



基于区块链非同质化代币的软件订阅模型

张展鹏^{1,2,3}, 张亮^{1,2,3}, 彭凌祺⁴, 阚海斌^{1,2,3}

(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,上海 200433;
3.上海市智能信息处理重点实验室,上海 200433; 4.上海华虹计通智能系统股份有限公司,上海 201206)

摘要: 在软件销售场景中软件订阅和支持模式将成为主流,同时中心化授权在效率 and 安全性上受服务端限制。针对存储订阅信息的服务端可能遭受攻击导致盗版问题,提出基于非同质化代币的软件订阅模型,编写以太坊智能合约并利用去中心化方案保证订阅信息不可篡改。应用非同质化代币表示软件订阅,在区块链上映射软件订阅的生命周期,软件订阅流程依据代币原生操作完成,同时支持订阅购买者和销售商通过与智能合约交互实现安全的软件销售。根据订阅流程中智能合约调用的手续费开销,在模式设计层面选择发布-订阅区块链预言机,解决以太币汇率变化等相关问题,在系统设计层面设计链上存证和链下支付策略,使得订阅服务流程中的交易手续费不受以太坊主网行情的影响。实验结果表明,该模型通过非同质化代币完整映射软件订阅场景,并保障信息的可信、公开和不可篡改。

关键词: 软件即服务;区块链;智能合约;非同质化代币;数字凭证

开放科学(资源服务)标志码(OSID):



中文引用格式: 张展鹏,张亮,彭凌祺,等.基于区块链非同质化代币的软件订阅模型[J].计算机工程,2022,48(1): 24-32.

英文引用格式: ZHANG Z P, ZHANG L, PENG L Q, et al. Software subscription model based on non-fungible tokens in blockchain[J]. Computer Engineering, 2022, 48(1): 24-32.

Software Subscription Model Based on Non-Fungible Tokens in Blockchain

ZHANG Zhanpeng^{1,2,3}, ZHANG Liang^{1,2,3}, PENG Lingqi⁴, KAN Haibin^{1,2,3}

(1.School of Computer Science, Fudan University, Shanghai 200433, China; 2.Shanghai Engineering Research Center of Blockchain, Shanghai 200433, China; 3.Shanghai Key Laboratory of Intelligent Information Processing, Shanghai 200433, China;
4.Shanghai Huahong Jitong Smart System Co., Ltd., Shanghai 201206, China)

[Abstract] In the software sales scenario, software Subscription and Support (S&S) is becoming the mainstream. However, centralized authorization is limited in efficiency and security by the server end. The server storing the subscription data may be attacked, leading to privacy issues. To deal with the problem, a blockchain-based software subscription model is proposed, which employs the Ethereum smart contract and the decentralized scheme to ensure the subscription data to be tamper-resistant. By using Non-Fungible Token (NFT) to represent subscriptions, the procedure of software subscription can be mapped on the blockchain and completed by operations natively supported by NFTs. Subscribers and providers interact with the smart contract to implement secure software sales. According to the transaction fee of each step in the subscription process, at the mode design level, and a "publish-subscribe" Blockchain Oracle is selected to solve the problems related to changes in the exchange rate of Ethereum. At the system design level, by proposing the strategy of "on-chain depository, off-chain payment", the transaction fees are no longer affected by the Ethereum mainnet market. The experimental results show that the model can completely map the software subscription scenario through NFTs, and ensures the information to be credible, open and tamper-resistant.

[Key words] Software as a Service (SaaS); blockchain; smart contract; Non-Fungible Token (NFT); digital certificate
DOI: 10.19678/j.issn.1000-3428.0062500

0 概述

随着云计算技术的快速发展,软件销售模式正

在发生变化,软件订阅和支持(Subscription and Support, S&S)模式将逐渐取代软件存储介质和产品密钥封装销售模式。S&S是软件即服务(Software

基金项目: 国家重点研发计划(2019YFB2101703);国家自然科学基金(U19A2066);上海市科技创新行动计划(20222420800, 20511102200, 19511102500)。

作者简介: 张展鹏(1998—),男,硕士研究生,主研方向为区块链应用、分布式系统;张亮,博士研究生;彭凌祺,硕士;阚海斌,教授、博士。

收稿日期: 2021-08-26 **修回日期:** 2021-10-01 **E-mail:** 20210240045@fudan.edu.cn

as a Service, SaaS)的应用模式之一,用户可以自行下载和安装软件,在购买授权开始使用并根据意愿选择续订,在软件订阅期限内可以联系销售商获得支持和帮助^[1],改善了用户体验,提高了用户自由度。然而,S&S一般使用中心化形式授权,限制了授权效率并降低了安全性。如果授权服务器被攻击,则授权信息会被篡改^[2],用户很可能不能正常使用购买和订阅的软件,未经授权的用户则可以恶意使用和传播软件,从而存在盗版隐患。

区块链是一个去中心化数据库,通过分布式同步协议解决了信任问题,区块链数据具有公开性和不可篡改性^[3]。以太坊^[4]是一种被广泛应用的区块链技术,支持可编程节点,开发者通过编写和部署智能合约^[5]可以使节点完成指定功能^[4]。目前,区块链不限于作为点对点的电子现金系统提供支付服务^[3],已经成为无权威第三方的可信服务提供者^[6]。以太坊改进建议(Ethereum Improvement Proposals, EIP)由以太坊开源社区提出,改进了以太坊的方案并被并入以太坊的官方标准。EIP-721通过了新代币标准——非同质化代币(Non-Fungible Token, NFT)。NFT代表独特、不可替换的资产,具有唯一且不可分割性,能较好地映射真实世界的个人财产、虚拟藏品、资产负债等^[7]。本文将区块链技术应用到软件购买和订阅场景中,基于NFT的软件订阅模型,使真实世界的软件购买和订阅证书映射为NFT,通过NFT的铸造、授权、鉴权和销毁实现软件使用权限管理。

1 相关研究

1.1 软件支持和订阅

在市场经济背景下,软件销售模式从“卖产品”转向“卖服务”。20世纪90年代,IBM提出服务科学、管理与工程(Service Science, Management and Engineering, SSME)学科体系。Amazon、Google、Yahoo等公司纷纷转型成为软件服务公司^[8],SaaS快速发展。SaaS将应用程序当成Web服务交付,并根据用户使用计划收费,该模式推动了软件销售中永久许可向订阅许可的转变。

S&S是SaaS的典型代表,用户可以自行下载和安装软件,在完成购买或续订后连接服务器获得使用授权;用户向授权服务器提交登录信息,授权服务器完成用户认证后向用户颁发数字凭证。在订阅有效期内,客户端使用该数字凭证请求服务端资源。软件订阅降低了用户的拥有成本,用户能自由地使用软件并始终保持软件为最新版本。以S&S模式销售软件使销售商的资金流更平稳^[9],降低了经营管理风险^[1]。这一方面决定了S&S容易成为创业团队的选择;另一方面,创业团队的安全方案也容易受到客观条件限制。S&S依赖中心化的认证和授权,假

如中心化服务器被攻击,用户认证和订阅信息将被篡改,造成软件资源被盗用^[10]。

S&S模式已在软件销售中取得了很大进展,然而针对授权服务器的攻击是中心化场景的固有隐患,盗版问题也长期威胁软件服务业。在软件购买和订阅的授权环节探索和应用去中心化方案将对解决上述问题具有积极作用。

1.2 区块链

经过十余年的发展,区块链应用领域从可编程货币拓展到了可编程金融和可编程社会,已在数字世界中形成了可信赖、可溯源^[11]的生态系统^[12]。区块链是存储交易历史的分布式数据库,全体节点运行共识算法^[13-15],根据结果将交易历史储存在区块上。2008年,中本聪设计了一种点对点的电子现金系统,首次出现了对区块链的描述和应用。在比特币网络中,区块以单链表数据结构存储在节点上。节点可以参与通过暴力枚举寻找下一个合法区块头的竞争,俗称“挖矿”,优胜者将获得比特币作为奖励;当下一个区块被找到时,这个新区块会记录本区块全体交易和上一个区块的哈希。比特币的交易历史是不可篡改的^[16],原因是哈希逆运算极为困难,即使比特币网络超过50%的计算能力被作恶者控制,作恶者篡改交易历史也将付出指数爆炸的计算能力,作恶获利将小于参与“挖矿”的收益^[3]。

2013年,以太坊初创白皮书^[4]发布,被普遍认为开启了区块链2.0时代。以太坊是一个公开无需许可的区块链,实现了节点可编程。可编程节点的本质是一个在以太坊上自动运行的去中心化应用程序(Decentralized Application, DApp),由Solidity^[17]等图灵完备的高级编程语言实现^[18]。DApp继承了以太坊的去中心化、不可篡改、可验证、公开透明^[19]等关键特征。智能合约拓展了区块链的功能,允许开发者在继承以太坊关键特征的同时实现诸多新应用。以太坊的应用热点集中于去中心化金融(Decentralized Finance, DeFi)、NFT等新领域。一些优秀的应用项目赢得了研究者、开发者和投资者的青睐,例如去中心化交易所Uniswap、服务于链下数据获取的预言机Chainlink、基于NFT的养猫游戏《以太猫》(CryptoKitties)等。

1.2.1 去中心化金融

DeFi是指建立在以太坊上的点对点金融基础设施^[20],主要包括去中心化借贷、区块链预言机(Blockchain Oracle)和去中心化交易所。DeFi是在以太坊上开发的成功DApp之一,处于爆发式增长期,继承了以太坊的安全性和透明性,允许任何人或机构随时随地开展金融活动。去中心化借贷在理论上不需要身份认证,用户可以借出闲置资金,也可以有抵押或无抵押地借入数字资产,上述数字资产买

卖均须通过合约交互完成。区块链预言机提供了将链下数据可信地映射到链上的机制^[21],已被用来为链上金融活动服务,为去中心化借贷提供基准价格,防止抵押、兑换等价格偏离市场^[22]。去中心化交易所是完全匿名的,基础功能是去中心化借贷,通过引入稳定币、预言机等机制保证交易公平性。

DeFi的技术进步不仅体现在金融领域,而且证明了一种无权威第三方、可公开审计、完全开放的金融架构在技术上是可行的。同时,研究者和学习者也应该注意到去中心化金融活动是难以被监管的,合约安全漏洞、诈骗、非法金融活动等均会威胁 DeFi 生态。

1.2.2 非同质化代币

NFT是在EIP-721上被提出并通过的以太坊代币标准,也被称为ERC721代币^[7]。与ERC721代币相对的概念是ERC20代币^[23],即同质化代币。以太币(Ether,ETH)是典型的ERC20代币。与同质化代币不同,同种NFT之间不等同也不可互换,每个NFT不可再分也不可合并。NFT最初的思想是WANG等^[24]在2017年提出的染色代币设想,该设想用染色代币在比特币网络上表示真实世界资产。以太坊社区通过NFT实现了在以太坊上映射大量(至少 2^{128} 个^[7])同类资产。为了清晰地展示非同质化代币的特征,本文对ERC721代币与ERC20代币进行了比较,如表1所示。

表1 ERC721代币与ERC20代币的比较

Table 1 Comparison between ERC721 and ERC20 tokens

特征	ERC721代币	ERC20代币
可互换	否	是
可再分	否	是
可合并	否	是
唯一表示单个资产	是	否
支持元数据拓展	是	否

第一个被熟知的基于NFT实现的以太坊DApp是于2017年被推出的《以太猫》。以太猫的本质是以太坊上的NFT,通过递增的ID保证每一只以太猫都是独一无二的,即NFT的tokenId^[7],ID越小的以太猫价值越高。从2017年起,NFT促进了艺术品、奢侈品等真实世界资产的通证化。另外,借助NFT,在以太坊上公开、不可篡改地标记票据、许可证等票证也变得容易实现。NFT虽然被广泛地应用于真实世界资产标记上链,但是相关基础研究及理论体系仍然处于欠缺状态^[19]。

2 系统设计

NFT从设计之初^[24]就适合在以太坊上表示真实世界资产,在软件订阅场景中销售商需要大量认证和分发激活令牌。本文设计一种基于NFT的软件订

阅认证系统 Smart-Subsc,通过 NFT 支持的原生操作^[7]对应实现软件激活和服务端资源访问权限的管理。与传统的中心化认证方案不同,基于 NFT 的订阅认证信息存储在以太坊,具有公开透明性和不可篡改性,这将有助于提升订阅者与销售商之间的信任。

2.1 Smart-Subsc需求分析

针对传统客户端-服务端模式的订阅和授权场景中,服务端在受到攻击后用户的授权信息可能被篡改的问题,Smart-Subsc将提供解决方案,实现下列功能:

- 1) 订阅信息不能被恶意篡改,订阅者已经购买的订阅不能被攻击者转让,也不能被恶意销毁;被订阅者取消和到期的订阅将永久失效,不能被恢复。
- 2) 订阅者和销售商的操作应该足够简单,订阅者可以自由地购买,也可以自由地提前取消并获得退款;销售商能够查询订阅状态,激活即将生效的订阅,在订阅到期后使订阅永久失效。
- 3) 软件订阅会定期生效,在生效前,订阅者可以取消订阅并获得退款;一旦开始生效,订阅者将无权限对订阅执行操作。在订阅有效期内,有且仅有销售商能对订阅执行操作。当订阅到期后,它会永久失效,任何人都不能操作它。在用户不主动转让订阅的前提下,除订阅者和销售商外,其他人不能对订阅执行任何操作。另外,订阅者可以自由地提前订阅,销售商只需要定期验证并激活即可。

2.2 Smart-Subsc功能设计

本文在传统客户端-授权销售商的基础上应用基于NFT的智能合约对软件订阅进行管理。订阅者可以调用合约方法完成购买和提前取消,也可以查询订阅价格等必要信息;销售商可以通过合约调用查询和按期确认订阅,以及使过期的订阅永久失效。

图1给出了Smart-Subsc主要功能时序图。订阅购买者需要拥有链下身份(例如用户名、电子邮箱)和链上身份(即以太坊地址)。在步骤1~步骤3中,订阅购买者向软件销售商提交自己的链上身份,软件销售商通过验证订阅购买者对消息的签名来确认链上身份真实性^[25]。在步骤4和步骤5中,订阅者通过与Smart-Subsc合约交互,在以太坊发起转账来购买订阅,Smart-Subsc合约会在购买成功后返回订阅对应NFT的ID。在步骤6中,订阅者在链下向销售商发送消息,提供自己已经成功购买的订阅及其对应的tokenId。在步骤7和步骤8中,销售商可以通过与Smart-Subsc合约交互来验证订阅者的购买,即验证对应NFT的存在性和拥有者。在步骤9~步骤11中,销售商可以按期再次验证和激活订阅者的订阅,在激活成功后,销售商向订阅者颁发该期软件订阅的激活数字凭证,授权订阅者在对应期限内正常使

用软件。文献[25]指出通过分发激活令牌作为使用订阅软件的数字凭证属于单点登录(Single Sign On, SSO)场景。在步骤12和步骤13中,当该期订阅到期时,销售商再次与合约交互,销毁对应的NFT,订阅者也不能再使用过期的令牌访问软件资源。

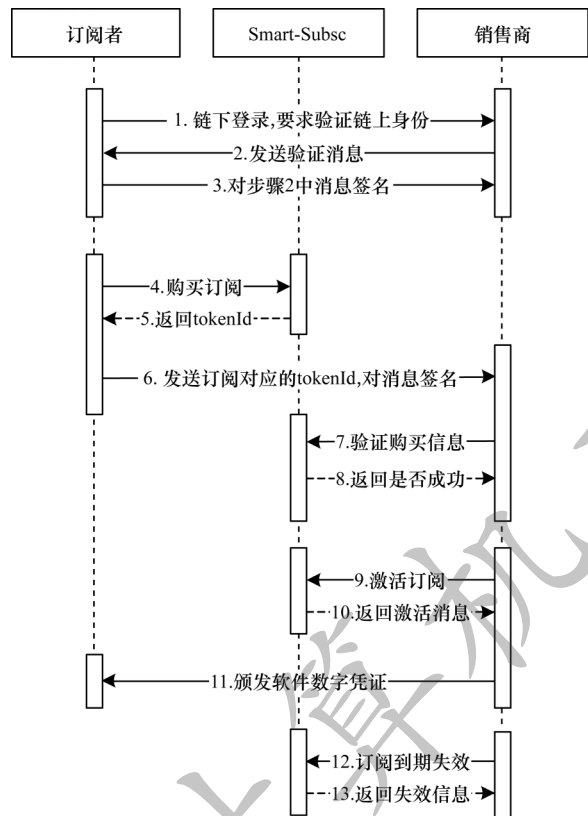


图1 Smart-Subsc主要功能时序图

Fig.1 Sequence diagram of the main functions of Smart-Subsc

相比文献[25]给出的模型中只能通过NFT余额验证订阅者是否完成购买最近一期订阅,本文模型允许订阅者提前购买任意多期订阅,也允许订阅者在任意时候提前取消。另外,本文模型基于NFT原生支持的操作权限管理充分实现了订阅的按期激活和失效,客观和完整地将软件订阅流程映射到链上。本节的功能设计要求订阅者在链上支付以太币购买订阅,针对以太币支付受到以太币对法定货币汇率波动的影响^[26],将在下文中讨论对应的解决方案。

2.3 Smart-Subsc合约设计

在传统客户端-授权服务端的基础上设计与实现了基于NFT的智能合约,支持订阅者通过与合约交互完成购买和取消订阅,销售商也能通过与合约交互查询、确认、激活订阅者的有效订阅,以及使过期订阅永久失效。Smart-Subsc智能合约去中心化地管理了全体订阅的状态,充分利用了NFT的特性。

Smart-Subsc合约存储了销售商的服务端地址server、当前订阅价格price、取款服务费率serviceFeeRate、下一个NFT的ID tokenIdMax和订阅者余额deposit,具体功能如下:

1)server。在合约构造时初始化为合约构造者,之后保持不变。

2)price。在合约构造时初始化,只能由销售商更新。

3)serviceFeeRate。在合约构造时初始化,只能由销售商更新;订阅者在提取余额时会扣除服务费来补偿合约支付的转账手续费。

4)tokenIdMax。初始化为0,每铸造一个新NFT时递增;tokenId是NFT的唯一标识,与订阅一一对应。

5)deposit。构造成订阅者地址到余额的映射,订阅者支付订阅的余额会存在合约,订阅者余额表给出了每个订阅者的余额;订阅者可以随时提取自己的余额。

当购买订阅时,订阅者可以调用合约的purchaseSubscription函数。购买订阅实际上会铸造(mint)^[7]新NFT,tokenId由递增的tokenIdMax指定。另外,新NFT只会批准(approve)^[7]销售商的以太坊地址执行转账操作。在购买过程中,订阅者的余额可以部分或完全抵扣支付金额。算法1给出了purchaseSubscription函数的调用流程。

算法1 purchaseSubscription函数调用

输入 订阅价格price、订阅者余额deposit、订阅者地址sender、销售商地址server、订阅者支付的金额value、下一个NFT的tokenIdMax

输出 订阅者是否成功购买订阅

//优先用余额抵扣购买,判断余额是否足够支付

1.newDeposit = deposit+value-price

2.if newDeposit < 0 then

3.return FALSE

4.end if

5.deposit = newDeposit//购买成功,更新余额

//铸造NFT,tokenId是tokenIdMax,拥有者是订阅者;销

//售商被批准对NFT执行操作

6._safeMint(sender, tokenIdMax)

7.approve(server, tokenIdMax)

8.tokenIdMax递增,保证tokenId唯一性

9.return TRUE

订阅者只被允许在订阅生效前取消订阅,通过调用合约的cancelSubscription函数实现。取消订阅实际上会直接由订阅者销毁(burn)^[7]对应的NFT。在订阅生效前,订阅者可以在任何时候销毁它。

在新的订阅周期开始时,销售商将根据订阅者的以太坊地址和NFT的ID来确认订阅者的订阅并充当NFT的消费者,将对应的NFT转账给自己。销售商将通过调用合约的activateSubscription函数激活新的订阅,activateSubscription在运行前须确认NFT的拥有者是即将订阅生效的订阅者,批准者是销售商的以太坊账户。在订阅生效且NFT从订阅者到销售商的转账完成后,订阅者将无权再对NFT做任何操作,即订阅者不能对已经生效的订阅进行取消操作。在订阅周期结束时,销售商会将到期订阅对应

的 NFT 销毁,通过调用合约的 `expireSubscription` 函数实现。这样任何人不能通过恶意篡改使已经过期的订阅重新生效。

在 `activateSubscription` 和 `expireSubscription` 执行前,合约会分别检查订阅对应的 NFT 是否满足被激活和被失效的条件,即校验函数 `subscriptionNotActivated` 和 `subscriptionNotExpired` 是否返回真。这样的校验是通过确认 NFT 的拥有者 (`owner`)^[7] 和批准者 (`approver`)^[7] 实现的。算法 2 和算法 3 分别给出了 `subscriptionNotActivated` 和 `subscriptionNotExpired` 函数的调用流程。

算法 2 `subscriptionNotActivated` 函数调用

输入 NFT 对应的订阅者 `owner`、销售商地址 `server`、`tokenId`

输出 订阅对应的 NFT 是否可以被激活

//NFT 拥有者对应订阅者,确认该 NFT 未被转账给其他人

```
1.if owner != ownerOf(tokenId) then
2.return FALSE
3.end if
//销售商能对该 NFT 执行转账操作,确认该 NFT 的拥有
//者未取消销售商的转账权限
4.if getApproved(tokenId) != server then
5.return FALSE
6.end if
7.return TRUE
```

算法 3 `subscriptionNotExpired` 函数调用

输入 销售商地址 `server`、`tokenId`

输出 订阅对应的 NFT 是否永久失效

//确认 NFT 已经被转账给销售商,有且仅有 NFT 的拥有者能销毁它

```
1.if ownerOf(tokenId) != server then
2.return FALSE
3.end if
4.return TRUE
```

Smart-Subsc 合约为用户提供余额账户,这样做可以允许订阅者更加灵活地选择支付时机。在购买订阅时,订阅者的余额可以用来支付;在取消订阅时,退款会自动转到订阅者的余额。订阅者可以在任意时候通过调用合约的 `withdraw` 函数提现余额,从余额账户取款到自己的账户,退款将被扣除服务费。

另外,Smart-Subsc 合约也提供了基本的查询函数,支持订阅者查询服务端地址、订阅价格、服务费率 and 订阅者余额。针对合约的管理者,即软件销售商,Smart-Subsc 合约提供了 `updatePrice` 和 `updateServiceFeeRate` 函数分别更新价格和服务费率。

图 2 给出了 Smart-Subsc 合约的软件架构。Smart-Subsc 合约通过调用 EIP-721^[7] 合约实现了各项功能;订阅者和销售商分别与合约交互完成各自的操作,更新价格、服务费率以及激活和使订阅过期的功能只允许销售商的以太坊账户操作。通过合约提供的余额账户 `deposit`,为订阅付款与订阅生效之间充分解耦,订阅者在购买和提前取消操作上具有更高的自由度。由于 NFT 的铸造、转账和销毁都通过向以太坊发送交易完成,订阅的状态改变会被永久且不可篡改地记录在链上,这些订阅历史也是可追溯且公开的。

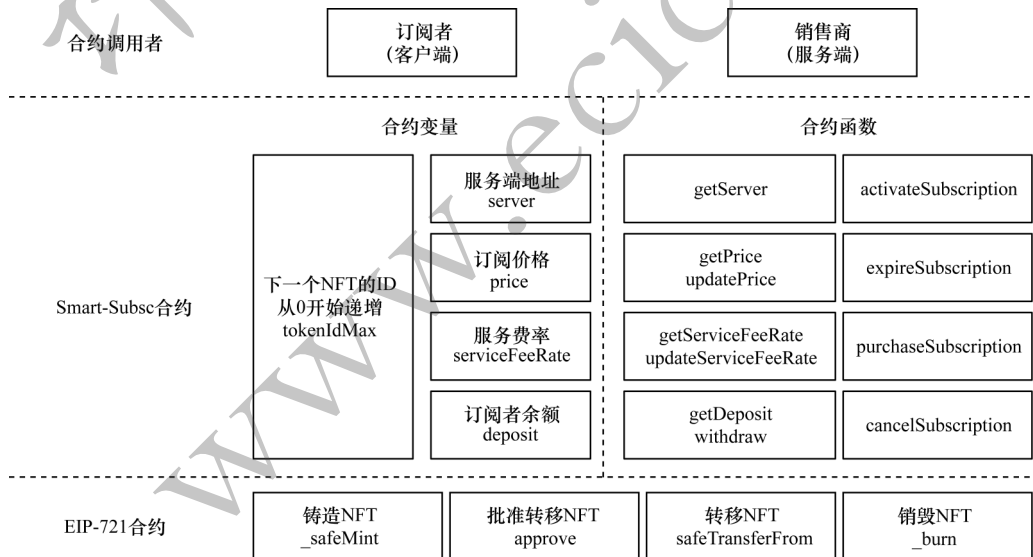


图 2 Smart-Subsc 合约的软件架构

Fig.2 Software architecture of Smart-Subsc contract

本文实现的合约支持以太币支付。文献[26]指出:以太币对真实世界中货币的汇率存在波动性。为了更好地将真实世界中软件销售映射到链上,一种可供考虑的方案是支持使用汇率波动更小的数字货币支付。事实上,Smart-Subsc 可以支持任意

ERC20 代币^[23],例如稳定币 USDT、DAI 等。稳定币可以更好地将订阅的真实世界价值映射到链上。虽然稳定币锚定了真实世界中法定货币的价值,但是它并非像以太币一样是以太坊账户在链上发起交易所必需的。为了使用稳定币支付订阅,订阅者通常

需要在交易所将以比特币兑换成稳定币,但这会导致额外的手续费开销。

2.4 区块链预言机应用研究

为了使 Smart-Subsc 更好地在链上映射链下的软件订阅销售,将从系统设计的角度讨论数字货币与真实世界中法定货币之间的价值映射问题。在 2.3 节中提及直接在合约支持稳定币支付会给订阅者带来额外的手续费开销,这可能从经济角度限制了其应用。

稳定币本身是受信任的机构发行的代币,类似地,提供比特币与法定货币之间汇率的区块链预言机也是受信任的第三方服务,它以合约形式被部署在链上,充当了链上数据与链下信息的中间层。相比支持稳定币支付,应用汇率报价预言机的方案会根据比特币与指定货币之间的汇率更新以比特币为单位的价格,订阅者在支付交易手续费的同时无须兑换其他数字货币,减轻了经济负担。文献[27]对区块链预言机的工作机制做了总结,一般地,区块链预言机的设计模式包括即时读、发布-订阅和请求-响应 3 种,具体如下:

1)即时读。实现方法为数据被静态存储在预言机,通过合约函数调用读取。优势为实现简单,用户几乎不需要支付手续费。劣势为以太坊存储成本高,难以适应大规模数据,且不适合存储高频变化的数据。

2)发布-订阅。实现方法为预言机以发送交易的形式广播数据,用户同步以太坊状态即接收数据。优势为适合高频变化数据,用户没有额外的网络资源开销。劣势为发布者将为每一次数据更新发送的交易支付手续费。

3)请求-响应。实现方法为用户主动请求预言机,在与链下资源交互获得结果后,预言机发送交易,异步返回结果。优势为充分地发挥了链下资源,例如服务器磁盘和 CPU 在存储和运算时成本较低。劣势为相比发布-订阅,请求-响应是有状态的,预言机需要记录用户与请求之间的对应关系,开销更高。

Smart-Subsc 销售的软件订阅以美元标价,需要获得比特币(ETH)对美元稳定币(USDT)之间的汇率 ETH/USDT,更新以比特币为单位的价格。由于很多数字资产交易所会提供受到相关投资者认可的汇率报价,因此假设 Smart-Subsc 找到了一个受信任的汇率报价预言机,可以从交易所的资金池等渠道获得相关投资者共识的 ETH/USDT 报价。

ETH/USDT 报价处于实时变化^[26],不适合应用即时读设计模式,否则预言机会被频繁更新,导致预言机拥有者支付大量手续费。关于请求-响应设计模式,客户端合约获得 ETH/USDT 报价的过程对预言机而言无须实现成有状态的,也不涉及链下大规模运算,因此请求-响应设计模式也是不必要的。发布-订阅设计模式适合用来获得 ETH/USDT 报价:一方面,用户订阅的汇率报价完全以交易形式在全网

广播,不存在额外的网络资源开销;另一方面,汇率报价更新是合约在相关函数被调用时主动完成的,软件销售商可以根据汇率变化控制更新时机,主动控制汇率报价更新带来的手续费开销。另外,价格更新过程发生在链上,对软件订阅者是公开透明的。

3 实验结果与分析

Smart-Subsc 合约使用 Solidity^[17]基于 Truffle 框架^[28]开发,由 Ganache^[28]提供以太坊测试网,测试文件遵循 Mocha 测试库^[28]标准。Smart-Subsc 调用了 OpenZeppelin 库^[29]对 EIP-721^[7]的实现,使用两两互异的 NFT 表示已购买的软件订阅,应用 NFT 支持的转账、批准和销毁操作实现订阅权限管理,并且开放了 Smart-Subsc 的源代码,参见 <https://github.com/KofClubs/Smart-Subsc>。针对最近比特币价格处于高位,以太坊主网中交易手续费价格较高的问题,本文给出解决方案。

3.1 Smart-Subsc 链上操作开销

与比特币网络^[3]类似,以太坊的状态转移也是交易驱动的^[4]。对 Smart-Subsc 合约,更新价格和服务费率,NFT 的铸造、转移和销毁本质上均会使以太坊发生状态转移,需要合约函数的调用者发送交易来实现,并支付交易手续费。以太坊的交易手续费以 gas 为单位,根据交易对应的以太坊虚拟机汇编指令计算得到^[6]。全网 gas 价格(gasPrice)^[4]在不断变化。一般地,在全网待处理交易(pending transaction)较多时,为了使自己的交易被更快确认,交易发起者需要提供更高的 gas 价格。

在本地以太坊网络测试 Smart-Subsc 构造函数(constructor)、updatePrice、updateServiceFeeRate、activateSubscription、expireSubscription、purchaseSubscription、cancelSubscription 和 withdraw 对应的交易手续费开销,以 gas 为单位。表 2 给出了上述合约函数调用者为交易支付的手续费开销范围。值得注意的是,调用 purchaseSubscription 支付的交易手续费与 NFT 铸造量和订阅者余额均有关。

表 2 Smart-Subsc 合约函数调用开销
Table 2 Call overhead of Smart-Subsc contract function

合约函数	调用者支付的交易手续费/gas
constructor	3 629 613
updatePrice	28 932
updateServiceFeeRate	28 568
activateSubscription	[28 192,56 447]
expireSubscription	[22 505,30 073]
purchaseSubscription	[72 651,132 651]
cancelSubscription	[25 632,36 327]
withdraw	[22 397,37 397]

对 Smart-Subsc 合约的部分函数,在不同情形下调用它们需要的交易手续费是不同的。造成这种差异的主要原因是以太坊的存储资源珍贵,把零值写成非零值将带来更多手续费开销,把非零值归零将

被退还部分手续费作为激励^[6]。分析造成不同情形下交易手续费差异的原因,参考 OpenZeppelin 库的实现,无论是 ERC20 代币^[23]还是 ERC721 代币^[7],它们记录每个以太坊地址对应的代币余额都会使用映射(mapping(address=>uint 256))数据类型。当代币合约被构造时,余额是初始零值。当某个地址被转入第一笔金额时,它的账户余额会从零值变成非零值,相比账户余额从非零值变成另一个非零值,前者会消耗更多的交易手续费。当某次交易后,某个地址的余额被清零时,这笔交易被退还手续费,导致实际的手续费开销更少^[6]。

以太坊虚拟机在更新映射数据类型的值时,会执行 sstore 指令。当 32 Byte 的以太坊虚拟机存储空间被从零值设置成非零值时,这步操作(sset)被收取的手续费是 20 000 gas;当相等长度的存储空间发生其他更新,即从非零值到零值或从非零值到另一个非零值时,这步操作(sreset)被收取的手续费是 5 000 gas。值得注意的是,当 32 Byte 的存储空间被清零时,这步操作(sclear)将被退还 15 000 gas,但是每次交易最多被退还一半手续费^[6]。

表 3 给出了不同情形下调用 purchaseSubscription 函数的交易手续费。可以看出:第一次铸造 NFT 会带来更多的手续费开销,因为 NFT 的 ID 是从 0 递增的;在购买过程中,超额支付的部分会被存入余额账户,向余额为 0 的余额账户存款会带来更多的手续费开销;如果直接使用余额抵扣购买,那么将余额账户清零所需要的交易手续费更少。另外,当 tokenId 不为 0 且订阅者未超额支付时,交易手续费与购买前余额是否为 0 无关。

表 3 不同情形下的 purchaseSubscription 函数调用开销
Table 3 PurchaseSubscription function call overhead under different situations

tokenId 为 0	余额抵扣	购买前余额为 0	超额存入余额	交易手续费/gas
是	不抵扣	是	是	132 651
			否	117 651
否	部分抵扣	无关	否	87 651
				72 651
	完全抵扣	无关	否	87 651
				102 651
				87 651

对其他涉及以太币和 NFT 转账的合约函数进行类似分析,总结影响交易手续费的主要因素:

- 1) 订阅者余额账户 deposit 是否从 0 开始被更新或归零,更新零值的手续费开销更高,归零将节省手续费。
- 2) 被操作的 tokenId 是否为 0,铸造首个 NFT 时,总发行量会由零值变成非零值,会产生更高的手续费开销。
- 3) 代币转账的发起者在转账后余额是否为 0,接收者在接收前余额是否为 0。向余额为 0 的账户转

入代币将消耗更高的手续费;代币余额归零将被退还手续费作为奖励。值得注意的是,销毁 NFT 的本质是向零地址转账^[7],是否被首个销毁也会影响手续费开销。

4) 订阅者以太币余额在支付后或退款前是否为 0,原理与因素 3 类似。

图 3 给出了订阅者大量购买订阅且其中一部分被提前取消的场景下,订阅者和销售商为每次购买订阅平均支付的手续费。假设订阅者购买了 1 000 次订阅并以不同退订率提前取消,对于每次购买订阅,订阅者平均支付的手续费是 87 681~113 345 gas,销售商平均支付的手续费是 0~71 520 gas。当购买退订率变高时,订阅者将为 cancelSubscription 函数调用支付额外的手续费,这导致平均手续费开销变高,销售商需要激活和使过期的订阅相对变少,合约调用手续费变低。结合退订率分析平均手续费开销有助于销售商更好地确定用户取款的服务费率。在理想状态下,订阅者从合约取款支付的服务费将被销售商用来支付合约调用产生的手续费。

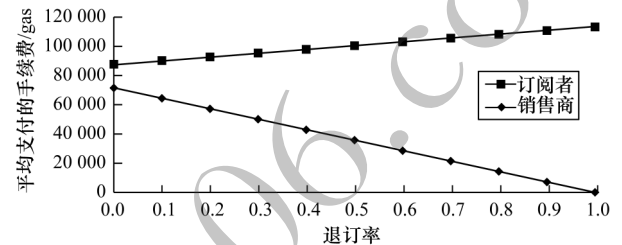


图 3 订阅者和销售商每次购买订阅平均支付的手续费与退订率之间的关系

Fig.3 Relationship between the average transaction fee paid by subscribers and sellers for each subscription purchase and the cancellation rate

3.2 针对高手续费问题的链下支付方案

本文实现与测试的 Smart-Subsc 合约需要支付以太币购买订阅。近期,以太坊主网 gas 价格约为 50 Gwei, ETH/USDT 汇率也处在历史高位,这会带来高昂的手续费开销。例如,订阅者购买订阅将至少支付 72 651 gas,折合约 75 元。若以 2019 年 11 月以太坊市场行情来计算^[30],订阅者购买订阅支付的手续费为 0.13~0.23 元。针对目前市场行情导致的高手续费问题,本文实现了在链下完成订阅购买并在链上存证的方案。相比 Smart-Subsc 链上实现和测试的方案,链下支付方案有如下改变:1) 订阅者不再使用以太币来购买软件订阅,即链下支付方案中订阅者不再向智能合约转账;2) 订阅者可以自行在链上生成订阅对应的 NFT,实际支付过程在链下完成,软件销售商需要确认订阅者是否实际购买了软件订阅并确认链上 NFT 的合法性;3) Smart-Subsc 将被部署在以太坊测试网或者私网上,因为目前只有以太坊主网的以太币具有公认的价值,所以在测试网上部署合约并发起交易消耗的以太币并不会带来实际的经济负担,同时以太坊测试网也是公开透明、

不可篡改的。

图 4 给出了 Smart-Subsc 链上存证和链下支付时序图。该方案不要求订阅者支付以太币来购买软件订阅,相应地,软件销售商需要确认订阅者铸造的 NFT 的合法性,即它们是否是软件订阅的有效存证。合法性包含两方面:订阅者是否已经成功地购买了订阅;订阅者铸造的 NFT 是否有效。软件销售商将有权直接销毁无效的 NFT。另外,在图 1 的基础上,图 4 简化了合约交互的表达。以太坊测试网与主网不存在功能性差异,Smart-Subsc 能够在以太坊测试网正常工作,不过这将增加软件销售商的管理负担,软件销售商必须逐一确认 NFT 作为链下交易存证的合法性。

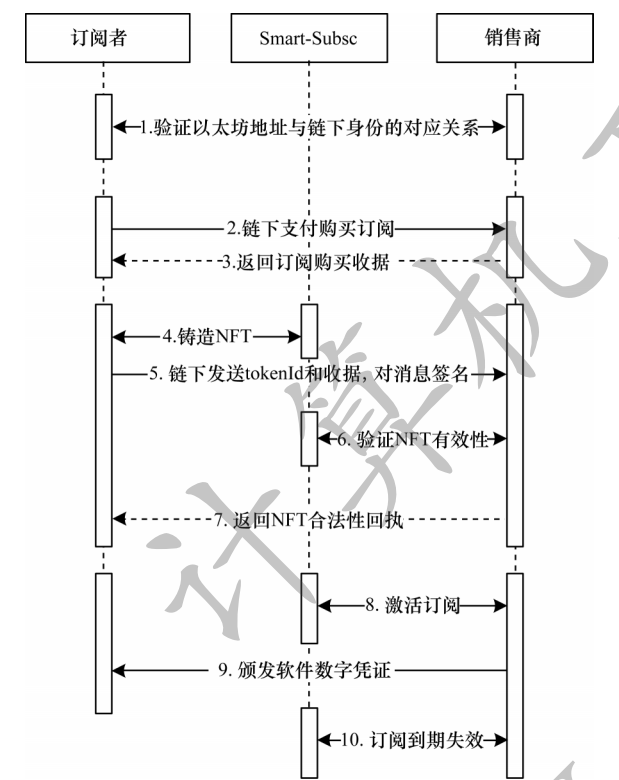


图 4 Smart-Subsc 链上存证和链下支付时序图

Fig.4 Sequence diagram of Smart-Subsc with on-chain depository and off-chain payment

3.3 Smart-Subsc 合约交互与测试

本文分别为订阅者和软件销售商实现了与合约交互的 Node.js 客户端,具有文本订阅者交互界面,使用者可以根据提示方便地完成链上操作。图 5 和图 6 以截屏形式分别给出了订阅者和软件销售商使用 Smart-Subsc 提供的客户端与合约交互的实例。Smart-Subsc 遵循 Mocha 测试库撰写了测试文件以测试在不同的使用场景下合约是否工作正常。图 7 是测试文件输出截屏,Smart-Subsc 对每个合约函数在不同场景下的工作状态进行验证。Smart-Subsc 将 NFT 的关键特性,即铸造、转账、销毁和操作权限管理完整地应用在软件订阅的特定场景中,提供了对应的文本交互界面并通过场景测试。



图 5 订阅者与 Smart-Subsc 的合约交互与购买订阅截图

Fig.5 Screenshot of contract interaction and subscription purchase of subscriber and Smart-Subsc



图 6 销售商与 Smart-Subsc 的合约交互与确认并激活订阅截图

Fig.6 Screenshot of contract interaction to confirm and activate the subscription of sellers and Smart-Subsc

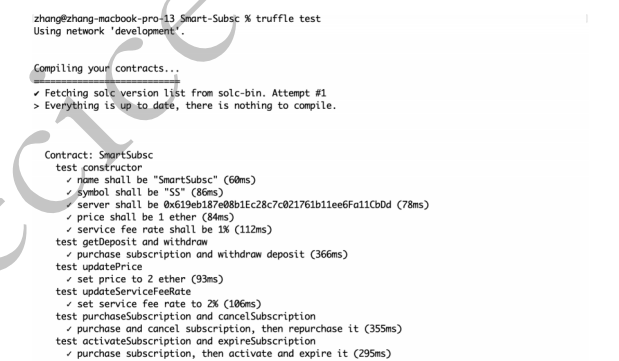


图 7 Smart-Subsc 通过多种使用场景测试的截图

Fig.7 Screenshot of Smart-Subsc has passed a variety of usage scenarios test

4 结束语

本文基于 NFT 的关键特性设计与实现软件订阅模型,计算与讨论订阅者和销售商的合约调用开销。针对当前行情下交易手续费相对昂贵的问题,提出链上存证和链下支付的解决方案,并对现有 DeFi 和 NFT 研究和应用进行相关补充。实验结果表明,该模型解决了中心化软件授权场景中订阅信息被篡改造成的盗版问题,实现了 NFT 与软件订阅和支持模式的有效结合。NFT 非常适合表示真实世界资产,后续可对其在更多资产管理场景中的应用进行研究。区块链预言机作为服务于区块链的可信信息提

供者,也将成为未来可编程社会发展中的热门研究方向。

参考文献

- [1] 高洪福. SaaS的春天来了吗?——“IT生存法则”之软件订阅和支持服务[J]. 网络安全和信息化, 2020(8): 22-23.
GAO H F. Is the spring of SaaS coming? Software subscription and support services of "IT Survival Law"[J]. Cybersecurity & Informatization, 2020(8): 22-23. (in Chinese)
- [2] 张江徽, 崔波, 李茹, 等. 基于智能合约的物联网访问控制系统[J]. 计算机工程, 2021, 47(4): 21-31.
ZHANG J H, CUI B, LI R, et al. Access control system of Internet of Things based on smart contract[J]. Computer Engineering, 2021, 47(4): 21-31. (in Chinese)
- [3] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[EB/OL]. [2021-07-21]. <https://bitcoin.org/bitcoin.pdf>.
- [4] BUTERIN V. A next-generation smart contract and decentralized application platform[EB/OL]. [2021-07-21]. <https://github.com/ethereum/wiki/wiki/White-Paper>.
- [5] SZABO N. Smart contracts[EB/OL]. [2021-07-21]. <https://bit.ly/2rLG2Nr>.
- [6] WOOD G. Ethereum: a secure decentralised generalised transaction ledger[EB/OL]. [2021-07-21]. <https://ethereum.github.io/yellowpaper/paper.pdf>.
- [7] ENTRIEN W, SHIRLEY D, EVANS J, et al. ERC-721 non-fungible token standard[EB/OL]. [2021-07-21]. <https://eips.ethereum.org/EIPS/eip-721>.
- [8] 蔡中魁. 基于服务设计的办公服务订阅系统的设计与研究[D]. 广州: 华南理工大学, 2016.
CAI Z K. The design and research of office service ordering system based on service design[D]. Guangzhou: South China University of Technology, 2016. (in Chinese)
- [9] ZHANG J, SEIDMANN A. Perpetual licensing vs. subscription of software: a theoretical evaluation[C]//Proceedings of the 42nd Hawaii International Conference on System Sciences. Washington D. C., USA: IEEE Press, 2009: 1-10.
- [10] WOOD T, CECCHETT E, RAMAKRISHNAN K K. Disaster recovery as a cloud service: economic benefits and deployment challenges[C]//Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing. Boston, USA: USENIX, 2010: 8-15.
- [11] KUMAR R, TRIPATHI R. Traceability of counterfeit medicine supply chain through blockchain[C]//Proceedings of the 11th International Conference on Communication Systems & Networks. Washington D. C., USA: IEEE Press, 2019: 568-570.
- [12] 张亮, 刘百祥, 张如意, 等. 区块链技术综述[J]. 计算机工程, 2019, 45(5): 1-12.
ZHANG L, LIU B X, ZHANG R Y, et al. Overview of blockchain technology[J]. Computer Engineering, 2019, 45(5): 1-12. (in Chinese)
- [13] WATANABE H, FUJIMURA S, NAKADAIRA A, et al. Blockchain contract: a complete consensus using blockchain[C]//Proceedings of the 4th Global Conference on Consumer Electronics. Washington D. C., USA: IEEE Press, 2015: 577-578.
- [14] ZHENG Z, XIE S, DAI H, et al. An overview of blockchain technology: architecture, consensus, and future trends[C]//Proceedings of 2017 IEEE International Congress on Big Data. Washington D. C., USA: IEEE Press, 2017: 557-564.
- [15] 谭敏生, 杨杰, 丁琳, 等. 区块链共识机制综述[J]. 计算机工程, 2020, 46(12): 1-11.
TAN M S, YANG J, DING L, et al. Review of consensus mechanism of blockchain[J]. Computer Engineering, 2020, 46(12): 1-11. (in Chinese)
- [16] FENG Q, HE D B, ZEADALLY S, et al. A survey on privacy protection in blockchain system[J]. Journal of Network and Computer Applications, 2019, 126: 45-58.
- [17] TIKHOMIROV S. Ethereum: state of knowledge and research perspectives[M]//IMINE A, FERNANDEZ J M, MARION J Y. Foundations and practice of security. Berlin, Germany: Springer, 2018: 206-221.
- [18] BECK R, CZEPLUCH J S, LOLLIKE N, et al. Blockchain—the gateway to trust-free cryptographic transactions[C]//Proceedings of the 24th European Conference on Information Systems. Berlin, Germany: Springer, 2016: 1-14.
- [19] REGNER F, SCHWEIZER A, URBACH N. NFTs in practice—non-fungible tokens as core component of a blockchain-based event ticketing application[C]//Proceedings of the 40th International Conference on Information Systems. Munich, Germany: Association for Information Systems, 2019: 1-8.
- [20] WERNER S M, PEREZ D, GUDGEON L, et al. SoK: Decentralized Finance (DeFi)[EB/OL]. [2021-07-11]. <https://econpapers.repec.org/scripts/a/abstract.pf?p=y;h=RePEc;arx=papers;2101.08778>.
- [21] MAMMAZADA K, IQBAL M, MILANI F, et al. Blockchain Oracles: a framework for blockchain-based applications[C]//Proceedings of the 18th International Conference on Business Process Management: Blockchain and Robotic Process Automation Forum. Berlin, Germany: Springer, 2020: 19-34.
- [22] MUEHLBERGER R, BACHHOFNER S, FERRER E C, et al. Foundational Oracle patterns: connecting blockchain to the off-chain world[C]//Proceedings of the 18th International Conference on Business Process Management. Berlin, Germany: Springer, 2020: 35-51.
- [23] VOGELSTELLER F, BUTERIN V. EIP-20: ERC-20 token standard[EB/OL]. [2021-07-21]. <https://eips.ethereum.org/EIPS/eip-20>.
- [24] WANG Y, KOGAN A. Designing privacy-preserving blockchain based accounting information systems[EB/OL]. [2021-07-21]. <https://ssrn.com/abstract=2978281>.
- [25] GRIFFIN J. Software licenses as non-fungible tokens[EB/OL]. [2021-07-21]. <https://medium.com/atchai/software-licences-as-non-fungible-tokens-1f0635913e41>.
- [26] RIMBA P, TRAN A B, WEBER I, et al. Quantifying the cost of distrust: comparing blockchain and cloud services for business process execution[J]. Information Systems Frontiers, 2020, 22(2): 489-507.
- [27] BENIICHE A. A study of Blockchain Oracle[EB/OL]. [2021-07-21]. <https://arxiv.org/abs/2004.07140>.
- [28] Truffle Suite. Sweet tools for smart contracts[EB/OL]. [2021-07-21]. <https://www.trufflesuite.com/>.
- [29] OpenZeppelin. Build secure smart contracts in solidity[EB/OL]. [2021-07-21]. <https://openzeppelin.com/contracts/>.
- [30] 张朝栋, 王宝生, 邓文平. 基于侧链技术的供应链溯源系统设计[J]. 计算机工程, 2019, 45(11): 1-8.
ZHANG C D, WANG B S, DENG W P. Design of supply chain traceability system based on side chain technology[J]. Computer Engineering, 2019, 45(11): 1-8. (in Chinese)