 GID 询问: $\mathcal{A}$ 询问用户 i, C 返回对应用户的 GID_i.

(2) 属性权威公私钥询问: $\mathcal{A}$ 询问任一属性权威的公私钥, C 返回 PK,SK.

(3) 属性密钥询问: $\mathcal{A}$ 提交 $<x,GID>$, C 执行属性密钥生成算法返回用户 GID 关于属性 x 的属性密钥 $K_{x,GID}$, $K'_{x,GID}$.

(4) 认证询问: $\mathcal{A}$ 提交 $<GID,tid,m,\{K_{x,GID},K'_{x,GID}\},(A,\rho)>$, C 执行匿名认证生成算法返回对应的认证 $\eta=(t_1,t_2,\sigma,\pi)$.

(5) 认证验证询问: $\mathcal{A}$ 提交 $<tid,m,\eta,(A,\rho)>$, C 执行验证算法返回验证结果.

(6) 链接询问: $\mathcal{A}$ 提交 $<M,M',\eta,\eta'>$, C 执行链接算法返回结果.

(7) 追踪询问: $\mathcal{A}$ 提交 η, C 执行追踪算法返回 GID.

(8) 用户秘密值询问: $\mathcal{A}$ 提交 GID, C 返回对应的 S_u.

4.2.1 匿名性

请求者和工作者在发布和执行任务时可以使用新的地址, 防止地址被链接. 匿名性要求工作者的回应过程及数据不泄露其身份信息, 在本方案中只有匿名认证部分与工作者的身份信息相关. 假设敌手 $\mathcal{A}$ 可以控制 $t-1$ 个追踪者, 匿名性的形式化定义如下.

游戏 1.0: 挑战者 C 构造身份为 GID_0 和 GID_1 的用户. 敌手 $\mathcal{A}$ 选取 $GID,M_i=\{tid_i,m_i\}$ 和访问控制策略 A_i 提交给 C, 询问认证 η. 同时 $\mathcal{A}$ 可以询问属性权威公私钥, $\mathcal{A}$ 可以询问 C 不多于 $p(\lambda)$ 次且可以与控制的 $t-1$ 个追踪者交互. $\mathcal{A}$ 选择 $(M^*,A^*)\notin\{(M_i,A_i)\}$ 且 M^* 和 $\{M_i\}$ 的 tid 不能相同, C 选一个随机 $b\in\{0,1\}$, 使用 GID_i 生成认证 η_b 返回给 $\mathcal{A}$, $\mathcal{A}$ 输出 $b'\in\{0,1\}$. $\mathcal{A}$ 赢得游戏如果 $b=b'$.

方案满足匿名性如果对所有 PPT 敌手, $|\Pr[\mathcal{A}wins]-1/2|\leqslant negl(\lambda)$, 其中 $negl$ 为可忽略函数, λ 为安全参数.

定理 1. 当 ZK 满足零知识性, 哈希函数 H_1,H_2 是随机预言机且 ElGamal 算法和 Pedersen 门限秘密分享算法满足安全性时, 方案满足匿名性.

证明: C 回应的认证为 $\eta_b=(t_1,t_2,\sigma,\pi)$, 考虑游戏 1.1, 游戏 1.1 与游戏 1.0 的区别在于认证中 π 被一模拟器所模拟. ZK 的零知识性保证该模拟器可在无 $\vec{w}$ 的情况下模拟 π 使得模拟器的输出 π^* 和 π 不可区分. 因此, 游戏 1.1 将证明换成 π^* 后, $\mathcal{A}$ 在游戏中的胜率不会发生不可忽略的变化. 现考虑游戏 1.2, 游戏 1.2 与游戏 1.1 的区别在于以 $g^z(z\xleftarrow{R}\mathbb{G})$ 替代 σ 中的 Ex_1. $\mathcal{A}$ 至多控制 $t-1$ 个追踪者, 因此由 Pedersen 门限秘密分享算法的安全性, $\mathcal{A}$ 无法解密 σ, 由于 ElGamal 算法在 DDH 假设下满足选择明文攻击下的不可区分性 (IND-CPA), Ex_1 在 $\mathbb{G}$ 中均匀分布, 与 g^z 不可区分. 因此, 游戏 1.2 将 σ 换成 σ^* 后, 与游戏 1.1 相比 $\mathcal{A}$ 的胜率不会发生不可忽略的变化. 游戏 1.2 中, C 给 $\mathcal{A}$ 的回应 $\eta_b=(t_1,t_2,\sigma^*,\pi^*)$, 因此除 t_1, t_2 外的所有信息都与 b 无关. 由于 H_1,H_2 是随机预言机, 因此 $\mathcal{A}$ 在游戏 1.2 中没有优于猜测的算法来选择 b' 使得 $b'=b$. 又因为游戏 1.2 和游戏 1.0 中 $\mathcal{A}$ 的胜率的差异可被忽略, 所以 $\mathcal{A}$ 在游戏 1.0 中的胜率为 $1/2+negl(\lambda)$, 因此方案满足匿名性.

4.2.2 属性隐私性

属性隐私性要求无法通过匿名认证得知用户的属性, 从而避免攻击者利用属性集定位到具体的用户来分析该用户的行为. 属性隐私性形式化定义如下.

游戏 2.0: 敌手 $\mathcal{A}$ 选取 GID_i,M_i, 访问控制策略 A_i 和满足策略的属性集合 Ω_i, 提交给 C, 询问认证 η_i, 可以询问不多于 $p(\lambda)$ 次. $\mathcal{A}$ 选择 $(GID^*,M^*,A^*)\notin\{(GID_i,M_i,A_i)\}$ 和两个满足 A^* 的属性集 Ω_0, Ω_1 发给 C. C 选一个随机 $b\in\{0,1\}$, 使用 Ω_i 的属性私钥生成认证 η^* 返回给 $\mathcal{A}$, $\mathcal{A}$ 输出 $b'\in\{0,1\}$. $\mathcal{A}$ 赢得游戏如果 $b=b'$.

方案满足属性隐私如果对所有 PPT 敌手 $\mathcal{A}$, $|\Pr[\mathcal{A}wins]-1/2|\leqslant negl(\lambda)$, 其中 $negl$ 为可忽略函数, λ 为安全参数.

定理 2. 当 ZK 满足零知识性时, 方案满足属性隐私性.

证明: C 回应的认证为 $\eta_b=(t_1,t_2,\sigma,\pi)$, 考虑游戏 2.1, 游戏 2.1 与游戏 2.0 的区别在于认证中 π 被一模拟器所模拟. ZK 的零知识性保证该模拟器可在无 $\vec{w}$ 的情况下模拟 π, 使得模拟器的输出 π^* 和 π 不可区分. 因此, 游戏 2.0 将证明换成 π^* 后, $\mathcal{A}$ 在游戏中的胜率不会发生不可忽略的变化. 游戏 2.1 中, C 给 $\mathcal{A}$ 的认证为 $\eta_b=(t_1,t_2,\sigma,\pi^*)$. 因此, $\mathcal{A}$ 在游戏 2.1 中没有优于猜测的算法来选择 b' 使得 $b'=b$. 又因为游戏 2.1 和游戏 2.0 中 $\mathcal{A}$ 的胜率的差异可被忽略, 所以 $\mathcal{A}$ 在游戏 2.0 中的胜率为 $1/2+negl(\lambda)$, 方案满足属性隐私性.

4.2.3 不可伪造性

不可伪造性要求没有符合策略的属性私钥的攻击者不能生成有效认证. 不可伪造性形式化定义如下.

游戏 3: 敌手 $\mathcal{A}$ 选取 GID^* 和属性集 Ω^*, 询问 C 用户秘密值和属性密钥. $\mathcal{A}$ 选取 GID_i, M_i, 属性集合 Ω_i 和访问控制策略 A_i 给 C, 询问认证 η_i. $\mathcal{A}$ 可以询问 C 不多于 $p(\lambda)$ 次. $\mathcal{A}$ 选择 GID^*, 消息 M^*, 访问控制策略 A^* 生成认证 η^*. $\mathcal{A}$ 赢得游戏如果: $A\{K_{x,GID^*},K'_{x,GID^*}\}\neq 1\wedge(GID^*,M^*,A^*)\notin\{(GID_i,M_i,A_i)\}\wedge Verify(M^*,\eta^*,(A^*,\rho),GP)\to 1$.

方案满足不可伪造性如果对所有 PPT 敌手 $\mathcal{A}$, $|\Pr[\mathcal{A}wins]|\leqslant negl(\lambda)$, 其中 $negl$ 为可忽略函数, λ 为安全参数.

定理 3. 当 RW 属性加密[28]算法满足安全性且 ZK 满足零知识性和知识证明性时, 方案满足不可伪造性.

证明: 假设存在一个敌手 $\mathcal{A}^*$ 能以不可忽略的概率赢得游戏, 代表 $\mathcal{A}^*$ 可以以不可忽略的概率达成至少以下一个条件: (a) 在 $A\{K_{x,GID^*},K'_{x,GID^*}\}\neq 1$ 的情况下, $\mathcal{A}^*$ 构造出了合法的属性密钥, 从而生成合法认证; (b) $\mathcal{A}^*$ 使用不合法的属性密钥生成合法认证. 由 RW 属性加密算法的安全性确保条件 (a) 达成的概率可忽略. 同时, 因为 ZK 满足零知识性, $\mathcal{A}^*$ 无法得到 GID_i 属性密钥的信息. 且 ZK 满足知识证明, 因此可以从合法认证中提取出 $\vec{w}$, 得到合法的属性密钥, 所以条件 (b) 达成的概率可忽略. 因此不存在这样的敌手 $\mathcal{A}^*$, 方案满足不可伪造性.

4.2.4 可链接性

可链接性要求任何有效的敌手都不能对 tid 相同的两个消息生成认证而不被链接. 可链接性形式化定义如下.

游戏 4: 敌手 $\mathcal{A}$ 选取 GID^* 和属性集 Ω^*, 询问 C 用户秘密值和属性密钥. $\mathcal{A}$ 选取 GID_i, M_i, 属性集合 Ω_i 和访问控制策略 A_i 给 C, 询问认证 η_i. $\mathcal{A}$ 可以询问 C 不多于 $p(\lambda)$ 次. $\mathcal{A}$ 对两个有相同 tid 的信息 M_1, M_2, 选择 A^* 生成认证 η_1^* 和 η_2^*. $\mathcal{A}$ 赢得游戏如果: $Verify(M_i^*,\eta_i^*,(A^*,\rho),GP)\to 1\ for\ i=1,2\wedge Link(M_1^*,M_2^*,\eta_1^*,\eta_2^*)\to 0$.

方案满足可链接性如果对所有 PPT 敌手 $\mathcal{A}$, $|\Pr[\mathcal{A}wins]|\leqslant negl(\lambda)$, 其中 $negl$ 为可忽略函数, λ 为安全参数.

定理 4. 当 ZK 满足零知识性和知识证明性, 且 H_1,H_2 是随机预言机时, 方案满足可链接性.

证明: 假设存在敌手 $\mathcal{A}^*$ 能以不可忽略的概率攻破方案的可链接性. $\mathcal{A}^*$ 控制 GID^* 对两个有相同 tid 的信息 M_1^*,M_2^* 生成认证 η_1^* 和 η_2^*. 因为 ZK 满足零知识性且 H_1,H_2 是随机预言机, 在 $\mathcal{A}^*$ 询问认证时, 无法得到 GID_i 用户秘密值和属性密钥的信息. 同时由于 ZK 满足知识证明性, 因此当 η_1^* 和 η_2^* 通过验证时, $\mathcal{A}^*$ 必须使用 GID^* 的用户秘密值和属性密钥来生成认证, 所以 $t_{11}^*=H_1(tid,S_u^*)=t_{12}^*$, 则两个认证可以被链接, 与存在敌手 $\mathcal{A}^*$ 能以不可忽略的概率攻破方案的可链接性矛盾, 假设不成立.

4.2.5 可追踪性

可追踪性要求任何有效的敌手在控制少于 t 个追踪者的情况下不能避免被追踪者追踪. 可追踪性形式化定义如下.

游戏 5: 敌手 $\mathcal{A}$ 选取 GID^* 和属性集 Ω^*, 询问 C 用户秘密值和属性密钥. $\mathcal{A}$ 选取 GID_i,M_i, 属性集合 Ω_i 和访问控制策略 A_i 给 C, 询问认证 η_i. $\mathcal{A}$ 可以询问 C 不多于 $p(\lambda)$ 次. $\mathcal{A}$ 选择消息 M^*, 访问控制策略 A^* 生成认证 η^*. $\mathcal{A}$ 赢得游戏如果: $Verify(M^*,\eta^*,(A^*,\rho),GP)\to 1\wedge Trace(\eta^*,\{TSK_i\})\to\perp$ 或 $GID_j,GID_j\neq GID_i$.

方案满足可追踪性如果对所有 PPT 敌手 $\mathcal{A}$, $|\Pr[\mathcal{A}wins]|\leqslant negl(\lambda)$, 其中 $negl$ 为可忽略函数, λ 为安全参数.

定理 5. 当 ZK 满足零知识性和知识证明性, 且 H_1,H_2 是随机预言机时, 方案满足可追踪性.

证明: 假设存在一个敌手 $\mathcal{A}^*$ 能以不可忽略的概率攻破方案的可追踪性. $\mathcal{A}^*$ 控制 GID^* 对消息 M^* 生成认证 η^*. 因为 ZK 满足零知识性且 H_1,H_2 是随机预言机, 在 $\mathcal{A}^*$ 询问认证时, 无法得到 GID_i 用户秘密值和属性密钥的信息. 由于 ZK 的知识证明性, η^* 通过验证时, $\mathcal{A}^*$ 必须使用 TPK 正确加密 GID^*, 因此执行 $Trace(\eta^*,\{TSK_i\})$ 可得到 GID^*, 假设不成立.

此外, 本方案还满足任务机密性和数据机密性. 任务通过 RW 属性加密, 只有工作者的属性密钥与访问控制策略相匹配才能解密. 同时工作者通过对称加密将工作结果加密后通过链下安全通道发送给请求者, 在请求者公开确认 $H(S_1)$ 后, 工作者发布密钥 S_2, 此时只有请求者拥有密文 S_1, 能解密得到工作结果, 链上其他用户只有 S_2 和 $H(S_1)$, 无法得到 S_1.

4.3 公平性分析

本方案参考文献 [23] 设计了请求者和工作者之间传输数据和分发奖励的协议来实现公平性.

匿名认证的验证和 S_2 一致性的检验均由智能合约执行, 因此请求者无法作恶, 下面分析请求者可能作恶的几种情况: (a) 在收到 S_1 后终止交易; (b) 在奖励阶段作弊. 对于情况 (a), 因为 S_1 是加密后结果, 请求者不能在没有得到 S_2 的情况下获得真实数据, 而工作者的押金会被退还; 对于情况 (b), 请求者如需按规则分配奖励, 需要提供零知识证明, 由 *ZK* 的可靠性来避免请求者作弊的情况.

分析工作者恶意的几种情况: (a) 对同一个任务发送多次工作结果; (b) 提交伪造的认证; (c) 提交伪造的 S_2. 对于情况 (a), 由于方案满足可链接性, 提交多次会被链接, 且押金不返还; 对于情况 (b), 由于方案满足不可伪造性, 无法通过验证且押金不返还; 对于情况 (c), 智能合约通过验证 S_2 的计算结果是否为 Tag 可以检测出这种伪造行为, 并使用追踪机制获取工作者的真实身份, 同时没收其押金.

5 实验与分析

本节将本文提出的可问责匿名属性认证方案与其他匿名认证方案进行对比分析, 同时将本文的众包方案与其他基于区块链的众包方案进行比较, 然后对本文方案进行了仿真实验.

5.1 功能特性对比

各匿名认证方案的对比如表 1 所示. 本文提出的可问责匿名属性认证方案在实现匿名性的同时提供了细粒度访问控制, 即只有当用户的属性满足指定的访问策略时, 用户才能生成认证, 同时如果用户针对同一任务认证了两个不同的消息, 则对应的两个认证将被链接. 此外还提供了门限可追踪, 当用户有恶意行为时, 追踪组成员通过协作可以获取用户的身份. 通过使用 RW 去中心化属性密码和智能合约, 本方案还实现了去中心化.

表 1 匿名认证方案比较

协议	属性密码	可链接性	可追踪性	门限可追踪性	去中心化
Guo等人[14]	×	×	√	√	×
Zheng等人[15,32]	×	√	√	×	×
Okamoto等人[33]	√	×	×	×	√
TCABRS[34]	√	×	√	×	×
Ding等人[13]	√	×	√	√	×
El Kaafarani 等人[35,36]	√	√	×	×	×
Hong等人[37]	√	√	√	×	×
本文方案	√	√	√	√	√

表 2 中比较本文提出的众包方案与一些现有的基于区块链的众包方案. 从对比中可以看出, 现有的基于区块链的众包系统解决了传统众包平台集中化的问题, 但仍存在一些问题, 如没有对工作者进行筛选, 没有提供工作者匿名机制, 或提供了匿名性但没有实现问责机制. 本文的众包方案允许请求者制定访问控制策略来筛选工作者, 从而在一定程度上提高了工作结果的质量. 同时本方案提供隐私保护, 除了任务内容受到保护外, 工作者的身份隐私也收到保护. 此外还实现了可链接和门限可追踪性, 确保在匿名的情况下, 相同工作者只能对同一任务提交一次, 且可以获取恶意用户的身份, 实现问责. 与 ZebraLancer 方案[21]相比, 本方案不需要可信的注册中心, 去中心化程度更高, 同时提供了任务保护、工作者选择和可追踪的能力.

表 2 基于区块链的众包方案比较

方案	任务保护	工作者筛选	匿名性	可链接性	可追踪性
ZebraLancer[21]	×	×	√	√	×
Ghaffaripour等人[38]	×	×	×	×	×
CrowdBC[39]	×	√	×	×	×
Feng等人[40]	×	√	√	×	×
Tan等人[4]	√	√	×	×	×
SecBCS[20]	√	√	√	×	√
本文方案	√	√	√	√	√

5.2 实验仿真

本节通过在 macos Monterey 12.5, CPU 为 Apple M1 Pro 的 PC 机上对本方案进行仿真实验, 使用 gnark 0.7.1[41] 来实现 zk-SNARKs, 运行在 Hyperledger Fabric 网络环境下. 表 3 列出了本方案各阶段理论上的计算开销与实际开销. 为了便于分析, 忽略序列化反序列化、智能合约调用和网络传输等方面的影响, 令 $|C|$ 表示任务加密涉及的属性个数, $|S|$ 表示任务解密时实际用到的属性个数, t 和 n 为门限秘密分享算法的阈值和总数, E 表示群 $\mathbb{G}$ 上指数运算耗时, P 表示双线性配对运算耗时, N 表示整数域上的指数运算, E_T 表示群 $\mathbb{G}_T$ 上的指数运算, H 表示哈希运算耗时. 实验模拟了 $t=2, n=3, |C|=1, |S|=1$ 时的场景. 由于系统初始化时要使用 *ZK.Setup* 算法, 因此耗时较长, 但由于只需要在初始化时执行一次, 因此是可以接受的. 此外, 生成匿名认证的过程也耗时较长, 但可以由工作者链下执行. 其他步骤的时间开销均为毫秒级, 可以满足实际应用场景的需求.

表 3 方案的时间开销

算法	理论时间开销	实际时间开销 (ms)
GlobalSetup	$P+ZK.Setup$	435 687.188
UserSetup	H	0.782
AASetup	成员 : $2n(t-2)N+E+E_T$ 智能合约 : $t(E+E_T)$	成员: 1.725, 智能合约: 1.466
TracerSetup	追踪者 : $n(t-2)N+E$ 智能合约 : tE	追踪者: 0.106, 智能合约: 0.205
KeyGen	成员 : $4E+H$ 申请用户 : $2tE$	成员: 0.262, 申请用户: 0.834
Encrypt	$E_T+\|C\|(2E_T+4E)$	4.196
Decrypt	$\|S\|(3P+E_T)+H$	1.991
Auth	$2E+2H+ZK.Prove$	14 115.249
Verify	$ZK.Verify$	1.947
Trace	追踪者 : E 智能合约 : tE	追踪者: 0.080, 智能合约: 0.138

图 2 显示了在属性权威机构和追踪机构的初始化阶段, 当 $n=30$, 门限值 t 从 10 逐渐增加到 20 时合约计算时间的仿真结果和其拟合曲线. 合约计算 *PK*, *TPK* 的时间和 t 成正比, 与理论结果相符.

图 3 显示了不同属性数量下, 分别使用全与和全或策略时加解密的运行时间, 可以看到加密的时间开销与访问控制策略中的属性个数呈线性关系, 且与逻辑控制符比或要复杂. 而解密的时间开销与实际使用的属性个数也呈线性关系, 符合理论结果. 加解密的运算均在链下执行, 在涉及属性数量为 32 的情况下, 全与加密所用时间为 53.206 ms, 而解密为 39.011 ms, 符合实际的使用要求.

为提高众包系统的性能, 智能合约的运算开销与链上的存储开销应尽可能小. 本方案在链上不存储工作结果的密文, 而是存储两个哈希值和密钥 S_2, 节省了存储开销. 匿名认证的生成在链下进行, 而验证则由智能合约执行, 同时链上要存储匿名认证, 因此 zk-SNARKs 证明的大小和验证效率对系统的性能影响很大.

通过实验将本方案的匿名认证大小、验证认证时间与生成认证时间和 ZebraLancer[21]方案进行了比较, 使用了 EdDSA 数字签名算法[42]来实现 ZebraLancer 方案中的证书. 由于 zk-SNARKs 的证明大小为 O(1), 本方案的匿名认证大小为 687 B, 几乎不受属性数量和策略的影响, 而 ZebraLancer[21]为 559 B. 图 4 和图 5 显示了 ZebraLancer

方案与本方案在全与和全或策略下验证认证与生成认证的时间开销, 由于 ZebraLancer 方案未实现属性认证, 因此将其与本方案属性数量为 0 的开销放在一起比较. 由于增加了可追踪和去中心化的特性, 本方案的证明大小、验证和生成认证时间的开销比 ZebraLancer 方案略有增加. 在零知识证明中, 双线性对的运算开销很大, 因此使用到的属性数量越多, 时间开销越大, 可以看到在全或策略中, 属性数目的增加对时间开销影响不大, 而全与策略的时间开销则随着属性数量的增加而增长.

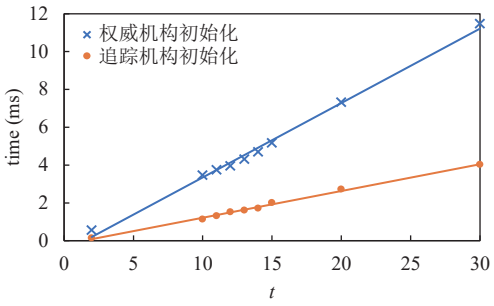

图 2　不同 t 下权威与追踪机构初始化的时间拟合曲线

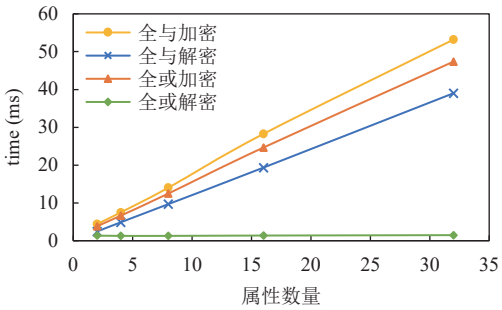

图 3　不同访问策略下加解密的时间

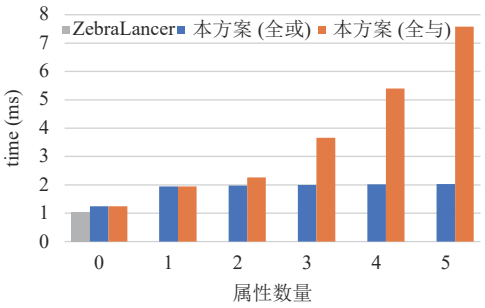

图 4　验证认证的时间开销

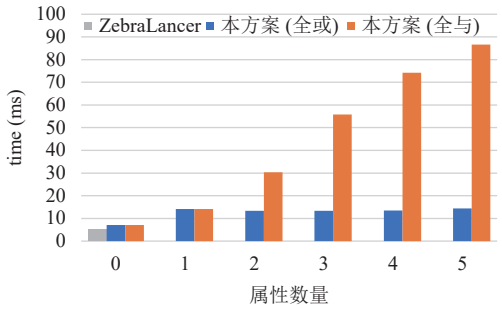

图 5　生成认证的时间开销

本方案的证明大小几乎不受属性数量和访问策略复杂性的影响, 验证和生成认证的时间受到影响但需要在链上执行的验证算法时间开销较小, 在涉及 5 个属性的全与策略下开销为 7.57 ms, 而生成认证过程虽然开销较大但由工作者链下执行, 因此符合实际使用需求.

6 总　结

本文提出了一种去中心化可问责属性认证方案, 并将其与区块链技术相结合设计了一个新型众包平台. 与传统众包方案相比, 本方案提供了一个去中心化的众包系统, 在保护用户隐私的同时实现了可链接性和可追踪性来处理工作者的恶意行为, 并通过门限秘密分享实现了权威组织和追踪者组织来分散风险、增加去中心化程度. 此外, 本方案对工作者进行了筛选以提高工作质量, 实现了请求者和工作者之间的公平交易. 当然, 本方案在计算和存储开销上都还有进一步优化的空间, 未来将进一步优化本方案的效率. 此外, 未来将研究增加声誉系统, 如设置信用评分等来扩展本系统的功能, 实现更好的激励机制.

References:

[1] Isaac M, Benner K, Frenkel S. Uberhid 2016 breach, paying hackers to delete stolen data. 2017. https://www.nytimes.com/2017/11/21/technology/uber-hack.html

[2] McInnis B, Cosley D, Nam C, Leshed G. Taking a HIT: Designing around rejection, mistrust, risk, and workers' experiences in Amazon Mechanical Turk. In: Proc. of the 2016 CHI Conf. on Human Factors in Computing Systems. San Jose: ACM, 2016. 2271–2282. [doi: 10.1145/2858036.2858539]

[3] Feng W, Yan Z. MCS-Chain: Decentralized and trustworthy mobile crowdsourcing based on blockchain. Future Generation Computer Systems, 2019, 95: 649–666. [doi: 10.1016/j.future.2019.01.036]

[4] Tan L, Xiao H, Shang XL, Wang Y, Ding F, Li WJ. A blockchain-based trusted service mechanism for crowdsourcing system. In: Proc. of the 91st IEEE Vehicular Technology Conf. Antwerp: IEEE, 2020. 1–6. [doi: 10.1109/VTC2020-Spring48590.2020.9128425]

[5] Chaum D. Blind signatures for untraceable payments. In: Chaum D, Rivest RL, Sherman AT, eds. Advances in Cryptology. Boston: Springer, 1983. 199–203. [doi: 10.1007/978-1-4757-0602-4_18]

[6] Camenisch J, Lysyanskaya A. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Proc. of the 2001 Int'l Conf. on the Theory and Application of Cryptographic Techniques. Innsbruck: Springer, 2001. 93–118. [doi: 10.1007/3-540-44987-6_7]

[7] Chaum D, van Heyst E. Group signatures. In: Proc. of the 1991 Workshop on the Theory and Application of Cryptographic Techniques. Brighton: Springer, 1991. 257–265. [doi: 10.1007/3-540-46416-6_22]

[8] Liu JK, Yuen TH, Zhou JY. Forward secure ring signature without random oracles. In: Proc. of the 13th Int'l Conf. on Information and Communications Security. Beijing: Springer, 2011. 1–14. [doi: 10.1007/978-3-642-25243-3_1]

[9] Li J, Au MH, Susilo W, Xie DQ, Ren K. Attribute-based signature and its applications. In: Proc. of the 5th ACM Symp. on Information, Computer and Communications Security. Beijing: ACM, 2010. 60–69. [doi: 10.1145/1755688.1755697]

[10] Tan SY, Groß T. MoniPoly—An expressive *q*-SDH-based anonymous attribute-based credential system. In: Proc. of the 26th Int'l Conf. on the Theory and Application of Cryptology and Information Security. Daejeon: Springer, 2020. 498–526. [doi: 10.1007/978-3-030-64840-4_17]

[11] Gu K, Wang KM, Yang LL. Traceable attribute-based signature. Journal of Information Security and Applications, 2019, 49: 102400. [doi: 10.1016/j.jisa.2019.102400]

[12] Kaaniche N, Laurent M. Attribute-based signatures for supporting anonymous certification. In: Proc. of the 21st European Symp. on Research in Computer Security. Heraklion: Springer, 2016. 279–300. [doi: 10.1007/978-3-319-45744-4_14]

[13] Ding SL, Zhao YM, Liu YY. Efficient traceable attribute-based signature. In: Proc. of the 13th IEEE Int'l Conf. on Trust, Security and Privacy in Computing and Communications. Beijing: IEEE, 2014. 582–589. [doi: 10.1109/TrustCom.2014.74]

[14] Guo YW, Tang HK, Tan AD, Xu L, Gai KK, Jia XW. A privacy-preserving auditable approach using threshold tag-based encryption in consortium blockchain. In: Proc. of the 6th Int'l Conf. on Smart Computing and Communication. New York: Springer, 2022. 265–275. [doi: 10.1007/978-3-030-97774-0_24]

[15] Zheng HB, Wu QH, Guan ZY, Qin B, He SY, Liu JW. Achieving liability in anonymous communication: Auditing and tracing. Computer Communications, 2019, 145: 1–13. [doi: 10.1016/j.comcom.2019.05.021]

[16] Hwang JY, Chen LQ, Cho HS, Nyang D. Short dynamic group signature scheme supporting controllable linkability. IEEE Trans. on Information Forensics and Security, 2015, 10(6): 1109–1124. [doi: 10.1109/TIFS.2015.2390497]

[17] Fujisaki E, Suzuki K. Traceable ring signature. In: Proc. of the 10th Int'l Conf. on Practice and Theory in Public-key Cryptography. Beijing: Springer, 2007. 181–200. [doi: 10.1007/978-3-540-71677-8_13]

[18] Au MH, Liu JK, Susilo W, Yuen TH. Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction. Theoretical Computer Science, 2013, 469: 1–14. [doi: 10.1016/j.tcs.2012.10.031]

[19] Rahaman S, Cheng L, Yao DF, Li H, Park JM. Provably secure anonymous-yet-accountable crowdsensing with scalable sublinear revocation. Proc. on Privacy Enhancing Technologies, 2017, 2017(4): 384–403. [doi: 10.1515/popets-2017-0055]

[20] Lin C, He DB, Zeadally S, Kumar N, Choo KKR. SecBCS: A secure and privacy-preserving blockchain-based crowdsourcing system. Science China Information Sciences, 2020, 63(3): 130102. [doi: 10.1007/s11432-019-9893-2]

[21] Lu Y, Tang Q, Wang GL. ZebraLancer: Private and anonymous crowdsourcing system atop open blockchain. In: Proc. of the 38th IEEE Int'l Conf. on Distributed Computing Systems. Vienna: IEEE, 2018. 853–865. [doi: 10.1109/ICDCS.2018.00087]

[22] Li P, Lai JZ, Wu YD. Accountable attribute-based authentication with fine-grained access control and its application to crowdsourcing. Frontiers of Computer Science, 2023, 17(1): 171802. [doi: 10.1007/s11704-021-0593-4]

[23] Chen F, Wang JH, Jiang CK, Xiang T, Yang YY. Blockchain based non-repudiable IoT data trading: Simpler, faster, and cheaper. In: Proc. of the 2022 IEEE Conf. on Computer Communications. London: IEEE, 2022. 1958–1967. [doi: 10.1109/INFOCOM48880.2022.9796857]

[24] Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613. [doi: 10.1145/359168.359176]

[25] Blakley GR. Safeguarding cryptographic keys. In: Proc. of the 1979 Int'l Workshop on Managing Requirements Knowledge. New York: IEEE, 1979. 313–313. [doi: 10.1109/MARK.1979.8817296]

[26] Pedersen TP. A threshold cryptosystem without a trusted party. In: Proc. of the 1991 Workshop on the Theory and Application of

Cryptographic Techniques. Brighton: Springer, 1991. 522–526. [doi: 10.1007/3-540-46416-6_47]

[27] Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proc. of the 13th ACM Conf. on Computer and Communications Security. Alexandria: ACM, 2006. 89–98. [doi: 10.1145/1180405.1180418]

[28] Rouselakis Y, Waters B. Efficient statically-secure large-universe multi-authority attribute- based encryption. In: Proc. of the 19th Int'l Conf. on Financial Cryptography and Data Security. San Juan: Springer, 2015. 315–332. [doi: 10.1007/978-3-662-47854-7_19]

[29] Lewko A, Waters B. Decentralizing attribute-based encryption. In: Proc. of the 30th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Tallinn: Springer, 2011. 568–588. [doi: 10.1007/978-3-642-20465-4_31]

[30] Ben-Sasson E, Chiesa A, Tromer E, Virza M. Succinct non-interactive zero knowledge for a von neumann architecture. In: Proc. of the 23rd USENIX Conf. Security Symp. San Diego: USENIX Association, 2014. 781–796.

[31] Elgamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. on Information Theory, 1985, 31(4): 469–472. [doi: 10.1109/TIT.1985.1057074]

[32] Zheng HB, Wu QH, Qin B, Zhong L, He SY, Liu JW. Linkable group signature for auditing anonymous communication. In: Proc. of the 23rd Australasian Conf. Wollongong: Springer, 2018. 304–321. [doi: 10.1007/978-3-319-93638-3_18]

[33] Okamoto T, Takashima K. Decentralized attribute-based signatures. In: Proc. of the 16th Int'l Conf. on Practice and Theory in Public-key Cryptography. Nara: Springer, 2013: 125–142. [doi: 10.1007/978-3-642-36362-7_9]

[34] Lu AR, Li WH, Yao YZ, Yu NH. TCABRS: An efficient traceable constant-size attribute-based ring signature scheme for electronic health record system. In: Proc. of the 6th IEEE Int'l Conf. on Data Science in Cyberspace. Shenzhen: IEEE, 2021. 106–113. [doi: 10.1109/DSC53577.2021.00022]

[35] El Kaafarani A, Chen LQ, Ghadafi E, Davenport J. Attribute-based signatures with user-controlled linkability. In: Proc. of the 13th Int'l Conf. Heraklion: Springer, 2014. 256–269. [doi: 10.1007/978-3-319-12280-9_17]

[36] El Kaafarani A, Ghadafi E. Attribute-based signatures with user-controlled linkability without random oracles. In: Proc. of the 16th IMA Int'l Conf. Oxford: Springer, 2017. 161–184. [doi: 10.1007/978-3-319-71045-7_9]

[37] Hong JN, Xue KP, Gai N, Wei DSL, Hong PL. Service outsourcing in F2C architecture with attribute-based anonymous access control and bounded service number. IEEE Trans. on Dependable and Secure Computing, 2020, 17(5): 1051–1062. [doi: 10.1109/TDSC.2018.2845381]

[38] Ghaffaripour S, Miri A. A decentralized, privacy-preserving and crowdsourcing-based approach to medical research. In: Proc. of the 2020 IEEE Int'l Conf. on Systems, Man, and Cybernetics. Toronto: IEEE, 2020. 4510–4515. [doi: 10.1109/SMC42975.2020.9283027]

[39] Li M, Weng J, Yang AJ, Lu W, Zhang Y, Hou L, Liu JN, Xiang Y, Deng RH. CrowdBC: A blockchain-based decentralized framework for crowdsourcing. IEEE Trans. on Parallel and Distributed Systems, 2019, 30(6): 1251–1266. [doi: 10.1109/TPDS.2018.2881735]

[40] Feng W, Yan Z, Yang LT, Zheng QH. Anonymous authentication on trust in blockchain-based mobile crowdsourcing. IEEE Internet of Things Journal, 2022, 9(16): 14185–14202. [doi: 10.1109/JIOT.2020.3018878]

[41] Botrel G, Piellard T, Tabaie T, *et al*. Consensys/gnark: v0.10.0. 2024. https://zenodo.org/records/11034183 [doi: 10.5281/zenodo.5819104]

[42] Josefsson S, Liusvaara I. Edwards-curve digital signature algorithm (EdDSA). 2017. https://www.rfc-editor.org/pdfrfc/rfc8032.txt.pdf

陶静怡(1999－), 女, 硕士生, 主要研究领域为区块链, 属性密码.

阚海斌(1971－), 男, 博士, 特聘教授, CCF 杰出会员, 主要研究领域为密码学, 编码理论, 算法与计算复杂性, 区块链.

张亮(1989－), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为区块链, 应用密码学.