Fcrowd

（支持细粒度重用和隐私保护的公平众包系统）

电子邮箱：cjwcier@gmail.com

摘要

众包在我们的日常生活中获得了许多发展和广泛的应用。传统的集中式众包系统遭受高管理成本和低效率的困扰。区块链技术的最新发展促成了去中心化众包系统的发明，它可以克服集中式系统的局限性，并使得众包解决方案的重用成为可能。但是，这样的系统也带来了新的安全和隐私挑战。例如，区块链上的交易是公开可见的，这可能导致众包解决方案的隐私泄漏。而且，在这样的平台上，公平交易更具挑战性。本系统研发了一种基于区块链的具有细粒度重用的隐私保护和公平的众包系统，以满足去中心化众包的安全性要求。具体来说，我们构造“仅地址”（OAO）身份验证以确保众包过程中参与者地址的唯一性。我们还通过承诺设计了一种提交然后公开的方法，以抵制在众包过程中的“搭便车”和虚假报道”攻击。因此，可以保证公平地交换财产。此外，为了确保数据机密性和解决方案项的细粒度重用，我们采用基于配对的加密技术来生成加密密钥并确保灵活的授权重用。在重用阶段，采用了秘密授权来保证隐私的访问授权。

安全性分析和实施结果表明，本系统可以有效地实现隐私保护和公平众包以及细粒度的众包重用。

第一章 系统概述

1.1 系统背景

近年来，众包得到了广泛的关注和广泛的应用价值，可以通过公开招募合格的工人来解决许多复杂的任务。许多众包应用如 Amazon MechanicalTurk 、Upwork 和 CrowdFlower ，旨在满足我们日常生活的要求。通常，这种众包系统由三组参与者组成：请求者、工作人员和集中式众包平台。请求者通过众包平台发布具有激励机制的任务，然后对这些任务感兴趣的一组工作人员将相应的解决方案提交给众包平台。最后，平台返回正确的解决方案，并将奖励分配给相应的工作人员。

专业的众包平台的出现，解决了某些特定行业内的现实需求，通过互联网完成任务获取报酬的形式，突破地域、时间限制，使用抽奖悬赏、在线招标，点对点雇佣等形式，促进了各种资源的优化利用。

1.2 需求分析

传统的集中式众包平台依赖于受信任的第三方模型，该模型易受不可避免的挑战的影响。

•由不同外部攻击（例如 DoS 攻击，Sybil 攻击和远程劫持）引起的单点故障。

•中心化系统内部行为不当导致损害平台、员工或攻击者的自身利益。

•平台经常无法解决请求者与工人之间的纠纷，这可能导致“搭便车”（不诚实的工作者在不执行实际任务的情况下获得奖励）和“虚假报告”（不诚实的请求者在不付款的情况下获得解决方案）。

•整个众包过程中的敏感信息由集中化平台托管，这可能会导致隐私泄漏。

•由于第三方中介机构的存在，雇主会增加支出成本，工作者则会减少收益成本。

目前传统的众包解决方案都需要依赖可信第三方来设计相应的信任机制、激励机制与结果质量评估，随着越来越多的第三方出现信任危机，开发一套不需要依赖不可信第三方的众包系统成为非常必要的事情。区块链技术的发展为上述需求提供了一种替代选择，可以克服当前集中式众包系统的安全挑战。然而，这项

新技术也带来了一些新的弊端，这在集中式众包系统中是前所未有的。

•区块链本身的透明性特征违反了数据隐私性，然而许多众包数据可能是隐私敏感的。

•区块链具有延迟确认的性质，在解决方案提交到网络后，区块矿工可能需要花费几分钟才能将交易打包到区块中。恶意工作人员可以轻松地从他人那里复制解决方案，并提交与自己相同的答案以发起“搭便车”攻击。

•区块链的匿名性。在区块链中，每个人都可以通过注册许多地址而不暴露真实身份来加入区块链。因此，不诚实的工人可能会通过不同的地址多次将解决方案提交给同一任务以要求获得比预期更多的奖励，或者恶意的请求者可能会匿名向他/她自己提交大量合谋的解决方案以降级诚实的工人。

在上述基本安全和隐私问题上，本系统研发了一种基于区块链的具有细粒度重用的隐私保护和公平的众包系统，以满足去中心化众包的安全性要求。我们构造“仅地址”（OAO）身份验证以确保众包过程中参与者地址的唯一性。我们还通过承诺设计了一种提交然后公开的方法，以抵制在众包过程中的“搭便车”和虚假报道”攻击。因此，可以保证公平地交换财产。此外，为了确保数据机密性和解决方案项的细粒度重用，我们采用基于配对的加密技术来生成加密密钥并确保灵活的授权重用。

1.3 相关工作分析

Li 等人通过智能合约提出了一个基于区块链的分散式众包框架，称为CrowdBC，但是未考虑安全风险和隐私保护要求。Lu 等人实现匿名和私人众包，由 zk-SNARK 设计了基于区块链的去中心化系统 Ze-braLancer。然而在他们的工.作中没有讨论众包的再利用。Zhang 等人提出了一种使用承诺和同态加密技术的基于区块链的公平、安全的众筹方案。朱等人构建了基于混合区块链的众包平台zkCrowd。具体地说，他们采用了混合区块链结构和相应的技术来确保安全的通信和隐私保护。Lin 等人通过组签名和基于密文策略的基于属性的加密，提出了一种基于安全的基于区块链的众包系统 SecBCS。综上所述，我们在表 1 中介绍了上述基于区块链的众包方案的安全性比较。

表 1-基于区块链的众包方案的比较

系统	数据可靠	认证	公平	隐私	细粒度重用
CrowdBC	×	×	√	×	×
Zebralancer	√	√	√	√	×
Zhang et al.	√	×	√	×	×
ZkCrowd	×	√	×	√	×
SecBCS	√	√	√	√	×
Fcrowd	√	√	√	√	√

1.4 特色及主要技术难点

本系统的主要特色在于以下几个方面：

•去中心化

本系统目的在于建立基于区块链的去中心化的众包系统，该系统无需依赖第三方中心化服务器和管理商，不存在单点故障问题，系统的安全性保证由成千上万个分布式的节点共同来维护。

•智能合约

智能合约是以一套数字形式定义的承诺，包括合约参与方可以在上面执行这些承诺的协议，本系统将用户管理、任务的发布、任务的接收、任务的重用等通过智能合约的形式来自动执行与完成，且合约本身在区块链平台无法被篡改。

•认证

只有经过身份验证的参与者才能在我们的分散框架中发布任务或提交解决方案。每个用户都可以通过隐私保护方式的合法注册参与者来验证交易。此外，在重用阶段，只有经过授权的重用用户才能获得众包解决方案。

•公平

在众包中，公平是指根据原始激励政策及其实际贡献来支付薪酬的工人的薪酬。在我们的框架中，我们采用严格的逻辑设计公平的交易众包流程，以确保交易的公平性。

•隐私保护

为了完成众包流程，请求者发布任务，工作人员通过交易向区块链提交解决方案，所有这些信息都是公开的。为了保证隐私，提交的解决方案只能由相应的请求者打开。此外，在重用阶段，为了确保隐私感知的访问授权交付，我们应该在以太坊中实现不可链接性，即对于任何两个传出的交易，它们的接收者是否相

同是未知的。此外，授权信息只能被授权的重用者访问。

•细粒度重用

只有授权的重用用户才能恢复授权的信息，并根据细粒度的重用访问相应的解决方案项。

本系统的主要难点：

•智能合约虽然是图灵完备的语言，但是由于合约的发布和执行都涉及交易费，无法将所有的过程都在智能合约中实现，如何设计有效的智能合约来描述任务众包的过程是本系统的难点之一。

• 我们构造 OAO 身份验证算法以确保众包过程中参与者地址的唯一性，还设计了一种先提交然后公开的方法来防止搭便车和虚假报告攻击。在细粒度重用阶段，我们采用了一种隐匿通信算法来保护参与者隐私并使用基于配对的加密技术实现对重用内容的授权控制和隐私保护。以上算法中涉及到的 Zokrate、Perdersen-commit、Eddsa、Pairing、AES 等工具和算法的具体应用和编码实现是本系统需要着重考虑的。

第二章系统设计与实现

2.1 系统总体方案

本系统是基于区块链的可重用的隐私保护和公平众包系统，为自由职业者提供众包服务，与传统的方案不同，该平台无需依赖可信第三方服务平台，任务发布者与接收者之间通过智能合约的形式达成一致，支持细粒度的任务重用并保障众包六成的交易公平和交易隐私。在该系统中，通过以太坊上的智能合约为用户之间达成一致性约定，运用密码学算法实现认证授权和隐私保护。

本系统主要分为三层结构：应用层、区块链层、隐私数据存储层。应用层作为用户接口操作层，将用户的输入信息进行密码学变换后与区块链平台进行对接，包括用户注册、任务管理、任务重用三个主要部分。区块链层在本系统中选择以太坊作为系统实施平台，主要运行本系统集成了用户注册、任务管理、任务重用等功能的智能合约，并存储众包过程中产生的相关公开数据，防止数据修改。隐私数据存储层在系统方案设计中采用安全保护的本地数据库，用于可选择地安全

存储用户个人隐私密钥数据并受数据主人的的完全访问控制，帮助维护管理用户在众包过程中必要产生的密钥，减轻了参与众包流程的复杂程度，维护了系统的简洁易用性。在使用本系统时，得益于分层的系统结构，复杂的密码操作和密钥管理工作对用户是透明的。

2.2 系统实现原理

本系统实现原理主要按照系统的总体架构，主要分为三层结构：应用层、区块链层、隐私数据存储层

• 应用层

应用层属于面向用户的接口层，主要功能包括：用户管理模块、任务管理模块、任务重用模块、密码操作模块、Web3py 智能合约管理模块、以及数据库管理模块。本系统开发了基于 Web3py 的智能合约管理模块，Web3py 支持以太坊的 JSON-RPC 客户端 API 的实现，利用 Python 即可完成对智能合约的读写操作。应用层以 Web 浏览器的形式运行起来，服务端通过 Django 3.0 进程提供 Web 服务，用户端浏览器访问即用。由于隐私保护的需要，系统后台需要响应复杂的密码学操作，服务端要求配置好相关密码库的 Python3.6 环境，采用 Linux 系统部署。

请求者、工作者、重用者在使用该平台之前首先进行注册，注册主要是将用户账号、密码、以及钱包地址等相关信息导入到本系统中，同时通过智能合约在区块链中完成注册。注册账号成功之后，用户需要初始化生成自己的密钥对并保存，随后以系统返回的注册证书和地址绑定消息线下计算 Zokrates-proof 并上传至系统。此后，请求者通过应用层接口发布任务，并规定任务完成的时间、需求、访问控制粒度、以及预先承诺的奖励等信息。工作者可以通过应用层查询到当前待完成的任务列表，并按照需求进行投标。重用者通过应用层在任务列表搜寻感兴趣的已达成交易的可重用任务，使用隐匿消息与目标任务发布人协商重用意向，达成重用交易。

• 区块链层

区块链层属于中间层，本系统将以太坊（Ethereum）作为区块链平台，以太坊可以建立和发布新一代的分布式应用平台，支持图灵完备的智能合约。智能合约预先定义了 Fcrowd 相关功能接口和数据结构。应用层经过编译后的合约通过

RPC 接口发送到区块链层，矿工对合约有效性进行确认，确认通过则写入区块链数据层。Fcrowd 将众包过程的信息全部上链存入智能合约，并通过与智能合约交互推进任务状态的变化。区块链层基于以太坊平台提供的共识机制来保证所有节点所存储信息的一致性。智能合约提供了众包数据存储的功能，用户使用 Web3py 与智能合约交互，获取合约数据状态，查询任务信息，比如任务截至日期、承诺状态、Proof 身份验证状态、提交状态等等。同时，用户以提交一个新的交易的方式合规触发任务数据状态的改变。

• 隐私数据存储层

隐私数据存储层是系统认证授权模块的底层数据库，是一个安全受控的链下本地数据库系统（Mysql），专用于安全存储管理个人用户的密钥对等信息。本系统通过众包数据链上存储，密钥隐私数据链下本地存储的方式，保证众包公平同时简化密码操作时密钥管理、密钥输入等操作。隐私数据存储层在系统方案设计中采用本地数据库，用于可选择地安全存储用户个人隐私数据并受数据所有者的的完全访问控制，帮助维护管理用户在众包过程中必要产生的密钥，减轻众包流程参与的复杂程度。在使用本系统时，得益于隐私数据存储层的设计，密钥管理工作对用户可以是透明的。

2.3 系统功能设计

• 用户注册管理

在用户使用本系统进行任务发布及接收之前，需要预先进行注册，注册的信息包括：用户名、密码、钱包地址等，同时系统为新注册用户设定相关参数的默认值，包括已发布任务、已提交任务等。

模块编号	001
模块名称	用户注册
输入	用户名、密码、确认密码
处理	将以上用户信息输入通过前台传到后台 Controller，后台利 Web3py 将信息写入到 Fcrowd 合约
输出	注册成功/失败

• 用户登陆管理

用户登陆之前需要进行注册，注册完成的每个用户都会有对应的用户名及密码，而且此信息通过加密的形式保存在区块链中，登陆过程的验证也是通过

Web3py 进行接口调用来验证用户是否在 Fcrowd 合约中注册，验证成功则登陆任务管理界面，失败则提示用户用户名或密码错误。

模块编号	002
模块名称	用户登录
输入	用户名、密码
处理	将以上用户信息输入通过前台传到后台 Controller，后台利 Web3py 将登录信息发送到 Fcrowd 合约进行查询，确认过程会调用合约的已注册用户确认函数，并返回对用户名及密码的判断结果
输出	登录成功/失败

• 用户信息管理

一个用户支持有多个账户，在发布任务和接收认识时支持选择不同账户进行支付。查看用户个人账户，会显示用户相关的个人信息，绑定钱包余额等基本信息参数。

模块编号	003
模块名称	用户信息
输入	无
处理	根据当前已登陆用户名，读取区块链中 Fcrowd 合约中的用户信息、钱包绑定信息，钱包余额等
输出	读取成功/失败

• 任务列表管理

所有已经发布的任务以列表的形式在系统中呈现，供所有用户检索查看，该列表还提供排序查找等功能。任务列表信息包任务 ID、任务发布人、任务需求 C、任务截止时间 T、任务粒度（Item）、承诺验证状态、提交状态等信息。

模块编号	004
模块名称	任务列表
输入	无
处理	后台 Controller 首先会使用 Web3py 调用 Fcrowd 合约函数 getTaskID 获取当前保存在合约中的任务 ID 数组。所有已经发布的任务通过 mapping 池保存在 Fcrowd 合约中。随后通过调用合约函数 getTasks（ID）获取所有任务信息并返回给前端

输出	任务列表信息

• 任务发布

已注册的用户可以在平台中发布任务信息，发布任务信息包括：任务需求描述 C、任务激励策略 P、模糊的激励策略 PS、任务完成时限 T、任务粒度（Item，用于重用阶段对任务提交内容进行细粒度的访问控制授权）。发布任务的过程需要将用户输入的信息进行一定的密码学操作后发送至区块链中，其他满足要求的工作者在任务信息上链之后可进行任务查询接收操作。激励策略 P 适用先提交后 Open 验证的策略，任务发布时预先提交对 P 的承诺信息 comP，发布者需要在任务截至时间 T 内上传对激励策略 P 以供其他用户通过验证承诺的真实性检验任务发布者在发布任务时是否诚实地公布了奖励策略 P。

模块编号	005
模块名称	任务发布
输入	任务需求描述 C、任务激励策略 P、模糊的激励策略 PS、任务完成时限 T、任务粒度（Item）/任务 ID
处理	将以上任务发布信息通过前台传到后台 Controller，后台随机产生承诺钥匙 crsP 对激励策略 P 生成承诺信息 comP，对每个 Item 预生成用于解密的私钥集 Rs 和共享密钥集 As。密钥数据存储层存储 P、Rs、crsP 等秘密信息。后台使用 Web3py 将 PS、comP、As、Item、C、T 等任务信息上传到 Fcrowd 合约。已经发布的任务通过 mapping 池保存在 Fcrowd 合约中。承诺提交阶段后台通过任务 ID 检索隐私数据库，将正确的承诺信息 P、crsP 通过 web3py 调用合约函数 subComP（ID）上传到合约中对应的任务条目中
输出	发布成功/失败；提交成功/失败

• 任务提交

已注册的工作者可以在平台中接受并提交任务，提交任务信息包括：提交内容 RP、任务 ID。提交任务的过程需要将用户的提交信息进行密码学操作后同步至区块链中，其他用户在提交内容上链之后可看到相应的密文提交信息。提交内容同样采用先提交后 Open 验证的策略，任务提交时预先提交对 enRP（密文 RP）的承诺信息 comRP，任务发布后发布者需要在任务截至时间 T 内上传密文 enRP 以供其他用户验证承诺的真实性以检验任务提交者在提交任务过程中是否诚实。

模块编号	006
模块名称	任务提交

输入	提交内容 RP，任务 ID/任务 ID
处理	将以上任务发布信息通过前台传到后台 Controller，后台调用合约函数 getAs(ID)获取 RP 加密密钥参数 As，使用线性配对操作产生密钥 K，并对 RP 加密生成 enRP。随机产生公共参考串 crsRP 对 enRP 生成承诺信息 comRP。隐私数据库存储 enRP、crsRP 等秘密信息。后台使用 Web3py 将 comRP 上传到 Fcrowd 合约。已经提交的任务内容通过 mapping 池保存在 fcrowd 合约中。承诺提交阶段后台通过任务 ID 检索隐私数据库，将正确的承诺信息 enRP、crsRP 通过 Web3py 调用合约函数 subComRP（ID）上传到合约中对应的任务条目中
输出	提交成功/失败；提交成功/失败

• 任务查询

任务查询模块提供了以任务 ID 为关键字检索任务提交信息的功能，检索信息包括提交状态、加密的提交内容 enRP、承诺随机串 crsRP、承诺 comRP、身份证明 Proof、Proof 验证状态\承诺验证状态、以及 RP 明文。拥有解密私钥的任务发布者可以成功打开 RP 明文，其他用户只能查询到其他非敏感信息。

模块编号	007
模块名称	任务查询
输入	任务 ID
处理	后台 Controller 调用合约函数 getSub(ID)获取所有任务提交信息并对提交人身份 Proof、comRP 进行验证。后台向密钥数据库请求当前系统登录用户的相关私钥集 Rs 对 enRP 作解密操作，解密失败返回空白，解密成功返回 RP 明文
输出	任务提交相关信息以及 RP 明文信息

• 任务重用

任务重用申请阶段，重用者首先在任务列表搜寻感兴趣的已达成交易的可重用任务，然后通过隐匿传输算法在区块链网络构建安全匿名的信息传输通道向重用目标任务发起人协商重用意向。重用意向的协商内容包括重用任务 ID、重用代价 Price、重用授权粒度 Item、重用类型 Type。

模块编号	008
模块名称	重用申请
输入	任务 ID、重用代价 Price、重用授权粒度 Item、重用类型 Type

处理	Web3py 调用合约函数 getKeyAB(Add)获取任务发布人 Add 的隐匿传输公钥 PA、PB。后台 Controller 随机产生密钥对（r1，R1）、（r,R）。系统调用隐匿通信公钥生成函数 getStealPK（r1，PA,PB）生成加密密钥 PK 和隐匿标签 lab，然后使用 PK 加密 ID、Price、Tiem、Type、R 生成密文 C。Web3py 调用合约函数 BroadInfo（C||R||lab）将 C、R 和 lab 广播到区块链
输出	消息发送成功/失败

• 重用查询

重用交易达成后，重用者会收到含有重用授权密钥 C1 的专属隐匿消息。重用解密过程需要输入密钥 C1、重用任务 ID，如果密钥正确，返回与重用申请时重用授权粒度 Item 相对应的任务提交结果。

模块编号	009
模块名称	重用查询
输入	任务 ID、重用授权密钥 C1
处理	后台首先利用 Web3py 调用 getSubmit（ID）、getTask（ID）合约函数获取该任务的密文提交信息 comRP 和共享密钥信息 As。随后对 C1、As 使用线性配对操作恢复 comRP 的解密密钥 K，使用 K 解密 comRP 得到 RP 并返回给前台
输出	满足解密条件的任务提交结果

• 隐匿消息

隐匿消息的传递基于隐匿传输算法，该算法可以安全匿名隐私地在区块链网络构建通信。任务重用者与任务发起人在协商重用细节时，都是以互相发送隐匿消息的方式进行。用户点击查询可以在隐匿消息中检索他人发给自己的秘密消息。

模块编号	010
模块名称	隐匿消息
输入	无
处理	后台 Controller 首先从隐私数据库读取当前登录用户的隐匿传输私钥对(ka、kb)，持续监听区块链中的隐匿广播消息（lab、R、C）并尝试提取并匹配计算消息中的隐匿标签 lab==H(ka•R)，如果匹配成功计算解密密钥 SK=lab+kb。使用 SK 对隐匿消息 C 进行解密并返回给前台
输出	匹配的隐匿消息

• 重用密钥

任务发布人通过查看隐匿消息接收重用意向申请（重用人公钥 R、重用人地址 Add、任务 ID、重用代价 Price、重用授权粒度 Item、重用类型 Type、），如果同意重用意向的协商内容，则生成对应的重用授权密钥 C1 并发送给重用申请人。

模块编号	011
模块名称	发送重用密钥
输入	重用者地址 Add、重用者公钥 R、重用授权粒度 Item、任务 ID
处理	Web3py 调用合约函数 getKeyAB(Add)获取地址 Add 的隐匿传输公钥 PA、PB。后台 Controller 随机产生密钥对（r1，R1）。后台从隐私密钥数据库读取当前登录用户任务 ID 的私钥集 Rs,调用隐匿通信公钥计算函数 getStealPK（r1，PA,PB）生成加密密钥 PK 和隐匿标签 lab，然后使用 PK 加密 C1（由 Rs、Item、R 经过线性配对操作生成的重用授权密钥）生成密文 C。Web3py 调用合约函数 BroadInfo（C||R1||lab）将 C、R1 和 lab 广播到区块链
输出	消息发送成功/失败

• 初始化

在用户初次登录本系统时，需要初始化用户密钥参数生成账户绑定的密钥对。后续任务发布、提交、重用等功能的密码操作都需要使用用户私钥。用户也可以通过密钥重置模块随时更新重置账户的密钥信息、提高账户使用的安全性。用户点击确认密钥重置后，系统会返回当前账户的私有 witness 信息，用户线下使用 Zokrates 工具上计算 witness 信息生成零知识身份证明 Proof，并上传将其上传到本系统。

模块编号	012
模块名称	密钥重置
输入	无/proof
处理	后台 controller 随机生成密钥对(PA,ka)、(PB,kb)、(ski,pki)，ka、kb 则存储到本地隐私数据库。Web3py 调用合约函数 setKey(PA,PB)将 PA、PB 上传到智能合约。后台对 pki 生成签名信息 certi，生成地址绑定信息 t1、t2，返回 witness(ski,pki,certi,t1,t2)。上传 Proof 过程，Web3py 调用合约函数 upProof (Proof)将 Proof 上传到智能合约用于保存账户信息 mapping 池
输出	消息发送成功/失败;上传成功/失败

• 区块链浏览器

用户可以通过区块链浏览器查看、检索区块链信息，如区块信息、交易信息等。

模块编号	013
模块名称	区块链浏览器
输入	无/块高/交易 hash/区块 hash
处理	浏览器后台通过 Web3py 接口实时监控区块链的出块信息和交易打包详情，根据用户的浏览要求将相关信息整理后推送到前台。使用检索功能时，后台依据检索关键字查询返回相应的交易详情和块信息
输出	实时更新的区块链出块、交易信息

• 数据大屏

数据大屏实时统计显示 Fcrowd 系统任务发布、接受相关数据，以图表的形式呈现相关任务交付、任务奖励分配、任务重用的统计数据。

模块编号	014
模块名称	数据大屏
输入	无
处理	后台对 Fcrowd 统计信息作分析处理，以图表的形式渲染推送到前台
输出	数据大屏信息

2.4 系统软件流程

当前本系统为试运行开发版本 Fcrowd 0.9.1，后续会推出 Docker 项目部署包。软件运行的整体流程图如下，从系统开始，首先需要部署服务端程序，服务端程序为基于 Django 开发框架的 web 应用程序。打开页面首页之后，如果用户还未进行过注册，首先参照用户注册管理的要求进行注册。注册成功后用户进行登陆进入相应的 Fcrowd 功能导航界面，功能导航主要包括七个部分：发布任务、提交任务、查询任务、隐秘消息、重用密钥、任务重用、重置密钥。首次登录系统的用户都需要使用重置密钥功能进行密钥信息初始化操作后才能继续正常使用其他功能。如果是请求者可以选择任务发布功能，填写任务信息将任务公布出去。如果是工作者则进行任务接受，选择提交任务功能，接收任务后提交相信任务内容。所有用户都可以在任务查询页面查询任务信息。如果是重用工作者，选

择任务重用功能，发送重用申请并监听隐秘消息，取得相应的重用授权后使用重用查询功能获得重用内容。重用任务接受者通过监听隐秘消息与重用工作者协商重用意向，达成共识后使用重用密钥功能计算重用授权密钥并秘密发送给重用申请人。

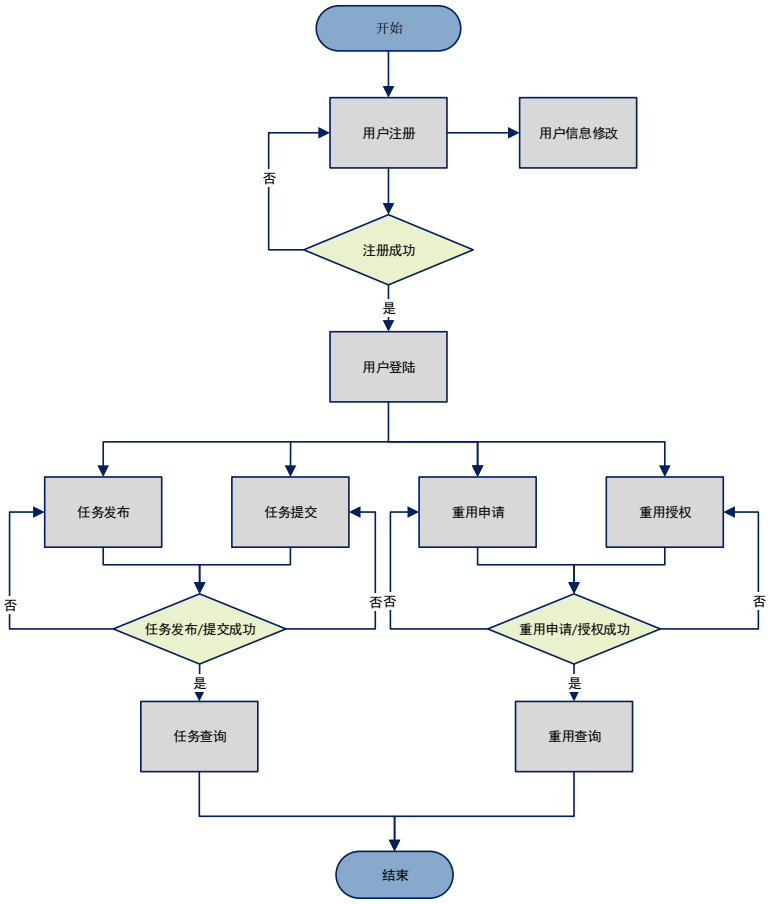


图 2.4.1 软件运行流程图

2.5 系统运行指标

• 最大用户注册数，由于客户端可以在本地部署运行，最大用户注册数对客户端服务不构成压力。服务端为以太坊，所以理论上可以支持的最大注册用户同以太坊账户数目一致，若不考虑以太坊的并发处理能力至少支持 1000000 用户的注册。

• 最大并发用户数，由于服务端为以太坊服务端，以太坊是分布式的区块链平台，支持上万用户的并发访问。

• 每秒能处理的请求/事务的数量受以太坊区块大小和出块时间的限制，以

太坊主网目前能力为 7-15 笔交易每秒，查询服务则不受限制可达 100000 笔每秒。

• 请求响应时间，系统每次请求的响应时间与以太坊操作时间相匹配，写入或者修改交易信息需要秒级的响应时间，查询类服务毫秒级时间。

• 隐匿消息匹配时间，一个新的以太坊区块通常打包数十个交易，全部进行隐匿匹配操作的时间在毫秒级别。

• 任务信息大小，由于任务信息保存全部保存在区块链，任务信息的大小受区块大小限制，单次提交信息不超过 1MB。后续版本会升级采用 IPFS 存储任务信息，只将 HASH 校验值上链，理论上任务信息大小不受限制。

• 交易确认时间，以太坊主网每 14 秒产生一个区块，因此交易确认最小时间为 14 秒。

第三章系统测试与分析

3.1 测试总体方案

在 Ubuntu18.04 系统下搭建系统运行环境，并测试整个系统所实现的所有功能。

3.2 测试环境与配置

• 测试环境

操作系统	Unbuntu18.04
客户端	Chrome 浏览器 90.0.4430
服务端	Ganache CLI v6.10.1 测试网 Python 3.6 开发语言 Django 3.0.6 开发框架 pip3 Python 包管理工具

• 环境配置（由于后端代码逻辑需要密码库支持，开发环境配置较为复杂，后续会推出 Docker 一键拉取的环境部署包）

1. ubuntu18.04、python3.6、django3.0、pip3 开发环境的安装配置可以参考网络，在此不详述。
2. 以下分别是 Ganache 客户端、Zokrates 零知识 python 工具库、Zokrates Linux 客户端工具、secp256k1 椭圆曲线 Pythony 支持库、Pypbc 线性配对 Python 支持库。环境安装具体步骤可参考各自 github 地址给出的使用文档，在此不详述。以下为 github 源码地址：

   https://github.com/trufflesuite/ganache

   https://github.com/Zokrates/pycrypto

   https://github.com/Zokrates/ZoKrates

   https://github.com/ludbb/secp256k1-py

   https://github.com/debatem1/pypbc
3. 启 Ganache-cli 客户端。命令行输入 ganache-cli 出现 Listening on 127.0.0.1:8545 提示说明启动成功。

```
HD Wallet
==================
Mnemonic:      that tuna love auto duck aunt ordinary check cram manual bless just
Base HD Path:  m/44'/60'/0'/0/{account_index}

Gas Price
==================
20000000000

Gas Limit
==================
6721975

Call Gas Limit
==================
9007199254740991

Listening on 127.0.0.1:8545
```

4. 导入数据库。本系统基于本地数据库进行隐私数据存储操作。打开源码包，将 mysql 文件夹下的 Fcrowd.sql 文件导入 MySQL 数据库，内含一个数据库 crowDB，四张数据表 reuse、subtask、task、user 用于保存各个众包阶段的隐私密钥信息。
5. 合约部署。首先启动 Ganache-cli 客户端。然后在浏览器打开 remix IDE 网页，ENVIRONMENT 选择连接到本地 ganache-cli 区块链网络，将源码包 contract 文件夹下的 Fcrowd.sol、proof.sol 编译部署到 Ganache 网络中。并更新源码 web3init.py、proof.py 的合约地址。

web3init.py > ...

```
1   from web3 import Web3
2   import json
3   web3 = Web3(Web3.HTTPProvider("HTTP://192.168.140.1:7545"))
4
5 > abi =[ ...
609 ]
610 address = "0x4B3e2d5eE094Ec73073cc242EeE7C151169205C5"
611 crow = web3.eth.contract(address=address, abi=abi)
612 web3.eth.defaultAccount = web3.eth.accounts[0]
613 #print(web3.eth.defaultAccount )
614 ##print(contract.functions.login("user3","pass3"))
615 #print(web3.eth.getBalance(web3.eth.defaultAccount))aa
```

proof.py > ...

```
1   from web3 import Web3
2   import json
3   import time
4   from datetime import datetime
5   web3p = Web3(Web3.HTTPProvider("HTTP://192.168.140.1:7545"))
6   #web3p = Web3(Web3.HTTPProvider("HTTP://127.0.0.1:8545"))
7
8 > abi = [ ...
43  ]
44  #address = Web3.toChecksumAddress("0x927933a2ad12dfe346715391eb6bbdde0e8a8e7a")
45  address = "0x64f213Fc5c8ae354Be98d40FF7c47E6Aa18EcDB1"
46  proof = web3p.eth.contract(address=address, abi=abi)
```

6. 启动 Fcrowd Web 工程

命令行进入源码包主目录下：python3 manage.py runserver 127.0.0.1:8000

本地测试 URL：http://localhost:8080/app/login

3.3 测试用例

本次测试主要以系统模块为单位进行测试，测试内容见下表，测试截图如图

3.3.1-3.3.12 所示，测试视频见附件。

序号	测试场景	功能要求	测试结果
1	账号注册	在账号注册界面实现个人账号注册	通过
2	账号登录	通过登录界面成功登入账号进入任务列表主界面，能正常显示账号信息	通过
3	密钥初始化	通过密钥重置界面初始化密钥信息并上传 proof 信息	通过
4	任务发布	通过任务发布界面成功发布任务，提交承诺信息	通过
5	任务提交	通过任务提交界面成功提交任务解决方案并提交承诺信息	通过
6	任务查询	通过查询任务界面查看当前账户已发布、已接受任务，查询已发布任务的提交结果	通过
7	隐秘消息	通过隐秘消息界面成功查看秘密信息	通过
8	任务重用	通过任务重用界面发送重用申请，在输入重用授权密钥后能够对重用任务进行解密查看	通过
9	重用授权	通过重用授权界面向重用申请人发送重用授权密钥	通过
10	账号登出	点击登出，退出当前账户的登录状态，返回初始登陆界面	通过
11	区块链浏览器	正常显示区块信息、交易信息，检索功能有效	通过
12	数据大屏	正常显示 Fcrowd 系统数据大屏信息	通过

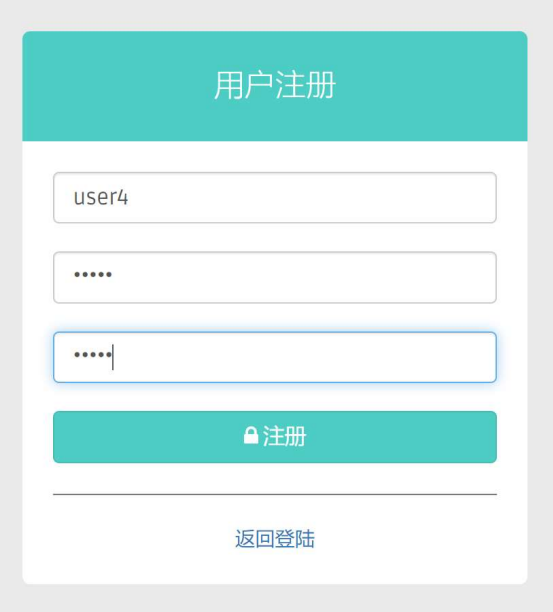


图 3.3.1 账号注册测试

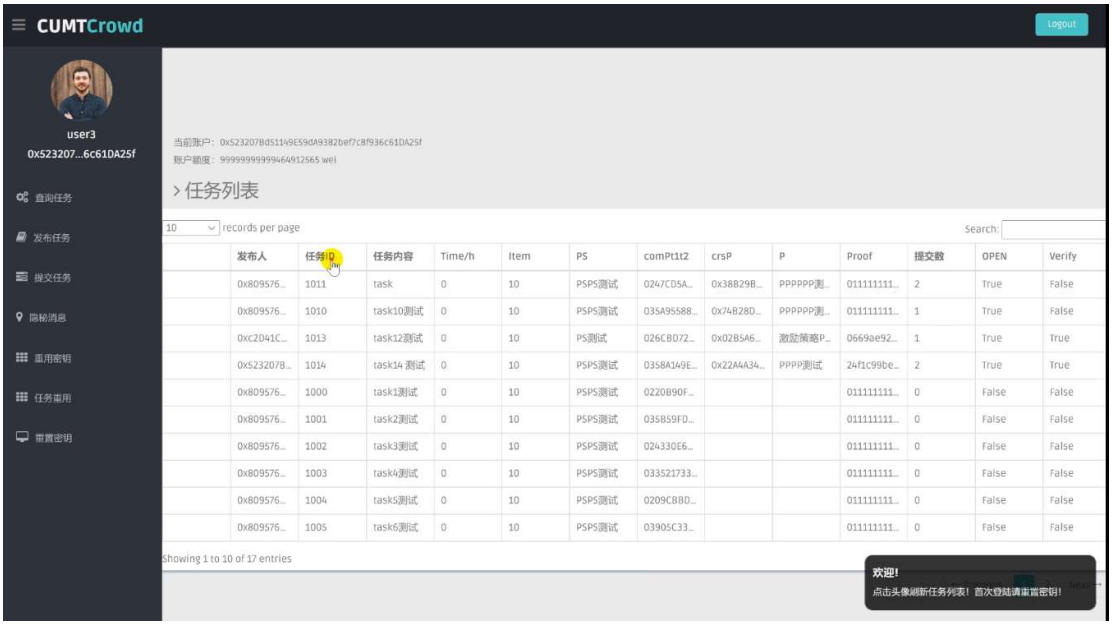


图 3.3.2 账号登录测试

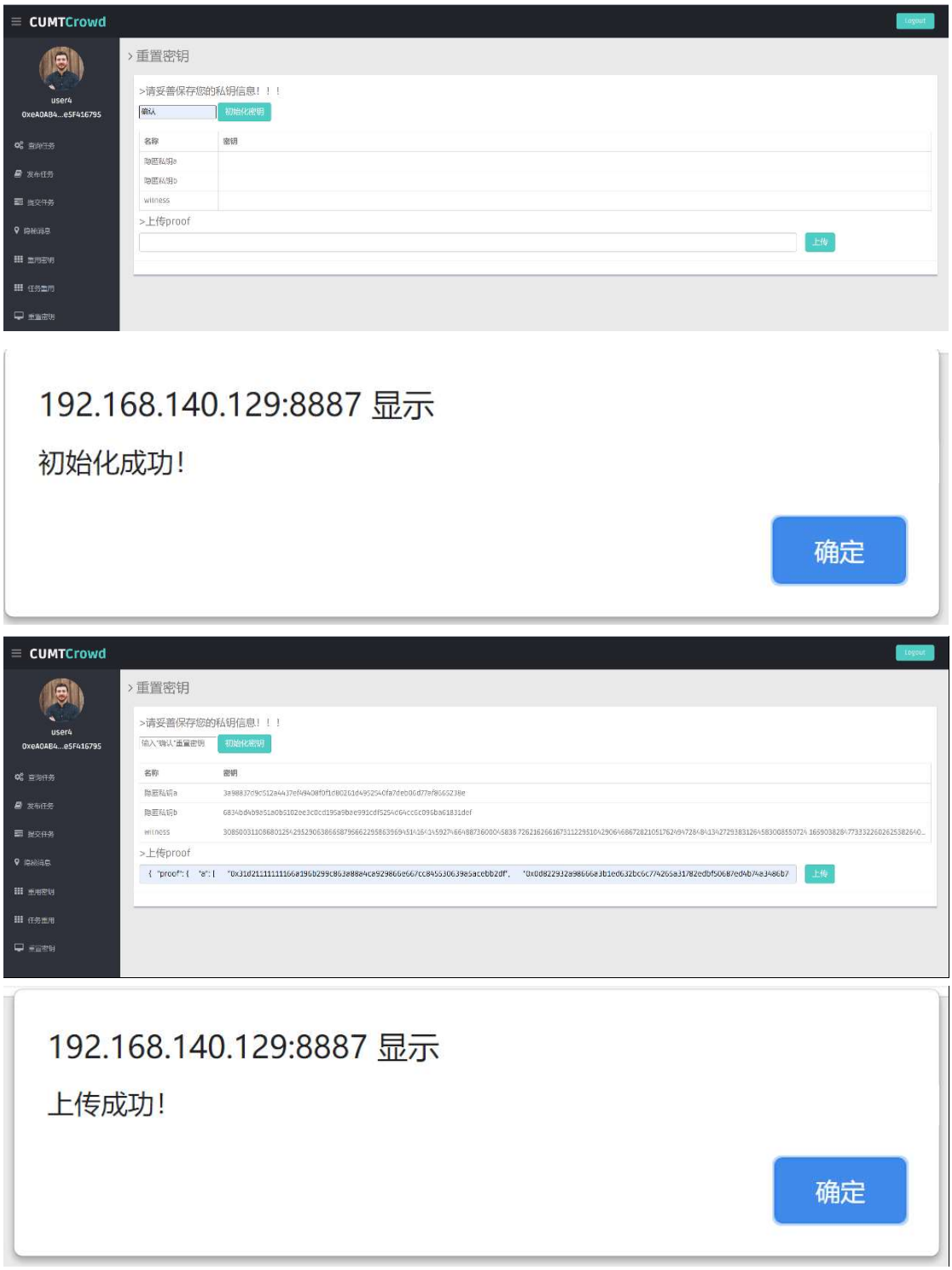
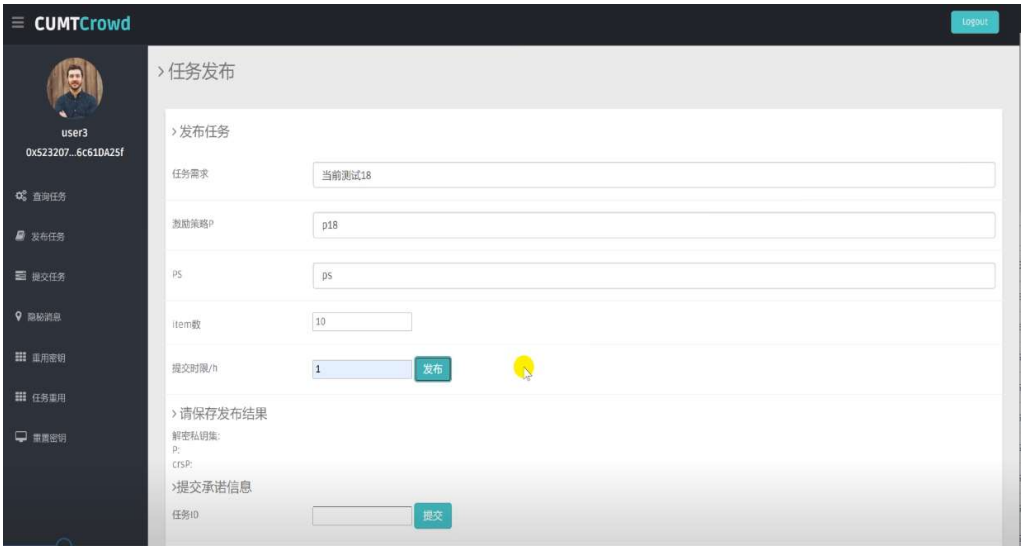


图 3.3.2 密钥初始化测试


CUMTCrowd
Logout
user3
0x523207...6c61DA25f
查询任务
发布任务
提交任务
隐秘消息
重用密钥
任务重用
重置密钥
› 任务发布
› 发布任务
任务需求
当前测试18
激励策略P
p18
PS
ps
item数
10
提交时限/h
1
发布
› 请保存发布结果
解密私钥集:
P:
crsP:
›提交承诺信息
任务ID
提交


192.168.140.129:8887 显示
发布成功!
确定

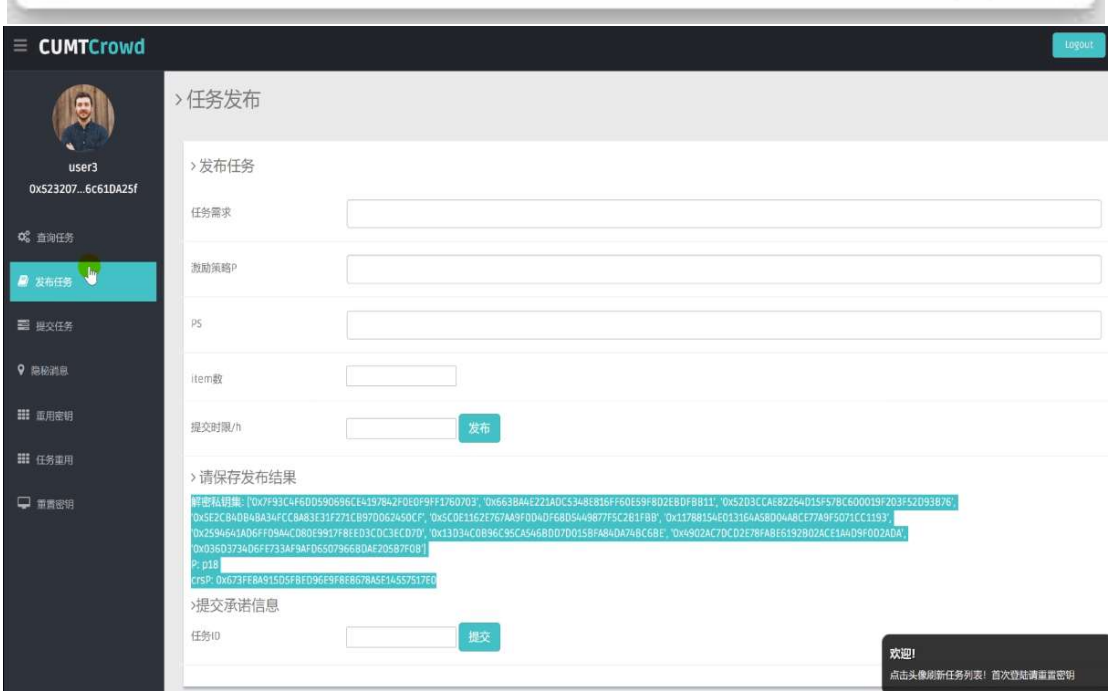

CUMTCrowd
Logout
user3
0x523207...6c61DA25f
查询任务
发布任务
提交任务
隐秘消息
重用密钥
任务重用
重置密钥
› 任务发布
› 发布任务
任务需求
激励策略P
PS
item数
提交时限/h
发布
› 请保存发布结果
解密私钥集: ['0x7F93C4F6DD590696CE4197842FOE0F9FF1760703', '0x663BA4E221ADC5348E816FF60E59F8D2EBDFBB11', '0x52D3CCAE82264D15F57BC600019F203F52D93B76',
'0x5E2CB4DB4BA34FCC8A83E31F271CB97006245OCF', '0x5C0E1162E767AA9F0D4DF6BD5449877F5C2B1FBB', '0x1178B154E01316A5BD04A8CE77A9F5071CC1193',
'0x2594641AD6FFD9A4CD80E9917F8EED3CDC3ECD7D', '0x13D34C0B96C95CA546BDD7D015BFA84DA74BC6BE', '0x4902AC7DCD2E7BFABE6192B02ACE1A4D9F0D2ADA',
'0x036D3734D6FE733AF9AFD65079668DAE205B7F0B']
P: p18
crsP: 0x673FE8A915D5FBED96E9F8E8678A5F1A557517EC
›提交承诺信息
任务ID
提交
欢迎!
点击头像刷新任务列表! 首次登陆请重置密钥

图 3.3.4 任务发布测试

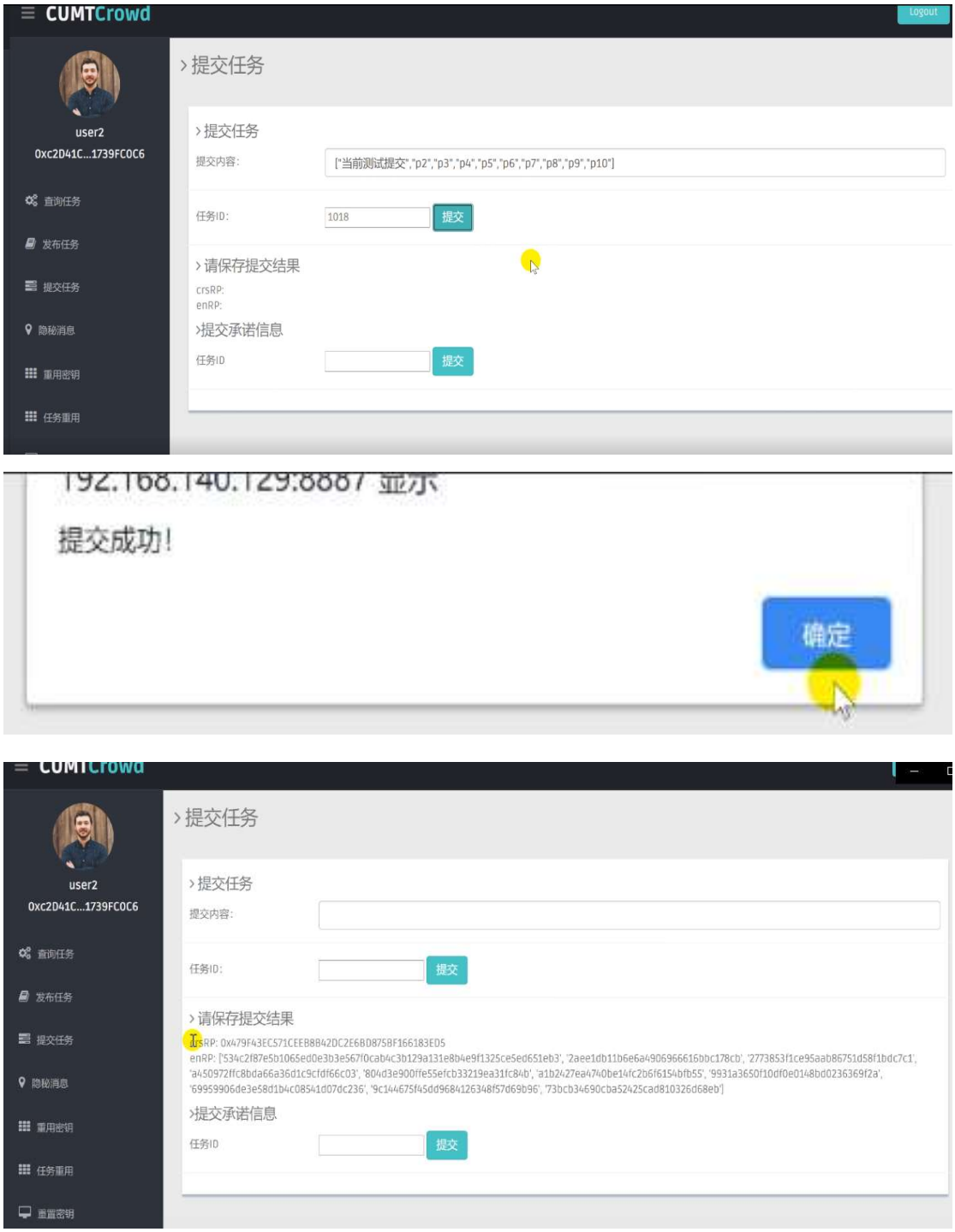
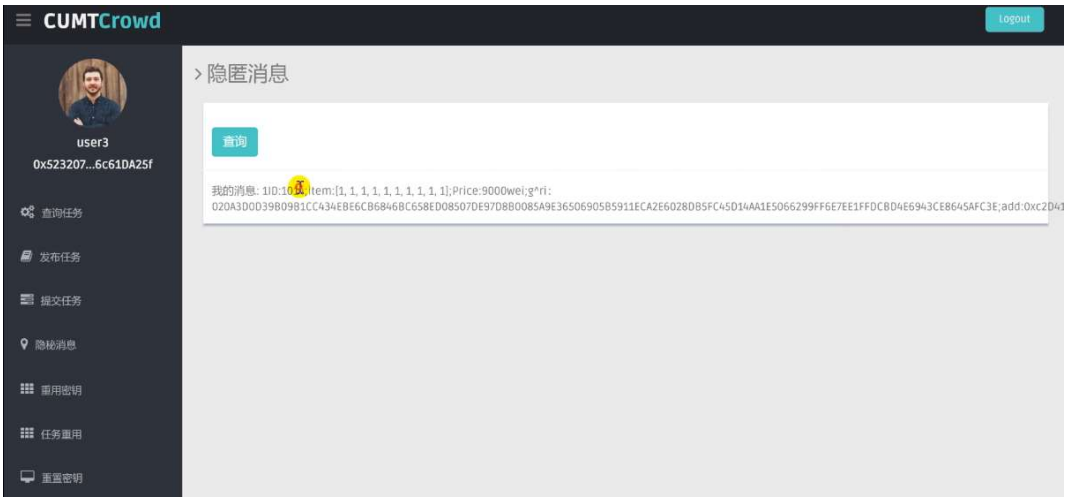

CUMTCrowd
Logout
user2
0xc2D41C...1739FC0C6
查询任务
发布任务
提交任务
隐秘消息
重用密钥
任务重用
›提交任务
›提交任务
提交内容:
["当前测试提交","p2","p3","p4","p5","p6","p7","p8","p9","p10"]
任务ID:
1018
提交
›请保存提交结果
crsRP:
enRP:
›提交承诺信息
任务ID
提交
192.168.140.129:8887 显示
提交成功!
确定
CUMTCrowd
user2
0xc2D41C...1739FC0C6
查询任务
发布任务
提交任务
隐秘消息
重用密钥
任务重用
重置密钥
›提交任务
›提交任务
提交内容:
任务ID:
提交
›请保存提交结果
crsRP: 0x479F43EC571CEEB8B42DC2E6BD875BF166183ED5
enRP: ['534c2f87e5b1065ed0e3b3e567f0cab4c3b129a131e8b4e9f1325ce5ed651eb3', '2aee1db11b6e6a4906966616bbc178cb', '2773853f1ce95aab86751d58f1bdc7c1',
'a450972ffc8bda66a36d1c9cfdf66c03', '804d3e900ffe55efcb33219ea31fc84b', 'a1b247ea4740be14fc2b6f6154bfb55', '9931a3650f10df0e0148bd0236369f2a',
'69959906de3e58d1b4c08541d07dc236', '9c144675f45dd9684126348f57d69b96', '73bcb34690cba52425cad810326d68eb']
›提交承诺信息
任务ID
提交

图 3.3.5 任务提交测试


CUMTCrowd
Logout
user3
0x523207...6c61DA25f
查询任务
发布任务
提交任务
隐秘消息
› 任务查询
› 我的任务
我发布的: 1012 1014 1017 1018
我提交的: 1011 1013 1015
› 提交结果查询
1018
查询解密
查询结果:
提交人
任务ID
enCRP
crsRP
comRPt1t2
proof
open
verivy
解密RP
CUMTCrowd
Logout
user3
0x523207...6c61DA25f
查询任务
发布任务
提交任务
隐秘消息
› 任务查询
› 我的任务
我发布的: 1012 1014 1017 1018
我提交的: 1011 1013 1015
› 提交结果查询
任务ID
查询解密
查询结果:
提交人
任务ID
enCRP
crsRP
comRPt1t2
proof
open
verivy
解密RP
0xc2041C897C862...
1018
['534c2f87e5b106...
0x479F43EC571CE...
020F4A15A8A5DE4...
0669ae92682166...
True
True
['当前测试提交', 'p...

图 3.3.6 任务查询测试


CUMTCrowd
Logout
user3
0x523207...6c61DA25f
查询任务
发布任务
提交任务
隐秘消息
重用密钥
任务重用
重置密钥
› 隐匿消息
查询
我的消息: 1ID:1011,Item:[1, 1, 1, 1, 1, 1, 1, 1, 1, 1];Price:9000wei;g^ri:
020A3D0D39B09B1CC434EBE6CB6846BC658ED08507DE97D8B0085A9E36506905B5911ECA2E6028DB5FC45D14AA1E5066299FF6E7EE1FFDCBD4E6943CE8645AFC3E;add:0xc2041

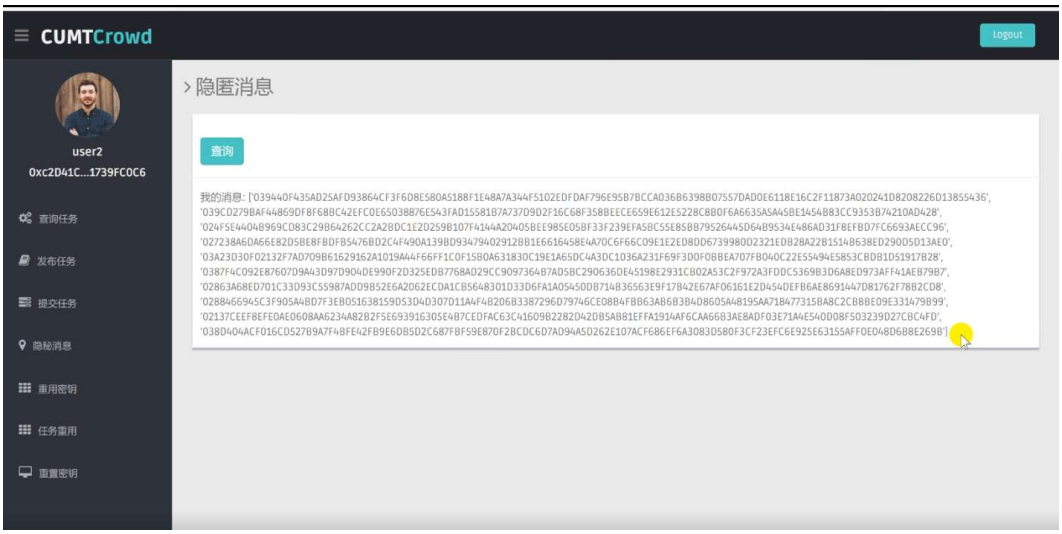
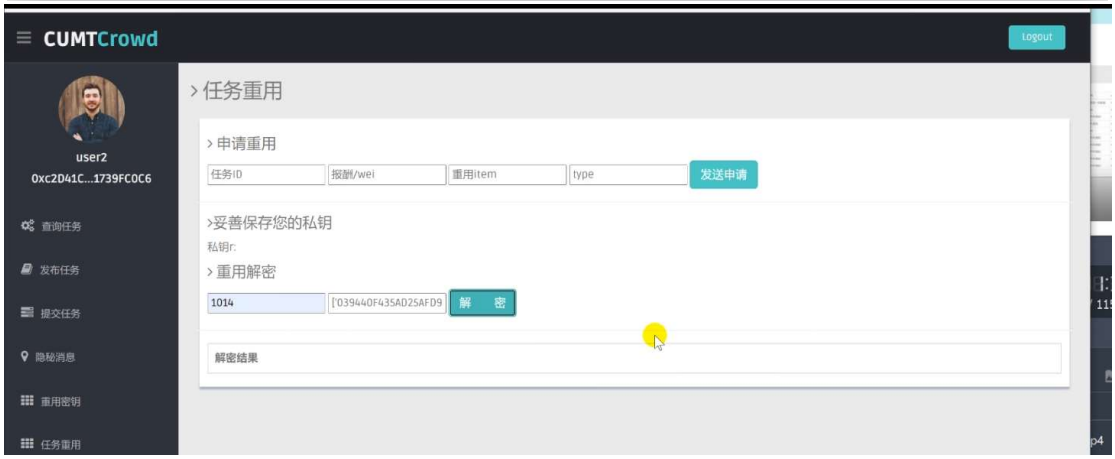

CUMTCrowd
Logout
user2
0xc2D41C...1739FC0C6
查询任务
发布任务
提交任务
隐秘消息
重用密钥
任务重用
重置密钥
隐匿消息
查询
我的消息: ['039440F435AD25AFD93864CF3F6D8E580A5188F1E48A7A344F5102EDFDAF796E95B7BCCA036B639BB07557DAD0E6118E16C2F11873A020241D8208226D13855436',
'039CD279BAF44869DF8F68BC42EFC0E65038876E543FAD15581B7A737D9D2F16C68F358BEECE659E612E5228C8B0F6A6635A5A45BE1454B83CC9353B74210AD428',
'024F5E4404B969C0B3C29B64262CC2A2BDC1E2D259B107F4144A20405BEE985E05BF33F239EFA5BC55E85BB795264450D64B9534E486AD31FBEFBD7FC6693AECC96',
'027238A6DA66E82D5BE8FBDFB5476BD2C4F490A139BD93479402912BB1E661645BE4A70C6F66C09E1E2E0B0D673980D2321EDB28A22B1514B638ED290D5D13AE0',
'03A23D30F02132F7AD709B6162916A1019A44F66FF1C0F15B0A631830C19E1A65DC4A3DC1036A231F69F3D0F0BBEA707FB040C22E5549 4E5853CBDB1D51917B28',
'0387F4C092E87607D9AA3D97D904DE990F2D325EDB7768AD29CC9097364B7AD5BC290636DE45198E2931CB0ZA53C2F972A3FDDC5369B3D6A8E0973AFF41AEB79B7',
'02863A68ED701C33D93C55987ADD9B52E6A2062ECDA1CB56483010D33D6FA1A0545 0DB714B36563E9F17B42E67AF06161E2D454DEFB6AE8691447D81762F7BB2CD8',
'0288466945C3F905A4BD7F3EB05163815 9D53D4D307D11A4F4B206B3387296D79746CE08B4FBB63AB6B3B4D8605A4B195AA718477315BA8C2CB8BE09E331479B99',
'02137CEEF8EFE0AE0608AA6234A82B2F5E693916305E4B7CEDFAC63C41609B2282D42DB5A881EFFA1914AF6CAA66B3AEBADF03E71A4E54OD08F503239D27CBC4FD',
'038D404ACF016CD527B9A7F4BFE42FB9E6DB5D2C687FBF59E870F2BCDC6D7AD94A5D262E107ACF686EF6A3083D580F3CF23EFC6E925E63155AFF0E048D6B8E269B']

图 3.3.7 隐匿消息测试


CUMTCrowd
Logout
user2
0xc2D41C...1739FC0C6
查询任务
发布任务
提交任务
隐秘消息
重用密钥
任务重用
任务重用
> 申请重用
1014
9000
[1,1,1,1,1,1,1,1,1,1]
1
发送申请
>妥善保存您的私钥
私钥r:
> 重用解密
任务ID
重用密钥c1
解 密
解密结果


192.168.140.129:8887 显示
发送成功!
确定


CUMTCrowd
Logout
user2
0xc2D41C...1739FC0C6
查询任务
发布任务
提交任务
隐秘消息
重用密钥
任务重用
任务重用
> 申请重用
任务ID
报酬/wei
重用item
type
发送申请
>妥善保存您的私钥
私钥r:
> 重用解密
1014
[039440F435AD25AFD9
解 密
解密结果

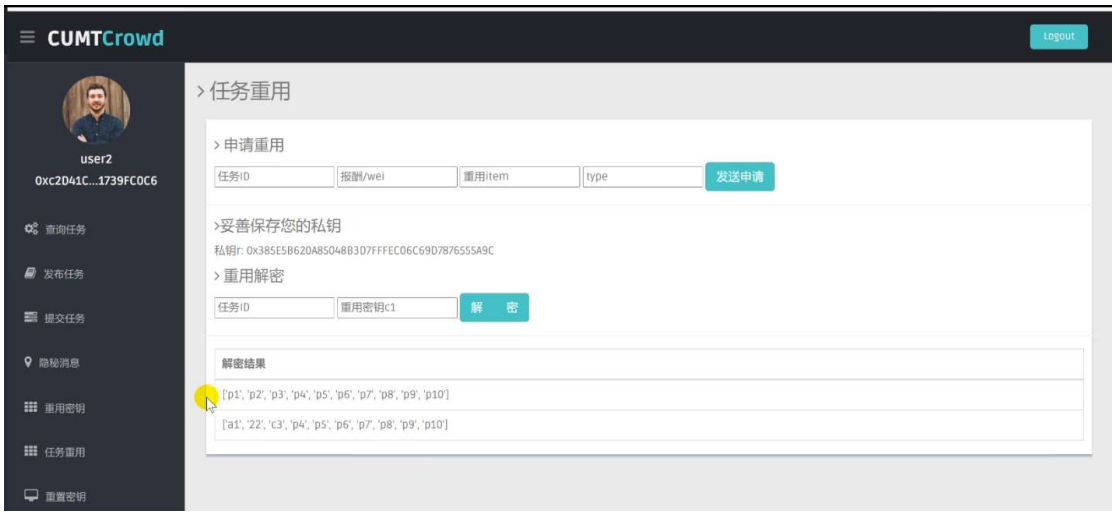


图 3.3.8 重用申请/解密测试

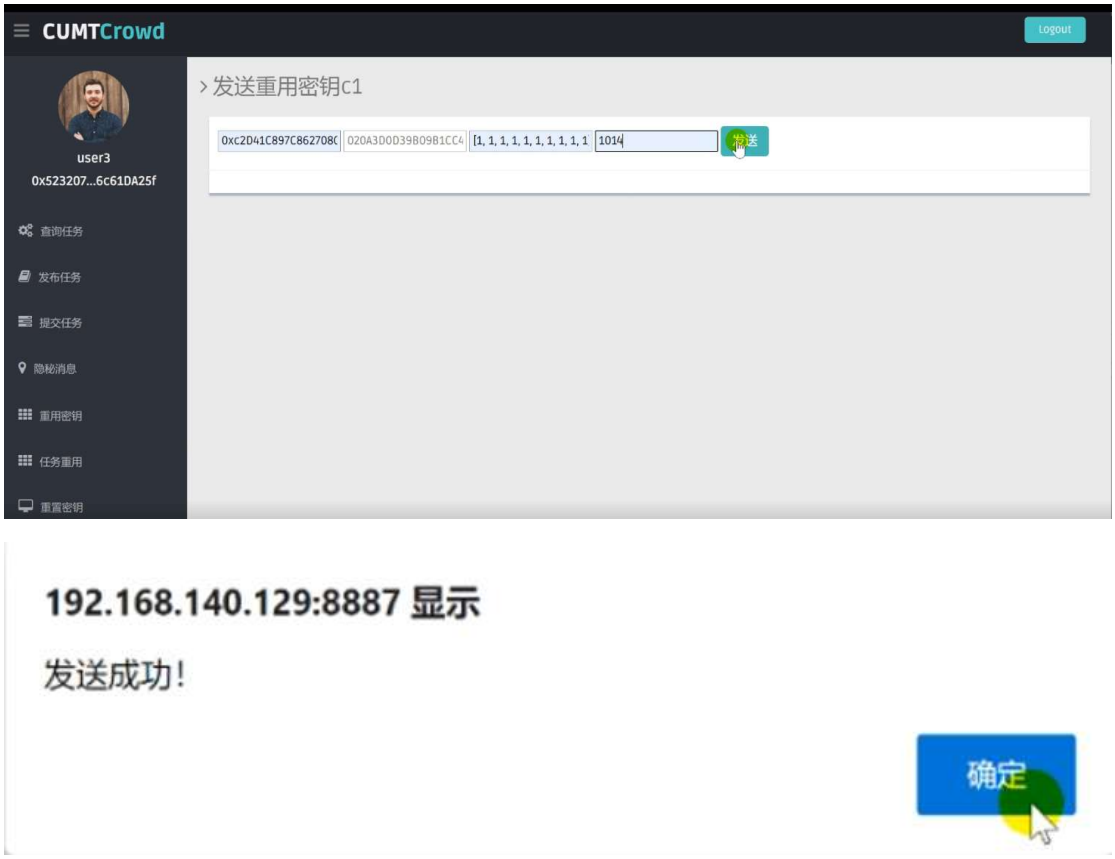
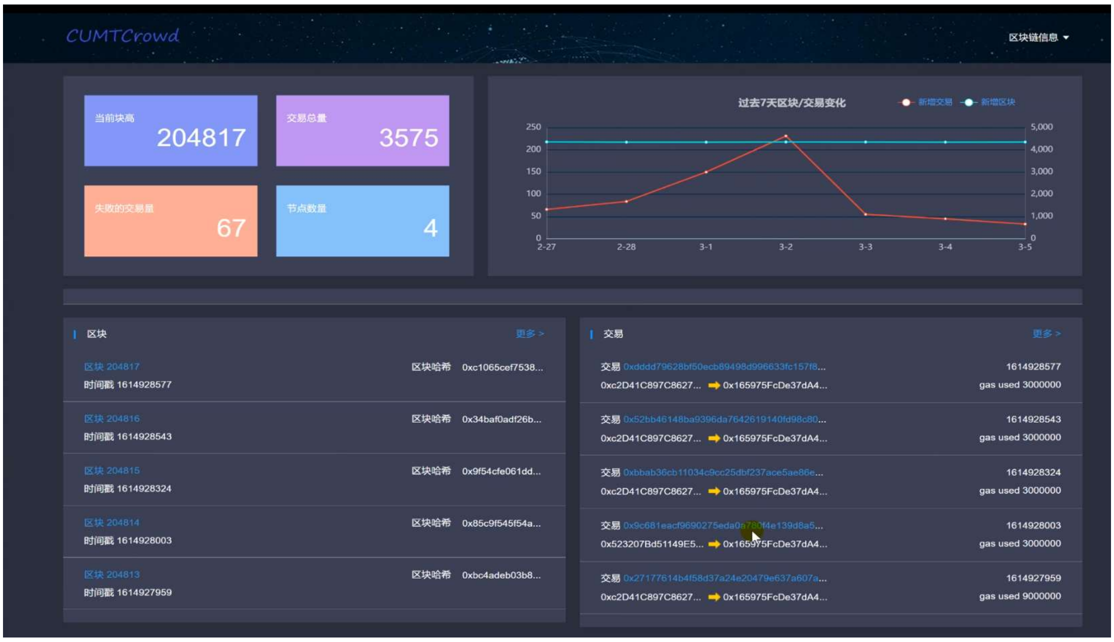
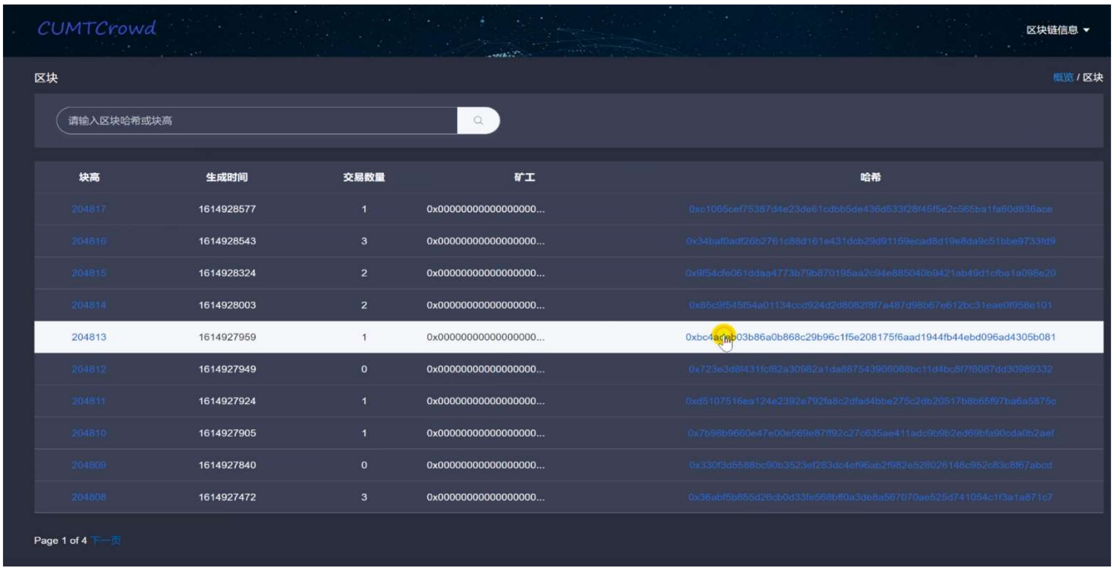
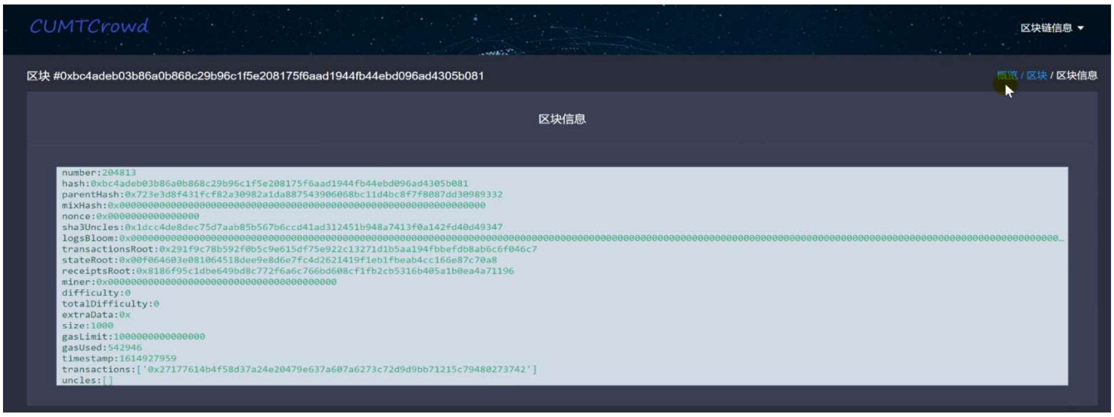


图 3.3.9 重用授权测试

图 3.3.10 账号登出测试


CUMTCrowd
区块链信息
当前块高
204817
交易总量
3575
失败的交易量
67
节点数量
4
过去7天区块/交易变化
新增交易
新增区块
区块
更多 >
区块 204817
时间戳 1614928577
区块哈希 0xc1065cef7538...
区块 204816
时间戳 1614928543
区块哈希 0x34baf0adf26b...
区块 204815
时间戳 1614928324
区块哈希 0x9f54cfe061dd...
区块 204814
时间戳 1614928003
区块哈希 0x85c9f545f54a...
区块 204813
时间戳 1614927959
区块哈希 0xbc4adeb03b8...
交易
更多 >
1614928577
gas used 3000000
1614928543
gas used 3000000
1614928324
gas used 3000000
1614928003
gas used 3000000
1614927959
gas used 9000000


CUMTCrowd
区块链信息
区块
概览 / 区块
请输入区块哈希或块高
块高
生成时间
交易数量
矿工
哈希
204813
1614927959
0xbc4adeb03b86a0b868c29b96c1f5e208175f6aad1944fb44ebd096ad4305b081
Page 1 of 4 下一页


CUMTCrowd
区块链信息
区块 #0xbc4adeb03b86a0b868c29b96c1f5e208175f6aad1944fb44ebd096ad4305b081
概览 / 区块 / 区块信息
区块信息
number:204813
difficulty:0
totalDifficulty:0
extraData:0x
size:1000
gasUsed:542946
timestamp:1614927959

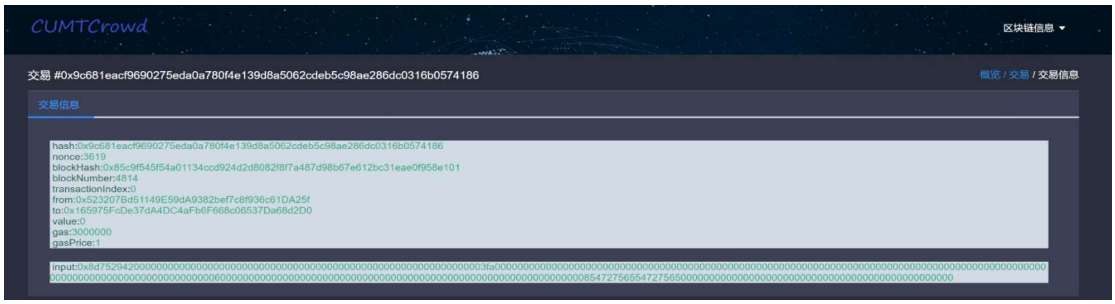


图 3.3.10 区块链浏览器测试

图 3.3.10 数据大屏测试

项目地址：https://github.com/ieCcc/mysite115-9.27