文章编号：2096-1472(2018)-10-34-04
DOI:10.19644/j.cnki.issn2096-1472.2018.10.010

# 基于区块链技术的个人信息管理

黄小菊，徐文起，章 涛，宫学庆
(华东师范大学计算机科学与软件工程学院，上海 200062)

**摘 要**：现有的个人私密信息管理工具一般基于独立的软件或云服务，存在着数据难以共享、易被泄露等问题。区块链技术具有去中心化、数据难以篡改、可追溯等特点，能用于数据管理，但目前缺乏成熟的应用模式。本文对利用以太坊智能合约存储和管理数据的特点进行了分析，设计了一个基于区块链技术的个人信息管理系统。其不依赖于任何服务商实现个人信息数据的存储和分享，能够有效解决数据丢失、数据泄露和数据滥用的问题，同时也为区块链技术应用于数据管理提供了一种可行的解决方案。

**关键词**：区块链；智能合约；个人信息管理
**中图分类号**：TP3 **文献标识码**：A

# A Study of Personal Information Management Based on Blockchain

HUANG Xiaoju,XU Wenqi,ZHANG Tao,GONG Xueqing
( *School of Data Science and Eningneering*,*East China Normal University*,*Shanghai* 200062,*China*)

**Abstract:**The existing personal private information management tools are generally based on independent software or cloud services,and there are problems such as difficulty in sharing data and being easily leaked.Blockchain technology has the characteristics of decentralization,falsified data prevention,and traceability.It can be used for data management,but currently there is no mature application mode.We analyze the characteristics of using Ethereum smart contracts to store and manage data,and design a block-chain-based personal information management system.It does not rely on any service provider to store and share personal information,and can effectively solve the issues of data loss,data leakage,and data abuse. Furthermore,it provides a feasible solution for blockchain technology applied to data management.

**Keywords:**blockchain;smart contract;personal information management

## 1 引言(Introduction)

随着互联网和智能手机应用的普及，大量的数字化个人信息已经成为人们最重要的个人资产，如个人联系方式、通讯录，以及在各种APP中的用户名和密码等。现有用于个人信息管理的工具难以在保证数据安全性的前提下实现便利的数据共享。

区块链是基于P2P网络、共识机制和加密算法等技术的新型应用模式，具有去中心化、数据难以篡改和可追溯等特点。区块链技术的研究是当前学术界和工业界共同关注的一个热点问题，相关研究工作不仅包括共识算法[1]和智能合约[2]等区块链实现技术，还包括如何将区块链技术应用于各个领域，如物联网[3]、金融[4]和医疗[5]等。作为一种新兴的技术，其应用模式还没有公认的成熟方案，有待于进一步研究。

本文设计了一个基于以太坊[6]的个人信息管理解决方案，将个人信息数据通过以太坊的智能合约保存到区块链中，不仅能够实现数据的可靠存储、同步和分享，还能够避免数据泄露和数据滥用的问题。

## 2 区块链技术(Blockchain technology)

区块链技术是综合共识机制、加密算法、智能合约、P2P网络等技术的新兴技术框架，应用范围广泛，但目前尚没有公认的成熟应用模式。以太坊作为最主流的去中心化区块链应用开发平台，现有的开发框架和开发语言都相对较成熟，并且支持智能合约。本文构建的个人信息管理系统基于以太坊实现。

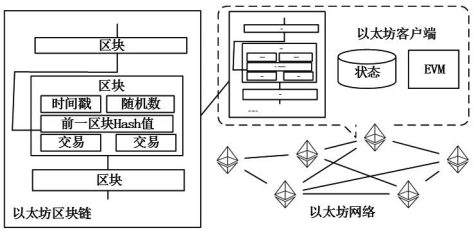


图1 以太坊网络结构

Fig.1 The architecture of Ethereum network

基金项目：国家自然科学基金项目（No.615721194）.

以太坊网络是由运行着以太坊客户端的节点组成的P2P网络，结构如图1所示。以太坊客户端中存储着以太坊区块链和以太坊的状态，同时也运行着EVM(Ethereum Virtual Machine)。以太坊区块链是按链表结构组织的区块集合，区块中存储着时间戳、随机数、前一区块数据的Hash值，以及大量的交易。交易中记录着以太坊账户发生的状态变化。以太坊状态是区块链中特定时间点上所有账户和数据的快照[7]。账户分为外部账户和合约账户两类，账户地址为20字节的16进制字符串。外部账户类似现实世界的银行卡账户，其中存储着账户余额等信息。合约账户即智能合约，存储着合约数据和合约代码。EVM是能执行交易、修改以太坊状态，其为智能合约提供了完全隔离的运行环境。

智能合约是一段运行在以太坊网络中的脚本，可通过变量赋值、函数调用来存储、操作数据。其中的数据存储在以太坊节点中，具有四个重要特点：(1)公开性。任何连接到以太坊网络的节点均可自由读取以太坊区块链中的所有数据。(2)数据管理成本较高。调用智能合约来存储、操作数据会消耗交易发起者一定的以太币。同时，以太坊网络中的所有节点都需要执行该合约，消耗了大量的计算资源。(3)数据操作是异步的。执行合约过程中需要向网络发送交易、产生新区块。只有在这些过程完成后数据操作才算成功。(4)不易丢失、难以篡改。智能合约中的数据存储在以太坊网络中，使得数据难以篡改。以太坊网络中每个节点均存储着智能合约数据副本，保证了数据不易丢失。

## 3　系统设计(System design)

为解决现有个人信息管理工具存在的问题，提出基于以太坊的个人信息管理系统。该系统具有以下特点：(1)能存储大量的个人数字化信息；(2)能提供可靠的数据存储；(3)能灵活控制数据的访问权限；(4)能实现数据共享。同时，该系统的实现也是对以太坊应用模式的探索。

### 3.1　系统总体架构

系统可以分为以太坊智能合约、系统节点两个部分，架构如图2所示。智能合约负责管理用户重要数据、控制数据访问权限；系统节点包含本地存储、以太坊客户端、数据管理服务、云存储服务四个组件，分别负责存储数据、连接以太坊网络、数据处理和云存储服务，由此管理、维护用户所有数据。

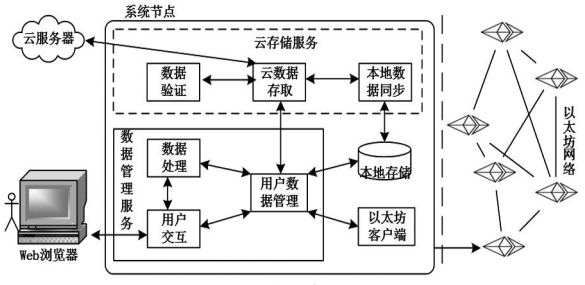


图2 系统架构

Fig.2 System architecture

### 3.2　以太坊智能合约的设计

我们设计了用于用户个人信息存储、访问控制的UPC(User Profile Contract)和用于共享数据的GPC(Group Profile Contract)两种合约，实现语言选用Solidity。GPC的数据共享类似QQ的群文件共享，以下认为有权限访问同一个GPC数据的用户处于一个群，群主为该GPC的创建者。

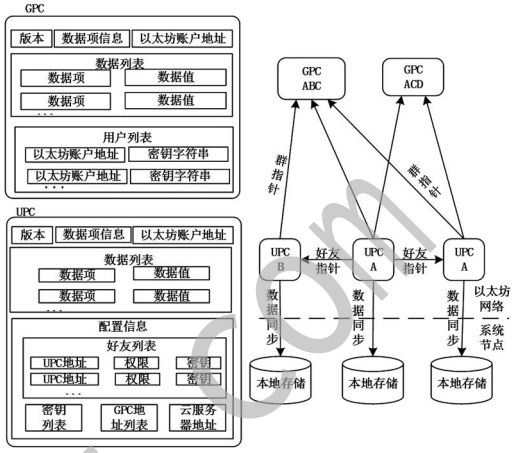


图3 智能合约结构

Fig.3 The structure of smart contract

UPC、GPC及本地存储的关系如图3所示，UPC、GPC存储在以太坊网络中，本地存储为系统节点的重要模块。本地存储中的数据和合约中的数据需要进行同步，用于保证数据的一致性；UPC中会记录用户所加入群的GPC地址，便于查找群内成员分享的数据；UPC中还会记录好友的UPC地址，使得合约之间互相关联，便于信息的访问。

#### 3.2.1　UPC

UPC由用户创建，用于存储和管理用户关键数据，从而可保证用户数据的可靠存储、安全共享。结合加密算法后UPC可用于控制数据的访问权限。

UPC的数据结构如图3所示，包含版本、数据项信息、以太坊账户地址、数据列表和配置信息。版本存储特定区块的时间戳，该区块中包含最近修改合约状态的交易，由此来标记数据的更新时间；数据项信息是一个字符串，其中包含数据列表中存储的所有数据项名，由此可从数据列表中获取到所有数据值；以太坊账户地址即为用户登录系统的账户地址；数据列表中存储着用户所有个人数据；配置信息中记录了好友列表信息、GPC地址列表、用户 地址，以及用户加密自身数据的所有密钥列表。其中好友列表中包括好友的UPC地址、用户提供给好友的密钥名称列表、好友提供给用户的密钥列表，由此即可查询到用户所有好友的信息、好友对用户信息的访问权限。配置信息在UPC中以加密字符串形式存储，用于系统节点初始化、获取合约数据。

存储用户数据时可以自主选择加密策略：不加密、仅加密数据值或数据项与数据值均加密，由此可平衡数据处理效率和数据安全性的要求。

UPC中主要包括两个方法：①setData(string_key,string_

value)用于添加、更新、删除数据。setData()中设置了自定义的onlyOwner Modifier，用于保证方法调用者只能是合约创建者。执行该方法需要向以太坊网络发送交易，消耗以太币；②getData(string_key)用于获取数据，不需要向以太坊网络发送交易，因此不消耗以太币。

#### 3.2.2 GPC

GPC主要用于群内成员互相分享数据。集中存储分享给特定群的数据能减少用户对大量无关数据的读取、解析，提高分享效率；针对同一数据项，GPC可向不同群分享不同的数据值，由此可简化合约中的数据存储，并实现灵活的数据分享。

GPC的数据结构如图3所示，包含版本、数据项信息、创建该GPC的用户账户地址、数据列表和用户列表。版本、数据项信息含义与UPC中的相同；以太坊账户地址即为创建该GPC的用户账户地址；数据列表中存储着群内成员分享的所有信息，它们的数据值均使用群密钥加密。该密钥由GPC创建者指定；用户列表中存储群内所有成员的账户地址及它们所对应的密钥字符串。该密钥字符串由GPC创建者使用用户公钥对群密钥加密得到，群内成员可从自己的密钥字符串中解析出群密钥，并用该密钥解密GPC中的数据或存储希望分享给其他成员的数据。

GPC中的主要方法包括setData(string_key，string_value)、getData(string_key)、setUser(address_friend,string_keystring)、getKeyString(address_friend)。其中setData()向Mapping写入数据时会在参数_key前加上调用者的账户地址，由此确定数据所有者。getData()和getKeyString()和UPC中的实现基本相同。

### 3.3 系统节点

由于以太坊智能合约中数据管理成本高，系统中较复杂的数据运算、大量数据的存储和处理都需在系统节点中完成，其开发使用Truffle框架完成。

#### 3.3.1 数据管理服务

数据管理服务负责维护系统中的所有数据，实现安全、可靠的数据管理，其三个功能模块之间的交互如图2所示。具体功能如下：

(1)用户交互模块负责为用户提供数据管理界面，由此向用户展示有访问权限的数据、收集用户提交的新数据。

(2)数据处理模块负责数据的加密、解密，以保证数据安全存储和共享。根据UPC中对数据类型的分类，对数据采用不同的加密算法处理。私密数据要求较高的安全性，需要使用非对称加密方法进行加密；对于模式公开数据，综合考虑安全性和数据处理效率，一般使用对称加密方法加密，使得用户向好友分享密钥即可实现分享数据。

当用户选择启用云存储服务时，对于部分安全性要求高且有共享需求的数据，可选用代理重加密方法进行加密。好友在线下向用户提出数据共享请求，用户即可使用重加密密钥生成算法产生重加密密钥，并通过云存储服务提交给云服务器。云服务器对数据密文完成重加密后，好友即可用自己的私钥解密获取的数据密文。由此可提供更高的数据安全性，并提高数据处理效率。

(3)用户数据管理模块提供了getData(string_key)、setData(string_key,string_value)两个数据访问接口，由此向本节点或其他节点的用户交互模块提供数据，由此实现数据的共享。同时，用户数据管理模块通过调用智能合约、本地存储的数据操作方法来存取、更新数据，以保证系统中所有数据的一致性。

数据管理模块中对于数据的加密粒度，以及加密方式都给了用户较大的选择空间，使得用户可以根据自身的需要在数据安全性和处理效率间权衡，带来较高的数据处理灵活性。

#### 3.3.2 以太坊客户端

以太坊客户端将系统节点连接到以太坊网络，从而完成发送交易、部署合约、调用合约函数等功能，本文使用的是go-ethereum客户端和MetaMask。

以太坊客户端会在用户首次登录时对用户提供的以太坊账户地址、密钥对进行检查，以此实现身份认证。之后，用户从系统节点发出的所有交易都由该账户发起，并由该账户提供发起交易所需的以太币。

#### 3.3.3 本地存储

本地存储用于在本地存储用户数据，从而提高系统数据管理效率。为实现与智能合约数据的同步且保证数据查询、更新效率，我们选用键值数据库HBase实现。

将存储在UPC和GPC上的数据分别存放在userProfileTable和groupProfileTable表中，并在合约地址、账户地址列上设置索引以便于查找。userProfileTable包含两个列族：contractInfo用于存储合约地址、版本、配置信息等合约相关信息；dataInfo用于存储用户个人数据。groupProfileTable包括contractInfo、dataInfo、userInfo三个列族，前两个内容与userProfileTable中的相同，userInfo中存储GPC的用户列表信息。

以上数据库模式的设计使得本地存储与智能合约进行交互时能快速实现数据的同步，同时支持大量数据的存储、查询、更新。

#### 3.3.4 云存储服务

云存储服务为可选功能模块，主要负责云服务器数据的管理，保证数据的可靠存储和安全共享。其三个功能模块的关系如图2所示。具体功能如下：

(1)云数据存取模块可连接云服务器，为数据管理服务提供管理云端数据的标准接口，包括getDataHash(string_key)、getData(string_key)、setData(string_key,string_value)、reEncrypto(string_key,string_encrptokey)，分别用于获取数据Hash值、获取完整数据、更新数据、重加密数据。

(2)数据验证模块负责对从云服务器中获取的数据进

行验证，以减少无效数据的传输。使用云数据存取提供的getDataHash()从云服务器中获取数据Hash值，确定和数据管理服务提供的Hash值相同时，再调用getData()请求完整数据，并返回给数据管理服务。

(3)本地数据同步模块用于检查云服务器和本地存储中数据的一致性，以防止云服务器或本地存储中的数据被篡改。同步在系统节点空闲时进行，通过验证数据Hash值的一致性来实现。

该模块的引入能在保证数据安全性的同时保证数据存储、分享的效率。

## 4 场景分析(Scenario analysis)

为完整描述利用智能合约完成数据存储、分享的过程，假设场景：A、B为好友，分别使用系统节点A和系统节点B将他们的个人数据存储在智能合约上。A、B均未启用云存储服务功能。B可访问A的“办公电话”“学习经历”等信息。现假设系统节点A发生故障，本地数据全部丢失，用户A仅记录了以太坊账户$Addr_{user}^{A}$、密钥$Key_{pri}^{A}$、UPC地址$Addr_{upc}^{A}$。系统节点B在本地存储了所有可访问的数据，且用户B记录了$Addr_{user}^{B}$、$Key_{pri}^{B}$、$Addr_{upc}^{B}$。在此场景下进行实验，以验证系统的有效性，图4描述了用户A，B使用系统进行数据管理的实验流程。

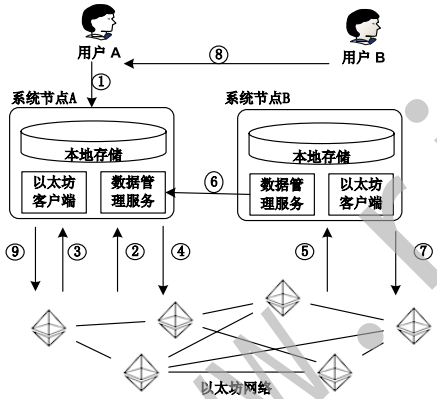


图4 数据管理流程

Fig.4 The procedure of data management

系统节点初始化：图4中第①到③步描述的是用户A初始化系统节点、获取数据的三个关键步骤。

①用户A向系统节点提供$Addr_{user}^{A}$和$Key_{pri}^{A}$，由系统节点对其进行验证，确认正确之后进入第二步，否则初始化失败。

②系统节点使用$Addr_{upc}^{A}$可访问A的UPC由此读取A的所有数据，并将其解密。解密后的数据包含A的个人信息列表、好友B的UPC地址，以及对应的密钥等，被存储到节点的本地存储中。

③根据上一步得到的合约地址可读取出其他用户分享给A的所有数据，并将其存储到本地。系统节点读取的数据展示给用户，初始化过程结束。

该初始化过程保证了用户在不丢失合约地址的情况下，关键数据不会丢失。

更新数据：用户A的“学习经历”信息中数据量较大，完整数据存储在本地存储中，智能合约中仅存储了该信息的Hash值。现需更新该数据，流程如图4中第④到⑥步所示。

④系统节点调用函数将加密的新数据同步到合约中。合约执行成功后，将最新的数据Hash更新到本地存储中。

⑤系统节点B定期从$Addr_{upc}^{A}$中读取A的版本信息，当发现其本地版本低于UPC中的版本时，重新读取合约数据，并进行解密。

⑥对于“学习经历”等存储在A节点中的数据，B向A节点的数据管理服务发送获取数据的请求。得到的数据密文由B使用$Key_{pri}^{B}$解密，并更新到系统节点B的本地存储中。

通过监控合约状态，用户可获取到好友的实时最新数据；通过标准的数据访问接口可以实现数据的分享。

群内共享数据：B希望向A分享“地址”“邮箱”等信息，但他不想分享$Addr_{upc}^{B}$中存储的私人地址和邮箱。因此，他使用GPC来向A分享数据，具体流程如图4中第⑦到⑧步所示。

⑦用户B通过系统节点的以太坊客户端新建GPC，得到的地址为$Addr_{gpc}^{B}$，并指定群密钥为$Key_{gpc}^{B}$中。B使用$Addr_{gpc}^{B}$调用addUser()将$Addr_{user}^{A}$、使用$Key_{pub}^{A}$加密的群密钥添加到GPC的用户列表中。

⑧用户B在线下向A发送$Addr_{gpc}^{B}$。用户A即可使用该地址创建GPC实例，从而读取B分享的数据，并能使用GPC提供的方法管理数据。

上述分享过程使得用户访问好友数据时不需读取、解析好友UPC中的所有的信息，提高了数据访问效率。同时，对群成员来说，获取群内其他成员分享的信息仅需读取一个合约数据即可实现，数据分享、访问更加便利。

收回共享权限：A不希望再向B分享自己的“办公电话”信息，收回访问权限过程如图4中⑨所示。

⑨A将“办公电话”信息使用新的密钥$Key_{po}^{A}$进行加密，并更新到合约中。成功后，A向其他对“办公电话”信息有访问权限的好友发送$Key_{po}^{A}$。由此，数据访问权限被收回。

上述过程中被更新的合约数据量较少且过程简单，实现成本较低，并且能保证数据访问权限被彻底收回。

## 5 结论(Conclusion)

互联网及手机应用的普及为人们带来大量的数字化个人信息，而现有的个人信息管理工具存在着数据难以分享、易被泄露等问题。区块链技术具有去中心化、数据难以篡改、可追溯等特点，能为解决个人信息管理中的问题提供新思路。本文提出的基于以太坊的个人信息管理解决方案将个人信息数据存储到以太坊智能合约上，并由系统节点对数据进行组织和维护，由此可实现大量的个人数字化信息的可靠存储、安全共享，并确保用户对数据访问权限的控制。

(下转第30页)