分 类 号: TP309.2　　密　　级: _____

学　　号: 16207035015

西安科技大学
XI' AN UNIVERSITY OF SCIENCE AND TECHNOLOGY

硕士学位论文

Thesis for Master's Degree

区块链智能合约在学位管理系统上的研究与实现

申请人姓名：党京

指 导 教 师：孙弋

学 科 门 类：工学

学 科 名 称：通信与信息系统

研 究 方 向：区块链

2019 年 6 月

西安科技大学

学位论文独创性说明

本人郑重声明：所呈交的学位论文是我个人在导师指导下进行的研究工作及其取得研究成果。尽我所知，除了文中加以标注和致谢的地方外，论文中不包含其他人或集体已经公开发表或撰写过的研究成果，也不包含为获得西安科技大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

学位论文作者签名：南京　　日期：2019.6.12

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西安科技大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文被查阅和借阅。学校可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西安科技大学。

保密论文待解密后适用本声明。

学位论文作者签名：南京　　指导教师签名：

2019 年 6 月 12 日

论文题目：区块链智能合约在学位管理系统上的研究与实现

学科名称：通信与信息系统

硕 士 生：党京　　（签名）党京

指导教师：孙弋　　（签名）孙弋

摘 要

高等教育已经成为各国发展科技实力和提高创新能力的重要途径。但教育管理信息的产生、修改及发布均由单一的管理机构集中处理，权限过度集中，教育信息存在泄露和篡改等潜在风险，这对个人教育信息的保密和社会诚信的公平公正带来了巨大安全隐患。区块链技术拥有去中心化、公开透明、防篡改、可追溯等特性，为教育管理应用系统的数据安全提供保障。智能合约具有按照用户意愿执行、无人能中断执行过程等优点。因此论文中提出一种基于区块链和智能合约技术实现风险可控的学位管理的解决方案。

论文中首先对去中心化的学位管理系统的整体架构进行设计，并对系统中学位注册、学位发布、学位召回、学位查询和交易/区块查询等业务进行分析。其次根据需求设计了学位交易的业务规则和流程，利用智能合约技术设计实现学位交易的合约模块。论文中根据不同对象控制不同权限功能以及节点的动态管理功能的基础合约，利用合约实现学生、学校、教育局之间相互交互的业务合约，并使用 Oyente 工具检测业务合约的安全性，对存在问题的合约改进并实现。为了提高系统在交易过程中的可靠性和执行效率，系统中使用安全性高、成熟度高的 ECDSA（Elliptic Curve Digital Signature Algorithm，椭圆曲线数字签名算法）算法对交易数据进行签名，并验证签名是否正确；使用 PBFT（Practical Byzantine Fault Tolerance，实用拜占庭容错）共识机制代替以太坊中效率低下、算力浪费的 PoW（Proof of Work，工作量证明）机制，PBFT 机制不需要节点挖矿，投票决定即可达成共识，与论文中基于联盟链的应用系统相契合。最后论文设计搭建以太坊网络区块链环境，通过 Node.js 将智能合约部署到链上，从而实现学位管理系统。在该系统交易的相关操作都是通过调用智能合约来实现。

通过测试验证了利用区块链和智能合约实现学位管理系统的可行性较高，区块生成时间短，运行效率快，为系统进一步扩展提供借鉴作用。

关 键 字：区块链；智能合约；学位管理；PBFT 共识机制

研究类型：应用研究

Subject : Research and Implementation of BlockChain and Smart Contract in Academic Degree Management System

Specialty : Communication and Information System

Name : Dang Jing **(Signature)** Dang Jing

Instructor : Sun Yi **(Signature)** Sun Yi

ABSTRACT

Higher education has become an important way for countries to develop their scientific and technological strength and improve their innovative ability. However, the generation, modification and publication of educational management information are centralized by a single management organization, with over-centralized authority and potential risks of leaking and tampering with educational information, which brings huge security risks to the confidentiality of personal educational information and the fairness and justice of social integrity. Block chain technology has the characteristics of decentralization, openness and transparency, tamper-proof, traceability and so on. It provides security for data security of educational management application system. Smart contracts have the advantages of executing according to user's wishes and no one can interrupt the execution process. Therefore, this thesis proposes a solution to realize risk-controllable degree management based on block chain and intelligent contract technology.

Firstly, the thesis designs the overall architecture of the de-centralized degree management system, and analyses the business of degree registration, degree publication, degree recall, degree query and transaction/block query. Secondly, according to the requirements, the business rules and processes of degree trading are designed, and the contract module of degree trading is designed by using smart contract technology. In this thesis, according to the basic contract of different objects controlling different permission functions and dynamic management functions of nodes, we use the contract to realize the interactive business contracts among students, schools and educational bureaus, and use Oyente tool to detect the security of business contracts, and improve and implement the existing contracts. In order to improve the reliability and execution efficiency of the system, ECDSA (Elliptic Curve Digital Signature Algorithms) with high security and maturity is used to sign the transaction data and verify whether the signature is correct; PBFT (Practical Byzantine Fault Tolerance)

Consensus mechanism is used to replace the inefficient and wasteful PoW(Proof of Work) inTaifang, PBFT mechanism does not need node mining, voting decision can reach consensus, and it is consistent with the application system based on alliance chain in the thesis. Finally, the thesis designs and builds the Ethernet network block chain environment, and deploys the smart contract to the chain through Node.js, so as to realize the degree management system. The related operations of transactions in this system are realized by calling smart contracts.

The test results show that the feasibility of using block chain and smart contract to realize degree management system is high, the block generation time is short, and the operation efficiency is fast, which provides a reference for further expansion of the system.

Keywords: BlockChain; Smart Contracts; Degree Management; PBFT Consensus Mechanism

Thesis : ApplicationResearch

目　录

1 绪论

1.1 课题研究背景及意义

1.1.1 研究背景

近年来，国家对教育领域制度改革和完善[1]高度重视，促使各大高等院校逐年扩招，接受教育的学生数量屡创新高。从 2001 年到 2018 年底，学生人数由 260 万上升到 980 万。学生人数的急剧增加，加大了各大高校处理学生业务的工作量与难度。随着学生学术成果申报、等级考试等应用场景逐渐增加到教育管理系统，由校园人工管理逐渐向数字化信息管理转变，使得我们现有的管理系统面临着效率乃至性能上的挑战。

同时在知识经济发展的大爆发时代，以学历证书和学位证书为代表[2]的各类文凭是一个人接受高等教育程度最直接、最客观的纸质化体现，也是各大企业招聘优秀人才的基本标准，更是一个人在社会求职过程中获得理想职位、谋求更高层次发展的重要条件。正是由于学位学历等文凭是现代社会人才教育背景的证明和创业就业的通行证，一些想要投机取巧的人就会采取学历造假方式来获得高额利益。中国史上著名人物蒋介石为满足高学历要求对自己学历文凭进行造假；新华都集团总裁兼 CEO 唐骏和雅虎首席执行官斯科特·汤普森(Scott Thompson)涉嫌学位造假事件引起世界轩然大波，最终以“引咎辞职”收场。“学历造假门”事件不仅引发了人们对各类事务和各行事业的诚信危机，背后更是反映出当前学历和学位信息管理系统存在诸多问题。目前中国高等教育学生信息网是管理学生学位的重要系统，但其存在的漏洞之所以能被恶意利用，根本原因在于系统是一个中心化的信息管理系统，主要弊端有四方面，其一，存储数据由中心管理员控制，权限过于集中，数据被篡改的可能性较大；其二，中心服务器被不法分子恶意攻击会导致数据泄露造成数据完整性问题；其三，为防止系统被攻击，增加维护系统安全设施，造成投资成本增加；其四，存储数据无法真正的做到公开透明，保证数据的客观真实性。

区块链技术及其产物智能合约的出现，给我们提供了新的思考方向。面对现有学位信息管理系统出现的问题，我们应该考虑如何将区块链与智能合约技术应用于该系统，解决系统效率以及学位信息安全等方面的问题，最终实现公开透明、去中心化的应用系统[4]。

1.1.2 选题意义

区块链技术最早是以比特币的底层基础技术框架形式出现，其概念由“中本聪”学

者在 2008 年所撰写的论文《Bitcoin: A Peer-to-Peer Electronic Cash System》[5]中首次提出，目的是为解决数字系统中出现的“双花问题”（双重支付）以及攻击者难以篡改系统交易问题，区块链技术最初应用是电子货币。随着对区块链技术进一步研究与创新，新一代以太坊区块链平台的出现，致使区块链不再局限于电子和数字加密货币交易，区块链已经在公正防伪、物联网、身份验证、预测市场、资产交易、电子商务、文件存储、社交通讯等方面得到了应用。同时基于以太坊区块链技术的应用领域已经朝着智能合约技术方向扩展。1995 年 Nick Szabo 首次提出智能合约概念，其类似于将法律条文写成可自动执行代码。Vitalik Buterin 将智能合约与以太坊[6]区块链平台结合，设计开发出一套可执行图灵完备脚本语言的虚拟机，通过在以太坊中编写智能合约，在虚拟机中运行智能合约，保证合约程序不被干扰，能够自动调用执行，最终在平台上实现各类去中心化应用（Decentralized Application，DApp）。智能合约不仅推动了区块链的多元化发展与应用，还保证了应用领域系统的安全可信。

区块链技术应用于各类系统的底层中，使系统具有安全、可信赖和普适性等优势。本课题基于上述背景考虑，提出以联盟链为基础，使用智能合约技术设计学位管理系统的业务逻辑，并验证合约的安全性，实现一个去中心化的学位管理系统。本系统设计合约的价值内涵主要体现在以下几个方面[7][8]：

1）用智能合约中的逻辑规则和表达能力实现学生、学校和教育局等不信任三方之间交易的安全交互，解决了协议规则被不法节点中断的问题。

2）存储到区块链的数据不可以被更改，无法作假。

3）降低了因第三方参与维护而产生不必要的成本，系统内部可信度高。

4）丰富了交易与外界状态的交互，企业和公司可以通过外部接口查询链上数据的交易（学位信息）。

1.2 国内外研究现状

如今，区块链在金融、教育、国家电网、医疗机构、政治等领域有着重要的应用价值。区块链技术作为新兴产业技术代表，其去中心化、安全可信、防篡改、可追溯等特性，促使各国对区块链的认识程度逐渐提高。国内外许多科研机构已经研究出区块链新的技术，并将其运用到相应工程。政府部门也开始对区块链技术给予高度关注，同时推出相应的发展规划。

2015 年 11 月，新加坡政府呼吁银行和监管机构密切关注区块链等新技术的发展。2016 年初，联合国社会发展部发布了《“加密货币”以及区块链技术在建立稳定金融体系中的作用》报告，提议应用区块链技术组建一个稳固的金融体系。2016 年 9 月，美国众议院通过一项严格要求区块链技术的无约束力的决议[9]，对区块链技术进行深入研究探讨[10]。2016 年年底，国家政府也将区块链技术纳入“十三五”国家信息化规划[11]。2017

年，欧洲中央银行开始探索如何将区块链技术应用于证券和支付结算系统中。2017 年 6 月，成都成立西南区《区块链创新发展联盟》。这一系列行为表明，区块链技术得到世界各国政策上的支持，这对发展区块链技术来说是一个强大助力。

同时，世界上各国家企业也对区块链项目进行开发试验，并为此投入了大量精力和财力。2016 年初，Yanislav 等人开发的 Aeternity 项目，很好的解决了区块链网络拥堵、速度慢、转账费用高等问题。2017 年 1 月，由中国中央银行推出的基于区块链的数字票据交易平台已测试成功。Linux 基金会支持的 Hyperledger 联盟推出了 Hyperledger Fabric 项目，项目主要利用区块链技术服务于企业级别的联盟团体。2017 年 3 月，蚂蚁金服借助区块链技术开发了为贫困地区人民服务的爱心捐赠平台[12]。2017 年 4 月，腾讯提出了可信区块链平台 TrustSQL[13]的整体方案，目的是为合作企业提供一站式应用服务。2018 年 7 月，迅雷为给区块链专门打造数据云存储区，自主研发了授权分发的迅雷链文件系统 TCFS（Thunder Chain File System）。

截止到目前，区块链技术的发展主要以智能合约为主导，在以太坊区块链平台编写任何去中心化应用，比如投票、域名、金融交易、众筹、知识产权等等。此外，国内外很多机构和政府都开始针对学生信用体系不完整、无历史数据信息链等问题，提出了利用区块链智能合约技术对学生信息进行存储，极力解决信息不透明及易被篡改问题，为建立良好的学生信息体系提供保障。不仅如此，其它各大领域也在积极利用区块链智能合约解决当前领域现存问题[14]。表 1 所示显示了目前区块链技术在各领域的应用场景。

表 1.1 区块链已应用领域

相关文献	应用的研究领域	描述
文献[15]	医疗领域	使用区块链技术的安全性处理 EMA 和利用区块链属性实现数据共享
文献[16]	智能化城市	区块链技术优势解决当前智能城市网络架构中出现延迟、带宽瓶颈、隐私安全性等问题
文献[17]	教育领域	区块链实现了学术界之外的教育声誉并使其民主化
文献[18]	数字版权领域	利用区块链中的 Nonce 值、哈希算法等对作品进行确权，追溯

基于智能合约的去中心化应用在今后更多场景使用，区块链智能合约的未来还存在着无限可能。因此，人们有理由期待在区块链技术和智能合约的契合下，解决金融领域及世界各行领域中出现的安全信任问题。这是又一次“大航海时代”的来临，也是对人类社会又一次重大重构。

1.3 课题主要研究内容

为解决当前中心化学位管理系统的弊端，论文利用区块链技术和智能合约的优点，将其结合应用于学位管理系统中。该系统的设计是基于以太坊联盟链架构，首先分析了学位管理系统中学生学位信息注册、学位发布、学位召回、学位查询的业务流程。利用智能合约设计了基础合约、数据合约以及业务合约，合约可完成前端系统业务与区块链之间学位信息交易的交互过程。交易执行过程由节点达成共识，将共识结果写入区块链，并对以太坊网络中动态节点增减算法和交易数据验证算法进行设计和实现。针对 PoW 算法的共识周期长、效率低下及算力浪费等问题，论文中使用 PBFT 共识算法代替，该机制只需全网 2/3 节点投票通过即可达成共识。基于上述分析，本文设计搭建以太坊区块链环境，并在其上部署智能合约，设计并实现基于六节点联盟的学位管理系统，以学位信息交易业务对区块链数据查询接口及系统平台进行测试。

针对以上所提出的研究内容，本论文章节与内容安排如下：

第一章绪论。简要介绍了课题的研究背景及选题意义，分析了现阶段学位管理系统的弊端和区块链技术在国内外目前的研究现状及应用领域，最后论述了课题的主要研究内容和各个章节安排。

第二章区块链相关技术。本章分析了区块链产生原理、分类及整体架构；对以太坊区块链技术原理进行深入探索，引出以太坊网络涉及的分布式网络、数字加密算法和存储结构 Merkle Tree，分析比较目前较常用共识机制的优缺点，为学位管理系统选择合适的共识算法提供依据；提出智能合约概念，对以太坊的智能合约做以简单阐述。

第三章学位管理系统总体设计。本章对学位管理系统的功能需求和非功能需求进行分析，对系统的整体架构进行设计，并分析了整个系统的业务模块。

第四章智能合约在学位管理系统中的设计与实现。本章节首先对学位管理系统的业务流程进行分析；其次研究了合约的设计原则，利用智能合约对学位管理系统的业务逻辑进行设计。根据学位数据在系统的交易流程，将智能合约与业务逻辑交互设计。验证合约安全性，并对合约改进与实现。对以太坊中核心算法进行设计与实现。最后设计实现基于六节点联盟的学位管理系统。

第五章学位管理系统测试。对系统进行完整性的功能测试，对底层区块链交易数据通过外设接口进行测试，搭建以太坊网络监测环境，监测交易、区块和智能合约的运行情况，最后对测试结果进行分析。

第六章总结与展望。对论文所做具体工作进行了总结，展望了后续可研究内容方向。

2 区块链相关技术

随着相关技术的发展，区块链和智能合约已经成为大家讨论和关注的热点。区块链技术避免了基础数据被篡改的客观性问题，防止了数据展示与数据对称（发布与变更）时产生变节性问题，使得存储区块链上的数据更加的透明和客观。因此各行各业积极探索适用于本行业的区块链技术，以期望开发出完全去中心化、安全性较高的专用应用平台。在介绍基于区块链智能合约的学位管理系统设计方案前，本章先从区块链的基本概念、区块链底层技术以及智能合约技术等方面全面剖析。

2.1 区块链

2.1.1 区块链产生原理

区块链技术本质上是去中心化的数据库，是比特币的核心技术和基础架构。它是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式[3]。基于区块链技术的体系不需要第三方去审计，而是以密码学方式去维护一份不可篡改和伪造的分布式账本，利用共识机制对去中心化节点系统中的交易达成一致，保证账本统一，从而解决了无第三方参与引起的信任和安全问题[19]。其相关概念主要包括以下几个基本知识：

交易（Transaction）：区块链上引起区块状态变化的操作均称为交易，每一次交易对应唯一的交易哈希值，之后会对交易打包；

区块（Block）：一定时间间隔内打包记录多组交易数据，是对当前账本的一次共识；

链（Chain）：根据区块上的时间戳顺序将区块连接成区块链。

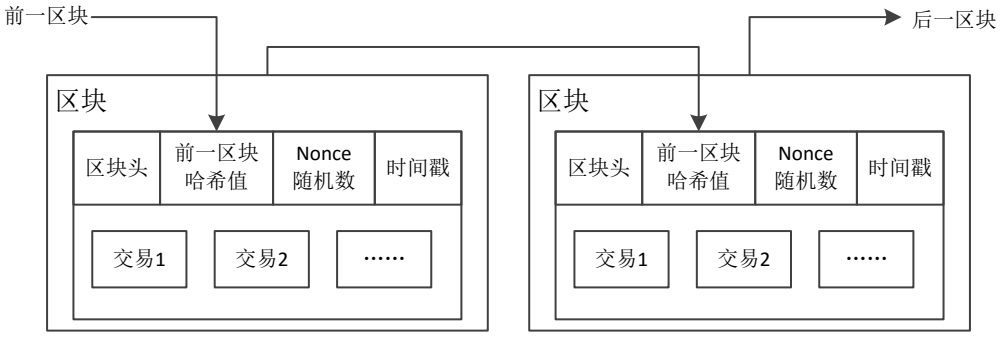

图 2.1 区块链数据结构图

如图 2.1 所示，区块是由表示这个区块的区块头和构成区块主体的一系列交易组成的，区块按照时间戳前后顺序连接起来就构成了区块链，区块从后向前有序的连接在该链条上。同时每一个区块头都必须运用哈希算法进行加密，得到一个哈希值，在区块链

上可以通过这个哈希值查询出相对应区块。区块是通过区块头信息进行标识，每个区块头中都含有一个指向父区块的指针，该指针是父区块的区块头进行哈希运算得到。我们利用区块头的“前一区块哈希值”字段引用前一区块，后面的每个区块都会包含前一区块的哈希值，故可以创建出一个可以追溯到第一个区块的链条。这个指向父区块的哈希值保证了交易数据的安全。如果想要改变一个已经生成的区块数据，必须达到两个要求才能对区块数据修改：一是在后面区块产生前，所有父区块的区块头的哈希值必须得到更改，这就需要网络节点拥有 51%的计算能力；二是修改速度超过分布式网络中其它节点产生新区块的速度。这种区块链数据篡改方式又称为 51%攻击，由于全网计算能力有限且运算量过大，该方式从根本上来说是不可行的。

2.1.2 区块链分类

目前区块链大体分为公有链、私有链和联盟链，下面对几种区块链简要介绍。

公有链就是公开、完全去中心化的区块链，意思就是所有人都可以访问区块链网络中的任何节点数据，每个节点都可以参与链上数据的读写、交易的执行、交易数据的有效验证以及区块在网络中的共识过程，共识过程即就是有权决定哪个区块可以添加到主链上并记录当前的网络节点[37]。公有链作为去中心化信任的衍生物，采用密码学机制来保障链上数据的安全，采用工作量证明机制及其奖励机制激励更多节点进行挖矿。比如区块在达成共识后，挖矿的节点就会获得一定数量的经济奖励。

私有链中存在着控制写入权限（如交易和验证权限）的管理节点，管理节点会依据情况对外设置读写权限或其它限制[38]。管理节点的存在让很多人对私有链产生争议，认为违背了区块链技术的初衷。但私有链最大的优势在于加密和审计，因此常用于机构和公司内部的数据库管理。

联盟链是去中心化或多中心化的区块链。在共识过程中，会提前选择某些节点控制成员区块以达成共识。就好比说，由 N 个金融机构构成的区块链系统，每个机构都将作为一个节点参与交易区块验证过程，只有 2/3 N 以上节点通过验证才会认可。联盟链中的读写权限、记账规则必须由选定的节点共同设置，节点数量有限，所以共识协议可以不用工作量证明的挖矿机制，选用 PBFT 算法投票决定，速度快效率高。同时联盟链对交易的时间、状态、每秒交易数有更高的安全和性能要求，因此本文选取联盟链区块链作为系统底层框架。

2.2 区块链架构体系

2013 年，Vitalik Buteria 提出“以太坊”概念[20]——一种能够被编程用以实现任意复杂计算功能的单一区块链。2014 年，以太坊项目被成功开发后，作为了下一代区块链平台。今天，以太坊是一个有智能合约功能的去中心化应用平台，智能合约内置监控 Event

事件机制，当合约中的状态机监听到满足条件的事件状态，就会选择相应合约动作自动执行。利用智能合约在以太坊平台开发的区块链应用，其安全性、可靠性和易用性的提升促使更多开发者研究并发布下一代去中心化应用系统[21]。以太坊区块链整体架构如图2.2 所示，主要分为三层：底层服务、核心技术层和顶层服务应用。

底层服务：保证以太坊区块链系统平稳运行，包括 P2P 网络、LevelDB 数据库、密码学算法以及分片（Sharding）优化等基础服务；

核心技术层：以太坊核心组成部分，包括区块链、共识算法和以太坊虚拟机等核心元件。区块链为主体，共识算法为辅助，在 EVM（以太坊虚拟机）运行智能合约；

顶层服务应用：基于以太坊平台，结合自身专有业务，包括 API 接口、智能合约以及去中心化应用等服务。DApp 发送交易通过 Web3.js 接口与智能合约交互，且所有合约运行在 EVM 上，而合约与合约之间、合约与区块链之间的交互都会通过 RPC 调用。

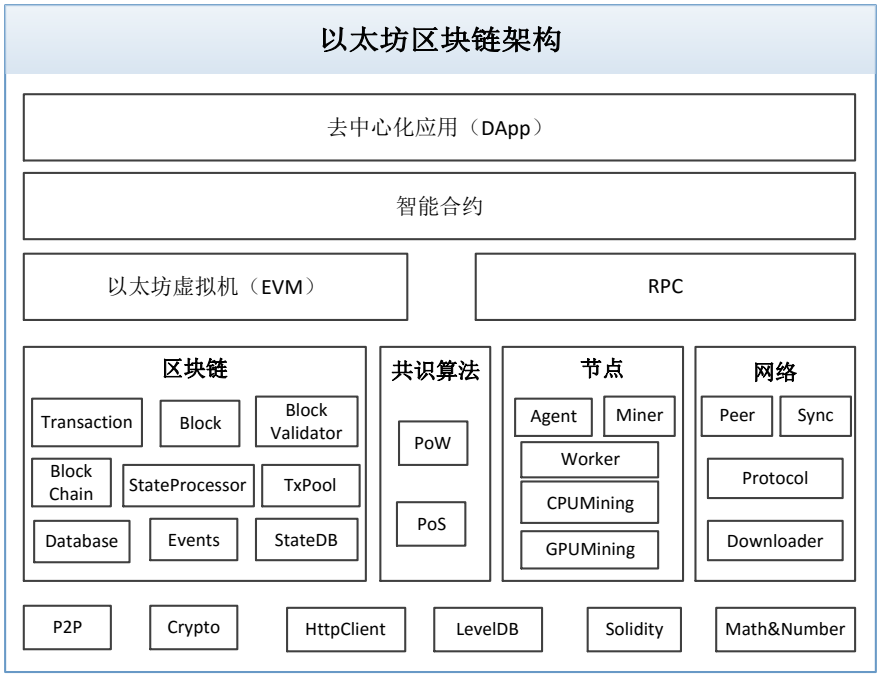

图 2.2 以太坊区块链架构

2.3 区块链技术

2.3.1 分布式网络

P2P（Peer to Peer）[22][23]协议是区块链去中心化系统中主要使用的网络协议。在 P2P 网络中，无中心化服务器，节点功能等同，这种等同表现在既可以是客户端也可以是服务器，既可以为其它节点提供数据服务，也可以享用其它节点提供的数据服务，不再只从中心化服务器中调用数据。当全网中部分节点丢失或被攻击，整个网络数据也不会受

影响，维护了整个区块链系统数据的安全性和完整性，如图 2.3 所示。

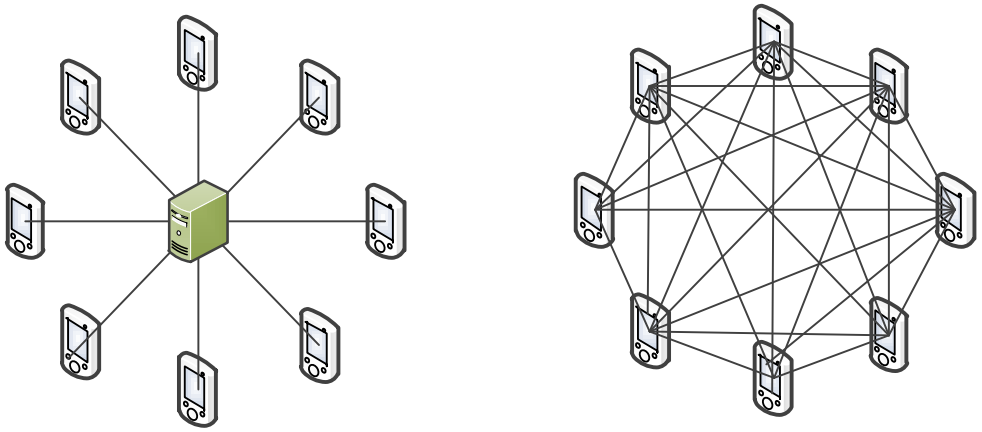

图 2.3 中心化网络与分布式网络

P2P 网络体系是去中心化、去信任、数据可靠和集体协作的网络体系，以太坊区块链具有以下特点：

去中心化。P2P 网络是分布式网络，各节点间数据交易有多个机构相互监督，避免了权限过于集中，数据被篡改的安全隐患，提高了数据的安全性。

去信任化。以太坊无中心节点，节点之间通过数字加密算法建立信任关系，保证交易的可靠性，因此分布式网络中节点间数据交换可信度高。

数据可靠。分布式网络中的所有节点均会时时更新并备份区块链数据，当全部节点同时损坏或者遭受攻击，系统数据才会被毁坏。单一节点被攻击时，不会影响整个系统数据的完整性。

集体协作。以太坊系统采用奖励机制保证全网所有节点均可参与到系统区块数据认可过程（比如比特币“挖矿”过程），利用共识算法选出某一节点将该区块加到链上。

2.3.2 数据加密算法

以太坊使用了多种数字加密算法，最主要的是哈希算法和非对称加密算法[24]，目的是在以太坊区块链中快速验证数据和确认用户的身份。

哈希算法（Hash 算法）是一种单向计算算法，其工作原理是用户可以通过 Hash 算法对任意长度的文本信息生成一段长度固定的唯一 Hash 值，却不能通过该 Hash 值反向获取到目标信息。哈希算法的特点是信息完全相同的两个文本，经过哈希算法加密后得到的 Hash 值也完全相同；但两个文本中即使有一个字符不相同，两文本加密后生成的 Hash 值均是杂乱无章且两 Hash 值之间没有任何关联性。目前以太坊区块链主要使用 SHA256[25]哈希函数处理区块中的交易数据，生成长度为 256bit（32Byte）二进制数。不同的哈希算法的特点如表 2.1 所示。

表 2.1 不同 Hash 算法比较

算法类别	安全性	输出大小（bit）	运算速度
SHA1	中	160	中
SHA256	高	256	略低于 SHA1
SM3	高	256	略低于 SHA1
MD5	低	128	快

非对称加密是由一组密钥对组成的加密方式[27]，此密钥对是唯一的。公钥是公开的，当对数据信息进行加密并实现机密信息交流时，使用公钥即可加密，私钥仅由信息的查看者拥有。解密信息时，拥有私钥唯一权限的用户才能获得解密信息，任何未经授权的用户（包括信息的发送者）都无法将信息解密。在以太坊系统中的每一笔交易都会有用户使用私钥去签名，区块链使用公钥去验证签名。只有当验证通过后该笔交易才合法，最终可以持久化存储到区块链上。如表 2.2 所示非对称加密算法的特点，本文出于系统的安全考虑，将使用 RSA-256 非对称算法对以太坊区块中的数据进行加密，使用椭圆曲线数字签名算法[26]（Elliptic Curve Digital Signature Algorithm，ECDSA）对区块数据进行签名和验证，主要证明数据的签名者为本人，无法代签，在技术上解决了信任问题。

表 2.2 不同非对称加密算法比较

加密算法	安全性	成熟度	运算速度	资源消耗	密钥长度	
					级别	长度（bit）
RSA	低	高	慢	高	80	1024
					112	2048
ECC	高	高	中	中	80	160
					112	224
SM2	高	高	中	中	80	160
					112	224

2.3.3 数据结构与存储

Merkle Tree[27]是比特币中的一种数据结构，其作用是将一个区块中的每个交易通过哈希算法生成整个交易的 Merkle 根值，使用这种数据结构对交易进行安全性存储和正确性验证。

Merkle Tree 基于哈希值、采用自底而上建立的二叉树模型。树中的每个叶子节点即一笔交易，每笔交易都对应一个特定的哈希值，之后将相邻两个哈希值再次进行哈希算法运算，得到的结果就是这两节点父节点的哈希值。一层一层叠加下去，直到树中只生成一个哈希值，即为整棵树的 Merkle 根。如图 2.4 所示，区块数据中有 L1~L4 四笔交易，

首先将 L1~L4 单元数据（交易）哈希化，得到每个叶子节点的哈希值分别为 Hash0-0、Hash0-1、Hash1-0、Hash1-1，再计算相邻两节点 Hash0-0 和 Hash0-1、Hash1-0 和 Hash1-1 的哈希值 Hash0 和 Hash1，不断叠加，直到计算出整棵树根节点的哈希值，将最终生成 32 字节的根节点哈希值存储到区块头中。该树最大优势就是如果底层数据被恶意篡改，那么对应叶子节点哈希值也会改变，最终导致其 Merkle 根值变化。当需要判断交易列表中某笔交易存在时，一个节点只需计算 $\log_2 N$ 个字节的哈希值，就可以得到一条从 Merkle 根到指定交易的路径，从而得到验证。使用 Merkle Tree 存储交易数据，在验证交易过程中，减少了数据的传输量和计算的复杂度。

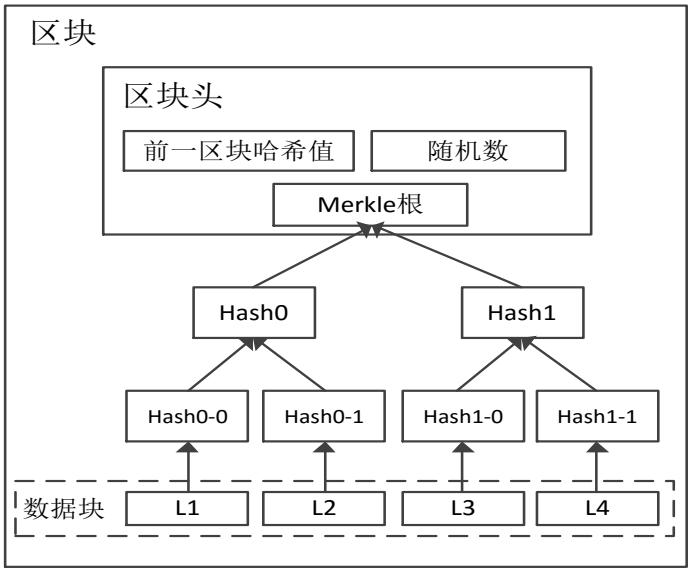

图 2.4Merkle Tree 结构

2.3.4 共识机制

区块链技术采用的是去中心化技术中由众多记账节点构成的 P2P 网络，但如何让全网中的节点就某一笔交易，通过一个规则将各自的数据保持一致是区块链技术中的研究重点。为此，开发者引入共识机制技术。所谓共识，简单理解就是指大家对某一件事情都表认可和赞同观点。共识机制技术则是网络节点在决策权高度分散的分布式账本中，按照某一协议，实现不同账本节点上数据的一致性和正确性。最初，比特币采用的是工作量证明（Proof of Work，PoW）机制，该机制主要依赖节点算力进行挖矿。随着区块链技术的改进，共识机制不断被创新，人们陆续提出了不过度依赖算力就能使区块链各节点达到全网一致性的机制，比如权益证明（Proof of Stake，PoS）机制、授权股份权益证明（Delegated Proof of Stake，DPoS）机制、实用拜占庭容错（Practical Byzantine Fault Tolerance，PBFT）算法等等。以太坊智能合约中使用最多的共识机制为 PoW，但由于 PoW 算法通过尝试输入 Nonce 值来决定节点记账权，计算量较大，电力和服务资源损耗

大。因此考虑使用 PBFT 共识算法，通过节点投票机制决定。以下对这几种共识算法进行简单介绍[29]。

（1）PoW

PoW 机制[12]利用复杂数学运算（算力竞争）来获取节点记账权，全网每一次达成共识都需要网络中所有节点同时参与竞争。1997 年，Adam Back 设计的“Hashcash”系统[6][30]最早使用 PoW 共识机制，后来中本聪将 PoW 共识机制应用于比特币区块链。

在 PoW 算法中，为了鼓励各节点（矿工）参与到网络节点竞争，PoW 设立了奖励机制，只有最先计算出复杂数学难题（挖矿）的节点能够获得奖励。在区块链中，参与竞争的节点针对某交易不断尝试各种随机数，对变更后的交易值进行 SHA256 哈希运算，计算出的哈希值小于等于给定的目标难度值，则挖矿成功。其它节点必须对该哈希值的正确性进行确认，验证通过生成区块头信息。对外广播新产生的区块，其它节点确认无误后新区块被添加到链上，主链长度加一，其它节点更新本地账本与主链数据信息同步，以维护全网统一。

该算法优点是算法简单、容易实现。缺点是挖矿过程中因计算出许多无用的随机数致使系统资源浪费，产生区块时间十分钟左右，区块确认时间长易导致 51%攻击。

（2）PoS

PoS 机制[32]要求网络参与者向其它节点展示持有的数字货币数量的所有权，是 PoW 的一种节能替代选择。只要有区块创建，节点就会产生一个“币权”交易，会给节点发放一些代币，PoS 根据每个节点的币龄值（即持有数字货币比例与货币持有时间的乘积）来改变挖矿难度。该算法不仅降低挖矿难度，计算寻找 Nonce 值速度也会提高，达成共识的时间也会减少。

PoS 优点是缩短共识时间，节省挖矿计算资源。缺点是挖矿成本低，网络可能会遭受攻击和破坏，易造成以太坊硬分叉问题。

（3）DPoS

DPoS 机制[33]与股份制企业类似，每一个节点代表一个股东身份，首先代币持有者通过投票的形式选出 101 名股东代表，这些代表彼此的权利是完全对等的，可以轮流生成区块并对区块进行验证，验证结果通过则产生的收益（交易代币）全部股东平分。如果某些代表未能及时生成区块，则会除去其股东代表身份，将由票数排名第 102 位股东替补，保证股东代表的总人数不变。

该算法优点是耗能低，达成共识速度快。缺点是易导致区块链分叉，且投票积极性不高，坏节点处理困难。

（4）PBFT 算法

PBFT 算法[31]是 Miguel 和 Barbara 设计的一种共识算法，目的是为了解决共识算法中复杂度高的问题，可应用于低延迟的存储系统，也可应用于吞吐量较小但需要处理大量交易事件的数字资产平台。PBFT 算法中公钥可以由网络中的主节点去发布，任何通过节点的消息都由各节点验证消息内容和格式，通过后进行签名认证。算法的优点在于能够容忍全网 N 个节点中的 f 个恶意节点，保证有 $2f+1$ 个信任节点，其中 $N=3f+1$，就可以达成共识并将信息添加到分布式账本中，保证全网数据一致性和安全性。

PBFT 算法共识处理流程如图 2.5 所示，即使部分节点被毁坏，网络仍可达成共识。

1）客户端会发送一笔交易给网络中的主节点。

2）主节点会将交易打包生成预准备消息广播给备份节点；

3）备份节点收到消息后，验证消息有效，生成准备消息发送到网络；

4）节点收到超过 $2f$ 个节点准备消息，验证通过生成提交消息广播到网络；

5）各节点收到 $2f+1$ 个提交消息后，就可以将该笔交易写入到区块中。

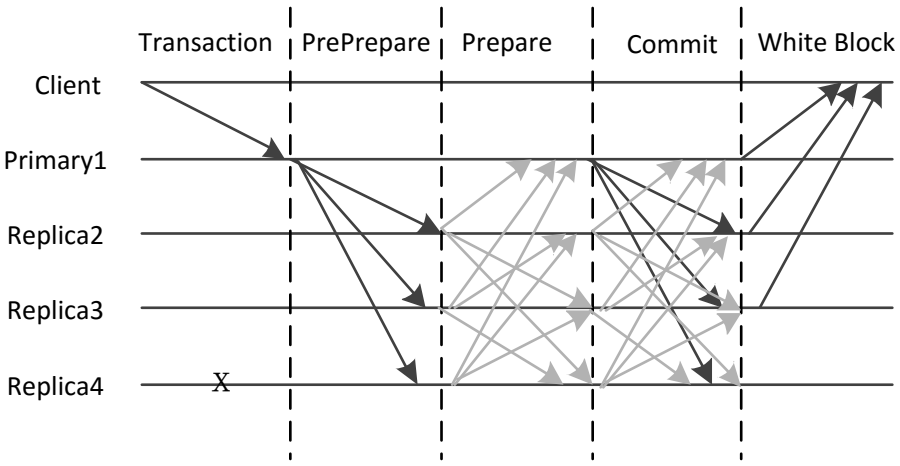

图 2.5 共识算法流程

该算法优点允许拜占庭容错（允许系统存在作恶节点），存在强监管节点，致使系统性能更高、耗能和容错性更低。

针对几种共识机制的优缺点进行分析比较[34]，PoW 和 PoS 机制在电子货币系统使用较多，其实现需要消耗大量的电力成本；DPoS 机制投票选举代理节点进行记账和验证交易，共识效率提高，适合代币存在的应用场景；PBFT 机制采取许可投票、少数服从多数原则，创建节点轮流记账的场景。根据本文需要实现的学位信息管理的业务场景，没有货币流通，可以允许少数节点拥有记账权（学校和教育局），且记账的节点可以轮流记账。基于上述原因，因此本文考虑使用 PBFT 共识算法，该算法监管性能更高，节点资源消耗较少，容错性更低，抗攻击性高，特别适合应用于学位管理系统。

2.4 智能合约技术

2.4.1 智能合约概念

1995 年，尼克·萨博（Nick Szabo）首次提出了智能合约概念，运用逻辑思维将已有的法律法规写成可执行代码。但将其运用到实际应用系统中，需要两个前提条件：一是系统中的应用如何触发智能合约并按照规范去执行；二是如何保证合约的运行环境不被外界不法分子破坏。这些问题一直阻碍智能合约的投入使用，直到区块链技术的问世，才让智能合约在实际应用场景中发挥作用。合约中的规则由分布式网络节点共同制定出一套双方都能遵守执行的协议，并将执行的数据结果写入区块链上，交易完成后，区块中就保存了无法篡改的、不会丢失的交易凭证。

因此相比较传统合约，区块链智能合约存在很多优势。

1）智能合约的条款是由代码确定的。由于代码逻辑明确，比起自然语言，更加不容易产生分歧。

2）智能合约部署到区块链网络，网络节点相互独立，都有一份数据账本，防止合约内容被篡改。合约执行记录也会被保存在区块链上，可作为永久的交易凭证。

3）合约的创建和执行都依赖区块链协议，合约执行的强制力可以保证。

以太坊区块链为智能合约的实现提供了平台，智能合约使得区块链的应用领域更加广泛，已经由最初使用的脚本系统发展为以太坊中的智能合约[36]。

2.4.2 以太坊中智能合约

以太坊中的智能合约是由 Solidity 语言编写，是一种类似于 JavaScript 的面向对象的高级编程语言，简单易懂。该合约被设计运行在以太坊虚拟机上，生成的 EVM 字节码，通过以太坊节点共识验证后上传到区块链上。从某种意义上讲，智能合约也可被认为是一份去中心化的合约。

在以太坊中，合约的智能化指出了区块链不仅包括合约代码，还有存储空间的虚拟账户。合约行为是由合约代码进行操控，账户存储始终保留着合约的状态。由于合约是由事件（交易）驱动的，因此我们需要获取发送者地址、接收者地址、数据区、Gas 值等属性，然后把交易的具体信息封装成事件，事件数据通过事务发送给合约，触发合约的状态机进行判断。若满足状态机触发条件（一个或多个），则会根据提前设定好的数据选择合约自动执行。智能合约概念模型如图 2.6 所示，当事务及事件 A 发送给合约，合约就会对 A 事件进行分析处理，然后根据触发条件，生成不同的事务和事件数据 B，然后将合约的状态和值记录到区块链上。智能合约就相当于裁判，负责判断和执行，可以将任何规则以事件的形式，依据用户的意愿正确、合理的去执行。

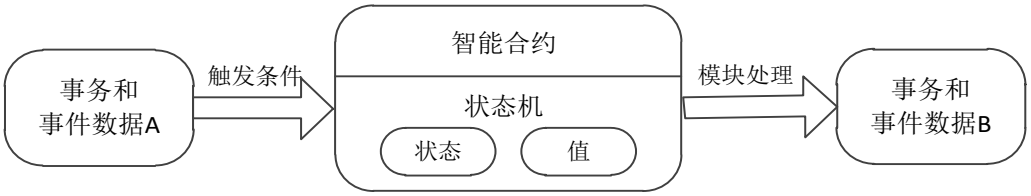

图 2.6 智能合约的概念模型

2.5 本章小结

本章首先描述了以太坊区块链产生原理，对区块链中的交易过程、区块以及区块如何连接成链进行了详细的分析；其次介绍了以太坊中底层分布式网络、交易过程中的数字加密算法、交易中数据如何存储才能保证存储数据安全可信；对当前区块链常见的共识机制 PoW、PoS、DPoS 和 PBFT 共识机制进行简要介绍，并对其性能及应用场景进行分析对比，确定选择 PBFT 机制作为本系统底层区块链的共识算法；最后对以太坊智能合约的概念和技术进行描述，为后续基于以太坊智能合约开发的学位管理系统做铺垫。

3 学位管理系统总体设计

本章对需要开发的学位管理系统做了整体的规划，首先从功能和非功能两个方面分析了系统具体需求，并根据需求设计了系统的整体架构和系统的具体业务模块，这为后续章节的实现提供了设计思路。

3.1 系统需求分析

需求分析的目的是对所开发的系统提出相应的要求和目标，这为后续系统的设计与实现起着重要的作用。本文研究区块链智能合约在学生学位信用体系的应用，目的是以区块链技术为底层框架实现安全、可信、不可篡改的学位管理系统。本文将从系统功能性需求、可用性、可扩展性和安全性等方面进行分析。

3.1.1 功能性需求分析

在去中心化的学位管理系统中，要求系统底层联盟链中各个不同对象之间不需要建立相互信任的关系，不需要第三方参与监管，学位信息就可以被安全执行、数据可被追溯，防止在交易过程数据作假问题。本系统是一个面向学生学位管理的系统，主要涉及学生、学校和教育局三个主体对象，该系统的功能主要包含以下几个部分：

1）系统的登陆退出，用户信息的新增、查询、修改、删除等；

2）用户的角色的区分，针对不同角色查看不同的系统功能；

3）学生模块需要包含学位信息的注册、学位信息查询等功能；

4）学校模块需要包含学位信息授予、学位信息召回及学位信息查询等；

5）教育局模块需要包含已经下发学位的学生，对学位召回的审核等。

同时在以太坊联盟链中，也需要对网络中节点内容进行检测，所以需要提供检测网络节点指标的相关接口。本文分析对以太坊节点中的区块数据、交易数据以及区块状态、交易状态等信息需提供了外部接口，以便交易出现问题可以快速定位。

3.1.2 非功能性需求分析

非功能性需求是确保系统安全、稳定运行的重要组成部分，本系统将从系统的可用性、可扩展性和安全性方面进行分析。

（1）可用性

本系统使用的用户有学生、教师和教育局管理人员，涉及多个年龄段的人群，因此系统界面简单友好，用户易操作，也可快速熟悉本系统。

（2）可扩展性

在学位管理系统开发完成之后，需要对数据或者其他信息留有接口，以便后期针对用户需求对系统进行扩展。

（3）安全性

学位管理系统需要存储大量的学生学位信息，因此要确保基本数据的安全性，主要有以下两部分组成：

数据交易过程对的安全性：在学位信息交易过程中，需对交易数据进行加密保护，本系统采用 ECDSA 算对数据签名验证，以防在交易期间数据被更改的可能性。

系统的安全性和可靠性：学位交易数据只有当网络中所有参与节点通过验证，并达成一致，才能存储到底层区块链中，保证了系统的可靠性。且存储到链上数据无法被篡改，保证了数据在系统中存储的安全性。

3.2 系统总体设计

在以太坊中，利用智能合约实现的系统应用称为去中心化应用（DAPP），DAPP 中的智能合约不仅是实现前端页面与底层区块链之间交互的桥梁，也是系统业务按照既定规则执行的保障。以太坊中提供了前端应用程序与后端智能合约代码交互的接口，为在以太坊区块链上快速开发去中心化应用奠定基础。

本小节内容通过对系统整体架构和业务功能两个方面对本系统的总体设计进行简要阐述说明。

3.2.1 系统整体架构设计

本系统是基于智能合约在以太坊平台上实现的去中心化应用学位管理系统，如图 3.1 所示，整体架构主要分为三层：底层服务、核心技术层和顶层服务应用。

1）底层服务：底层的物理节点服务器。本文采用 PBFT 算法至少需要 4 个节点方能达成一致，因此本文选取 6 台服务器可以构成分布式网络节点。

2）核心技术层：根据学位管理系统中的业务特性，对以太坊平台开发，涉及到的模块主要有以下几个层次。网络层：包含分布式网络、消息传播机制、数据验证机制等，确保节点之间安全交互。PBFT 共识算法层：实现 PBFT 共识算法，该算法可以确保全网不同节点中的区块数据一致。区块链数据层：定义交易数据结构、区块数据结构和共识数据结构。以太坊虚拟机（EVM）：智能合约代码是在 EVM 中运行，合约通过 EVM 执行来更改区块链上数据状态。

3）顶层服务应用：根据学位管理系统中的业务逻辑，分为合约层和应用层。合约层：利用智能合约实现学位管理系统中的业务逻辑合约和其他基础逻辑合约。去中心化应用层：用户与区块链系统交互的接口。该层需要提供前端页面，以太坊中通过调用 Web.js 与智能合约实现业务上数据的交互。同时在本系统前端页面中设置了学生密钥，学位注

册完成后页面中有学生密钥的生成，该密钥不在区块链公开，保证学生密钥的私密和安全。

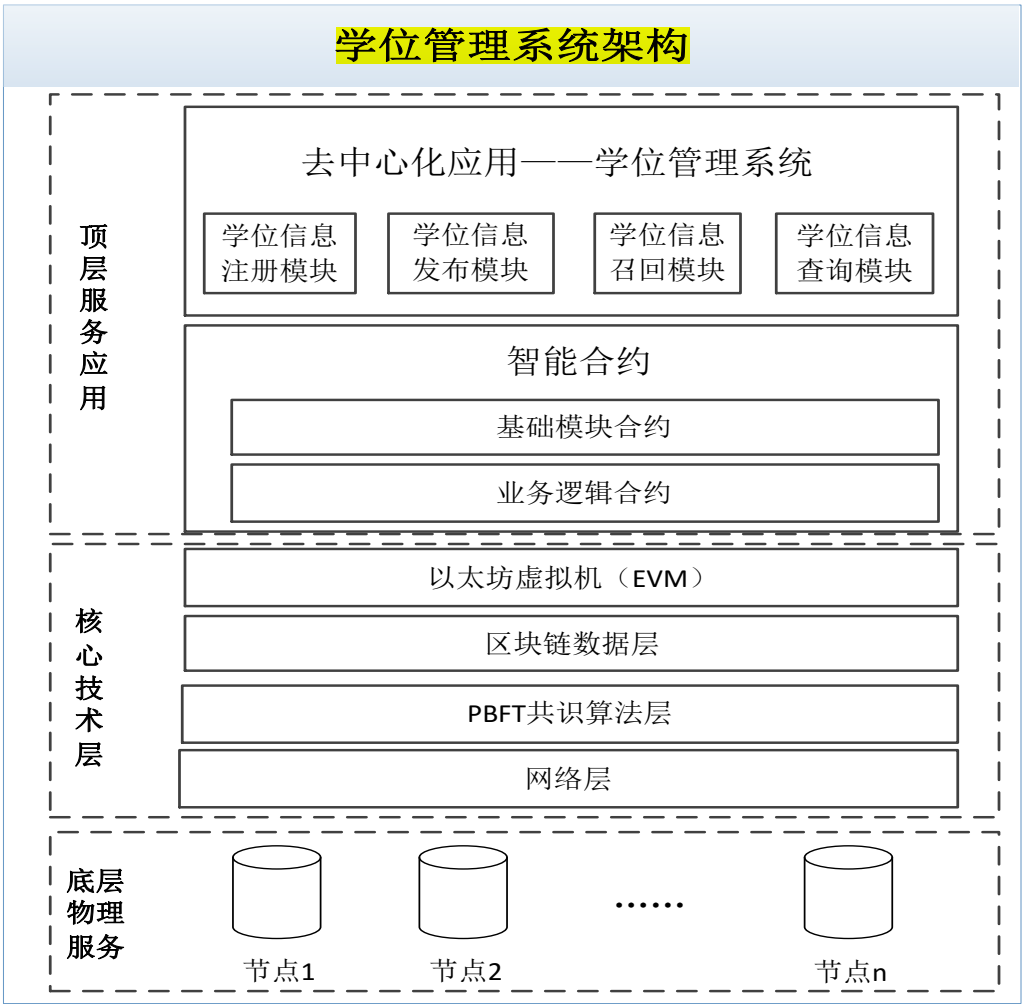

图 3.1 系统整体架构

3.2.2 系统业务功能模块设计

本系统将基于以太坊智能合约实现学位管理系统，该系统功能需求主要针对系统中学生、学校和教育局三类对象进行设计。学位注册模块主要针对即将毕业学生，通过该系统申请获取学位。学位发布模块是学校收到学生的学位申请，对学生学位进行审核操作，若学位信息合理则授予学位。学位召回模块是已获取学位的学生，若学位存在问题，学校可进行召回操作，并发送至教育局，教育局需要对学校提交召回操作的学位信息进行审核。学位查询模块是学生成功获得学位后，可查询得到自己的具体信息。由于区块链技术的特殊性，增加了交易记录查询、区块记录查询业务模块，其根据目前学生学位状态，对学生信息进行查询显示。系统的业务功能如图 3.2 所示：

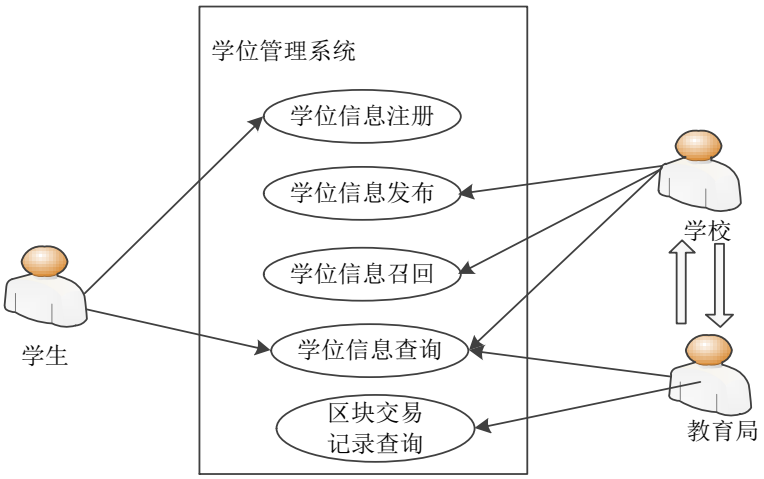

图 3.2 学位管理模块用例图

3.3 本章小结

本章节针对学位管理系统的业务场景，对系统的功能性需求和非功能性需求进行分析，同时设计了系统的总体框架，并分析了系统的业务功能模块。在该框架中，可知智能合约是系统前端页面和底层区块链交互的桥梁，这为下一步智能合约与系统业务的交互、合约与区块链交互的实现奠定了基础。

4 智能合约在学位管理系统中的设计与实现

本章节是整篇论文的核心。该系统在以太坊平台搭建联盟链，以学位管理作为应用场景，实现系统的去中心化。系统中的交易依托智能合约来执行，账户信息由区块链维护。因此本章首先对智能合约进行设计与实现，其次对系统中的核心算法设计实现。最后，将合约部署到区块链上，并完成整个系统平台的设计与实现。

4.1 学位管理系统业务流程

依据学位管理系统总体需求分析，设计了如图 4.1 所示学位管理系统的详细业务流程。

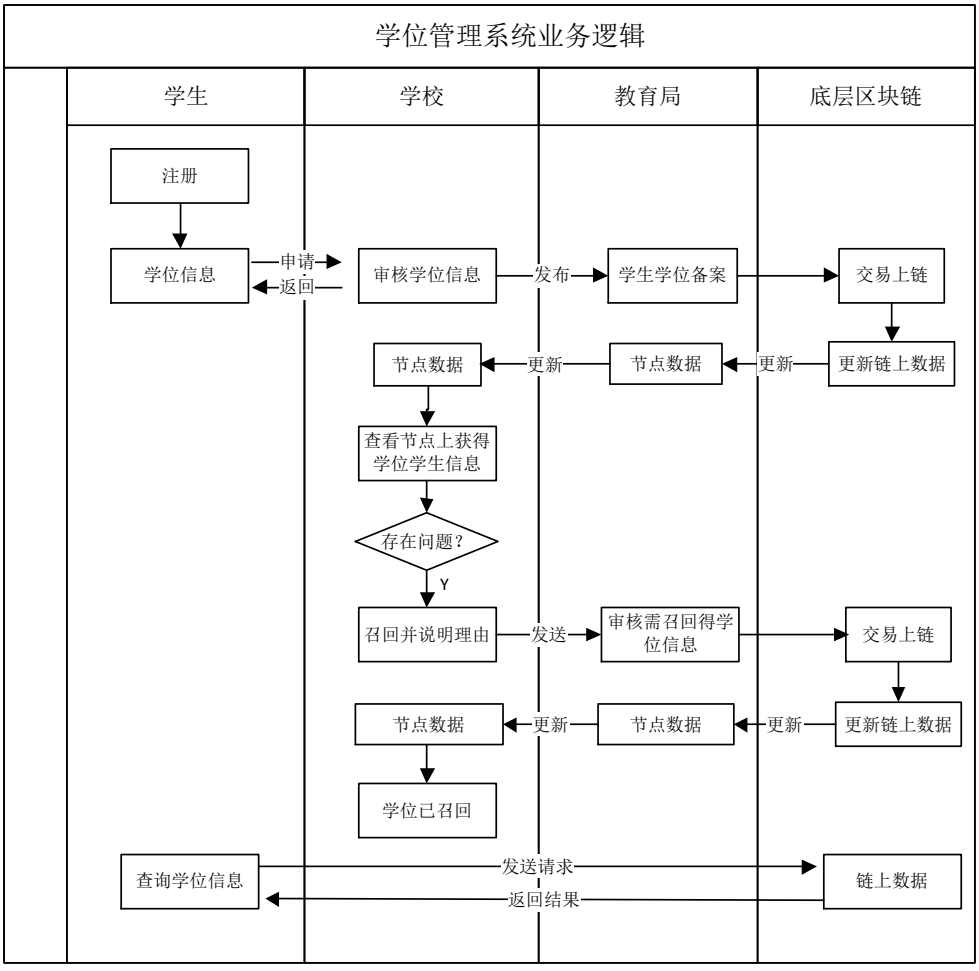

图 4.1 学位管理系统业务流程图

在系统中，学生首先会根据自己学习情况，判断自己是否满足申请学位条件，满足的话填写学位申请信息的相关内容，学校会收到来自学生学位申请的数据，通过审核学生在校学习情况的数据，无任何问题则学校对该生授予学位，从而实现一次学位发布。

当学校再次检查学生信息并发现遗漏问题，会对学生的学位进行召回并附上召回理由，提交至教育局，教育局认为召回理由正当，同意学校申请，则完成一次学位召回过程。系统为各企业机构提供查询已毕业生学位信息接口，可验证学位信息。整个过程信息流动在各个账户（学生、学校和教育局）之间，且每一个学位信息交互过程都可以被认为是账户之间的交易过程。

上述业务逻辑分析表明学位交易业务可以利用区块链和智能合约来实现。论文中对学位交易的流程进行了简化设计，这对使用智能合约实现学位管理业务的可行性更高。除此之外，在学位交易过程中产生的交易数据，是由网络中所有节点达成共识存储到区块链上，确保数据的不可篡改、安全可信。同时智能合约实现的学位管理系统，不需要中间机构审计就能保证交易的安全性，节省了大量经济成本。因此论文中将区块链和智能合约应用到学位管理系统中，其研究价值和现实意义较大。

4.2 智能合约详细设计与实现

4.2.1 智能合约运行框架

智能合约是部署在以太坊区块链中可自动执行的合约程序，主要作用就是在缺少第三方监管的情况下，智能合约能够同时运行在全网络所有节点，任何企业和个人都无权干涉，且存储到区块链数据库中的交易数据都无法篡改，并且可以追溯，保证了数据信息的公开透明与安全。

由于以太坊平台中内置图灵完备脚本语言的虚拟机，为部署在区块链上的智能合约自动执行复杂业务提供运行环境。虚拟机作为沙箱被封装起来，整个执行环境都被完全隔离，可以控制合约对程序、文件等访问权限，还将合约与合约之间隔离存储，从而限制合约的访问权限，避免因合约未授权进行访问和修改，保证合约之间交互的安全性。

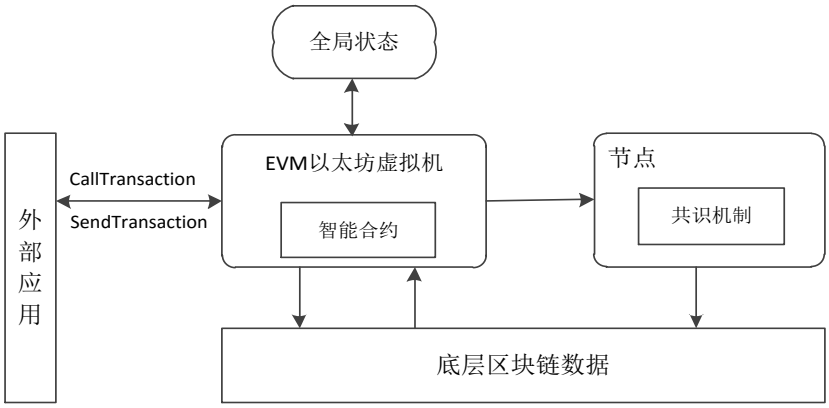

图 4.2 智能合约运行框架

智能合约与区块链整体框架，如图 4.2 所示。由于以太坊平台以 JSON-RPC 方式提供了外部应用访问区块链平台的操作入口，因此可通过 JSON-RPC 进行接口调用智能合

约，以太坊网络中所有节点收到调用智能合约的交易后，会在本地 EVM 通过合约地址和合约接口（ABI）找到并调用部署在区块链上的合约代码，运行完成之后，各节点之间会相互验证自身的运行结果，当所有节点通过共识机制达成一致，会将运行结果写入到底层区块链上[39]。

4.2.2 智能合约部署调用

以太坊是一个基于区块链技术的去中心化应用平台，在这个平台上，用户可以根据自己的业务需求设计智能合约。为确保在以太坊环境中智能合约的安全性，以太坊官方社区开发了 Solidity 智能合约的集成开发环境（IDE）：Remix（也叫 Broswer-solidity），是一种基于浏览器的 Solidity 编译器和集成开发环境，也提供了比较常用的开发框架 Meteor 和 Truffle。

由智能合约的运行原理可知，发送交易触发智能合约从而将智能合约部署到以太坊区块链中。如图 4.3 所示，详细展示了智能合约的部署和调用过程。

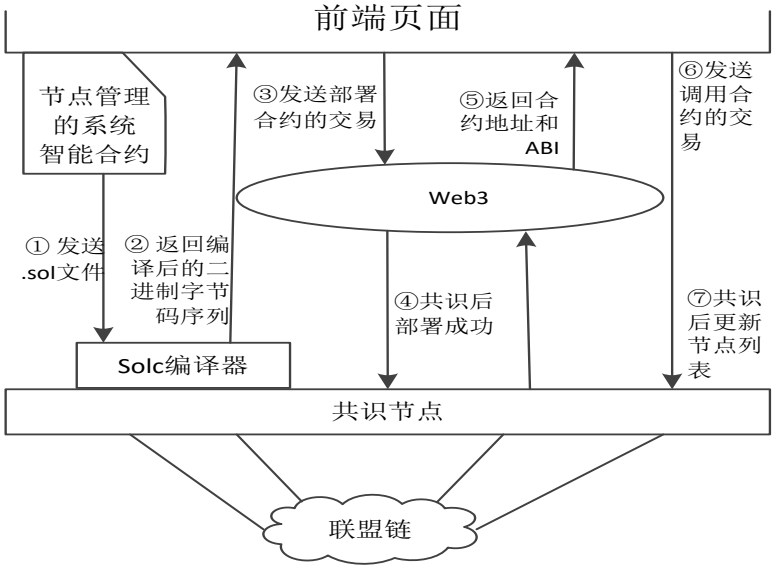

图 4.3 智能合约的部署及调用过程

将 Remix 编写的智能合约发送到 solc 智能合约编译器中，编译成 EVM 字节码作为交易的合约代码发送给共识节点。所有节点通过共识算法对交易的合约代码进行安全验证，达成一致后在 EVM 虚拟机中运行合约代码，这时合约就成功部署并存储到以太坊节点网络中，部署成功后，会返回部署成功的合约地址 Address 和二进制接口 ABI。浏览器可以通过调用智能合约地址 Address+ABI，发起一笔交易，调用部署在区块链上的合约，由于智能合约存在监听 Event 事件机制，状态机会根据触发合约的交易状态，将区块交易存储到链上，节点会根据交易接收方地址调用合约，智能合约代码分布式地运行在网络中每个节点的以太坊虚拟机中，当节点达成共识后更新联盟链网络节点数据，

从而完成浏览器与区块链数据库交互。

在功能丰富、逻辑强的智能合约中，除了通过发送交易的方式对以太坊区块链中的状态变量修改及进行序列化操作外，还允许合约在执行过程中进行消息调用，即通过创建一条“消息”的方式来调用其它合约，合约间消息调用过程如图 4.4 所示。

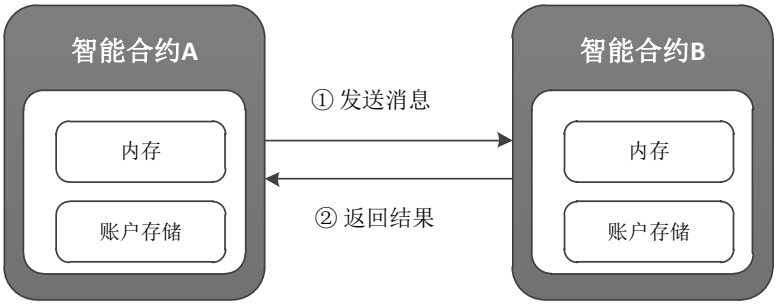

图 4.4 智能合约间消息调用

智能合约 A 创建一条消息发送给智能合约 B。消息的结构与交易类似，都由发送者地址、接收者地址、数据区、Gas 值等属性组成，但消息的调用属于交易执行的一部分，只对区块链上数据进行操作，不会改变合约中原有状态变量，也不会在区块链产生一条新的交易记录。当智能合约 B 收到消息后，就访问消息的数据区获取调用参数，执行合约代码，最终将结果返回给智能合约 A，并保存在智能合约 A 预先分配的一块内存空间。

4.2.3 智能合约详细设计

以太坊中，可以使用 Solidity、Serpent、LLL 等多种语言来编写智能合约，但 Solidity 语言使用较为普遍。Solidity 语法类似于 JavaScript 语言，其优点是规则相对简单，容易理解，但弊端也不少。比如：EVM 是基于栈的虚拟机，栈的最大长度为 1024，每个栈元素占 256 位，导致栈内存储受限；key-value 数据类型不能遍历，且无内置的函数库，不能直接被调用；区块中的 Nonce 值根据区块状态的改变随时变化，获取该值会影响系统编译速度和运行速度。

基于上述分析，考虑到合约的安全性和 Solidity 语言的弊端，根据以下几个原则设计出基础合约、数据合约和业务合约三个层次的智能合约。

1）合约中业务处理逻辑简洁，避免合约复杂，造成合约出错的风险；

2）尽量避免合约的外部调用，以防止出现 DAO 攻击，导致合约安全性低；

3）确保合约和函数模块化，数据存储与业务逻辑处理的合约分开设计，降低合约升级难度和应用数据迁移的难度。

基础合约：由于系统中的分布式网络节点进入退出系统，且用户较多，不同用户的权限不同，因此设置节点动态管理模块和角色权限管理模块。

数据合约：包含学位注册信息、学位发布信息、学位召回信息中的数据结构及类型，以及底层数据类型之间相互转化工具。

业务合约：实现学位管理系统中的业务，如学位发布、学位召回等。

（1）基础合约

本文基于以太坊联盟链平台实现，在智能合约中，合约可以根据系统所需特定功能来设计数据与逻辑业务。网络中允许节点加入或退出，为此添加了节点动态管理模块。系统业务中包含了三个不同对象，拥有不同的权限，为此设计了角色权限管理模块。本课题为这两模块设计了节点动态管理合约和角色权限控制合约，统称为基础合约。

1）节点动态管理合约：检查区块链网络中加入或退出节点，判断其是否被授予权限。公有链中的节点不需要任何人授予权限即可自由加入或退出，私有链和联盟链中的节点需要授予一定权限许可才可加入或退出。由于本系统基于联盟链开发，因此网络中参与共识过程的节点受到限制。节点动态管理合约中定义了该节点信息以及权限信息的结构体，并对外提供节点信息更新的接口。在分布式网络中，当有新的节点加入时，管理员会发起增加节点的交易，网络中其余节点确认该交易有效，达成共识后，该节点方可加入；当有节点退出时，管理员发起取消该节点的参与权限，广播到网络，其余节点达成一致，将结果存储到区块链中，节点将会从全网中断开。

2）角色权限控制合约：在交易被节点矿工打包进入区块前，判断用户是否有权执行这笔交易。由于系统中存在学生、学校和教育局三种不同的用户主体，因此，设计角色权限控制合约来划分权限。当用户发送一笔交易，需要调用智能合约，网络中的节点根据自身信息配置（动态管理方式）判断该用户是否有权调用。在该过程中，涉及到角色分组合约和筛选合约。角色分组合约存储了主体的权限信息（key：函数执行，value：权限信息），筛选合约中包含用户和用户分组的对应关系（key：用户地址，value：用户所在分组合约地址），只有当交易通过所有的筛选合约，才能输出该笔交易的执行结果，如图 4.5 所示。

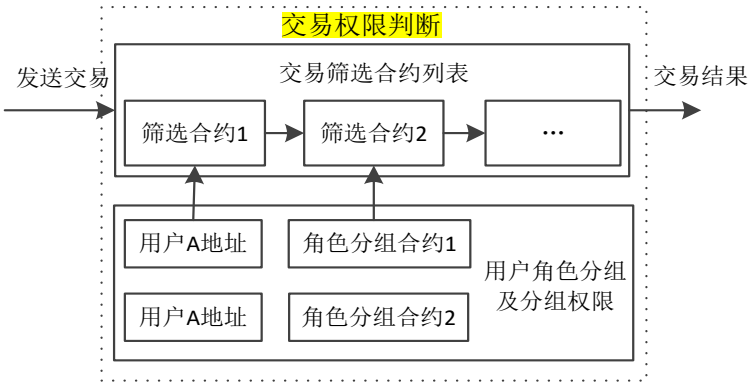

图 4.5 角色权限控制合约逻辑

矿工节点将用户发送的交易打包到区块前，需要对交易权限进行判定。应用程序二进制接口 ABI 将区块链外部与合约进行交互，因此 User（用户）将交易发送到 ABI 接

口合约中，将交易数据（From、To 和 data 等字段）转为二进制格式，依次调用执行交易中的筛选合约，筛选合约根据用户地址调用角色分组合约，查询该分组的权限，若获得到 User 权限，通过所有的过滤合约，才能将交易打包到区块，且分布式节点对结果达成一致，从而添加到区块链中。用户与角色权限控制合约的交互流程如图 4.6 所示。

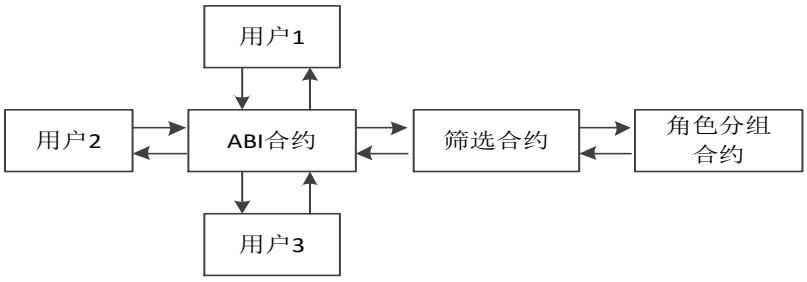

图 4.6 角色控制合约交互流程

（2）数据合约

Solidity 语言中有多种不同类型数据结构，当系统中涉及到多个合约并且合约中含有共同代码，则可将共同代码部署成一个数据合约。本文中的数据合约主要包含通用数据结构、数据类型转换和遍历算法，部署数据合约到以太坊区块链中只需操作一次，并通过执行 EVM 中 DELEGATECALL 复用合约代码。数据合约的作用如下所示：

1）根据对学位管理系统业务分析及语法规则，定义学生数据的结构体（如学生注册信息的数据结构、学位发布的数据结构和学位召回的数据结构等）。

2）根据数据结构体设计保存学生数据的合约，定义存储学生数据信息的 Map，实现创建学生信息、审核学生信息和查询学生信息的函数。

3）设计通用的底层算法，前端页面会不断与合约进行交互，涉及数据格式转换，需建立数据类型转换（如十六进制转换成 String）和遍历 Map 算法。

本文中的数据合约可以归类为业务合约中的一部分。用户发送交易会调用业务合约，业务合约可以通过 using for 来指令调用数据合约，调用一次数据合约后，数据合约会被部署到区块链中，之后调用业务合约就会自动调用数据合约，实现业务功能。

（3）业务合约

交易数据是区块链上最基本的数据[41]，也是智能合约中业务合约之间交互的最基本数据，本文业务合约依据第三章学位信息注册、学位信息发布、学位信息召回和学位信息查询的流程进行设计，将信息交易业务划分为用户合约、审核合约、管理员合约以及数据合约（已定义，包括数据类型结构存储和数据类型转换等）。

在业务合约中，主要包含以下几个合约，其主要功能如下：

1）用户（User）合约：不同类型用户在系统中拥有不同的操作权限，每个类型的用户都有相应的用户合约，因此，同一类型的用户只能调用对应的用户合约。学生用户可

在系统中注册个人信息和学位查询操作，学校用户可在系统中进行学位发布、学位召回等操作，教育局用户可在系统中审核学位召回操作。

2）审核（Audit）合约：在审核信息列表中，罗列全部学生学位信息。学位信息需由 key-value 形式的键值去存储数据。学生通过注册函数发送申请学位信息的交易，调用审核合约后显示高等院校及教育局对学位信息的审核详情，高等院校和教育局核对学生信息的正确性，并通过调用审核合约修改当前学位信息审核的状态。

3）管理员（Admin）合约：目前合约中主体对象为学生、各大高校、教育局。为确保整个系统的安全管理，将教育局作为一个管理员，并设置管理员合约管理整个交易系统。学校节点的动态增减需征得管理员同意方可进行。

以上分析了业务合约中各合约的功能及涉及的操作，可知合约与合约之间并不是独立存在的，本文将对业务合约的交互过程做以介绍。

由于管理员是系统中拥有最高权限的角色，因此搭建底层以太坊区块链环境配置时，在配置文件中设置管理员的最高权限，以便管理员能够在新增用户、审核信息、学位发布和召回等操作进行数据信息确认。由于以太坊中外部账户（以下简称“账户”，EOA）是操作以太坊的一把钥匙，每个账户都是由公私钥定义，并以地址为索引。钥匙对的编码存储为 JSON 格式的私钥文件，私钥可以对发送的交易进行加密，用户只有同时拥有钥匙文件和密码，才能执行交易。管理员合约与 EOA 进行交易的流程，如图 4.7 所示。

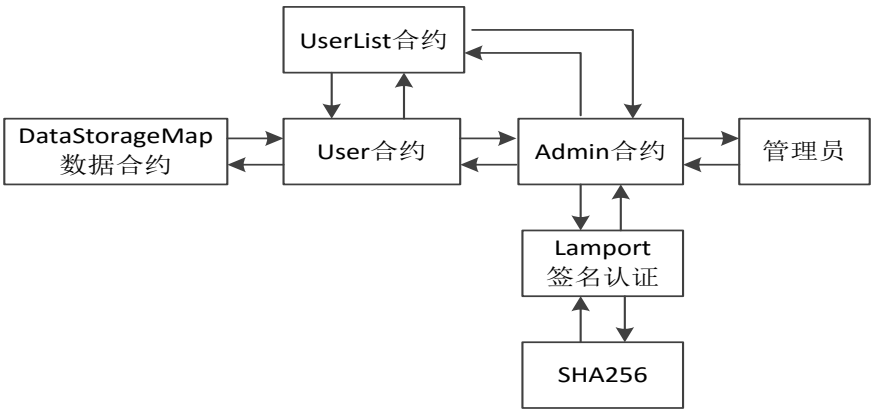

图 4.7 管理员合约的交互流程

管理员新增 User 时，首先该笔交易由私钥控制 Admin 中外部账户（EOA）调用 Admin 合约，在这个过程中，在 User 合约中设置 UserID、UserSort、UserPower 和 Address 的用户数据结构信息。此时 Admin 合约根据 User 合约中的 UserID、UserSort、UserPower 和 Address 与 UserList 合约对照，判断该用户是否已存在，如果这个用户不存在，将信息新增到 UserList 合约列表中。

学生通过学位信息注册将信息存储到学位信息列表中的交互过程为：学生学位信息注册时，该笔交易首先会调用 Admin 合约，根据学生学位注册的权限信息核查

StudentUser 合约，获取学生用户 Address，StudentUser 合约调用 ID 管理合约获取学位信息地址。成功获取后，StudentUser 合约调用数据合约 DataStorageMap，最终将学位信息写入学生信息列表中，具体流程如图 4.8 所示。

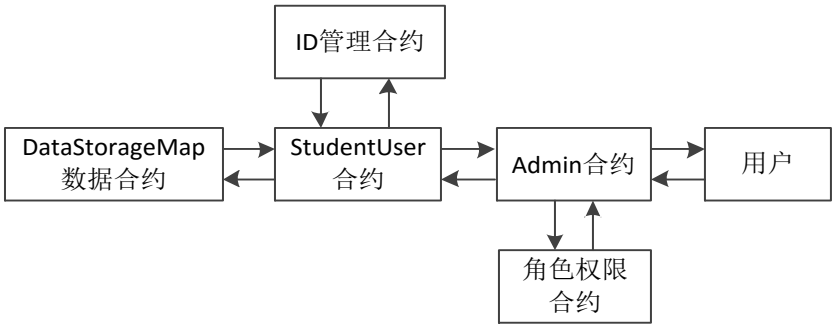

图 4.8 学位注册合约交互流程

学位发布过程中各合约的交互过程。成功注册到该系统的学生学位数据，需要经过学校审核学位信息后才可发布。学位发布过程中，合约间的交互过程为：

根据学生学位的注册信息，发送交易，触发外部账户调用 StudentUser 合约，StudentUser 合约将学生的学位信息发送给 SchoolUser 合约，SchoolUser 合约调用 SchoolAudit 合约审核学生的学位信息，无误后为该信息生成新的审核信息 ID，并将审核结果的状态重新插入到学生信息列表中。SchoolAudit 合约触发审核状态更改事件，返回学生学位信息地址给 StudentUser 合约和 EducationUser 合约，确认学位发布请求成功的消息返回到 StudentUser 和 SchoolUser，如图 4.9 所示。

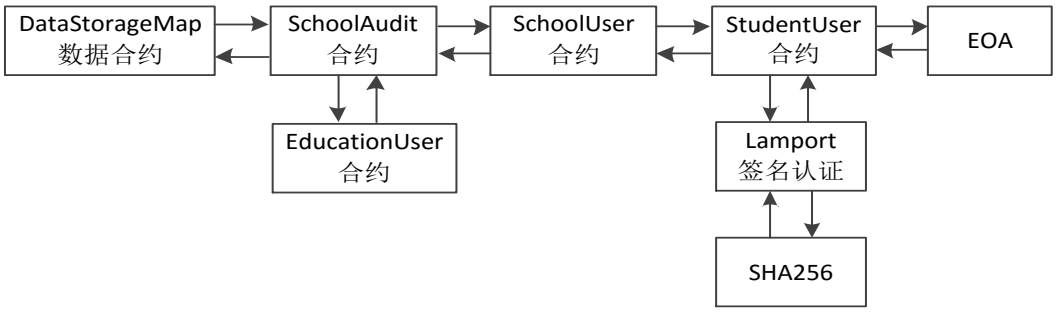

图 4.9 学位信息发布合约交互流程

学位召回同学位发布一样，合约之间进行交互。当学校发现已毕业的学生其毕业条件存在问题需要召回其学位证书时，其召回过程为：

SchoolUser 发出召回学位请求交易，触发 SchoolUser 合约，获得召回学位的权限，SchoolUser 合约调用 SchoolAudit 合约再次核对学生的学位信息，信息有误则执行召回操作，并生成新的审核信息 ID。SchoolAudit 合约更新审核学位信息中的召回状态，并将结果返回至 EducationUser 合约。 EducationUser 合约触发 EducationAudit 合约更改学生学位的召回，最后将召回学位信息审核结果返回至学生地址中，如图 4.10 所示。

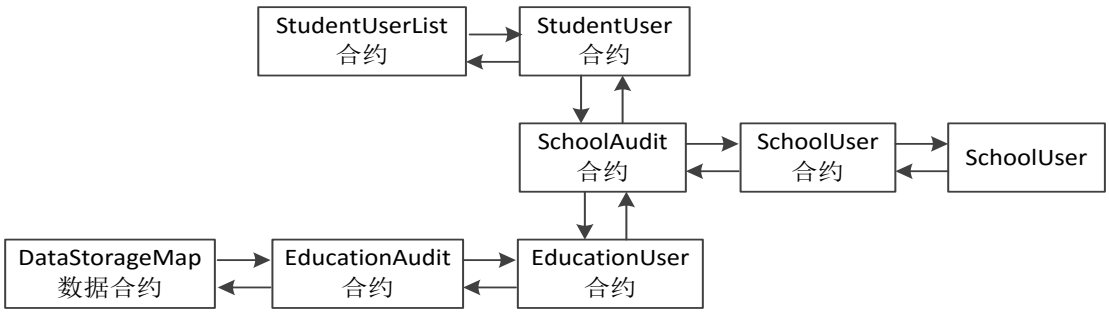

图 4.10 学位信息召回合约交互流程

4.2.4 合约安全性分析

区块链技术拥有无法篡改、数据透明的特点和永久运行的安全环境，完全符合智能合约运行自身逻辑业务的条件。近年来，智能合约的出现促使区块链技术在各大应用领域中展示出自身的价值。通过智能合约控制业务系统流程，自动触发相关操作，减少人为交互，提高了系统运行效率。正是由于智能合约在区块链中扮演着举足轻重的角色，智能合约也成为区块链安全事件的重灾之地。

- 2016 年 6 月 18 日，The DAO 众筹项目中，一技术黑客发现重复调用 splitDAO 函数可以从项目资产池转移出资产分给自己，这一重大漏洞导致 360 万个以太币丢失，价值逾 5000 万美元。
- 2018 年 4 月 22 日，BEC（美链）因为合约中 BatchTransfer 函数导致数据溢出的漏洞，导致大量 BEC 代币被盗，65 亿市值犹如泡沫般迅速归零。

虽然区块链技术不断在推进，智能合约自身的合约逻辑漏洞也逐渐在消除，但仍需进一步的改进和去除漏洞和陷阱。目前已知的漏洞和陷阱有私钥泄露、数据溢出、可重入性和竞争条件、Nonce 顺序依赖合约、RAM 漏洞和误操作异常等。常见的智能合约项目及合约中存在的漏洞如表 4.1 所示。

表 4.1 智能合约项目漏洞

智能合约项目	漏洞	漏洞描述
The DAO	重入性攻击	递归调用 splitDAO 函数
Random 项目	Nonce 依赖	矿工提前获取
SMT	数据溢出	输入参数可被攻击者操控
EOS	RAM 漏洞	嵌入垃圾信息到 row 锁定 RAM

由上表知，智能合约的漏洞总是以各种新颖的方式出现在世人面前，识别和消除合约中所有潜在的漏洞永远无法实现。与传统 IT 系统不同，智能合约的优势在于利用程序算法代替人为仲裁去执行合约，根据区块链的数据透明、不可篡改、永久运行特性，说

明合约一旦成功部署到区块链上是无法进行修改的。因此，在以太坊区块链网络上正式发布智能合约前，需要资深专业的程序审计人员去测试和验证合约代码中业务逻辑，只有确保合约中规避了目前已存在的漏洞和可预知的陷阱，才能将合约部署到链上，以保证智能合约在市场应用中的安全运行，避免经济损失。

4.2.5 验证结果及分析

在智能合约设计中，规避安全漏洞的方法可以由专业的人才去审计，但合约数量呈指数增长，导致人工审计合约成本增大，为此区块链开发人员和各企业相互合作探索出能够使用机器辅助验证的工具。目前区块链产业中形式化验证工具有链安科技的 Vaas 平台和 Certik 平台。由电子科技大学创建的链安科技形式化验证 Vaas 平台是全球第一个高度自动化的形式化验证平台，将已经编写好的合约代码导入该平台进行检查，即可自动生成合约中存在的安全漏洞报告提示。Certik 平台是哥伦比亚大学和耶鲁大学联合开发的，通过描述规范的逻辑语言和严谨的数学推演，是一种检查合约满足安全要求的形式化验证平台。

智能合约一般都是在 Remix 工具上开发，验证合约的工具最好可以集成到 Remix 工具中进行开发，因此本系统选择的验证工具为 Oyente。该工具能够使开发人员边编写代码边编译合约代码，然后通过自动化分析引擎检验代码合约是否存在漏洞，避免调用合约时出现的漏洞。Oyente 工具是基于符号执行的自动化验证，验证流程如图 4.11 所示。

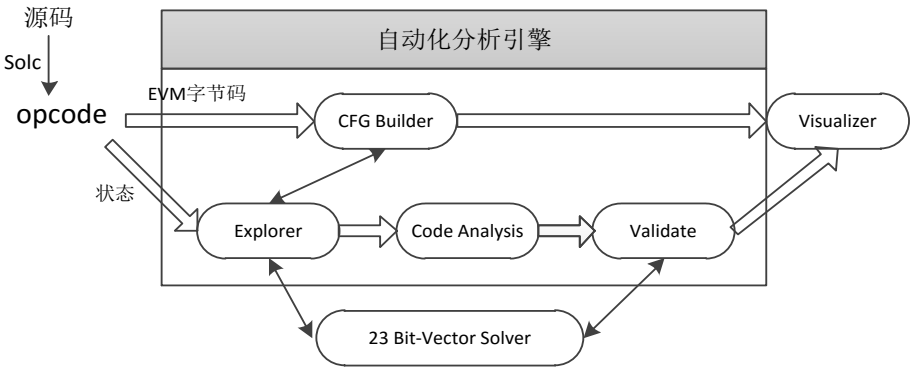

图 4.11Oyente 架构

智能合约编译后转为 EVM 字节码，首先会通过 CFG Builder（分叉执行）判断逻辑代码中是否存在执行验证符号，如果存在，Oyente 工具会将合约代码通过 EXPLORER 将逻辑流程全部验证，验证结果发送到 CODE ANALYSIS 进行合约逻辑的安全性检测，将检测出的合约漏洞通过 VALIDATOR 输出。

Oyente 工具与 Remix 开发工具集成后，如图 4.12 所示。Remix 开发界面中，界面左侧为合约文件列表；中间上方为合约代码的编辑器，下方为交易回执详细信息；界面右侧为调试工具栏。调试工具栏可以编译、执行、测试代码和分析代码，涉及漏洞问题的

代码会在代码分析（Analysis）中给出这些提示并准确捕捉合约中漏洞。

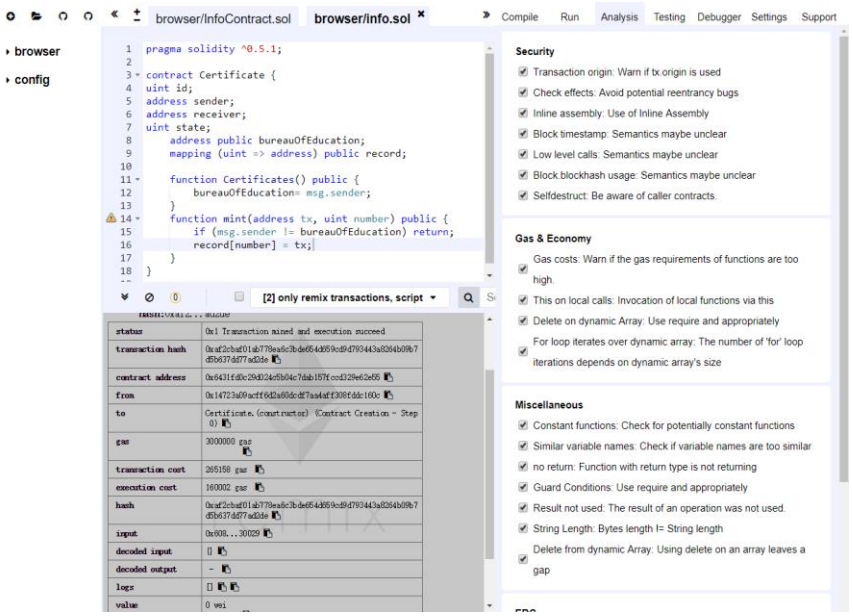

图 4.12 智能合约安全性检测

该系统中的合约程序借助 Oyente 工具检测出关于 User 合约中存在可重入的安全漏洞。学校对学生信息审核操作，先调用 studentList 函数获取学生列表，再调用更新学生学位信息状态函数 updateStudentInfo，之后会调用 SchoolAudit 合约。如图 4.13 所示，可以看出合约的外部调用过程出现了可重入性攻击漏洞。

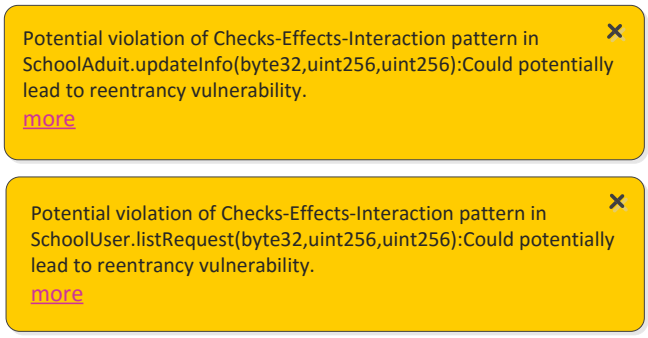

图 4.13 业务合约安全性检测结果

一个合约代码在执行过程中被打断，并且一次调用还未完全结束之前再次回调自身函数，这样就产生了可重入性漏洞。如上述检测结果显示，SchoolUser 合约在查询学生列表 studentList 时，通过调用 StudentUser 合约显示学生信息，之后 SchoolUser 合约调用 SchoolAudit 合约将审核后的学位信息插入到学生列表中，SchoolUser 合约在等待 SchoolAudit 合约执行完成前，SchoolAudit 合约又重新调用 StudentUser 合约，获取到 StudentUser 合约中之前的状态，导致重入攻击。合约中其它函数也存在相同的情况。

针对之前提到的以太坊智能合约开发时的安全陷阱，智能合约需解决系统中的安全性问题、可靠性问题和易用性问题，因此结合以上信息给出保证合约安全的建议。

1）使用 Checks-Effects-Interactions（检查--生效--交互）模式：首先判断相关操作权限，进行安全性检查，然后根据条件改变合约中状态变量，最后与其它合约进行交互。

2）代码轻巧模块化：尽量避免合约中一个函数设置太复杂的逻辑关系，复杂函数拆分为多个函数，工具性的逻辑封装成 Library。

3）使用 Fail-Safe（异常--安全）模式：在合约中增加一个自检查函数，当合约出现异常情况，能尽可能保障合约中数据的安全。

4.2.6 智能合约实现

在该学位管理系统中，以太坊区块链结合智能合约根据学位信息的实际业务进行实现。智能合约的实现模块中，利用 Solidity 语法规则，将系统中学位信息的交易数据（注册信息数据、发布数据和召回数据）等在数据合约中定义数据结构体，在业务合约中调用数据的存储 Map 类型，Map 数据结构如下：

```
mapping (uint => address)record
```

在该 Map 数据类型中，键值对（key-value）分别对应数据结构中的 uint（学生学位信息的存储地址 address）和 record（存储学位信息的数据）。

结构体定义完成之后，分析和设计智能合约在学位交易业务中的流程，主要是学生将学位信息进行注册并提交至学校，学校审核完成之后将审核结果返回给学生本人和教育局。学生可以根据交易地址 address 在查询接口中查看自己目前的学位信息状态。合约中学位发布算法流程如 4.2 所示。

表 4.2 学位发布算法

算法：学位发布算法
输入：地址 address
输出：交易请求处理是否成功
Begin：
1. if 已确认请求列表中存在输入的地址 address
2. break，处理失败
3. else
4. 实例化 SchoolAudit 合约对象
5. 调用 SchoolAudit 合约审核信息函数 auditInfo，返回审核结果
6. if true
7. return 审核通过，则学位发布成功
8. else
9. 将信息重新返回至 StudentUser 合约对象
10. return 未通过审核
End

用户学位发布成功后，若学校确认学历信息存在问题，需要召回时，学位召回算法流程如表 4.3 所示。

表 4.3 学位召回算法

算法：学位召回算法
输入：学校 ID，需召回学生地址 address，召回理由
输出：交易请求处理是否成功
Begin：
1. if 已确认请求列表中存在输入的地址 address
2. break，处理失败
3. else
4. 实例化 SchoolUser 合约对象和 EducationUser 合约对象
5. 调用交易双方 User 合约中处理请求函数 verifyList
6. 根据学生地址 address 获取学生学位详细信息 StudentInfo
7. 填写需召回学生学位理由 reason
8. SchoolUser 合约将 StudentInfo 发送到 EducationUser 合约对象
9. EducationUser 审核信息函数 auditInfo，返回审核结果
10. if true
11. 修改合约状态
12. return 审核通过，则学位召回成功
13. else
14. 将信息重新返回至 User 合约对象
15. return 未通过审核，学位召回失败
End

4.3 核心算法详细设计与实现

针对学位管理系统中，设计了增加动态节点增减算法和加密签名算法等技术的实现。同时本文共识机制使用 PBFT 算法，通过投票对节点达成一致，对区块达成共识过程不需要浪费过多的算力和时间。

4.3.1 P2P 网络动态节点增减算法

本系统是基于联盟链基础设计，在上一小节中，介绍了动态节点管理合约，利用合约控制各成员节点进出联盟网络，实现动态节点增减管理。在该过程中，需提前在配置文件中设置参与共识过程的节点参数，配置文件和智能合约相辅相成，共同实现节点管理。为此设计实现了动态节点增减的算法流程。动态节点增加（删除）主要处理函数如图 4.14 所示，其主要流程如下：

1）获得新增（退出）权限节点向全网所有节点发送请求；

2）接收到添加（删除）请求节点的管理员确认同意该节点增加（退出），然后向全网广播 AddNode（DelNode）消息；

3）当所有节点收到 2*f*+1 个 AddNode（DelNode）消息后，该节点更新连接信息，连接（断开）与请求节点的链接；并在连接（断开）后向全网广播 AgreeUpdateN 消息；

4）当节点收到 2*f*+1 个 AgreeUpdateN 消息后，更新节点系统状态。

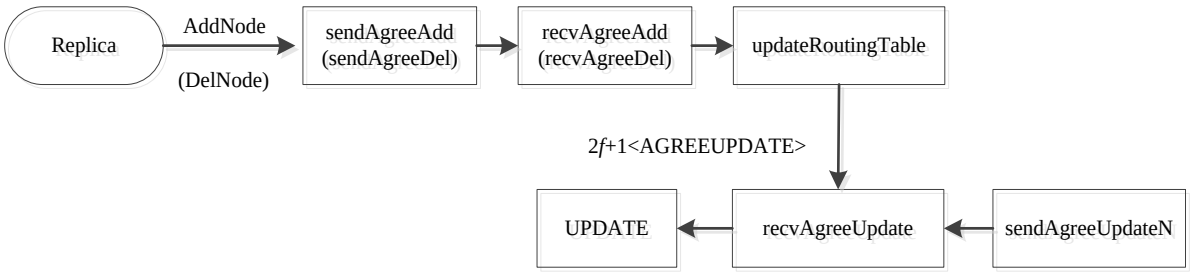

图 4.14 动态节点加入退出函数调用

4.3.2 交易数据签名验证算法

以太坊区块链中的数据是不可篡改的，在数据存储到链上之前，以太坊网络中的节点都必须对生成的每笔交易（Transaction，Tx）的有效性（如数据结构、数据内容）进行确认。若该 Tx 有效，将 Tx 打包到区块中，获得记账权节点广播区块到全网等待其它节点验证有效性（如区块头的 Nonce、StampTime 和 Data），全部节点达成一致，将区块数据添加到链上。在这一系列过程中，使用安全有效的公钥密码系统对交易信息进行签名认证，能够抵抗选择明文（密文）的攻击，从而保证交易安全。

本文采用数字签名算法是 ECDSA，该算法中 *d* 作为私钥，*Q*=（E_p，*G*，*n*）作为公钥。签名者利用私钥 *d* 对交易消息 *m* 进行签名，签名算法如表 4.4 所示。

表 4.4 生成数字签名算法

算法：生成数字签名算法
输入：公钥 *Q*=（E_p，*G*，*n*）
输出：数据的签名信息
Begin： 1. i=2 2. while(i<n-1) 3. d=rand(1，n-1) //选择一个随机整数 4. Q=d*G=(x_1，y_1)，r=x_1 *mod* n 5. if(r=0) continue 6. else

7. e=*Hash*(m)，s=k^{-1}*(e+r*d *mod*n)
8. if(s=0) continue
9. else
10. return(r，s)break //返回签名信息

End

验证者验证(r，s)是否为交易消息 m 的签名，签名验证算法如表 4.5 所示。

表 4.5 验证数字签名算法

算法：验证数字签名算法

输入：交易签名信息(r，s)

输出：签名验证结果

Begin：

1. i=2
2. while(i<n-1)
3. if(!(1<r<n-1&&1<s<n-1)) continue //验证 r，s 为[1,n-1]中的整数
4. else
5. e=*Hash*(m)，w=s^{-1} *mod* n，u1=e*w*mod n，u2=r*w* *mod* n
6. t=u_1*G+ u_2*G
7. if((x_1，y_1)!=t) continue
8. else
9. if(r!=x_1 *mod*n) continue
10. else
11. return TRUE break //验证结果成功

End

4.3.3 PBFT 共识算法

共识算法是保证以太坊区块链平台各节点数据一致的关键。目前以太坊平台中，分布式系统的一致性算法普遍采用 PoW 共识算法。该算法计算量较大，资源浪费严重，共识效率低下且共识周期长，研发人员对其进行技术改进，共识周期将一个区块打包时间从 10 分钟减少到 17 秒，但仍没有解决资源浪费问题且允许全网 50%节点出错，也就是 51%攻击。同时 PoW 算法还有可能出现区块链分叉，主链只会选择分支最长的链，这样会导致部分数据丢失。

PBFT 共识算法中，产生区块时，由于区块头中含由时间戳（Nonce），生成区块的节点作为主节点，主节点是唯一的，可避免出现分叉现象。PBFT 共识算法要容忍 *f* 个拜占庭错误，至少需要 3*f*+1 个节点，网络中的区块只要有 2*f*+1 个节点投票达成共识，即可存储区块信息到底层区块链数据层。本文使用 PBFT 共识算法预期一个区块生成的时

间为 15 秒，从而提高分布式网络节点的共识效率。而且该算法性能高，耗能低，容错性高，监管性强，容错性为 33%。因此本文基于该算法优点，结合系统的联盟链网络和学位信息交易业务，采用 PBFT 共识算法。PBFT 共识算法的实现如表 4.6 所示。

表 4.6 PBFT 共识算法

算法：PBFT 共识算法
输入：区块高度 BH
输出：共识后新区块高度 NewHeight
Begin：
1. H=BH+1，R=0，N=NewHeight
2. for(i=0;i<H;i++)
3. if(N==NewHeight)
4. 清空区块记忆池、主节点索引、投票等数据，更新本节点索引
5. N=NewRound
6. else if(N==NewRound)
7. 根据高度、轮次和参与共识节点数更新节点索引
8. N=Primary
9. else if(N==Primary)
10. if(内存中交易数量>=区块大小阈值)
11. 创建区块，将共识节点和交易结合，执行交易
12. if(本节点索引==主节点索引)
13. 创建并广播主节点内容
14. else 等待主节点的消息内容
15. N==Vote
16. else
17. N==Primary
18. else if(N==Vote)
19. if(交易是否合法)
20. if(未投赞成票)
21. 投赞成票，同时向节点广播交易内容
22. else if(未投反对票)
23. 投反对票，同时向节点广播交易内容
24. if ((赞成票+反对票) > 2/3*共识节点总数)
25. N=BlockVote
26. else if(N==BlockVote)
27. if(判断区块是否合法)
28. if(未投赞成票)
29. 投赞成票，同时节点广播区块内容

```
30.                 else if(未投反对票)
31.                     投反对票，同时节点广播区块内容
32.             if ((赞成票+反对票) > 2/3*共识节点总数)
33.                 N=Commit，R++，N=NewRound
34.         else if(N==Commit)
35.             将区块添加到链上，提交共识节点索引，向非共识节点广播区块和列表
36.             H++，R=0，N=NewHeight
37. return H-1
End
```

4.4 学位管理系统详细设计与实现

在系统实现运行之前，我们需要先将合约部署到区块链上。本系统为方便用户操作，设计了学位管理系统应用的前端界面，用户可以通过 Web 服务器提供的页面服务进行数据访问。为更好地实现系统业务与智能合约业务之间的交互，使用 Truffle 工具来实现对智能合约的开发。Truffle 支持对合约代码的单元测试，同时内置了智能合约编译器，只要使用脚本命令就可以完成合约的编译、部署、测试等工作，大大简化了合约开发周期。

系统中安装了 6 台 Ubuntu14.0 服务器作为底层区块链节点，并在配置文件中设置了六个节点的 IP 地址和区块链地址，使节点间建立 P2P 网络且达成共识。利用 Truffle 中的/migrations/文件部署智能合约，只需将合约部署到一个网络节点，其他节点就会更新自身数据并访问合约。合约部署成功后，开启本地服务器，通过运行 Node.js 中 npm run dev 启动项目即可将系统前端、业务合约和底层区块链之间进行交互，从而完成区块链数据查询接口以及学位管理系统中一系列功能操作实现。

4.4.1 区块链数据查询接口

在以太坊区块链上开发学位管理系统的目的就是保证学生学位信息在交易中数据的公开透明、防篡改和可追溯，设计了交易查询模块，并对外提供了数据查询接口，可以在实现过程随时关注交易数据，为后期系统的开发更新提供便利。本文提供的查询接口中，对交易和区块中的数据进行查询，也对交易和区块中返回结果的状态进行查询。

交易区块数据的查询，必须要有交易的存在。在学位信息注册、学位信息发布以及学位信息召回等业务中都会产生交易数据，该交易会调用智能合约业务，进而与区块链底层数据交互。因此可通过可以通过外设接口，根据交易 address 和区块 txhash 查询当前区块交易的具体数据。浏览器发送查询区块信息的 IP 地址请求链接，智能合约通过 Event 获取所有需要与合约交互的操作，当有查询请求时，会搜索已产生的有效操作，进而获取 Event 事件及相关字段值，提取重要数据反馈给当前节点。反馈到前端的数据

是一个 JSON 字符串，需要对 JSON 串解析，以获取字段值，最终在浏览器页面上显示。交易区块查询流程如图 4.15 所示。

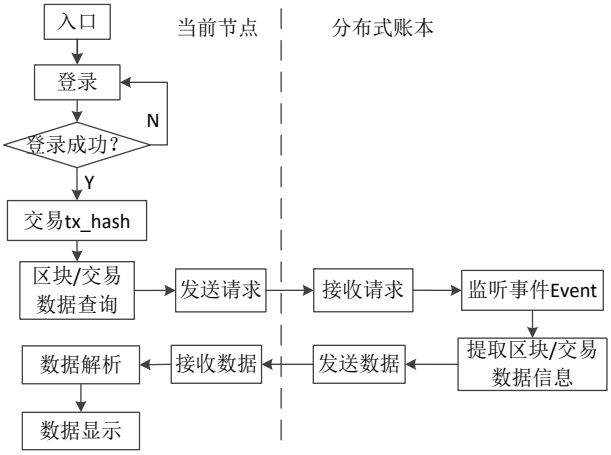

图 4.15 区块/交易数据查询的授予流程

4.4.2 学生注册学位信息

学位信息注册功能实现的是学生申请学位必须填写的数据信息，学生根据学校要求正确填写学生的姓名、性别和身份证号等重要字段，若学生信息输入格式出现问题或输入为 NULL，智能合约接收到注册请求后将会反馈给页面，弹出警示框注册失败。若个人信息注册成功，同样会与智能合约发生交互，生成交易，待网络中参与节点验证通过，会将该笔交易添加至新区块。新区块会被重新发布至节点网络，参与共识节点中有 2/3 以上节点投票（PBFT）认可新区块，则广播该区块，告知其余节点需要重新更新自身区块数据，保证区块链账本的一致性。根据该笔交易，会在页面生成保护学生数据信息的公钥、私钥和交易地址信息。密钥对作用是节点（学校、教育局）返回的信息中，需要运用自己的私钥去查看该节点返回的审核结果，生成的地址用于学生查询自己学位信息。学生学位注册模块流程如图 4.16 所示。

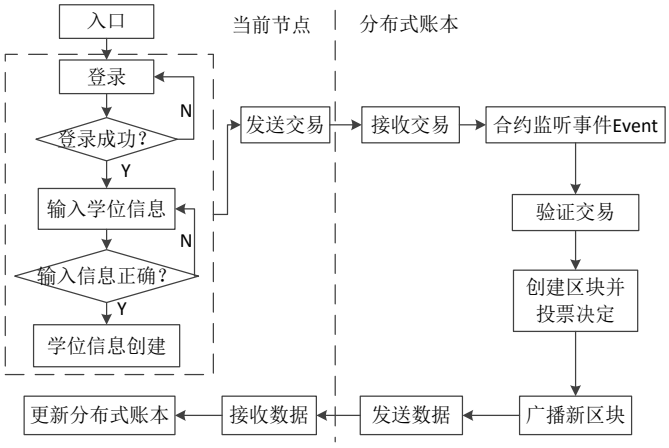

图 4.16 学生信息注册具体流程

4.4.3 学校发布学位信息

在学校授予学生学位时，授予学位的管理人员首先输入进入该页面的 IP 地址，同时必须获得进入该系统的权限，也就是得到这个学校节点的私钥（配置系统时已生成）。当学校节点私钥密码输入错误，无法进入到系统内部，因此进入学校节点的管理系统，必须获取到节点私钥。进入节点页面之后，学校首先会在学生“学位申请”导航栏接收来自学生提交的学位申请交易，先判断这笔交易是否有效，确认交易无误后，点击“授予学位”对学生学位信息进行授予操作，若存在问题，也可点击“取消”，表明该次学生学位申请失败，请重新填写信息提交申请。当学生学位授予成功，系统会将该交易结果返回至学生和教育局，将交易变为学生和教育局共有资产，从而完成学位信息发布，学校对学生学位授予成功。学生学位发布的流程如图 4.17 所示。

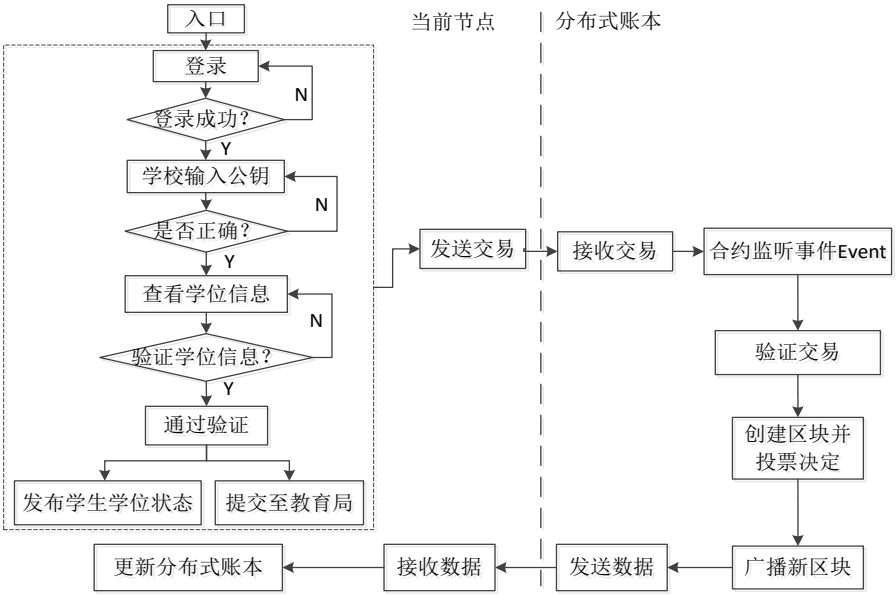

图 4.17 学位信息发布的授予流程

4.4.4 学校召回学位信息

当学生的学位信息一旦写入到区块链上，任何人都没有权利对学生的数据状态进行修正。学校授予学生学位信息时，将学位信息数据转变为学生和教育局共有，因此当学校发现并验证学生的学历出现问题，需要对学位进行召回请求。学校这时候就需要提交召回申请学位的交易请求，并发送给教育局。教育局输入教育局节点页面的 IP 地址进行审核操作，同样的，进入该页面的人员同样需要得到进入该系统的私钥。教育局审核验证学位召回理由成立，教育局就可以利用自己的私钥单方面将学位信息所有权返回给学校，学生不再拥有学位信息所有权，即学位被召回。学位召回交易过程也会与智能合约进行交互，只有当接收到合约发来的确认信息才可继续将信息返回给学校。学生学位召

回流程如图 4.18 所示。

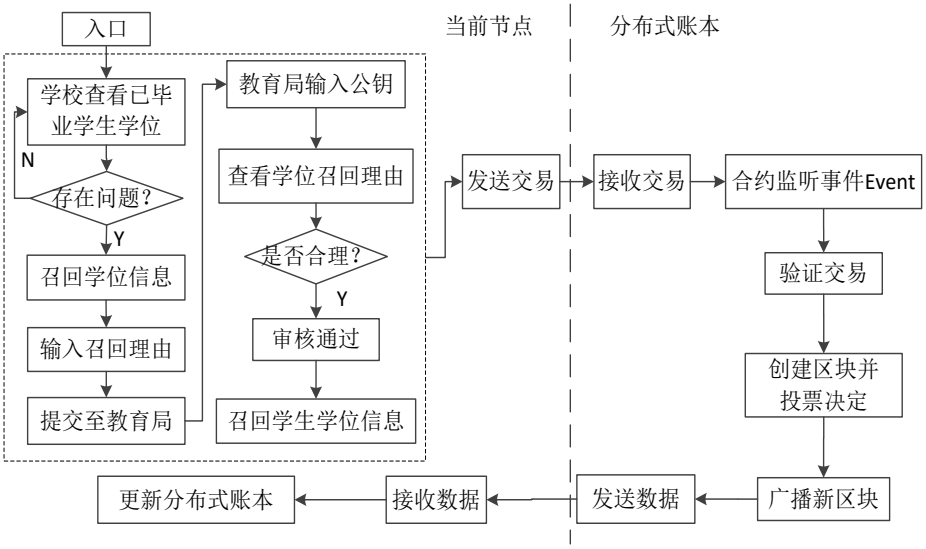

图 4.18 学位信息召回的授予流程

4.4.5 用户查询学位信息

学位信息查询模块，当学生的学位信息已被成功授予，可以通过提供的外部接口查询区块链上的学位信息数据。该模块访问者不需要被授予权限，任何需要查询学生学位信息的企业或者单位机构，都可以对学生学位的真实性进行验证。在系统中，若需要验证学生学位真实性，只要知道学生唯一的学位信息公钥地址 PublicKey+Address 就可以查询到学生的学位信息。点击“查询”会将查询学位的请求通过智能合约发送给区块链账本，后台会根据该请求取得 Event 事件，提供相应的学位信息并返回给节点，进而发送给浏览器 JSON 字符串，浏览器对收到的 JSON 数据进行解析，获取相应的信息，方便用户查看。学生学位信息查询流程如图 4.19 所示。

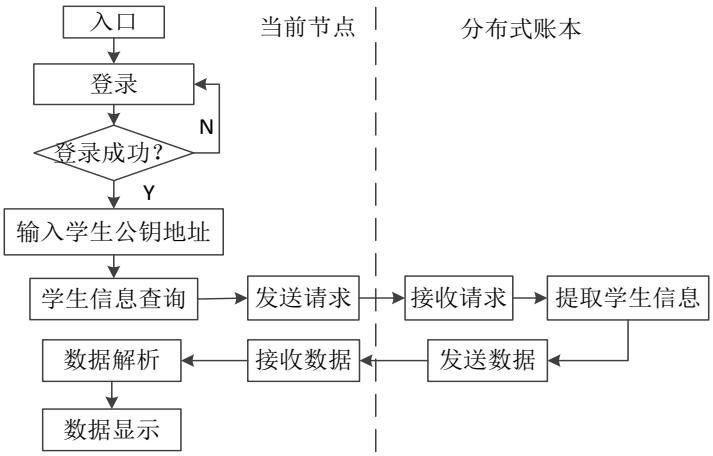

图 4.19 学位信息查询的授予流程

4.5 本章小结

本章首先介绍了智能合约的创建、部署、调用等过程。根据第三章学位信息交易流程，将学位信息管理业务设计为数据合约和业务合约，并结合系统的特定功能，设计了基础合约。使用 Oyente 工具对业务合约的安全性进行检测验证，分析合约出现安全漏洞的原因，对存在安全问题的合约改进并对合约进行实现。其次对系统中所运用的动态网络节点增减算法、交易数据验证算法和 PBFT 共识算法进行设计与实现。最后将合约部署到区块链，并对学位管理系统功能和区块链数据接口进行设计与实现。

5 学位管理系统测试

本章主要介绍了基于区块链的学位管理系统的测试环境，对系统中的功能模块和以太坊网络进行检测。进行这些测试的目的是为了确保系统在使用过程中的安全性和可靠性。

5.1 测试环境

本系统在 Eclipse 进行开发，采用 Java 语言中的 SSM（Spring+Spring MVC+Mybits）开发框架，对系统的业务进行编写，实现基本功能。在该系统中，Web 应用程序部署在 Tomcat 中，外部用户可以通过浏览器访问学位管理系统。学位管理系统基本配置环境需求如表 5.1 所示。

表 5.1 开发环境基本配置需求

软硬件设施	配置需求
操作系统	Windows10
内存	4GB
硬盘	500G
Java 程序开发	JDK1.8
数据库	MySQL5.7
Web 服务器	Tomcat8.0
区块链网络	Ethereum

5.2 系统测试

本节着重对系统功能和以太坊网络进行验证测试，测试功能模块主要判断系统是否满足基本设计需求，而以太坊网络状态则可以监测交易、合约和底层数据的运行情况，也可以查看每笔交易的具体信息。

5.2.1 功能性测试

本系统主要的功能是完成学位信息注册、发布、召回和查询，且在交易过程中确保交易数据的安全、公开透明，存储到区块链上的数据不可篡改、可追溯。

本文中对学位管理系统主要采用黑盒测试方式，将系统的功能按照模块的方式进行测试，检测系统功能的完整性与健壮性。通过测试各个模块，确保能够正确的写入和显示数据，对系统中已实现功能的可用性、页面的正确性、数据交互的正确性及验

证信息的有效性等进行相关测试。以下列举了系统部分功能的测试用例及测试结果展示（测试数据涉及人名均伪造）。

（1）学生学位注册模块

表 5.2 学生学位注册模块测试用例及结果

模块名	学位信息注册模块		
操作人员	学生	测试时间	2018-12-15
用例 ID	Case-01		
测试用例名称	学生学位信息注册功能		
前置条件	无		
相关用例	无		
测试目的	测试学生学位注册功能是否正常，错误输入是否有提示。		
操作步骤	1. 学校输入 IP 地址进入注册页面； 2. 在注册页面输入学生的学位信息； 3. 点击立即创建按钮。		
预期结果	1. 注册学生信息输入正确的 IP 地址进入到信息注册页面。 2. 学位信息填写格式正确，则学生学位信息注册成功，并显示学生公钥、私钥、地址信息。 3. 若学位信息已注册，弹出提示框。 4. 系统验证学生填写的学位注册信息进行，存在错误信息弹出提示框。		
测试数据	Test01	URL 地址：http://118.25.128.22:8080/#/student	
	Test02	学号：123；姓名：当当；身份证号：610234199602230058 学校：某大学；学院：某学院；专业：通信专业；年级：2015；年制：4	
	Test03	学号：123；姓名：当当；身份证号：610234199602230058 学校：某大学；学院：某学院；专业：通信专业；年级：2015；年制：4	
	Test04	学号：123；姓名：当当；身份证号：afafasgfsdgsd 学校：某大学；学院：某学院；专业：通信专业；年级：2015；年制：4	
实际结果	1. 学生成功进入学位注册页面。 2. 学生学位信息注册成功，成功显示学生的公钥、私钥和交易地址。 3. 学位信息已注册，请勿重复注册。 4. 用户身份证信息格式错误，系统提示，引导学生进行相关操作。		

学生学位注册完成页面如图 5.1 所示。

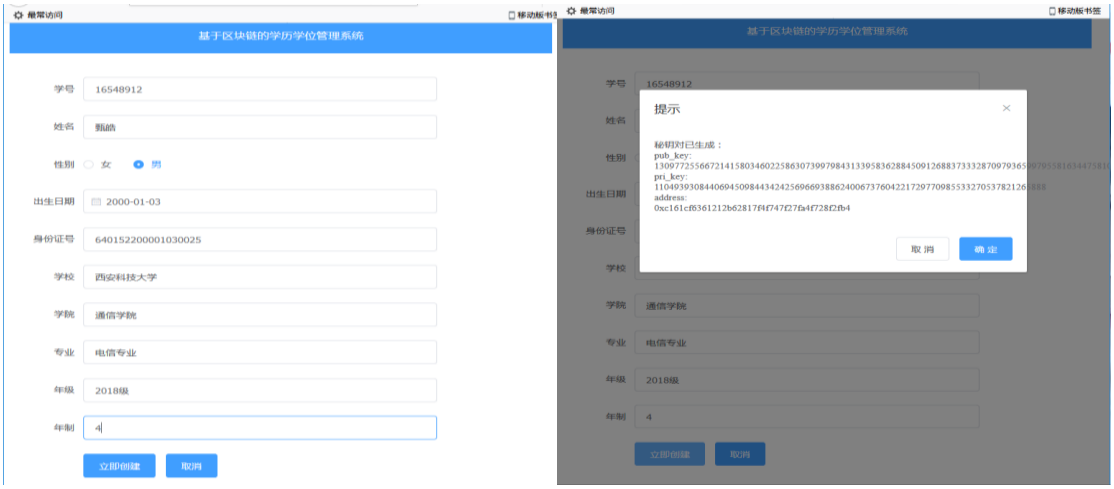

图 5.1 学生注册信息页面

（2）学校发布学位模块

表 5.3 学校发布学位模块测试用例及结果

模块名	学校发布学位模块		
操作人员	学校	测试时间	2018-12-15
用例 ID	Case-02		
测试用例名称	学校发布学位		
前置条件	学位信息已注册成功		
相关用例	学位信息注册模块		
测试目的	测试学校对于学生注册信息正确性的验证，决定是否授予学位，错误是否有相关提示。		
操作步骤	1. 学生输入 IP 地址进入学校审批页面； 2. 在学校页面输入授予进入该页面的权限的私钥； 3. 点击待审核导航栏，验证学生学位注册信息； 4. 若学生学位申请信息正确，则点击授予学位。		
预期结果	1. 学校输入正确的 IP 地址进入到学校节点页面。 2. 学校输入正确的私钥，获得操作该页面的权限。 3. 学生学位注册信息正确，则学校授予学位。		
测试数据	Test01	URL 地址：http://118.25.128.22:8080/#/school	

	Test02	1. 私钥信息： 02b361ad11f5a2cf82d137a6c8927b0f1722d15bb460c20a14c38e0659a19e4f 2. 验证待审核页面的学生信息。
实际结果	1. 学校输入正确 IP 地址进入教育节点页面； 2. 在学校页面输入了正确的私钥密钥获得进入该页面的权限； 3. 审核学生的学位申请信息； 4. 学位信息验证结果正确，同意授予学位。	

学校对学生学位进行授予过程如图 5.2、5.3 所示。

图 5.2 学校审核人员进入学校系统页面权限

图 5.3 学校对学生学位信息授予页面信息

（3）学校学位召回模块

表 5.4 学校学位召回模块测试用例及结果

模块名	学校召回学位模块		
操作人员	学校/教育局	测试时间	2018-12-15
用例 ID	Case-03		
测试用例名称	学校召回学位		

前置条件	学位信息已注册成功、学生已成功获得学位	
相关用例	学位信息注册模块、学校发布学位信息模块	
测试目的	测试已成功获得学位申请的学生，学位出现问题，学校进行召回操作过程，教育局确认召回信息，错误是否有相关提示。	
操作步骤	1. 学校在已通过导航栏选择需要召回的学生，并填写召回理由； 2. 点击召回按钮将召回申请提交给教育局； 3. 教育局在召回学位申请导航栏中查看需要召回学生信息，并审核召回理由是否合理；	
预期结果	1. 学校对需召回学生理由合理。 2. 教育局同意学校提交的学生学位召回请求。 3. 学生学位信息已被召回。	
测试数据	Test01	1. 选择召回学生：甄皓； 2. 输入召回理由：在校学位学分未达到学校要求。
	Test02	教育局 URL 地址：http://118.25.128.22:8080/#/jiaoyuju
	Test03	1. 教育私钥信息： c1007ec09f47c32af87b6f81efb340107b6140ef44afd3f08876 31e869cb1181 2. 验证召回学位申请页面的学生信息。
实际结果	1. 学校对需召回学生理由合理。 2. 教育局同意学校提交的学生学位召回请求。 3. 学生学位信息已被召回。 4. 召回理由输入有误，系统提示，引导进行相关操作。	

学校对学生学位进行召回等操作如图 5.4、5.5、5.6 所示。

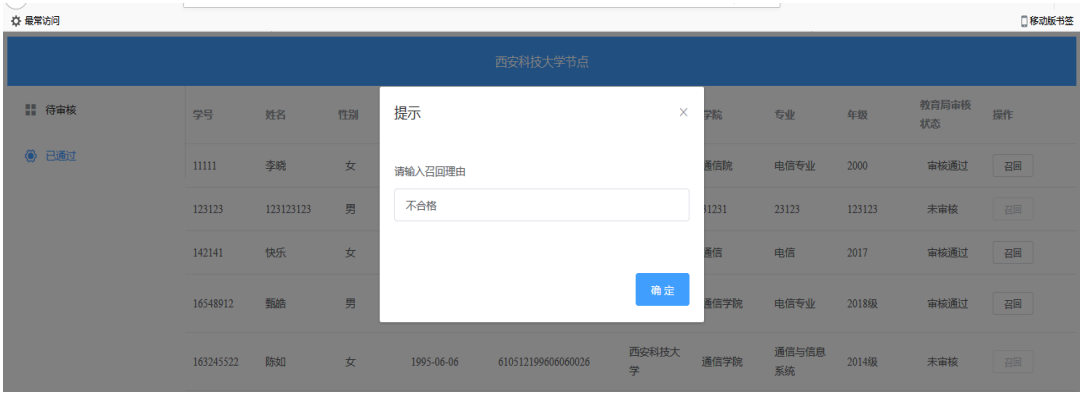

图 5.4 学校对学生学位进行召回操作

图 5.5 教育局管理员进入学校系统页面权限

图 5.6 教育局审核召回学位申请页面

（5）用户学位查询模块

表 5.5 用户学位查询模块测试用例及结果

模块名	学位信息查询模块		
操作人员	用户	测试时间	2018-12-15
用例 ID	Case-04		
测试用例名称	学位信息查询		
前置条件	学位已成功授予		
相关用例	学位信息注册模块、学校审批学位信息模块、学校审批学位信息模块、学位召回模块		
测试目的	测试外部企业人员或者用户在是否可以查询到学生学位信息		
操作步骤	1. 用户输入 IP 地址进入学位查询页面； 2. 在该页面用户输入学生的学位地址； 3. 点击查询。		

预期结果	1. 用户输入正确的 IP 地址进入学位查询页面。 2. 查询到获得学位证的学生的学位信息。 3. 查询不到已召回学生学位信息。	
测试数据	Test01	URL 地址：http://118.25.128.22:8080/#/select
	Test02	获得学位的学生地址： 0xc161cf6361212b62817f4f747f27fa4f728f2fb4
	Test03	召回学位的学生地址： 0xe7965269f9a6bd0e8a40de199553b36339e73f3d
实际结果	1. 用户输入正确的 IP 地址进入学位查询页面。 2. 查询到获得学位证的学生的学位信息。 3. 查询不到已召回学生学位信息。 4. 用户输入错误的地址信息，系统提示，引导进行相关操作。	

学生学位查询模块页面显示如图 5.7、5.8 所示。

图 5.7 学生学位查询页面信息

图 5.8 学位信息查询验证信息

（5）交易/区块查询模块

在以太坊浏览器 Etherscan 中，我们可以根据交易的哈希值查询交易的具体列表信息，如表 5.6 所示。

表 5.6 交易/区块查询模块测试用例及结果

交易数据字段	交易数据值
Hash	0xeca623ef431c71aaec9e35c8ddc8c3c7c6190073aed78989866597e24d4b95be
BlockHeight	3525607
TimeStamp	15sec
From	0xbe1085bc3e0812f3df63deced87e29b3bc2db524
To	0x40af244c94e679aebf897512720a41d843954a29
Transaction Fee	0.000074511 Ether

如图 5.9 所示，显示的是已生成区块中某交易的具体数据，其中包含交易所在的区块高度，生成区块的时间，当前区块中交易的哈希值以及前一区块哈希值等信息，该信息表示对交易内容完全透明化，体现了区块链数据的公开透明。

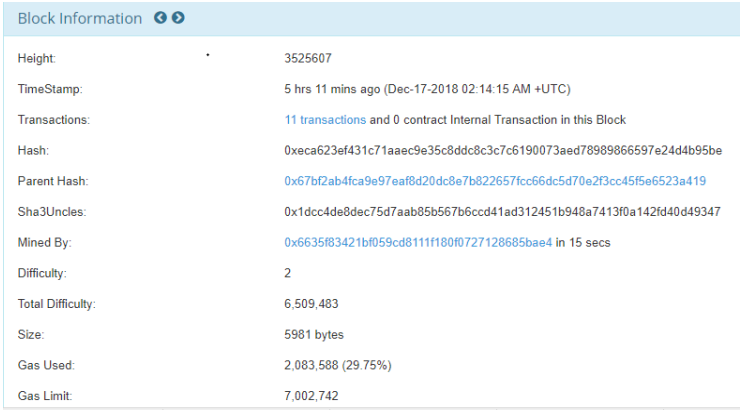

图 5.9 区块中的交易数据

本系统在学位管理系统中对外设置了区块链数据查询接口，用户通过 Browser 页面输入 http 请求，对交易和区块中的数据进行查询，也对交易和区块中返回结果的状态进行查询，以下介绍了区块交易数据查询接口信息。

① 查询交易数据

查询学位信息的交易数据是在查询接口中输入交易的 address，利用 http 中 GET 请求方式查询交易地址格式：http://[ipAddress]:[port]/getTransaction?[address]，查询结果中可以看到交易中学生的具体数据，查询请求结果如图 5.10 所示。

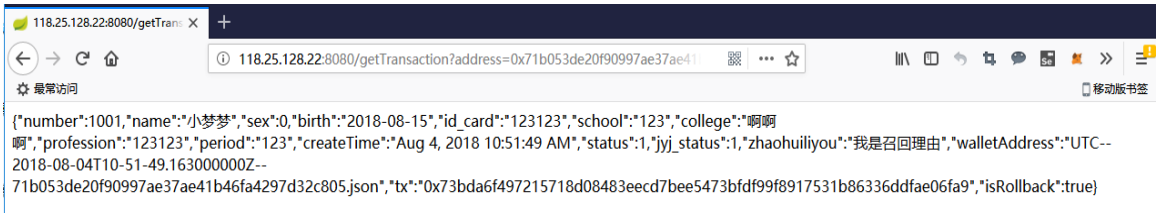

图 5.10 学位信息交易数据

② 查询区块数据

查询区块中的交易内容是在查询接口中输入区块的 txhash，利用 http 中 GET 的请求方式查询区块数据地址格式为：http://[ipAddress]:[port]/api/txinfo?[txhash]，查询请求结果如图 5.11 所示。从查询的结果中得到，区块数据中包含区块的 id、blockHash 字段表示区块中目前交易的全部信息，S 字段表示区块生成的签名信息。

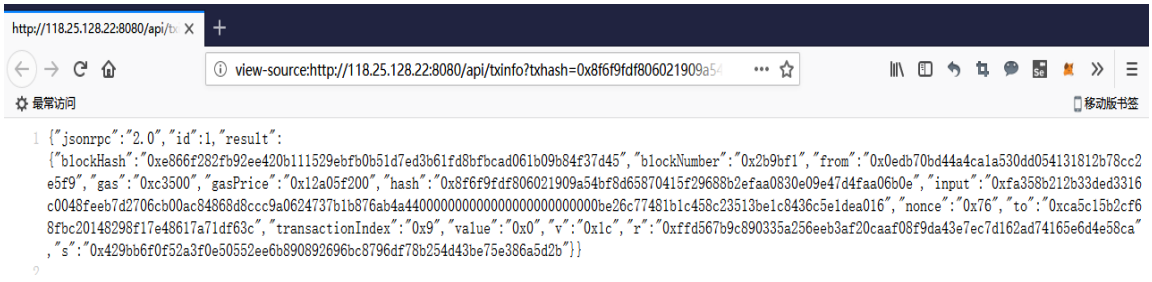

图 5.11 区块数据

③ 查询交易/区块状态

查询学位信息数据的交易状态是在查询接口中输入交易 txhash，利用 http 中 GET 的请求方式，查询交易数据状态结果的地址格式为：http://[ipAddress]:[port]/api/getStatus?[txhash]，查询请求结果如图 5.12 所示。

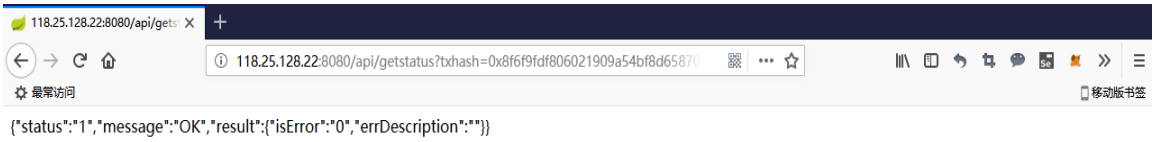

图 5.12 交易状态结果

查询区块状态是在查询接口中输入区块的 blockNumber，利用 http 中 GET 的请求方式，查询区块数据状态结果的地址格式为：http://[ipAddress]:[port]/api/blockreward?[blockno]，查询请求结果如图 5.13 所示。

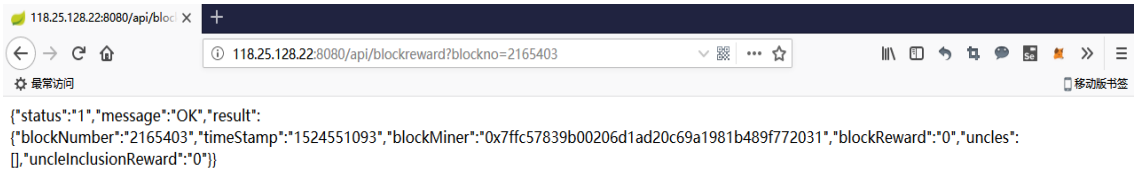

图 5.13 区块状态结果

5.2.2 以太坊网络检测

学位管理系统是基于以太坊区块链实现的，以太坊网络中各个节点组成一个联盟体，在系统运行时，需要对以太坊节点进行检测，并能提供检测指标的接口，不仅需要包括对区块的查询、账户信息查询以及交易信息，还需要对系统初始化时提供数据迁移功能。

本文利用开源的前端框架搭建了 Ethereum Network Status（以太坊区块链网络状态）检测界面，如图 5.14 所示，目的是查看系统、智能合约与区块链数据之间交互的状态变化。从图中我们可以看出，当前区块的高度为 4155，且生成区块的平均时间在 15 秒左右，而使用 PoW 算法生成区块时间为 10 分钟左右，因此，使用 PBFT 算法加快了产生区块的速度。此页面还显示了交易笔数、交易费用以及以太坊网络中六个验证节点连接情况。交易笔数说明有交易产生，在图中 TRANSACTIONS 区域中的柱状图表示智能合约正处于稳定执行过程，该检测界面主要用于观察区块链网络整体运行情况。

图 5.14 以太坊区块链网络检测界面

5.3 测试结果分析

通过本文进行的实验，可以看出本文设计的基于区块链智能合约的学位信息管理系统能够实现设计的功能，该系统能正常运行。

1）本系统能够通过智能合约来实现前端页面与底层区块链之间的数据交互，确保学位信息能被正确存储到链上。如果智能合约监听到事件交易处理时，合约执行遇到错误，则会执行合约还原，交易不被处理，也就不会写入链中，保证了系统数据的安全执行。

2）基于联盟链实现的学位管理系统，在保证去中心化的同时，采取新的共识机制PBFT 算法，区块生成时间短，效率高，系统运行速度快，提高了监管的便捷性。

5.4 本章小结

本章节首先介绍了基于区块链智能合约的学位管理系统的测试环境，并对系统中的学位信息注册、学位信息发布、学位信息召回、学位信息查询以及区块链数据查询等模块进行测试。同时还搭建了以太坊网络监测环境，可以实时监测区块、合约等的运行情况。最后测试结果表明在 PBFT 共识算法的基础上，区块生成时间短，系统的可行性强，执行效率高。

6 总结与展望

6.1 总结

区块链技术是金融科技领域乃至整个 IT 领域的重大技术创新，该技术中的去中心化、去信任化、数据可靠、集体协作等特性，从技术层面上保证了链式账本所存储的数据安全可信，适用于解决多方业务协作场景、维护信用增加监管资源成本的问题。基于上述原因，本文在以太坊区块链平台进行开发，利用智能合约设计学位信息交易业务，最终实现学位管理系统。本文工作内容如下：

1）分析了学位管理系统的功能与非功能需求，设计了适合本系统的整体架构。根据系统模块的业务逻辑，进行智能合约设计。本文基于去中心化联盟链设计的学位管理系统，将以太坊中原有的 PoW 共识机制被 PBFT 共识机制代替，全网节点只需 2/3 节点投票通过即可达成共识。在智能合约的设计中，分别设计了基础合约、数据合约和业务合约。基础合约中增加了动态节点增减机制，保障了系统安全；数据合约中设计了相关数据结构、底层数据转换算法和 Map 遍历算法，提高系统运行效率；业务合约中不同的用户可以根据合约设置的权限进行学位注册、学位发布、学位召回等不同操作，保证系统可靠执行。最后对智能合约的安全性进行验证，对存在问题的代码修改并实现。

2） 对学位管理系统中的数据签名验证 ECDSA 算法和 PBFT 算法进行设计与实现，该算法确保了交易数据的安全，以及在分布式网络节点中，各节点数据达成一致，集体来维护唯一的数据账本。

3）本文将后端代码、合约代码与前端界面代码进行交互，完成区块链智能合约在学位管理系统上的应用。所有与智能合约交互的业务都以交易的形式被记录在区块，保证了数据可追溯。使得学位信息在数据可靠性方面得到了极大提高，减少第三方监管成本，避免学位造假现象，解决学位管理系统的公信力问题。

6.2 展望

本文基于区块链智能合约的学位管理系统，解决了目前学位系统管理员权限过大、学位数据易被篡改和中心系统数据被攻击造成数据泄露等问题。该系统基本实现了系统前期设计的要求和目标，但系统仍然存在不足，尤其在以太坊区块链数据大量存储、海量交易数据的处理效率以及系统合约的隐密性等方面仍需要完善，该论文下一步的研究可从以下几个方面进行：

1）本系统区块链中交易数量过多时，网络处理交易效率较低，周期较长。在后续的研究中可引入分片技术，在网络节点中定义整理器的节点，整理器可根据协议选择分片

k 上的交易。不同的分片可以致力于解决以太坊区块链中所有交易效率的问题。

2）本系统中部署在区块链上的智能合约，外部可查看，不能保证合约内容的隐私。后续研究可以利用 Hawk 项目中 zkSNARK 和多方计算来解决匿名合约安全计算和内容隐私等问题。

致　谢

任时光匆匆，我仍是我。回首过去，昨日已不复存在，但西安科技大学这所我就读了整整七年的母校，以及在这七年期间母校带给我快乐与知识、教会我为人处世和教导我永不言弃的精神毅力等品质，在我人生记忆中烙下深刻的痕迹。让我懂得了，今日即将过去，逝去的终归留不住，只有抓住眼前才是最重要；明日即将到来，任何事都需未雨绸缪，才能在未来更好的展现自己。临别之际，我想要对那些在我学业、生活、寻找工作中给予我支持和鼓励的人表示真诚的感谢！

首先衷心的感谢我的导师孙弋教授。三年的硕士求学，老师在学习和生活上都给予我悉心的照顾和莫大的帮助。老师渊博的知识、专业的技能、积极的人生态度、高瞻远瞩的慧眼和对新技术敏锐的洞察力都让我为之钦佩。不仅如此，从论文的选题到具体的需求分析，再到系统的实现，每一步都有老师的指导以及对我成果的肯定，让我信心满满，继续前行；相应的，不足之处也提出宝贵的修改意见，确保成果的完善。“一日为师终身为父”,老师对我孜孜不倦的教诲,是我现在乃至未来学习生活中最宝贵的人生财富。

其次感谢我的同门师兄妹在我研究生三年给予我的帮助。当我学习遇到瓶颈的时候，是你们不厌其烦的帮助我，为我指点迷津，给予我精神上的鼓励和支持，让我继续前行。感谢你们，也感恩命运能够让我们相聚在这温馨的实验室，相互探讨和学习。感谢我的室友们，因为有你们的存在，我们宿舍生活丰富多彩，充满欢乐，还有你们积极的生活态度也感染了我，让我变得越来越优秀。

然后感谢为我无私付出的父母和亲人们。在我失意、无措之时，是你们对我一次次的开导、鼓舞，让我相信自己，勇于面对生活的挫折、敢于去尝试新事物。这种精神和毅力也让我延续到对科研学习中，坚信自己永不放弃，最终使得自己顺利完成研究生学业生涯。

最后感谢各位专家教授评审老师对本论文提出的意见和建议！

参考文献

[1] 徐才千. 新形势下高校后进生教育存在的问题及对策[J].教育探索,2010(11):87-88..

[2] 刘华倩. 浅谈市场经济背景下职业资格与学历相互并存的重要性[J].赤子(中旬),2013(11):260-261.

[3] 蔡亮,李启磊等. 区块链技术进阶与实战[M]. 人民邮电出版社, 2018(4).

[4] 曹阳,薄珺. 区块链技术与互联网音乐作品版权保护[J].南海法学,2018,2(03):77-87.

[5] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J].Consulted, 2008.

[6] Buterin V. Ethereum white paper: a next generation smart contract&decentralized application platform[J]. www3. ethereum. org Nick Szabo, Formalizingand Securing Relationships on Public Networks.

[7] Luu L, Chu D H, Olickel H, et al. Making smart contracts smarter[C]. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016:254-269.

[8] Juels A, Kosba A, Shi E. The ring of gyges: Using smart contracts for crime[J]. aries，2015,40:54.

[9] 2016-2020 年区块链技术深度调研及投资前景预测报告[DB/OL], http://www.cn-bigdata.cn, 2016(07).

[10] 张健. 区块链：定义未来金融与经济新格局[M], 机械工业出版社,2018.

[11] 国务院, “十三五”国家信息化规划[J].电子政务,2017(01):40.

[12] 黄俊飞,刘杰. 区块链技术研究综述[J].北京邮电大学学报,2018,41(02):1-8.

[13] Tencent FiT, Tencent Research Institute. White paper for tencent trustSQL[EB/OL]. White Paper,2017(in Chinese). https://www.jianshu.com/p/8f032879b395

[14] 陈亚飞. 基于区块链智能合约的仓单交易平台研究与实现[D].郑州市:郑州大学,2018.

[15] Azaria A, Ekblaw A, Vieira T, et al. MedRec: Using Blockchain for Medical DataAccess and Permission Management[C]. International Conference on Open and Big Data.IEEE, 2016:25-30.

[16] Pradip Kumar Sharma, Jong Hyuk Park. Blockchain based hybrid network architecture for the smart city[J]. Future Generation Computer Systems, 2018:650-655.

[17] Sharples M. The Blockchain and Kudos:A Distributed System for Educational Record,Reputation and Reward [M].Adaptive and Adaptable Learning. 2016.

[18] 华劼. 区块链技术与智能合约在知识产权确权和交易中的运用及其法律规制[J].知

识产权,2018(02):13-19.国务院.

[19]叶俊洪. 区块链在网络安全中的应用研究[J].现代信息科技,2018,2(11):144-146.

[20]Lijie Zhu. New Trend in Blockchain:From Ethereum Ecosystem to Enterprise Applications[A]. 联合国教科文组织知识社会局（Knowledge Societies Division(KSD),UNESCO）、世界高科技协会（World High Technology Society(WHTS)）.2017 第四届全球知识经济大会会刊[C].联合国教科文组织知识社会局（Knowledge Societies Division(KSD),UNESCO）、世界高科技协会（World High Technology Society(WHTS)）:百奥泰国际会议(大连)有限公司,2017:1.

[21]Buterin V. A next-generation smart contract and decentralized application platform[J]. white paper, 2014.

[22]管磊. P2P 网络监管中的网络视频节目信息发现技术研究[D].北京市:北京工业大学,2010.

[23]陆伟杰. 计算机对等网络 P2P 技术的探讨[J].信息与电脑(理论版),2016(17):149-150.

[24]J. Gobel, A.E. Krzesinski，H. P. Keeler，et al. Bitcoin Block Chaindynamics: The selfish-mire strategy in then presence of Propagation delay[J]. Performance Evaluation, 2015, 104: 23-41.

[25]Nikolic I, Biryukor A. Collisions for Step-Reduced SHA256[M], Fast Software Encryption. Springer Berlin Herlin Heidelberg, 2008: 1-15.

[26]任强,赵德平. 椭圆曲线数字签名算法下的公钥密钥验证[J].计算机与数字工程,2011,39(03):98-101.

[27]Merkle R C. Protocols for public key cryptosystems[C]. In IEEE Symposium an Security and Privacy, 1980: 122-134.

[28]刘通,王凤英. 基于 Merkle 树的起源完整性解决方案[J].山东理工大学学报(自然科学版),2012,26(03):68-71.

[29]Mark, 区块链核心技术演进之路—共识机制演进（1）[EB/OL]. http://www.8btc.com/blockchain-tech-consensus-mechanism,2016(12).

[30]Back A. Hash-A Denial of Service Counter-Measure[C]. USENIX Technial Conference.2002.

[31]Zbierski M. Parallel byzantine fault tolerance[J]. Soft Computing in Computer andInformation Science. Springer International Publishing, 2015: 321-333.

[32]KING S, NADAL S. Ppcoin: Peer-to-Peer crypto-currency with proof-of-stake[J]. self-published paper，August 2012，19.

[33]LARIMER D. Delegated Proof-of-Stake (DPOS) [EB/OL]. 2014, [2017-05-07]. http://www.bts.hk/dpos-baipishu.html.

[34]宋焘谊,赵运磊. 区块链共识算法的比较研究[J].计算机应用与软件,2018,35(08):1-8.

[35]Elli Androulaki, Ghassan O. Karame. Hiding Transaction Amounts and Balances inBitcoin[M]. Springer International Publishing: 2014: 6-15

[36]贺海武,延安,陈泽华. 基于区块链的智能合约技术与应用综述[J].计算机研究与发展,2018,55(11):2452-2466.

[37]Wikipedia, 区块链，[EB/OL]. http://zh.wikipedia.org/zh-hans/区块链，2017.

[38]Croman K，Decker C，Eyal I，et al. On scaling decentralized blockchains[C]. Proc. 3rd Workshop on Bitcoin and Blockchain Research. 2016.

[39]杨茜. 基于区块链的智能合约研究与实现[D]. 绵阳市:西南科技大学,2018.

[40]Mainelli M, Milne A. The Impact and Potential of Blockchain on Securities Transaction LIFECYCLE[EB/OL].https://doc.mbalib.com/view/c6aff6af955a52d3b7586b7db3d14140.html

[41]顾燕. 基于区块链的身份认证系统的设计与实现[D]. 北京市:北京邮电大学,2018.

附　录

攻读学位期间发表的论文：

[1] 党京，孙弋基于区块链的电子投票系统关键技术的实现[J].软件，2018,39(11):140-144.

攻读学位期间获得奖项：

[1] 第三届西安科技大学移动终端应用设计大赛二等奖，2017 年 6 月 24 日.

[2] 第五届研究生数学建模竞赛二等奖，2018 年 7 月.