一种基于区块链的可信数据共享解决方案

汪 菲[1],沈苏彬[2]
(1.南京邮电大学 计算机学院,江苏 南京 210046;
2.南京邮电大学 通信与网络技术国家工程研究中心,江苏 南京 210046)

摘 要:在数据共享不可阻挡的大趋势下,现阶段的主要共享方式是基于云的数据共享,它是将数据全部存储到云中再进行数据交易从而产生的数据共享,但这种数据共享方式会降低共享数据的安全性和隐私性,如数据被篡改、假冒等。区块链的去中心化以及区块上数据不可篡改的特性,可安全地记录双方之间的交易而无需可信任的第三方机构。文中将区块链技术应用到数据共享技术中,设计了一种去中心化、可信数据共享的智能合约,构造了共享数据的存储块结构,提出将共享数据的信息可信地存储到区块链上实现数据共享的解决方案,最终在以太坊平台上模拟真实场景。实验表明,基于区块链的数据共享技术实现了去中心化的目标,在一定程度上增强了共享数据的可信性、安全性、隐私性。

关键词:区块链;去中心化;数据共享;可信性;智能合约
中图分类号:TP309.2 **文献标识码**:A **文章编号**:1673-629X(2020)09-0115-07
doi:10.3969/j.issn.1673-629X.2020.09.021

A Solution for Decentralized Data Sharing Based on Blockchain

WANG Fei[1],SHEN Su-bin[2]
(1. School of Computer Science,Nanjing University of Posts and Telecommunications,
Nanjing 210046,China;
2. National Engineering Research Center on Communication and Networking,Nanjing
University of Posts and Telecommunications,Nanjing 210046,China)

Abstract:In the unstoppable trend of data sharing,the main sharing method at the present stage is cloud-based data sharing,which is the sharing of data generated by storing all the data in the cloud for data transaction. However,this way of data sharing maybe reduce the security and privacy of shared data,such as data tampering,counterfeiting and so on. The decentralization of the blockchain and the non-tampering of data on the block allow transactions between two parties to be recorded securely without the need for a trusted third party. Applying blockchain approach in data sharing,we design a decentralized and trusted data sharing smart contract,build a shared data storage block structure,and propose a solution to realize data sharing by storing the information of shared data credibly on the blockchain. This solution is simulated and tested on the Ethereum platform. It is showed that the data sharing approach based on blockchain achieves the goal of decentralization,which enhances the credibility,security and privacy of shared data to a certain extent.

Key words:blockchain;decentralization;data sharing;credibility;smart contract

0 引 言

共享数据的采集、传递、存储过程中,如若没有严密的防范措施,就可能存在数据假冒、被伪造的可能性。采集、传递数据过程中,从手工记录到使用无线电遥测等高效方式[1],再到数字跟踪设备[2],误差率的骤降显著地降低了伪造数据的可能性。如今,强大的计算和存储能力[3]使得用户更愿意依托云来存储或共享数据,借助会议密钥协商协议加密数据所有者的共享数据[4],使云中多个参与者能够自由成组共享数据,但大量信息交互和高度集中的计算资源使云面临着严峻的安全挑战[5],例如,云系统可能会受到恶意用户或云提供商的攻击。

如何在数据共享的同时还能保证数据存储更安全、数据信息更保密就成为新的核心问题。研究发现,数据越来越多地发生在与所有者相关的网络服务器站点上,即数据所有者和访问者之间。集中查询数据[6]

收稿日期:2019-10-31 **修回日期**:2020-03-04
基金项目:江苏省未来网络前瞻性研究项目(BY20130951108)
作者简介:汪 菲(1993-),女,硕士研究生,研究方向为区块链;沈苏彬,博导,教授,CCF高级会员(E200005482S),研究方向为物联网及其应用、未来网络及其应用。

是不经济的,不仅要解决数据庞大、分布式管理等问题,还要解决数据的安全性和隐私性[7-8]等问题,而在数据共享中,双方成为了数据共享的最小范围。2008年,中本聪发表了一篇名为《比特币:一种点对点的电子现金系统》的技术白皮书[9],具体阐述了交易、时间戳服务器、工作量证明、网络、加密、区块链技术等去中心化电子货币框架的重要理念,并上线了比特币的系统。区块链技术源于此系统,是比特币的底层数据存储技术,也是近年来最具有革命性的技术之一。区块链技术被称为实现比特币交易的公共账本,其主要特点就是去中心化[10],由网络节点共同维护这一公共账本,且区块上的数据也具有不可篡改、撤销的特性,由此区块链技术成为共享数据达到去中心化、增强共享数据安全性[11]和隐私性的关注点。

文中提出利用区块链的自身特性实现数据共享的去中心化,将区块链技术应用到数据共享技术中。首先要解决的问题是如何设计基于区块链的共享数据的存储结构,其次是如何使区块形成链,最后是基于区块链如何实现数据共享,提出去中心化的数据共享机制。最终形成将共享数据的信息存储到区块链上实现数据共享的解决方案,并在以太坊平台上模拟真实场景。

主要贡献如下:

(1)提出了面向区块链的共享数据的块结构。

(2)提出了去中心化的数据共享机制,利用区块链的自身特性实现数据共享的去中心化。

(3)在以太坊平台上模拟真实的场景,设计了去中心化、可信数据共享的智能合约并具体实现,基于区块链的数据共享模型增强了数据的隐私性和安全性。

1 基于区块链的数据共享结构设计

目前所提及的是区块链技术不再是单纯的底层数据存储技术,它还包含了对等网络技术、加密和数字签名技术、共识协议和实现机制、数据完整性存储等多个方面综合的技术,区块链技术可以实现可信数据公开、可信数据回溯、可信数据管理的设计方法。由于基于云的数据共享技术涉及第三方的参与,可能会降低共享数据的安全性和隐私性,文中提出了一种基于区块链的去中心化数据共享解决方案,去除第三方的参与,设计了一种去中心化、可信数据共享的智能合约,合约中包括共享数据的存储块结构,共享数据的区块链结构。

1.1 共享数据存储块结构

一般来说,区块链是由多个节点共同参与、带有时间戳的块链。它是由一串按照密码学方法产生的数据块和数据包组成,即区块。对每个区块数据信息都自动加盖时间戳,从而计算出一个数据加密数值,即哈希值。每一区块都包含块头(Header)和块体(Body)两个部分,块头包含前一个区块的哈希值、当前区块的时间戳、一次性数、难度系数、Merkle 根哈希值;块体是采用二叉树的形式存储交易数据,值得注意的是:这里并不是存储所有的交易数据,而是存储每笔交易通过加密算法得出的一个哈希序列,将交易的哈希序列作为叶子节点按二叉树形式加密存储得到一个根值,而这个值就是 Merkle 根哈希值。时间戳是断定相邻区块之间继承关系的时间标签,是用来保证区块的有序性,并且有效地防止了对区块数据进行篡改和欺骗行为。一次性数(Nounce)是一个随机哈希序列的值,用来完成工作量证明。难度系数代表了验证区块的工作量,可以自行动态调整的同时保证在预期时间内被计算出的时间控制。

图 1 是详细的区块结构。

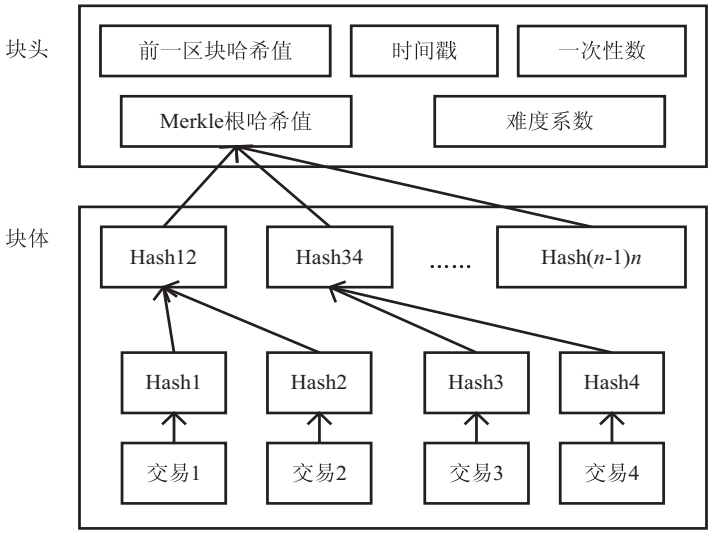

图 1 比特币中区块的结构

使用密码哈希函数 Keccak256[12]实现数字签名来进行节点的真实性身份验证。数字签名是通过数据名称、IP 地址、端口号的数据信息与私钥共同生成的。数字签名需要一对密钥(公钥和私钥),而密钥来自于椭圆曲线算法[13]。数字签名用于签署共享的报文数据。交易[14]是指数据上传与下载、变更网络状态的过程。在去中心化的数据共享解决方案中,共享数据的存储被视为交易。Merkle 树[15]是用来验证区块中独一无二的交易,如果交易数据的任何内容有一点点的变化,哈希序列都会产生很大的变化。用户可通过 Merkle 根哈希值追溯到数据共享区块链中的任意一项共享记录以及当前状态,实现共享数据的追溯。可信的时间戳用于记录当前区块数据的写入时间,证明了某些数据在某个时间存在某个状态,并且之后没有被更改。通过使用区块链技术保证和验证数据的完整性,增强共享数据的不可篡改性、隐私性。

通过使用数字签名,可以抵御恶意节点的 IP 欺骗和 Sybil 攻击。由于每个节点能够通过数字签名的加密、解密过程来证明其他节点的身份,因此有效防止了将数据名称与恶意修改的 IP 地址组合,从而对坏节点进行 IP 欺骗。同样,使用数字签名能够防御 Sybil 攻击。因为签名是由唯一私钥创建的,当作为许多伪造节点发送签名时,诚实节点通过使用伪造节点的公钥解密签名而得到结果是错误的,因此有效防止了单一节点具有多个身份标识。

支持基于区块链进行共享数据发布和共享数据追溯的两个重要部分就是时间戳和哈希树根值。在区块结构中,获得新区块开采权的矿工在打包数据块时,会在区块头中加盖时间戳,用于记录当前区块数据的写入时间。时间戳可以作为存在性证明的重要参数,它能够证实特定数据必然在某特定时刻是的确存在的,区块链在很大程度上保证了存储的共享数据不变性,并且随着时间的推移这个保证越来越强,因而适用于时间戳应用程序。而区块体中的哈希树将会对每一笔交易进行数字签名并加盖时间戳,最终由哈希算法得出一个哈希树根植存储于区块头中,用户可以通过该值追溯到数据共享区块链中的任意一项共享记录与当前状态,实现共享数据的追溯。

用户的原始数字数据是巨大的,可以是千兆字节甚至太字节,一般使用 URIs 或 DOIs 进行标识[16]。若原始数字数据直接发布,不仅不能保证其所有权,且对网络来说也是一个巨大的负担。元数据[17]是一种新的数据类型,定义元数据的格式可完全描述原始数据,且显著地减轻网络负担。一般情况下,元数据的功能是作为描述性信息存在的。现许多系统支持各种形式的元数据管理和数字资源保存,但前提是数据库所有者是可信任的。在不受信任的环境下,区块链技术依靠加密技术和共识协议提供持久的、不可篡改证明的公共账本,很好地克服了这个问题。因为数据标识具有唯一性,数字资源可通过唯一识别内容的机制进行识别,且很难推断出与其相关的任何内容。

文中提出的实现方法是在智能合约中存储一些与源数据相关的元数据以及数据所有者可共享的相关必要数据信息。区块中的完整的数据不是数据本身,而是数据所有者提供给访问者的一个 URL,或者是一个 ftp 入口,数据本身是存储在数据所有者的本地服务器上,共享数据的链接入口信息直接存储在区块上。该方法通过在智能合约中增加一个 Data 结构实现,这个结构包含一些附加信息也就是元数据,有共享数据的关键词(bytes name)、完整共享数据的链接入口(bytes hash)、数字签名(bytes dsign)、时间戳(uint256 timestamp)、数据所有者的地址(address owner)。具体定义如下(使用 solidity 语言):

```
struct Data{
bytes name;
bytes hash;
bytes dsign;
uint256 timestamp;
address owner;
}
```

1.2　共享数据的区块链结构

基于区块链的共享数据存储实际上就是共享数据形成区块后的上链过程。共享数据存储交易通过数字签名的验证后加入到当前区块中,当前区块经过工作量证明后,添加到区块链的尾部。以太坊采用工作量证明机制来实现共识协议,即所谓的“挖矿”,负责新区块的生成,因为区块链副本存在于大量的节点中。共享数据区块链的结构如图 2 所示。

每个共享数据的节点通过计算一次性数(Nounce)来争夺记账权,求得正确的数值以生成区块的能力是节点计算力的具体表现。共享数据存储在区块上,所有的节点都要同意其内容,并且这个内容是不可更改、持久的[18],工作量证明机制的本质就是一节点一票,“大多数”的决定表示为最长的链,因为最长的链包含了最大的工作量。如果大多数的节点被诚实节点控制,那么诚实节点将以最快的速度延长,并超越其他竞争链条;如果想要篡改已出现的区块,攻击者必须重新完成该区块的工作量,并完成该区块之后的所有区块的工作量,最终还要赶上并超越诚实节点的工作量,而这个过程代价巨大。区块中的难度系数代表了验证区块的工作量,可以自行动态调整的同时保证在预期时间内被计算出的时间控制。区块链的安全性通过工作量证明得以保证。主观上,挖矿节点获取了

奖励;客观上,区块链基于此规则得到了持续稳定的共同维护。

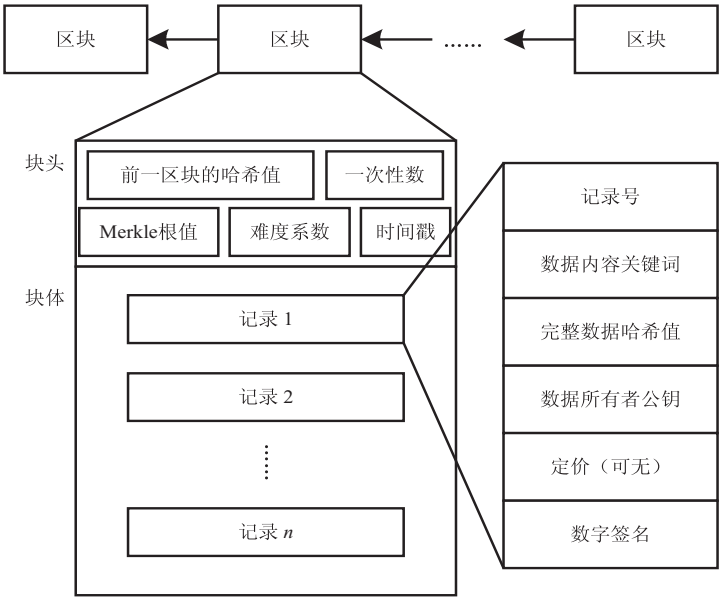

图 2 基于区块链的数据共享的区块链结构

当共享数据需要更新时,数据所有者应重新提交数据存储交易,链里的记录内容是无法更改和删除的,只能新增记录。比特币系统对区块的创建者提供了比特币的激励,对验证交易的一方也提供了比特币的激励。区块中的第一笔交易会产生一个新的比特币奖励给区块的创建者。这会激励节点验证交易,然后将比特币投入流通领域,因为没有中央机构发布比特币。这个第一笔交易称为创币(coinbase)交易。使用这种方法,可以激励节点保持诚实。比特币网络大约 10 分钟产生一个区块。除了基于区块创建产生的奖励,节点通过验证交易也会被奖励比特币。智能合约的执行是需要消耗一定的燃料(gas),共享数据形成的区块成功上链会得到一定的 gas,可理解为执行代码消耗的计算工作所给出相对应的奖励,即存在相应的激励机制,实际上 gas 就是交易的费用。如若当前区块执行异常,区块链状态将恢复为初始状态,就像一切没有发生一样。

2 基于区块链的数据共享机制设计

最初的第一代区块链应用案例——比特币的成功,到后来第二代的区块链发展重点在于注册、验证、合约或财产,智能合约[14,19]的发展使得区块链技术进一步成长为一个可编程的平台,它把业务规则转化成在区块链平台自动执行的代码。当一个预先编好的条件被触发时,智能合约自动执行相应的合约条款,这个过程不需要人的干涉,也无法进行干涉,除非改变智能合约的内容。智能合约对外提供服务接口来处理数据,智能合约编译后会产生 ABI,通过与智能合约交互可以实现共享数据的存储和访问请求与服务。

基于区块链的数据共享机制中的智能合约记录两大主要信息:共享数据存储 Data 类以及用于授权认证的 AuthoList 类,该两类在以太坊虚拟机中实现。Data 类中,记录区块的具体信息,不同共享数据是以数组的方式存储,记为 Datas。用于共享数据存储的 store 函数允许通过数据标识包含的元数据传递给此函数并存储到区块链中,其中元数据包含共享数据关键词,完整数据的链接入口,数据所有者的公钥地址、数字签名,数据的定价(可不定价)。节点通过使用 keccak256 算法联合时间戳对共享数据关键词,完整数据的链接入口,数据的定价这三者进行加密计算从而得到基础共享数据包,结果记为 message。通过密码哈希函数对完整数据的链接入口进行哈希运算得到哈希值,并对这个哈希值进行数字签名,得到数据所有者的数字签名 dsign。利用去中心化时间戳,将共享数据的关键词 name、完整共享数据的链接入口 hash、数字签名 dsign 以及交易发生的时间戳 timestamp 和数据所有者 owner 都作为附加信息添加到 Data 结构中。最后将这个结构代替数据哈希值映射到智能合约中。同样的,使用 mapping 类型的 AuthoList 以 256 位字节大小的键值存储来进行授权列表的存储,它在合约的执行期间可以使用值类型或者引用类型进行读取或者写入,键是数据请求方的公钥地址,值是数据请求方是否已经被授权的布尔值。在无信任的环境下,数据请求者首次请求区块链的共享数据需要通过 AuthoList 的验证。

数据所有者 A 与数据请求者 B 共享数据时,首先要与 B 达成共识以制定约束(数据范围等),即 A 已将共享的数据存储在区块链中。在基于区块链的数据共

享系统中,共享数据区块成功上链后,系统中的数据库会提取出区块中 A 的公钥地址和主题信息关键词,B 可以通过查询此数据找到想要的数据信息。设定访问者 B 已经找到想要的共享数据信息,得到关键词和数据所有者的公钥地址。用于共享数据的访问请求和服务的 query 函数允许 B 通过 A 的公钥地址传递给此函数发送共享请求到区块链中,区块链验证 B 是否已被授权,即在 AuthoList 列表中查询其布尔值是否为真。若找不到该数据或者布尔值为假,需要通过区块链的身份验证,在 AuthoList 中给予授权。若布尔值为真,输出区块中存储的该数据所有者的共享数据信息,其中完整数据的链接入口是用 B 的公钥通过不对称加密算法进行加密后而输出的。当 B 得到区块的相关信息时,用自己的私钥对加密的完整数据链接入口进行解密。至此,A 和 B 成功共享数据。

目前比特币采用的公钥加密算法是椭圆曲线加密法(ECC),共享数据的加密方式也采用了该方法。ECC 是建立在基于椭圆曲线的离散对数问题上的密码体制,它选定了一条椭圆曲线 $Ep(a,b)$,取椭圆曲线上的一点作为基点 G,由一个私有密钥 k 生成公开密钥 $K=kG$。A 将共享数据链接入口的明文编码到 $Ep(a,b)$ 上一点 M,并产生一个随机整数 r。计算点 $C_1=M+rK$;$C_2=rG$,再将 C_1、C_2 传给用户 B。用户 B 接到信息后,计算 C_1-kC_2,结果就是点 M。因为 $C_1-kC_2=M+rK-k(rG)$,再对点 M 进行解码就可以得到明文。在这个加密通信中,如果有一个第三方 H,他只能看到 $Ep(a,b)$、K、G、C_1、C_2,而通过 K、G 求 k 或通过 C_2、G 求 r 都是相对困难的,因此,H 无法得到 A、B 间传送的明文信息。相比较于 RSA 和 DSA 算法,ECC 安全性更高,速度更快,占用存储空间更小。

区块链技术优势在于强容灾能力和防篡改机制,即数据不会丢失、信息不会被单方面修改。共享数据产生的交易是记录在区块链的侧链上,共享数据存储和共享数据交易的结构不同,因此在上传时不会存在上链错误的情况。最终表现为:对于数据本身来说,数据由数据所有者存储和管理,即谁的数据谁控制;交易过程中,数据仅能被访问者获得,其他任何第三方无法获取数据。对于交易双方来说,访问的数据要保证其真实性,即明确知道共享数据的所有权是由谁提供的;交易要保密,信息查询或购买行为本身就是隐私,不能除双方外的任何人或机构知晓。

3　仿真实现与测试

设计的去中心化、可信数据共享的智能合约包含了共享数据存储阶段,共享数据访问请求和服务阶段,最终完成双方之间的数据共享。智能合约应为双方各自与区块链进行交互提供调用接口以完成各项功能。一旦智能合约被部署,其预定义的代码就作为一个合约账户存在于网络中,智能合约账户开放其预定义的接口函数供其他账户调用,在各节点验证交易签名后,根据交易参数执行合约,对参与方和交易参数进行验证更改区块链账本。

3.1　智能合约的运行

文中选择在以太坊平台上进行仿真和测试,选择由 solidity 编程语言 truffle 框架下的 webpack 模版来完成可信数据共享的实验和可视化私有链客户端 ganache。在写自己的智能合约时要删除 contracts 目录下原有的 ConvertLib. sol 和 MetaCoin. sol 文件,编写自己的 DataSharing. sol 文件。

首先要启动一个以太坊节点(如 Geth),使用智能合约编程语言(如 Solidity)编写智能合约(后缀为. sol)。通过编译智能合约获得应用二进制接口 ABI 和字节码 bytecode,ABI 是账户和智能合约间交互的接口,bytecode 是 EVM 可以执行的字节码。账户使用 bytecode 作为交易的参数,广播创建合约交易。智能合约的执行同样以交易的形式存在。账户通过 ABI 接口获取合约,之后节点通过 EVM 虚拟机运行 bytecode 执行智能合约的代码逻辑,将运行的结果打包成交易存储于区块链中。将编译好的合约代码部署到以太坊区块链中,这个过程需要消耗燃料,并且需要合约发起者使用自己的私钥对合约进行签名,通过工作量证明验证后,将合约代码存于以太坊区块链上。图 3 表示去中心化、可信数据共享的智能合约部署成功,并且消费了一定的 ETH。

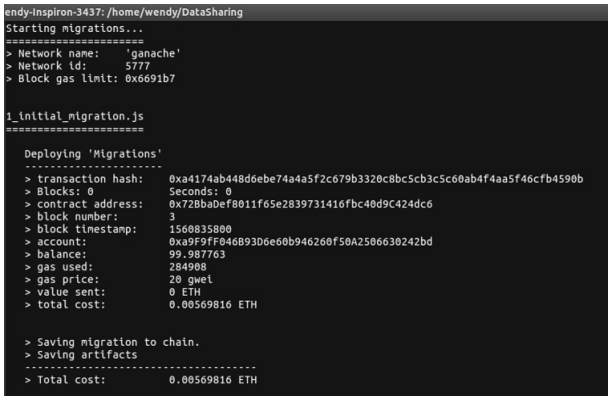

图 3　智能合约部署结果

3.2　共享数据的存储阶段

实验使用了一种将两种存储数据方法结合的方式。一种存储数据方法是利用账户存储数据,另一种存储数据方法是利用事件存储数据,也称为日志存储。这种结合提高实验的存储效率及精确度。以太坊区块链上的智能合约允许数据作为变量直接存储在合约中。区块链中数据的不易篡改特性可保证数据的更改

方式只能使用重新部署智能合约的方式进行更改,并且每次和智能合约的交互都会被看作一个交易记录在区块链上。区块链技术可安全地在双方之间记录交易,无需可信任的第三方机构,同时提供了一种使用可变的公钥作为身份来保护隐私,而区块链安全问题是通过工作量证明挖掘新的区块来解决加密难题从而得到保证。

具体实现的智能合约包含一个 store 存储函数,该函数允许通过数据标识包含的元数据传递给此函数并存储到区块链中,其中 name、hash、dsign、timestamp、owner 分别代表共享数据的主题信息关键词、完整数据的链接入口、数字签名、时间戳和数据所有者。运行结果如图 4 所示。

```
contract DataSharing {
function store(bytes32 name,bytes hash,uint256 timestamp,
bytes dsign) public returns (bytes) {
        if(msg.sender! =owner)
throw;
        address owner=msg.sender;
        bytes32 message=prefixed(keccak256(name, hash, tim-
estamp));
        address recovered=recoverSigner(message, dsign);
        require(recovered==msg.sender);
        Datas.push(Data({
    name:name,
    hash:hash,
        timestamp:timestamp,
        owner:owner,
        dsign:dsign
    }));
        Dsign(name,hash,timestamp,owner, dsign);
        return dsign;
    }
}
```

```
root@wendy-Inspiron-3437:/home/wendy/DataSharing/app# node store.js
0x3625f20d379e634facff0b31a6c0e1d4be3f4f882b49ada5403e80f159b1cc14
Result {
  '0': '0xaf28969b756812a3f5c3f34dd970efc0eb839374e72938c7c68de439a01f5f1b',
  '1': '0x5e603f4f54abc3c2a9967d89b8c8658a8e8cb8c694e1f16d674eb69aa1529da8',
  '2': '1561084060214',
  '3': '0x3325763862cf80FacE488569c14a243f182B09B1',
  '4': '0x6b2b4177ca9c3540e01ec436f8b52f278802da2f5a2009f3142ff9e5b9736c4334de340aefcb621d744c4b522
de2250ac71944bd376d1e3208d5eca5141196f21b',
  name: '0xaf28969b756812a3f5c3f34dd970efc0eb839374e72938c7c68de439a01f5f1b',
  hash: '0x5e603f4f54abc3c2a9967d89b8c8658a8e8cb8c694e1f16d674eb69aa1529da8',
  timestamp: '1561084060214',
  owner: '0x3325763862cf80FacE488569c14a243f182B09B1',
  dsign: '0x6b2b4177ca9c3540e01ec436f8b52f278802da2f5a2009f3142ff9e5b9736c4334de340aefcb621d744c4b5
22de2250ac71944bd376d1e3208d5eca5141196f21b' }
```

图 4 数据存储阶段的实验结果

3.3 共享数据访问请求和服务阶段

智能合约包含一个共享数据访问请求和服务阶段,该阶段具体分为两部分:一部分是检索,找到目标数据所在的区块,本论文的前提是根据共享数据的主题信息关键词已找到所需共享信息的数据所有者的公钥地址,在基于区块链的数据共享平台中,提供这样的数据检索模块;另一部分是对区块链发送数据访问请求和服务。在访问请求中数据所有者首先要验证访问者的身份,即在 AuthoList 中查看是否被授权,如若已被授权,则返回区块中存储的相应共享数据信息,其中完整数据的链接入口是用访问者的公钥通过不对称加密算法进行加密后而输出的。访问者得到区块的相关信息时,用自己的私钥对加密的完整数据链接入口进行解密。否则,要先对其进行授权,通过全网节点的验证认可而进行授权。具体实现的智能合约包含一个 query 请求函数,运行结果如图 5 所示。

```
function query (address owner) public returns (bytes32,
bytes32,uint256,address,bytes){
    bool autho=AuthoList[msg.sender];
      if(autho==true){
      uint len=Datas.length;
    for(uint i=0;i<len;i++) {
      bytes memory sig=Datas[i].dsign;
      if(isEqual(sig,dsign)){
    return (Datas[i].name,recover(Datas[i].hash),Datas[i].
timestamp, Datas[i].owner, Datas[i].dsign);
    }
            }
        }else{
        addAuthorize(msg.sender);
    }
    }
```

由以上实验可以看出,共享数据的链接入口不再是原来的值,且其字节长度和公钥地址长度是一致的,这是因为共享数据的链接入口数据用访问者的公钥进行加密而形成。在授权情况下,请求方的数据请求报文能够得到区块中已经存入的共享数据信息,也证实是能够将区块链技术应用到数据共享中,从而达到去中心化的目的。

```
root@wendy-Inspiron-3437:/home/wendy/DataSharing/app# node query.js
null Result {
  '0': '0xaf28969b756812a3f5c3f34dd970efc0eb839374e72938c7c68de439a01f5f1b',
  '1': '0x7c51e230f4f5f3abc03d89e1c260a4e4c3c2a692e1f16d674eb69ac2a1529da5c2739f
41de2c5afcd69967a1fc8933eb7102db2528a13c23760ff410e12205fc1a',
  '2': '1561084060214',
  '3': '0x3325763862cf80FacE488569c14a243f182B09B1',
  '4': '0x6b2b4177ca9c3540e01ec436f8b52f278802da2f5a2009f3142ff9e5b9736c4334de34
0aefcb621d744c4b522de2250ac71944bd376d1e3208d5eca5141196f21b'}
```

图 5 请求阶段的实验结果

4 结束语

区块链的去中心化以及区块上数据不可篡改的特性,可保证安全地记录双方之间交易而无需可信的第三方机构,这改变了基于云的数据共享模式。提出的基于区块链的共享数据存储区块设计、共享数据的区块链结构,以及基于区块链的去中心化数据共享机制,通过设计一种去中心化、可信的智能合约在以太坊平台上成功模拟真实场景。实验表明,基于区块链的数据共享实现了去中心化的目标,增强了数据的可信性

与安全性,同时保证了数据的隐私性。但是针对信息检索部分,当前基本没有任何区块链项目支持用户数据的自定义索引,但是从本质上看没有任何根据表明当前的区块链项目结构无法在其上构建通用索引能力(包括 B 树索引、位图索引、全文检索等)。希望未来的工作可以对此有所优化。这一点在未来的通用型区块链项目中一定会被弥补的。

参考文献:

[1] GATTESCH V, LAMBERTI F, DEMARTINI C, et al. To Blockchain or not to Blockchain: that is the question[J]. IT Professional, 2018, 20(2): 62-74.

[2] FREY R M, HARDJONO T, SMITH C, et al. Secure sharing of geospatial wildlife data[C]//Proceedings of the fourth international ACM workshop on managing and mining enriched geo-spatial data. Chicago, Illinois: ACM, 2017: 5.

[3] RAMAN R K, VARSHNEY L R. Dynamic distributed storage for Blockchains[C]//2018 IEEE international symposium on information theory (ISIT). Vai: IEEE, 2018: 2619-2623.

[4] SHEN Jian, ZHOU Tianqi, HE Debiao, et al. Block design-based key agreement for group data sharing in cloud computing[J]. IEEE Transactions on Dependable and Secure Computing, 2019, 16(6): 996-1010.

[5] 闫桂勋,刘　蓓,程　浩,等. 数据共享安全框架研究[J]. 信息安全研究,2019,5(4):309-317.

[6] 吴兆立. 高校共享数据中心关键技术[J]. 电子技术与软件工程,2018(23):153.

[7] LI Bin, WANG Yijie, SHI Peichang, et al. FPPB: a fast and privacy-preserving method based on the permissioned Blockchain for fair transactions in sharing economy[C]// 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE). New York: IEEE, 2018: 1368-1373.

[8] CHOWDHURY M J M, COLMAN A, KABIR M A, et al. Blockchain as a notarization service for data sharing with personal data store[C]//2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE). New York: IEEE, 2018: 1330-1335.

[9] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[R/OL]. 2008. https://bitcoin.org/bitcoin.pdf.

[10] 郝　琨,信俊昌,黄　达,等. 去中心化的分布式存储模型[J]. 计算机工程与应用,2017,53(24):1-7.

[11] 李善青,郑彦宁,邢晓昭,等. 科学数据共享的安全管理问题研究[J]. 中国科技资源导刊,2019,51(3):11-17.

[12] GARCÍA-BARRIOCANAL E, SÁNCHEZ-ALONSO S, SICILIA M A. Deploying metadata on Blockchain technologies[C]//Research conference on metadata and semantics research. Tallinn, Estonia: [s. n.], 2017: 38-49.

[13] 沈苏彬. 网络安全原理与应用[M]. 北京:人民邮电出版社,2005.

[14] KHAN N, LAHMADI A, FRANCOIS J, et al. Towards a management plane for smart contracts: Ethereum case study[C]//NOMS 2018 - 2018 IEEE/IFIP network operations and management symposium. Taipei: IEEE, 2018.

[15] 刘竹松,何　喆. 基于 Merkle 哈希树的云存储加密数据去重复研究[J]. 计算机工程与应用,2018,54(5):85-90.

[16] ROGAWAY P, SHRIMPTON T. Cryptographic hash-function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance[C]//International workshop on fast software encryption. Berlin, Heidelberg: Springer, 2004: 371-388.

[17] YVES-ALEXANDRE D M, EREZ S, WANG S S, et al. OpenPDS: protecting the privacy of metadata through SafeAnswers[J]. PLoS ONE, 2014, 9(7): e98790.

[18] 翟社平,李兆兆,段宏宇,等. 区块链关键技术中的数据一致性研究[J]. 计算机技术与发展,2018,28(9):94-100.

[19] BARTOLETTI M, POMPIANU L. An empirical analysis of smart contracts: platforms, applications, and design patterns[C]//International conference on financial cryptography and data security. Sliema, Malta: [s. n.], 2017: 494-509.