分 类 号_____ 学 号 M201572872

学校代码 10487 密 级_____

华中科技大学

硕士学位论文

基于区块链的
安全支付平台的设计与实现

学位申请人：张 舒

学 科 专 业：计算机技术

指 导 教 师：金 海 教授

答 辩 日 期：2017 年 5 月 23 日

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

The Design and Implementation of Secure Payment Application Based on Blockchain

Candidate ： Shu Zhang

Major ： Computer Technology

Supervisor ： Prof. Hai Jin

Huazhong University of Science and Technology

Wuhan 430074, P.R.China

May, 2017

独创性声明

本人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取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到，本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：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，允许论文被查阅和借阅。本人授权华中科技大学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。

保密□ ，在_____年解密后适用本授权书。

不保密□。

（请在以上方框内打“ √ ”）

学位论文作者签名：　　指导教师签名：

日期：　　年　　月　　日　　日期：　　年　　月　　日

摘 要

随着信息技术的不断发展，信息存储的方式也在不断的变革和更新。原有的中心化数据存储方式，存在着不可信、不透明以及清算程序复杂、用时冗长的缺点，已渐渐不能满足对效率需求日益上升的信息时代。各种行业都急需寻找一种新的存储技术来解决中心化数据存储带来的问题，区块链技术就是一种能够解决上述问题的存储技术。因此，设计基于区块链的安全支付平台来保证旅游景区交易的可信、透明以及提高交易的速度。

基于区块链的安全支付平台包括使用密钥管理机制来验证用户身份；设计权限管理智能合约控制不同角色的操作范围，保证高权限操作的安全性；设计零确认机制来实现现场购票功能，省略大量区块链节点确认时间，提高购买门票的交易速度，同时保证交易数据的透明、可靠性。

在使用Protocol Buffers对结构化数据进行序列化的过程中，为解决Map类型序列化后顺序改变的问题，设计了转换函数将Map类型转换成数组类型；密钥管理机制中用椭圆曲线算法由私钥生成公钥和钱包地址，将公钥作为验证用户身份的唯一标识；零确认机制中利用零确认标识区分现场购票和在线购票，在智能合约中产生和使用零确认标识并将过程记录在区块链上。

安全分析和测试结果表明，基于区块链的安全支付平台使旅游景区的交易变得可信、透明，在线购票相比第三方购票节省约一天的时间，大额度转账相比银行等交易平台节省一至三天的时间，具有更高的交易速度。

关键词：区块链，信息存储，安全支付平台，序列化，密钥生成

Abstract

With the continuous development of information technology, information storage methods are constantly changing and updating. The original centralized data storage mode has the shortcomings of not credible, not transparent and the liquidation of accounts is complex. Therefore, a variety of industries are in desperate need to find a new storage technology to solve the problem of central data storage. Blockchain technology is a solution to the above problems of storage technology. The secure payment application based on blockchain is designed for the travel company to ensure the transactions credible, transparent and improve the speed of the transaction.

The secure payment application based on the blockchain includes the key management mechanism to verify identity of user. The authority management is designed to intelligent control the operation scope of different characters to ensure the security of the high privileged operation. The zero confirmation mechanism is designed to realize the on-site ticket and saves confirm time to improve the speed of the speed of the transaction, while ensuring the transparency and reliability of transaction.

The conversion function is designed to solve the problem of unordered Map type in serialization with converting the Map type into array type. The key management mechanism uses the Elliptic Curve Cryptography algorithm to generate the public key and user address from private key, the public key is uniquely identified for authentication of user identity. Zero confirmation mechanism uses zero confirmation flag to distinguish between the on-site ticket and online ticket, generates and uses the zero confirmation flag in smart contract and records the process on the blockchain.

Safety analysis and test results show that the transaction on secure payment application based on blockchain are more credible and transparent. Moreover, the time of buying online tickets compared to other applications saves about one day, and the time of transferring large amount of money compared to other platforms saves one to three days.

Key words： Blockchain, information storage, secure payment application, serialization, key generation

目 录

1 绪论

近年来随着科技日益广泛的运用到人们的实际生活中，大量信息与隐私数据需要存储在网络上，如何能提供安全、透明以及更有效率的交易方式已经成为信息界和金融界共同关注的热点。本章首先介绍基于区块链的安全支付平台的研究背景，然后讨论国内外对基于区块链的交易平台的研究现状，举例若干以区块链为底层的数字货币应用，总结他们的优点和缺点，接着介绍本文的主要研究内容及主要实现的技术，最后梳理本篇论文的组织结构。

1.1 研究背景

随着网络的快速发展，金融行业和第三方支付软件迅速崛起，根据商务部 2015 年的统计，全国网络零售交易额为 3.88 万亿元，同比增长 33.3%，根据中国网络空间研究院在乌镇发布《中国互联网 20 年发展报告》显示，中国网络零售交易额跃居全球第一，而手机网购用户规模已达 2.70 亿人[1]。人们对网上银行、支付功能的要求越来越高，如何在更短的时间内处理更多的交易，同时又能保障较高的安全性，如何能让两个用户直接交易而省去中间第三方检验的繁杂过程，得到效率上的大幅提升，都成为交易改革的讨论热点[2]，因此基于区块链技术的账本数据存储有着重要的发展前景。

传统的网络交易需要借助第三方支付中介，对于用户来说，用户交易信息由中间商来保存在中心化的数据库中，用户需要信任中介商，才会使用该中介商提供的支付平台。支付公司为了保障交易的可信需要收集非常多的用户隐私信息，用以确定用户账户的可信，而对用户来说透露越多的隐私信息就越不安全。近几年私密信息泄露的情况时有发生，其范围包括医疗保健、法律和房产等涉及资产的行业，根据《2015 年度数据泄露调查报告》报告称：2015 年，全球 61 个国家出现 79790 起数据泄露事件，其中 2122 起已经得到确认[3]，我国多家知名酒店数据泄露，其中包括万豪酒店、喜达屋酒店等，这些酒店的数据泄漏导致所有曾在这些酒店消费的客户订单信息以及个人信息可以被黑客轻松的获取，被想要利用这些信息的人一览无余，这种用户隐私信息的泄露无疑是对顾客的敏感信息的不负责任，同时也对酒店的名声和市场价值的损害。对于交易中介平台来说，为了保障账目的可信性，清算大量的交易信息需要高昂的费用和冗长的时间，尽管如此，在巨大的交易量下，完

全的避免错误是不可能的，所以又需要付出更多的费用和时间来弥补这些错误，这会对用户造成不必要的资金浪费和时间损失。

另一方面，传统的中心化的管理方式[4]，是由银行或者第三方支付公司完全监管账目，这些中介公司记录的交易过程对用户是不可见的，账本数据的透明和可信难以保障。因此，随着信任危机的出现，众多研究者想要建立一种无需信任的、分布式的隐私保护中介平台，尽管现在已经有许多用户隐私保护算法，例如差分隐私[5]、k-anonymity[6]、t-closeness[7]，也有分布式存储系统，例如 Google File System[8]、BigTable[9]等，但是没有一个是分布式、无需信任的实际应用[10]。去中心化的数字货币早在 1982 年由 Chaum 在其论文中提出[11]，接着出现了点对点货币[12]，然而直到比特币的出现才引起全世界研究者的重视[13]。总结来说，现今的交易平台存在下列问题：

1．无法解决信任风险。如果用户想要通过某个机构进行交易，就必须先信任这个机构，该机构拥有操控用户资金的权利，只有信任该机构能保障用户的隐私和交易隐私信息，同时信任该机构的结算系统，才会使用该系统，但仍然要承担一定的风险。

2．银行会计清算系统需要较长的时间。一般金融理财产品，产品购买到期后都不能立即得到收益，需要 3-5 个工作日的清算时间，银行大账目转账也需要等待 1-3 日，跨境支付则需要更长的时间。

3．交易额度的限制。事实上因为清算的难度，很多微小额度的交易都难以进行，而大额度交易也有较长的清算时间。

4．中心化监管方式。随着对效率的追求和对数据安全的需求越来越高，中心化的监管方式是自上而下的监管模式，需要层层递进，效率上不够快捷。而在数据存储上又容易出现某个节点被侵入的危险，如果被侵入的是中心节点，更会造成巨大的危害。

5．交易数据不透明。对于许多的金融机构和政府机构，保密数据是非常重要的，但是对于普通的投资者来说，数据的透明性却是最重要的，数据透明能够使用户了解每一笔资金的来去，保障自己不受到交易机构的欺骗。

6．容易造成用户隐私的泄露。通常金融机构需要收集较多的用户隐私信息来确保使用该账户的用户的合法性和安全性[14]，在大数据时代利用网络的各类信息元素

可以拼凑和查询出任何私人信息包括医疗健康信息[15]、聊天记录等，这样无疑也增加了用户隐私泄露的风险。

综上所述，如何解决现金交易系统存在的数据不透明、清算效率低等问题已经成为金融界和信息技术界共同关注的热点话题，世界需要用很长的时间去普及一种全新的交易模式，来适应对效率和安全要求越来越高的交易用户。

1.2 国内外研究现状

从比特币开始，数字货币已经引起了国内外经济学家以及计算机学者的关注[16]，也渗透在各大经济市场当中[17][18]。区块链技术提供了一种分布式、透明和基于拜占庭算法的交易机制[19]，使用智能合约的机制自动完成复杂的协议[20]，有望成为新一代互联网的基础设施，包括支持匿名在线服务和数字资产[21]。国内外已有一些基于区块链信任机制的应用，例如最早提出区块链技术的比特币，加入智能合约机制的以太坊，支持私有链和联盟链的超级账本，花旗银行推广的花旗币，以及蚂蚁金服的爱心捐赠平台。

1.2.1 区块链的产生

随着网络技术的发展，网络支付已经成为一种重要的交易手段，金融机构通过网络技术实行更有效率的支付方式。而现有的支付方式大多是建立在对金融机构和软件的信任基础上的应用，虽然大部分交易都得到了安全保障，但仍然存在着一定的交易风险，比特币是一种能有效降低这种交易风险、提高交易效率的交易形式。区块链技术最初诞生于比特币，是比特币的底层结构。比特币最早出现在一篇由中本聪发表的论文中[2]，它使用了P2P的网络结构，使相互交易的双方能够直接通信，免去了中间第三方交易机构的一系列清算过程，同时利用数字签名来防止双花的问题，这种没有发行机构的交易，避免了第三方机构对货币的操纵。第一个比特币交易发生在2009年1月，到2011年6月为止吸引了大约一万的用户，产生了650万的比特币交易量。比特币使用去中心化的网络结构，将交易过程和数据存储在分布式数据库上，所有的记录的存储都需要经过验证过程[22][23]，想要改变单个节点的数据，必须得到其他节点的确认，同时这个确认过程也是挖掘比特币的机制[24][25]，最早是在B-money中提出的[26]，这种基于P2P网络、去中心化的系统结构保证了交易的自由性和安全性。比特币是一个匿名系统[27]，在这个匿名系统中，使用私钥和公钥的映射

来保证用户交易的安全，私钥只有用户自己知道，交易平台用公钥验证用户的身份。比特币能够吸引大量用户的原因也在于比特币低廉的支付成本，完全的自动化免去了繁琐的手续费，使众多用户可以在各种各样的平台上挖掘比特币。许多科技公司已经开始接受使用比特币购买虚拟物品，比特币已经成为一种真正的流通货币。

2014年2月25日，比特币交易平台Mt.Gox下线，比特币的市值迅速的下滑，造成比特币的迅速衰落的原因，其一是因为交易平台Mt.Gox被攻破，其中85万个比特币凭空消失，比特币的网络虽然能确保交易的安全性，但是比特币交易平台不足以保护用户的隐私安全[28]，其二是因为区块链处理数据的速率与区块大小和区块间隙有关，得不到较大的提升[29][30][31]，其三是比特币的商业价值没有得到控制，币的价值更适合收藏升值，而不适合在网络中做虚拟交易。

但是由于比特币在网络交易中的优势，对比特币的存储模式区块链的研究并没有因为比特币的衰落而停止，在比特币的基础上，扩展了如Bitcoin-NG[32]一类新的区块链协议，区块链的应用市场也在不停的拓展。

1.2.2 区块链与智能合约

智能合约是由以太坊提出的一个概念，一套以数字形式定义的承诺，包括合约参与方可以在上面执行这些承诺的协议，执行过程透明可跟踪、不可攥改。如果说比特币是分布式账本，它的交易符号是比特币，那以太坊就是一个没有交易符号的分布式平台，它在比特币的基础上增加了一个核心的智能合约的概念[33]。最初的比特币机制没有可扩展性，它是一个只能运行比特币交易的应用，如果有新的开发商想要在区块链的基础上开发新应用，需要完整的重做区块链。以太坊是一个提供给应用商的平台，开发者只需要支付一定的以太币，在这个平台上建立不同的合约，就能运行不同的运算，形同不同的业务，它不再只是记录交易，而在合约中加入更多的参数信息。通过建立自己的块链，在每一个区块中保存一个表示当前每个地址的平衡账目的清晰的“状态树”和一个表示当前区块和上一个区块间交易的“交易表”[34][35]，以太坊解决了可扩展性问题。以太坊支持的业务范围包括：投票、金融交易所、知识产权管理、众筹、人工智能[36]等。以太坊提供了一个平台，能够让区块链更简单便捷的支持众多的业务，不再需要自己设计全部的区块链存储，为了支持更多的应用，以太坊的合约提供尽可能简单的合约形式，使得任何程序员都能写出能在以太坊上运行的程序。智能合约将被允许在持久内存中存储数据，这样的内

存加上图灵完备的脚本语言将使在单个合约中编码一个完整的货币成为可能。相比于没有智能合约的比特币，有了智能合约特性的以太坊未来的发展方向更广，能支持更多的应用和、更大的交易量，再加上本身效率快的特性，会获得更长远的优势。

但是以太坊也有明显的缺点，即应用本身的代码和应用产生的代码都在一个区块中，长时间的积累数据量增大会造成区块的膨胀。同时在利用合约形成应用的过程中，多个应用和合约可能被同一机构控制，这又会造成和现有金融机构相同的经济现象，由国家或者一家机构控制以太币整个系统，这样一来区块链交易自由性不复存在。

1.2.3 私有链形式的区块链

比特币、以太坊等项目是公有链形式的项目，公有链的特点是无保密性、无法溯源、确认时间长、无最终性、吞吐量低，这些无法满足商业用途的需求，于是就出现了私有链形式的区块链。超级账本提出了支持基于私有链、联盟链的区块链形式，是一种新的适合商用的区块链应用，比公有链形式的区块链项目更注重隐私和保密性。超级账本是基于区块链的企业级分布式账本技术，用于构建各种行业的商业应用平台，具有模块化、高性能和可靠性，同时提供了商业友好许可[37]。相对于比特币的数字货币系统、以太坊的通用公有链平台，超级账本提供的是通用联盟链平台，由Linux基金会主要管理，使用公开或者私有的网络，能为开发商提供隐私性，也继承了以太坊智能合约的理念， 支持使用Go、Java等多种开发语言编写智能合约，而在超级账本上把这种智能合约叫做链上代码，以此来区分区块链上的合约和书面合约。允许企业以即插即用的方式使用共识机制和服务。从超级账本项目发起后，不断有新公司加入超级账本的阵营[38]，其中包括三星的IT服务子公司、中国重型机械制造商三一重工、华为等大型企业。对超级账本贡献巨大的fabric子项目，提供了基本的隐私性、机密性和可审计性，使用PBFT（Practical Byzantine Fault Tolerance）共识算法[39]，提供了交易的机密性，权限管理和控制功能，同时分离了共识机制和记账的职能，其中节点可以动态的伸缩，拥有可升级的智能合约，有望提升吞吐量。而另一个项目sawtooth Lake，最早是由Intel主要贡献和主导的区块链平台，支持全新的共识机制PoET（Proof of Elapsed Time）。

现有的区块链有一些商用缺陷，比如无法监管、缺乏隐私保护等，超级账本目标是创建一个开源的区块链社区，针对现有的区块链的不完善机制进行改革和规范，

让不同的区块链开发商都提供自己的开源代码，以此来吸引更多的开发商，同时促进区块链的技术发展，创造一个企业级的开源分布式记账框架和代码库。目前私有链和联盟链形式的超级账本正在迅速的吸收更多的公司和项目，各类项目也在规范和壮大中，有望为更多的私有和联盟社区提供安全保护机制，让更多的开发商实现合作共赢，促进区块链发展。

1.2.4 区块链与数字货币

目前许多金融机构和银行开始研究以区块链为基础的数字货币，其中比较有代表性的一家银行是花旗银行，其前身是“纽约城市银行”。经过近两个世纪的发展和并购，花旗银行已经成为了美国最大的银行之一。据《国际财经时报》网站报道，作为银行界的巨头，花旗银行正在研究三套基于区块链的分布式记账技术[40]，这三套技术相互独立，都处于正在研究和实验的阶段，并没有流通真实货币，花旗银行的创新实验室主管摩尔说，研究这些系统的目的不在于立马能运用到实际场景中，而是保证其数字货币技术处于领先水平，能够随时用区块链数字货币的技术抓住新的机遇。摩尔表示，可以将区块链作为跨境转账的方式，形成一个内部的挖矿网络，各个节点可以对网络进行维护也可以利用网络完成计算工作。花旗银行发表的演示文稿中提到，他们认为数字货币是一种必然趋势，未来一定会发生，但是用哪种特定的货币还不一定，比如比特币的前景就不太明朗。

1.2.5 区块链的应用前景

区块链所涉及的业务范围，已经从简单的虚拟币交易扩展到政府管理、食品安全、公益事业等。2016年7月，蚂蚁金服在杭州举办的首届全球XIN公益大会上表示，支付宝的爱心捐赠平台即将使用区块链技术上线。因为区块链的透明性和安全性，可以解决公益事业不透明的问题，为公益透明度打开新的局面。蚂蚁金服表示，要将每一笔款项的生命周期都记录在区块链上，这样的机制下，爱心人士捐赠的每一笔款项都能知道来去，每一笔款项都将被公开，包括管理费在内。这种模式有利于公益事业自给自足，不需要第三方繁杂的确认过程，让公益事业更透明，也能让更多人放心参与到公益事业中，有助于解决整个公益社会的信任问题。未来无论是即时通信、社交网络、媒体，还是银行、电商、公共服务等等，都将被区块链思维重构，一个崭新的区块链应用时代正在到来。

1.3 主要研究内容

根据现有区块链底层技术和区块链平台，综合分析多种技术的优缺点，选定在基于超级账本fabric子项目的私有链的基础上开发一个综合旅游景点交易平台，使门票、转账等交易数据具有可信性、透明性、安全性，同时留出可扩展接口，方便扩展游戏、租车等可以使用通用货币的第三方应用。具体研究内容和实现功能可概括如下：

第一，针对密钥的生成和存储，设计合适的加密手段和存储方法，使用尽可能可靠的方式保管密钥。

第二，针对权限管理模块，设计判断不同角色权限的方法，减小高权限操作被不合法利用可能性。

第三，针对交易模块，包括在线购票、现场购票和转账，实现现场购票零确认合约机制，使游客购票的等待时间尽可能的减少。

第四，解决不同平台传输过程中出现的序列化Map结构不固定的问题。

第五，详细设计密钥生成算法和保存机制。

第六，详细设计零确认的产生和使用方式。

1.4 论文组织结构

论文的章节结构具体安排如下：

第一章绪论。这一章首先描述了研究背景，简述了本课题的来源，研究目的及研究意义，接着论述了国内外基于区块链的应用平台的实现情况，最后介绍了本论文主要研究内容，需要解决的主要问题，需要实现的关键技术。

第二章支付平台系统设计。首先描述系统的总体构架，介绍系统底层、中间层和上层的组织结构，接着从模块化的角度，分析系统主要实现的功能并详解每个功能的具体使用流程，紧接着介绍了系统底层智能合约的运行机制。

第三章系统主要技术点。这一章首先概述了系统使用的编程框架和思想，以及框架的优势，接着解决字典序列化不固定导致的存储问题，接下来对地址生成与密钥管理做详细的分析，包括地址和密钥的生成算法、加密备份和安全恢复功能，最后介绍用零确认机制实现现场购票功能。

第四章安全分析与系统测试。这一章首先分析了基于区块链的安全支付平台相对于其他中心化的平台的优势与特点，然后进行功能数据测试和分析。

第五章总结与展望。最后一章对本文现有的研究工作进行总结，提出需要改进之处，并对未来的扩展开发工作做出展望。

论文最后是致谢和参考文献。

2 支付平台系统设计

这一章主要对基于区块链的安全支付平台做一个概述，设计系统的主要功能，分析使用该系统以及与系统有关联的各种角色。首先分层概述系统的总体架构，梳理每一层结构具体运用的机制和实现的功能；然后介绍系统功能模块，列出每个功能模块的操作流程图；接着从底层区块链的角度，运用区块链技术智能合约的特质，设计适用于不同模块的智能合约机制。

2.1 系统总体架构

本系统的主要思路在于利用区块链的中心化的存储模式，为旅游公司提供一个可信、透明的账本存储结构，利用区块链上的合约特点，设计新的智能合约完成权限管理；为游客提供更加快捷的购票功能、转账功能。在众多区块链的开源项目中，基本都是在比特币的基础上进行的改进，包括架构、智能合约、共识机制等。而本系统是在超级账本的基础上进行改进的，原因是在于超级账本除了拥有区块链底层的结构以及智能合约服务之外，还实现了公示算法插件化，还实现了会员机制，所谓的会员机制即是使用户只能在注册后进行网络的访问和交易。超级账本是基于银行设计的，而本系统应用场景是旅游景点和周边娱乐，在区块链的交互过程中涉及的功能更为广泛。

如图2.1所示，系统主要分为交互功能层、平台内核层与区块链层。直接与用户进行交互的是交互功能层，交互功能层主要具有的功能有注册登录、交易、信息查询。注册登录包括注册、登录、查看地址二维码，在用户第一次注册时，就会随机生成密钥和用户地址，用户地址使用二维码形式显示在客户端上，通过用户地址二维码进行购票和转账；交易包括转账、在线购票、现场购票，其中现场购票采用了零确认合约机制，缩短了用户购票的交易时间；信息查询包括余额查询、订单查询、个人信息查询和账单查询。平台内核层主要任务是保障数据的安全传输过程，是介于与用户直接交互的功能层和区块链信息存储层之间的系统结构，平台内核层主要具有的功能有密钥管理和权限管理。密钥管理包括密钥生成、密钥备份、密钥恢复，权限管理是利用区块链上的智能合约管理不同角色，保证其操作的合法性。最底层是去中心化的区块链结构，是在超级账本的基础上进行改进的存储结构，主要存储旅游公司订单信息和账户数据，包括分布式账本结构、各类合约和共识机制。由于

区块链是以时间顺序写入的链式存储结构，所以具有可追溯性，该存储模式可以减少审计员的账目清算时间。

图2.1 系统构架

本系统主要适用于对账目数据有透明与可信要求的旅游公司。项目的目标是建立一个以旅游公司为主，同时可以加入游戏、第三方服务商等周边业务的信息透明、可信的生态圈。

2.2 系统模块设计

本节按功能将系统分为三个模块，分别是密钥管理模块、权限管理模块、交易模块。一个模块有可能涉及到多个功能，但同一个模块的功能具有关联性，接下来介绍每一个模块的具体使用流程。

2.2.1 密钥管理模块

为了保障数据在传输过程中不被修改以及用户账户不被盗用，系统采用了非对称加密的技术。私钥是只有自己知道的密钥，公钥是所有人都知道的密钥，当一个对象需要发送数据时，首先需要用发送方的私钥加密传输的数据，接收方再用发送方的公钥解密传输数据，当一个对象需要接收数据时，发送方需要用接收方的公钥给

数据加密，接收方收到数据后再用自己的私钥解密数据。为此需要设计一套密钥管理模块，密钥管理模块主要分为密钥生成以及地址的生成、密钥备份和密钥恢复。

用户注册系统时，根据用户手机号随机生成一个私钥，再使用椭圆曲线算法由私钥生成对应的公钥，经过一系列的哈希计算之后，用base58表示法生成用户地址[41]。用户地址是用来转账和交易的账户地址，地址呈现给游客的形式并不是一连串的密码，系统中地址变换成两种可视化、便于记忆的形式，其中一种是地址二维码，二维码的作用是支持面对面转账功能、现场购票功能等一切需要提供用户身份的场景，另一种形式是手机号，因为用户地址与手机号是一一对应的，所以将手机号和地址对应存储在服务器中，可以直接输入手机号进行转账工作。而又因为密钥生成地址的算法是不可逆的，破解椭圆曲线算法难度极高，目前还没有人能够通过地址破解私钥，所以在这样的密钥机制下，信息交互的安全可以得到保障。

为了避免用户因意外丢失手机而丢失私钥的情况，系统提供了密钥备份和密钥恢复的功能，该功能需要用户信任服务器。当用户不能信任服务器，可以选择自己保存密钥，当用户选择信任服务器时，可以将密钥加密备份后存入云端，密钥将使用用户口令加密。当用户更换手机后需要重新恢复密钥时，曾经存入云端备份过的用户可以输入口令获得密钥，没有存入云端备份的用户，需要自行输入密钥。

2.2.2 权限管理模块

整个系统目前设置了六种角色分别为管理员、游客、景区服务、商户、注册机构、审计员，如图2.2描述了各个角色之间的关系，每个角色都有不同的功能权限，通过设计权限管理智能合约实现对权限的控制与验证工作。各个角色具体拥有的功能权限如下：

1.管理员：管理员是拥有最高权限的角色，可以直接管理区块链节点，也可以为系统中其他角色授权，在区块链数据层初始化时，管理员就制定一份初始权限合约，在合约里分类标记出所有角色以及他们所拥有的权限，并将这些角色的公钥保存在合约里，当用户提出操作请求时，先用智能合约里的公钥验证请求者的角色，再验证请求是否符合其角色权限。

2.游客：游客是通过手机客户端与区块链上的节点进行交互的，管理员授予游客权限，在交互操作过程中，先验证该游客是否拥有该操作的权限，游客的功能权限包括：

（1）注册登录功能，注册时需要向注册机构发送请求，登录时需要将账号密码用私钥加密后传送至区块链节点进行验证。

（2）购票功能，用户通过此功能进行园区的门票购买，包括在线购票和现场购票。

（3）转账功能，用户通过扫描二维码或者输入手机号进行转账。

（4）个人信息查询：随时能查到自己的订单信息和余额。

3.景区服务：景区服务是针对景区服务商提供的身份，主要通过与游客建立智能合约完成售票功能以及提供游戏等，是智能合约主要内容的制定者与修改者，景区服务的功能权限包括：

（1）注册登录功能，注册时需要向注册机构发送请求，登录时需要将账号密码用私钥加密后传送至区块链节点进行验证。

（2）购票订单处理，包括在线购票和现场购票订单的查询，为了完成现场购票功能景区服务需要制定一个合约机制，让用户购票后不必等待交易确认过程就可直接使用门票，缩短交易时间。

4.商户：商家指的是借用该系统的扩展接口提供景区周边服务的角色，商家将第三方应用数据保存在区块链上，可以在该应用上使用虚拟币，商家有权限构建自己的智能合约内容，但需要经过管理员审查与测试，商家的功能权限包括：

（1）注册登录功能，注册时需要向注册机构发送请求，登录时需要将账号密码用私钥加密后传送至区块链节点进行验证。

（2）转账功能，商家可以与游客通过转账功能进行交易。

（3）信息查询功能：商家可以查询带自己服务编号的区块链交易。

（4）提供服务：商户有权限制定和更改自己的智能合约，并分配合约运行带来的虚拟币的奖励。

5.注册机构：注册机构主要包括对游客和商户个人信息的验证和存储，在用户丢失账户密码，可以验证身份找回账户和虚拟币。

6.审计员：区块链的时间链式结构使得其内的账本容易清算和追溯，这有利于减少审计员的账目清算时间。

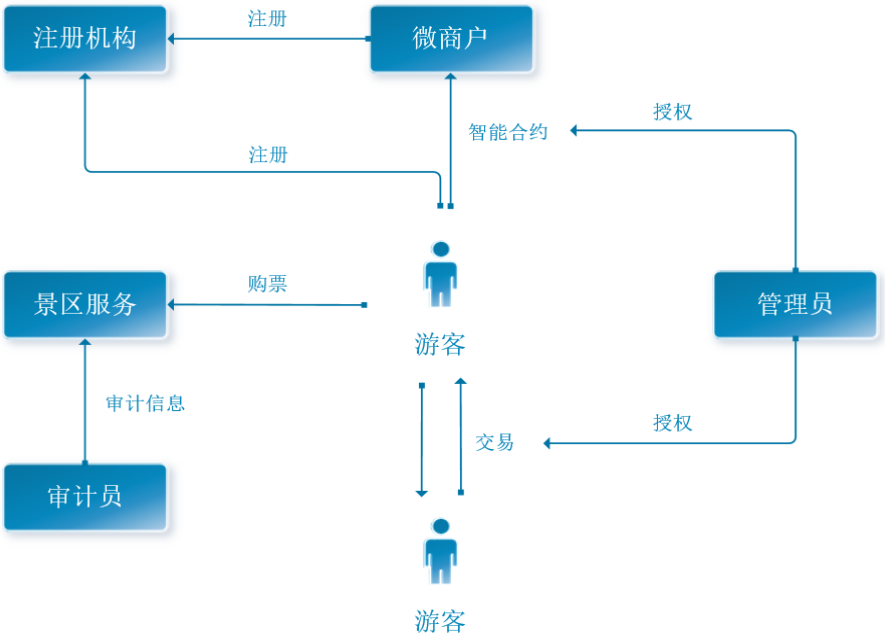


图2.2 角色关系图

2.2.3 交易模块

交易模块分为转账和购票两部分，购票是本系统的主要功能之一， 游客购票有两种方式，在线购票和现场购票。在线购票执行普通的区块链交易流程，即游客先购买门票，等待区块链上所有节点验证过程，验证游客地址里存在足够的虚拟币，大约需要二十分钟以内的时间（六次确认时间），所以在线购票需要游客提前二十分钟左右购票。现场购票设计了零确认合约机制，省去了确认过程，让用户买到票后可以立即入园，缩短了交易时间，如图2.3为现场购票的流程图，现场购票的具体流程如下：

1．游客用现金进行现场购票时，区块链节点无需先验证账户余额是否充足，而是给游客提供一个由旅游公司密钥生成的临时地址，临时地址也属于游客的账户。

2．临时地址中有足够支付门票的虚拟币，使用临时账户购买门票不需要确认余额，可以直接完成交易。

3．购票成功时临时地址中用于购票的虚拟币被加锁，同时游客与旅游公司签订零确认合约，合约内容是：在规定时间内使用门票，即可完成合约，扣除相应虚拟币，若在规定时间内未使用门票，虚拟币解锁，不会扣除游客账户虚拟币。

在转账功能中，小额交易使用零确认方式立即到账，而大额度交易采用六个确认

的方式，大约二十分钟到账。本系统相比于其他系统转账的优势在于：第一，用户密钥的机制和去中心化的区块链账本技术保证了交易的可信与透明，第二，区块链的可追溯特点使中间商很容易查询到任何时间的账单信息，而且能保证其可信，省略了大量的账本清算时间。

在本论文第三章第四小节，将具体阐述零确认机制的实现方法。

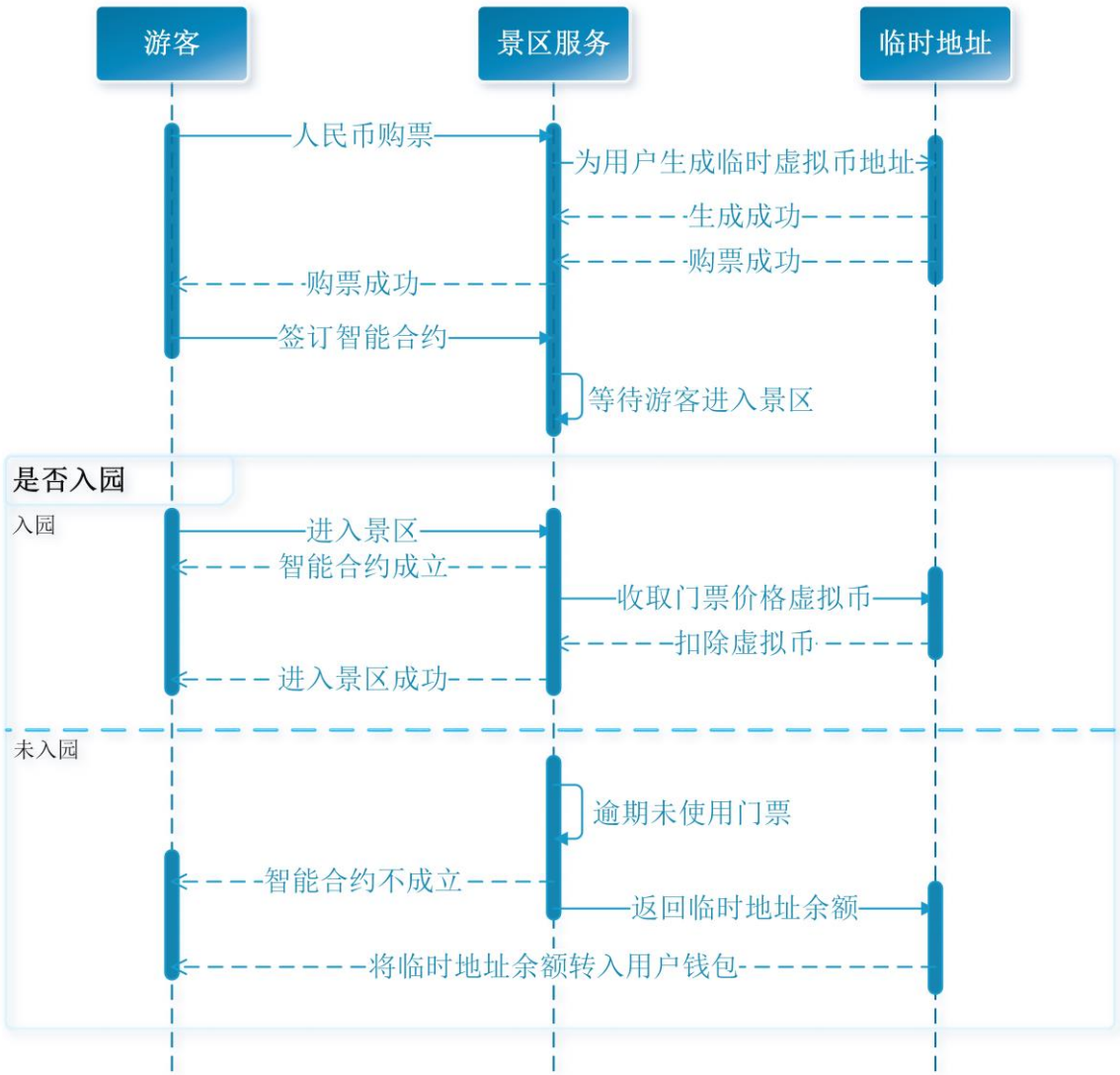


图2.3 现场购票序列图

2.3 智能合约设计

“智能合约”这个术语至少可以追溯到1995年，是由多产的跨领域法律学者尼克·萨博提出来的，他在发表在自己的网站的几篇文章中提到智能合约的理念[44] [45]，

他定义：“一个智能合约是一套以数字形式定义的承诺，包括合约参与方可以在上面执行这些承诺的协议。”一套合约定义了参与方同意的权利和义务，这些定义的权利和义务决定了合约的本质和目的，以一个销售合约为例，卖家承诺发货和卖家承诺支付货款就是一种合约，而智能的意义在于合约必须以数字形式写入计算机可执行的代码当中，因为合约的成立和运行都是由计算机自动执行的过程。

智能合约是区块链最重要的特性，也是区块链能成为颠覆性技术的原因，各大银行和企业想要利用区块链来发行数字货币都是因为区块链拥有智能合约，智能合约让编程货币和编程金融更加简单，会使金融界的结构发生巨大的变化。

在本系统应用过程中，使用超级账本Fabric的底层区块链结构，超级账本的智能合约又叫链上代码，链上代码是一个去中心化的交易程序，在验证节点上执行。链上代码是在一定的规定时间内完成的合约，如果超过规定的时间，表示合约不成立，就不会将交易数据写入区块链中，区块链上的账本数据也不会有任何改变。在交易进行时，区块链上的数据会维持原先的状态，直到所有节点都达到共识，变量才会提交到数据库，之后整体状态会改变，如果有一个节点没有达到共识，数据库就不会改变，整个区块链也不会更改。

本系统主要是在fabric智能合约的基础上设计实现权限管理合约和购票合约。权限管理合约中，与用户达成的协议是：当用户提出的请求符合其角色的权限范围时，允许用户执行角色，否则拒绝用户提出的请求。在权限管理合约代码中需要存储所有角色的权限范围，同时分类存储所有特殊角色的公钥来验证用户的身份，以完成整个合约的协议内容。购票合约中，与用户达成的协议是：当用户购买门票时，检测其是在线购票还是现场购票，若为在线购票，即验证所有节点的后完成交易，若为现场购票，即给用户分配一个拥有足够虚拟币的临时地址，达成零确认机制，省去节点确认时间直接完成交易。

2.4 本章小结

本章主要首先介绍了基于区块链的安全支付平台的系统总体构架，论述了系统的主要适用场景和与传统系统相比的优势，说明了系统具有交互功能层、平台内核层与区块链层三个层次，并对每个层次的构造做了详细的图解。

接着按照功能将系统分为三个主要模块：密钥管理模块、权限管理模块、购票模块，并具体介绍了每个模块的使用流程图，在权限模块说明了与本系统相关的六

种角色，阐述了各个角色之间的关系和主要的功能权限。

文章最后介绍了区块链的最主要技术特质智能合约，并阐述如何利用智能合约完成主要功能的设计思路。

3 系统主要技术点

这一章主要论述基于区块链的安全支付平台的技术点。首先介绍系统使用的编程构架以及该构架的优势；然后详细描述信息在手机客户端和区块链后台间交互产生的序列化问题，并给出解决方案；接着讨论地址生成的过程和密钥管理方式；最后论述使用零确认机制实现现场购票的方法和过程。

3.1 安卓架构

在设计和搭建一个安卓应用程序时，安卓构架可以提供基本的程序构架，其中存在着很大的灵活性，这种灵活性为程序员带来了很大的便捷，但也会带来很多问题，包括会导致程序中出现过大的类、命名方案不一致、不匹配以及缺少部门结构等，这些问题会导致应用程序难以测试、维护或者扩展[45]，安卓构架蓝图可以为上述存在的设计问题提出技术解决方案。传统的安卓应用是MVC（Model-View-Controller）结构的，如图3.1所示，MVC即表示模型、视图和控制器，一般模型为class函数，视图是xml文件，控制器由Activity实现，这样一来，Activity中就占据了大量的代码，不利于测试、维护和扩展，安卓构架蓝图是由谷歌设计的官方新架构MVP（Model-View-Presenter）[46]，如图3.2所示，是对MVC架构的改进，所谓的MVP即为模型、视图和主导器，模型实现数据存储和业务逻辑，视图提供与用户的交互接口，主导器相当于于传统的控制器，但是比控制器更为灵活。

图中可以看出视图和模型没有直接关联，是通过主导器进行交互的，而视图和主导器是可以相互调用的，主导器又可以调用模型的函数接口。调用过程和顺序可以概括为视图中的所有功能都要委托给主导器，主导器再调用模型完成任务，最后主导器调用视图更新结果反馈在界面上。

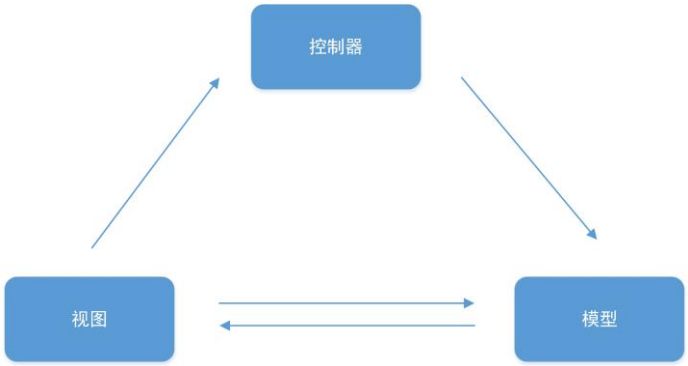


图3.1 MVC架构

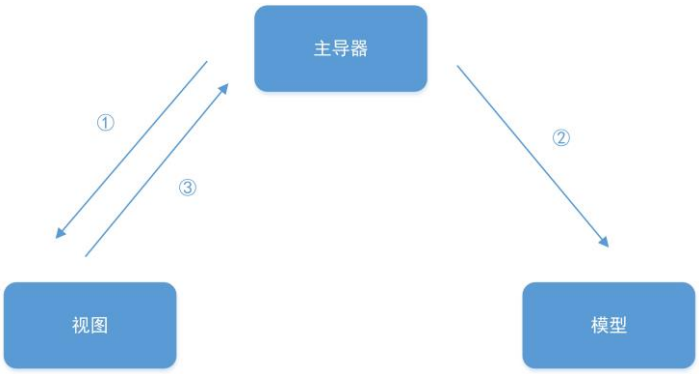


图3.2 MVP架构

3.2 解决 Map 序列化不固定问题

账户信息要传输到区块链上需要进行序列化，而数据结构map的序列化后结果是不固定的，会导致区块链上的各个节点的数据不同，从而导致区块链各节点无法达到共识，严重时会引起节点崩溃。为了解决Map序列化不固定的问题，设计了以下解决方案。

3.2.1 序列化传输

在数据传输的过程中，要将数据从一个平台传输到另一个平台，为了保证数据在传输过程中的稳定性，以及在更换平台和编程语言之后的数据可用性，在数据传输过程中进行序列化是必不可少的过程。序列化传输指的是将数据信息以及状态信息转化为可以传输的形式，以某种存储形式使自定义的对象能适应任何环境，同时使程序具有可维护性。序列化是不同的代码可以查看相同的数据实例，代码在本地计算机上存储时会被授予各种权限，但是从网络上下载的代码不会存在权限，而序列化后的数据实例不需要权限就可以访问查看。随着互联网的飞速发展，不同的计算机设备种类越来越多，在这些设备中传输数据进行通信的需求也越来越高，而这些设备之间通信需要采用不同的协议，序列化和反序列化就是协议的一部分，所谓序列化即是指将数据对象存储到文件、缓冲区等介质中，或者是转化成二进制的方式进行传输的过程，而反序列化即是指将序列化的文件、缓冲区等介质中的数据转化成数据对象，或者是将二进制串转化成数据对象或者数据结构的过程。在不同的编程语言中，数据结构、对象和二进制串的表示方法都不尽相同，例如Java是一种完全面向对象的编程语言，所以里面的数据都是类的实例化对象，但是C++不是完全面向对象的编程语言，它的数据结构是用struct表示，而对象是用class表示的。在Java

中二进制是以byte[]表示的，是一种原生态的数据类型，Java中的String数据类型就不能存储二进制数据，但是在C++语言中String就是以‘\0’结尾的二进制数据，是可以直接被传输层使用的。

序列化拥有不同的协议，每种协议都具有优缺点，在选择序列化协议的时候应该考虑需要序列化的根本原因，结合具体的适用场景综合对比各个协议的优势与缺点，选择尽可能合适的方案。在选择适用的序列化协议时具体有如下衡量标准：

1.通用性：通用性有两方面的定义，一方面是指对通讯双方平台的支持，另一方面是指是否是一种常用的序列化协议，常用的序列化协议通常具有更完善的跨平台、跨语言公共包，通讯双方的平台工程师也更容易学习。

2.成熟性：一个协议不可能支持所有的语言和平台，在不同的平台和语言的需求有所冲突的时候，必须做出选择牺牲对一些不常用平台的支持，这样的协议才会是成熟的，一个成熟的协议是从最初的制定到实施，要经过大量的测试，需要考虑到方方面面的，尽可能的提高整个协议的稳定性和高质量服务。

3.可读性：序列化之后的数据是二进制形式，所以在序列化以后工程师是无法读取数据信息的内容的，也无法验证序列化的正确性，往往需要很长的时间去进行调试和测验，因此为了让使用序列化协议的人员能够方便的识别序列化的结果，往往序列化协议还需要提供一个反序列化的程序，这也是十分复杂的。如果序列化之后的数据是可读的，这将大大提高工程师的测试效率。

4.性能：性能包括时间和空间的开销，在时间上复杂的序列化程序会导致需要较长的时间解析，在空间上序列化的过程需要在数据结构中加入新的信息字段，如果因此使空间占用量过大，会造成网络流量的浪费和拥塞。

5.可扩展性：网络时代的迅速发展使可扩展性变得尤为重要，在应用到实际场景时，根据不同的需求可以加入更多类型字段的服务，又不影响旧的协议，将大大提高协议的可用性。

6.安全性：当序列化后的数据在互联网上传输时，需要考虑安全性的问题，一般是被限制为基于HTTP/HTTPS的80和443端口，有成熟的传输层协议的支持会使序列化传输协议的可用性增高。

3.2.2 Protocol Buffers 结构数据序列化的问题

Protocol Buffers是谷歌公司2008年公布的一款开源项目[47]，其目的是支持各种编

程语言和平台的结构化数据串行化，可以用作数据存储和通信。从发布至今得到了众多开发工程师的认可，已经成为主流通用的数据串行化方法，经过了大量的实际应用测试，支持几乎所有语言的拓展包，是一个成熟的数据描述语言，并且使用了代码生成器生成了非常简单易懂的代码来表示存储的数据，甚至可以直接更新这些数据[48]。想要定义一个数据结构，首先需要在.proto文件中定义数据结构的内容，包括数据的值类型和字段名字，值类型可以是基本数据类型字符串、布尔型、整型等等，数据结构还可以分层级来存储，分为必选字段、可选字段和重复字段。尽管ProtoBuf（Protocol Buffers）数据描述语言拥有上述优势，但也有不适用于本系统的缺点，在序列化过程中ProtoBuf没有保留map类型数据内默认顺序，在反序列化的时候则出现了顺序不固定的情况，这可能对于其他系统来说是可行的，但是对于本系统的区块链节点确认来说，是不可不解决的问题。如图3.3定义了一个简单的账本形式的数据结构，有基本的地址、余额和需要传输的状态信息，而需要传输的状态信息存放在map类型中，但是在ProtoBuf的序列化过程中，map类型的值会在反序列化的时候出现不固定的问题，因为map本身是没有顺序的，所以在每一次反序列化时会得到不同的结果，导致状态信息的混乱问题。

为了解决Map数据类型序列化不固定的问题，设计了一个message转换函数和一个message反转换函数。 message转换函数将map里的数据进行排序，再将排序后的map数据按顺序放入数组中，并返回如图3.4所示的拥有数组的另一个message，生成数据内容一致但数据类型不同的结构，再将新的结构化数据进行序列化，而message反转换函数是将数组类型的数据重新解析成map类型，返回一个拥有map类型的新的message结构。

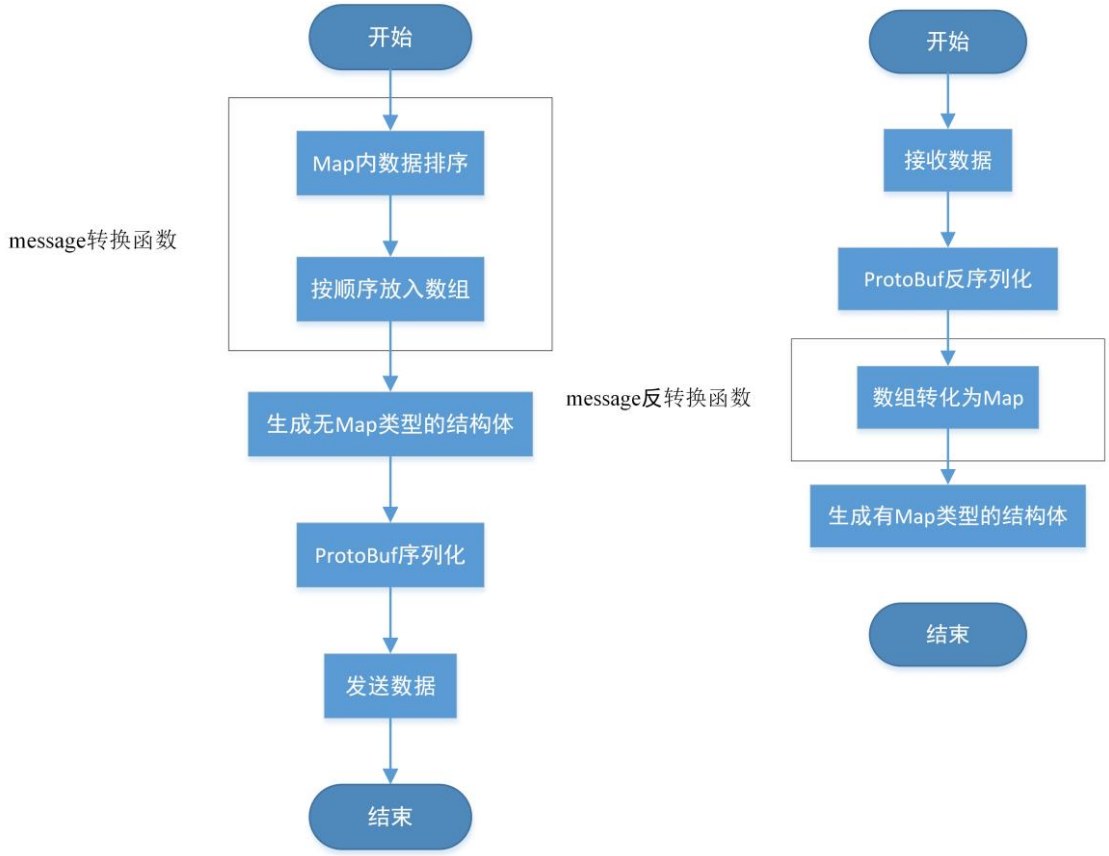
```
message Account{
    string addr = 1;
    float balance = 2;
    map<string, TX.TXOUT> txouts = 3;
}
```

图3.3 带有Map类型的结构体

```
message AccountSlice{
    string addr = 1;
    float balance = 2;
    repeated TxoutMap txoutmap = 3;
}
```

图3.4 带有数组类型的结构体

在节点与客户端之间通信的过程中，首先发送消息的一方需要使用message转换函数将map类型转换成数组类型的数据结构，接着序列化数据结构，最后通过网络发送数据至底层区块链节点；而在区块链上节点收到序列化数据之后，首先反序列化数据结构，再使用message反转换函数将数组类型转换成map类型，就可以使用收到的数据了。图3.5描述了数据发送和数据接收时的序列化过程。



(a)数据发送　　(b)数据接收

图3.5 数据序列化过程

3.2.3 数据存储与访问过程

使用客户端访问区块链上的数据的另一个问题是数据在进行序列化之后，某些字节无法使用post方式传输，为了防止这些字节在传输中变成无效的ASCII码，在发送数据之前，还需要使用最常见的用于传输的编码方式之一的base64编码方式来编码数据，将较长的标识信息转换为可读的字符串形式。如图3.6所示数据存储与访问的具体过程为，发送数据方先将结构化数据使用ProtoBuf进行序列化，再使用base64编

码将序列化后的二进制数据加密为易于传输和识别的字符串，最后发送数据；接收数据方收到数据后先使用base64解密数据，再使用ProtoBuf进行反序列化，最后得到可读的结构化数据。

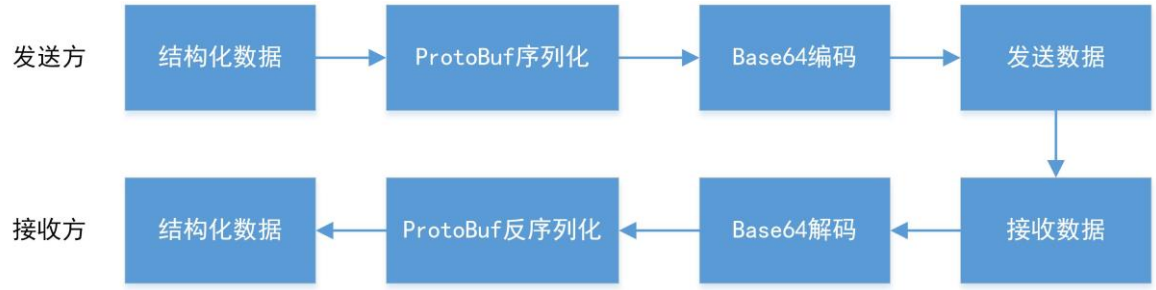


图3.6 数据交互过程

3.3 地址生成与密钥管理

为了保障用户账户的安全性，每一个用户在注册时就生成一组公私钥，通过公钥生成用户的钱包地址，钱包地址可用于花费虚拟币。在本系统中使用已经较为成熟的比特币私钥、公钥和地址的生成方法。

3.3.1 地址和密钥的生成

为了防止景区账本数据遭到盗窃而影响整个景区区块链系统的安全，必须严格验证每一个交易者的身份信息，而密钥和地址就是保障用户身份不被盗用的有效手段。在最早的区块链应用比特币平台中，生成公私钥的具体流程是：先随机生成一个字符串作为用户的私钥，在通过椭圆曲线加密算法ECDSA（Elliptic Curve Digital Signature Algorithm）生成公钥，使用公钥做SHA256变化，将得到的结果使用RIPEMD-160进行哈希计算，在结果前面加入地址版本号得到一个中间值，再次计算两次SHA-256哈希值，取结果中的前20个byte，将这20个byte加在中间值的后面作为校验，最后用base58将二进制地址变为字符串形式的地址。

本系统仿照比特币地址生成方式，生成150位的私钥和91位的公钥，如图3.7为地址生成的关键代码，主要的生成步骤包括：

1.提取用户的手机号信息加上字符串“_secret”和“_public”分别转换成二进制形式的pri和pub，此方法相对比随机生成二进制数的好处在于不会有相同的手机号重复注册，所以生成的私钥是唯一的。

2.使用PKCS8EncodedKeySpec函数加密pri为base64形式，产生私钥。

3.使用X509EncodedKeySpec函数生成pub的base64形式，使用椭圆曲线算法生成公钥。

4.将公钥进行SHA-256和哈希160的计算。

5.计算结果加上版本信息生成新的二进制串，并使用base58编码为字符串，得到最终的地址。

```
KeyFactory fact = KeyFactory.getInstance("ECDSA", "SC");
PublicKey publicKey = fact.generatePublic(new X509EncodedKeySpec(
        keyPair.getPublic().getEncoded()));
PrivateKey privateKeyc = fact.generatePrivate(new PKCS8EncodedKeySpec(
        keyPair.getPrivate().getEncoded()));

byte[] pub = publicKey.getEncoded();
byte[] secret = privateKeyc.getEncoded();


bytes = pub;
//SHA-256 哈希160
bytes = CryptUtil.sha256hash160(bytes);
CryptUtil.printHexString(bytes);

//base58编码
String address = CryptUtil.toBase58(bytes, version);
```

图3.7 地址生成关键代码

从比特币私钥、公钥和地址的生成过程可以看出，私钥是其中最重要的部分，公钥和地址都可以由私钥生成而来，而且整个生成过程是不可逆的，虽然从理论上讲是可以通过破解地址来得到私钥，但是破解的难度十分巨大，暂时没有能够逆推导的方法出现，在一定程度上来说，只要用户不泄露自己的私钥信息，别人就无法盗用其账户进行交易。由此也可以看出私钥的重要性，所以在生成私钥依赖的随机算法必须是安全的随机，所谓安全的随机就是指结果是不会重复的，如果生成私钥的算法不是安全的，那会导致有相同的私钥出现，从而会造成账本信息的错乱。而在本系统的使用手机号生成私钥的方式，避免了随机的不安全性，保障了私钥的唯一性。

3.3.2 密钥加密备份

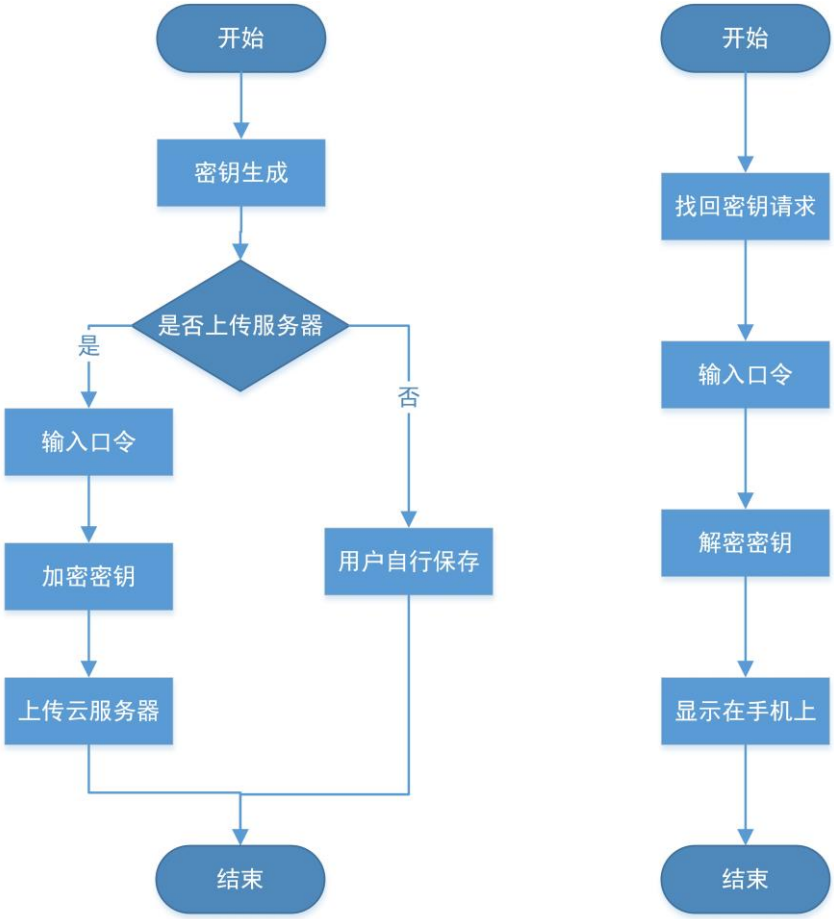
目前区块链的密钥都是以文件的形式存储在手机上，没有使用CA管理而是通过手机客户端管理证书，证书中的私钥是身份认证最重要的标识，如果私钥丢失，用户账户的安全就得不到保障，虚拟币很容易被盗取。以文件的形式存储在手机上，对于私钥的保存来说是不够安全和方便的，不够安全是因为私钥存放在手机中有可

能被其他软件窃取，也有可能因为手机损坏而丢失，不够方便指的是当客户更换终端设备，用户就需要重新输入一长串的私钥字符串。为此，一些用户会将私钥记录在笔记本上，有些会保存在云盘中，既能防止密钥被盗用又能防止密钥被遗忘。

在本系统中密钥是与手机绑定在一起的，所以当用户丢失手机导致密钥丢失时，区块链系统和手机客户端是无法帮助用户找回密钥的，只有用户自己知道自己的私钥，这就造成了一旦客户丢失密钥，客户的钱包地址就丢失了，其中的虚拟币也无法追回了，如此一来系统的整个钱包的实用性很低。在安全性与实用性的权衡之后，设计了密钥加密备份的功能，加密备份的功能不是强制性的，是需要建立在用户信任数据库安全的基础上的，用户也可以选择保存在硬盘或者记录在笔记本上。密钥备份的流程如图3.8(a)，当用户选择备份到云端数据库时，需要用户设置口令来验证其身份，这个口令与登录密码不同，设置好口令后，密钥将使用md5算法加密存储到云端数据库中。使用这样的密钥管理方法，在用户密钥丢失或者更换手机时，避免了手动输入长串密钥字符串的不便，可以直接从云端下载密钥部署到新的客户端中，当然这也牺牲了一定的安全性，如果想要完全的安全，用户可以自己记录私钥而不是信任第三方数据库安全。

3.3.3 密钥安全恢复

与密钥加密备份功能对应的，设计了密钥安全恢复功能，在用户需要恢复密钥到新的设备中时，具体流程如图3.8(b)，首先需要用户使用手机号和密码登录客户端账户，在功能选项中选择恢复密钥，此时数据库会首先验证用户是否是已登录的用户，再验证用户是否备份过密钥，如果用户已经登录并且备份过密钥，就让用户输入密钥备份时输入的口令，验证口令后下载并解码私钥到手机客户端中，并且完成公钥的生成和地址的计算，找回账户中的虚拟币。



(a)密钥备份　　(b)密钥恢复

图3.8 密钥备份与恢复流程图

3.4 零确认机制

目前的网络购票系统，在购买景区门票时都需要提前一天的时间，如果游客想要当天去景区游玩，就只能在景区门口现场买票而不能享受网络购票带来的优惠。本系统提供的在线购票只需要提前二十分钟购票，完全可以满足游客当天游玩的需求。同时，除了在线购票之外，还提供了一种现场购票的方式，现场购票不需要游客充值虚拟币，也不需要通过网银等支付手段，它是一种以现金支付的购票方式，其优势在于解决了区块链交易确认的等待时间长的问题，缩短了用户使用基于区块链的交易系统进行交易的时长。现场购票的实现方式是在智能合约中加入零确认机制，零确认机制的原理是在用户使用现金买票之后，智能合约为该订单做上一个标记，随后将分配一个临时地址到该订单的发起用户的账户中，这个临时地址中有足够支付现场门票的虚拟币，所以不需要经过节点确认用户余额，智能合约用临时地

址中的虚拟币进行买票，最后确认智能合约完成购票，节省了交易的多次确认的时间。下面主要介绍零确认是具体实现过程。

3.4.1 产生零确认标记

客户端会给区块链层发送两种标记请求，一种是invoke_produce请求产生零确认标记，一种是invoke_use请求使用零确认标记。在客户端处理用户请求后，将现场购票的订单请求信息封装好，订单请求信息中包括用户的地址、购票的类型以及invoke_produce请求类型。如图3.9为产生标记的关键代码，当智能合约收到客户端发送的现场购票订单请求信息后，首先使用base64解码消息，再进行数据的反序列化，接着检测账户地址和状态信息的有效性，检测数据是否有哈希冲突，如果上述检测都没有问题，就将带有标记的合约请求写入区块链中，这就是合约中产生零确认标记的过程。产生零确认标记后，合约分配有足够余额的余额地址给游客，并使用该地址中的虚拟币购买门票，生成订单信息，此时的订单是在合约中标记过的订单，订单中的门票在一天以内是可以使用的，如果游客在一天以内没有使用该门票，则余额地址中的虚拟币返还给游客，如果游客在一天之内使用门票，则在合约中执行零确认标记使用过程。

```
//base64解码
txDataBase64 := args[0]
txData, err := base64.StdEncoding.DecodeString(txDataBase64)
...
//反序列化
tx, err := ParseInstantTXBytes(txData)
...
//获取账户并检测账户有效性
instantAccount, err := store.getInstantAcount(tx.Addr)
...

//检测哈希冲突
_, ok := instantAccount.InstantTx[txhash]
...

//写入区块链中
instantAccount.InstantTx[txhash] = tx
err = store.putInstantAcount(instantAccount)
...
```

图3.9 产生零确认标记关键代码

3.4.2 零确认标记使用

当游客使用门票二维码入园时，客户端处理用户请求后，将入园请求封装好，入园请求中包括用户的地址、购票的类型以及invoke_use请求类型。如图3.10为使用标记的关键代码，当智能合约收到客户端发送的入园请求信息后，首先使用base64解码消息，在进行数据的反序列化，接着检测账户地址和状态信息的有效性，监测数据是否有哈希冲突，如果上述检测都没有问题，再检测状态信息中的零确认标记，判断这个标记是否在时间限定内，如果在购买门票的一天之内，则同意入园，如果超出一天，则拒绝入园，并将零确认标记删除，保证每个标记只能使用一次，最后将零确认使用过程写入到区块链中，此时合约完成。普通交易需要经过六次确认，但智能合约只需要一次确认时间，使用合约完成现场购票过程，大大的缩短了购票交易时间。

```
//base64解码
txDataBase64 := args[0]
txData, err := base64.StdEncoding.DecodeString(txDataBase64)
...
//反序列化
tx, err := ParseUseTXBytes(txData)
...
//获取账户，并检测账户有效性
instantAccount, err := store.getInstantAcount(tx.Addr)
...

//检测标记，判断这个标记是否在可用时间内
if instantAccount.InstantTx[tx.Txhash].Before > 0 {
    beforetime := time.Unix(instantAccount.InstantTx[tx.Txhash].Before, 0).UTC()
    if !beforetime.After(time.Now().UTC()) {
        delete(instantAccount.InstantTx, tx.Txhash)
        temp := fmt.Sprintf("The coin time is out: %v:%v", tx.Addr, tx.Scenic)
        return nil, errors.New(temp)
    }
}

//标记中钱是否足够
if tx.Usecoin > instantAccount.InstantTx[tx.Txhash].Instantcoin[tx.Scenic] {
    logger.Errorf("There is not enough coin to be use instant: %v:%v", tx.Addr, tx.Scenic)
    temp := fmt.Sprintf("There is not enough coin to be use instant: %v:%v", tx.Addr, tx.Scenic)
    return nil, errors.New(temp)
}
//删除标记
delete(instantAccount.InstantTx[tx.Txhash].Instantcoin, tx.Scenic)
if len(instantAccount.InstantTx[tx.Txhash].Instantcoin) == 0 {
    delete(instantAccount.InstantTx, tx.Txhash)
}
//提交修改到区块链
if err := store.putInstantAcount(instantAccount); err != nil {
    logger.Errorf("putInstantAcount error:%v", err)
    return nil, err
}
```

图3.10 零确认标记使用关键代码

3.5 本章小结

本章主要介绍了系统的主要技术点，包括一个非传统的安卓架构MVP，添加了主导器作为视图和模型之间的交互，相对于传统的MVC架构具有更好的测试性、维护性和扩展性。在不同的平台上传输数据时使用序列化进行数据的统一编码，在序列化的过程中发现了Map序列化不固定的问题，并设计函数将结构体中的Map数据排序后固定在数组中，解决了Map序列化不固定的问题。使用手机号生成私钥，再使用私钥生成公钥，利用公钥进行哈希变换生成用户钱包地址，私钥只有用户自己知道，保障了账户的安全性。最后描述使用零确认合约机制减少了现场购票的交易时间的具体方法。

4 安全分析与系统测试

本章首先分析目前的交易平台存在的安全风险，再分析本系统可以减小的风险类型，分析系统的可信与透明性；接着进行系统测试，主要是功能测试与性能测试；最后对本章进行总结。

4.1 安全分析

基于区块链的安全支付平台，其最大的优势的是交易信息的可信与透明性。除了拥有可信、透明的底层数据存储结构之外，系统还设计了密钥管理模块和权限管理模块来保障交易平台的安全性。用户数据泄露的事件在近几年中时有发生，2016年9月雅虎上至少五亿用户数据被黑客盗取，2016年10月根据乌云漏洞平台的漏洞显示网易邮箱数据泄露，泄露的信息包括用户的密码、生日、密保信息、聊天信息等。保证交易系统的安全，不仅关系到用户的利益，同时关系到一个公司的信用度问题。2015年12月苹果公司拒绝与美国警方合作透露用户隐私在分析本系统的可信与透明性之前，2016年上半年微软因为不满美国政府对其用户电子邮件的秘密搜查而起诉了美国政府。这些事件都能表明无论是对于用户还是公司来说，隐私保护都是非常重要的，而本系统的可信与透明性满足了用户和公司对安全的需求。在具体分析本系统可信和安全性之前，总结目前交易平台存在的安全风险如下：

1.中间商不可信。目前的交易模式都存在中间商，例如淘宝和支付宝，都需要用户信任支付平台，由支付平台处理交易中间的验证程序，一旦用户信任了不诚信的支付平台，就有可能丢失财物。

2.盗用身份。不健全的身份认证机制会导致用户的密码一旦被破解，使用该账户余额进行交易，甚至利用账户身份信息实施违法行为。

3.信息泄露。隐私信息泄露的事件层出不穷，目前各大网站中心化的数据存储方式，让黑客只需要攻破中心节点就能获取甚至上亿的用户信息。

4.数据造假。正因为中心化的数据存储方式，使得应用开发商可以随意更改应用数据，包括手游数据造假、直播观众量造假等，严重影响了观众的网络体验，欺骗消费者。

5.资金去向不明。政府部门公共资金去向不明，爱心捐款项目无法使民众信服，很难查询到每一笔资金的去向。

下面将分析本系统的可信性与透明性。

4.1.1 可信性

本系统是一种基于区块链的安全支付平台。首先，系统中的密钥管理模块采用了本地生成和本地保存密钥的模式，只有用户的手机保存着私钥，其他人没有私钥就无法冒充用户身份进行交易，除非用户手机丢失，但是系统提供的密钥备份功能，让用户在丢失手机后能很快将密钥输入新的手机中，丢失的手机里的账户将无法登陆使用，也无法查看到密钥，从而保证了进行交易的用户的真实性。使用椭圆曲线算法生成公钥和用户钱包地址，以目前的解码技术来说，没有人能够通过用户的公钥和地址逆推导出用户的私钥，所以用户的私钥是安全的，用户的身份是可信的。

其次，系统中的权限管理模块中保存了高权限用户的公钥，只有当高权限用户使用其私钥发送操作请求时，才能通过权限合约的验证。一般用户是无法进行高权限操作的，黑客也无法利用有效的身份进行攻击和窃取用户信息，在很大程度上增强了系统的可信性。

第三，区块链的底层结构本身就是可信的数据存储模式，其分布式系统中的每个节点都记录了全部的账本信息，如果要更改信息，需要广播给所有节点，经过所有节点的核对之后，再记录到节点中。分布式账本结构免去了用户对中间商的信任需求，让交易双方直接联系而不需要中间商的确认。

由上述可知，本系统在一定程度上解决了目前交易平台存在的安全风险中的前三项：中间商不可信、盗用身份、信息泄露。

4.1.2 透明性

本系统是一种基于区块链的安全支付平台，其底层区块链的最基本特点是将带有时间戳的交易数据记录在链式区块结构上，具有时间的可追溯性，可以查询到每一笔资金的过去和未来走向。分布式的账本结构使得如果有人想篡改区块链的上数据就一定会被记录在区块链上，想要不被发现的更改数据是不可能的，所以区块链上的交易都是真实的不可篡改的。虚假的游戏用户数据和不明确的捐款去向在区块链的结构中是不可能发生的，用户可以体验一个无欺诈的网络世界。

由上述可知，本系统在一定程度上解决了交易平台存在的安全风险中的后两项：数据造假、资金去向不明。

4.2 系统测试

系统测试是软件生命周期的一个重要的阶段，是保证系统质量的关键步骤。本节首先介绍系统的测试环境，再进行系统的功能测试和性能测试。

4.2.1 测试环境

所有测试都使用安卓客户端进行，后台交易数据存储在服务器上，服务器上运行区块链底层存储结构，其中硬件环境和软件环境配置如表4.1所示。

表4.1 环境配置

		详细参数
硬件环境	服务器	4台，intel Xeon E5-2640V3*2 (2.4GHZ)
	手机	高通骁龙820，3GB RAM，32GB ROM
软件环境	服务器	Ubuntu Server 16.04，Fabric 0.6，Go，Docker，rocksDB，MySQL5，PHP5，Nginx，phpMyAdmin
	手机	Android 5.0

4.2.2 界面测试

界面测试中使用AndroidUITestRunner来测试单个UI界面的演示效果，测试布局的准确性，已经再不同型号的手机上的适配性，AndroidUITestRunner是一个类似Junit的测试接口，提供用户编写测试函数的接口，自动生成列表界面，打开列表中的一项就是运行一个界面的测试。本系统的主要界面包括注册登录界面、主页界面、用户信息界面、购票转账界面、订单信息界面。

测试界面满足下列结果：

1.窗体大小合适，空间布局合理，没有文字被覆盖。

2.窗口调用流畅，可以返回上一个界面。

3.在输入错误时有相应的提示信息。

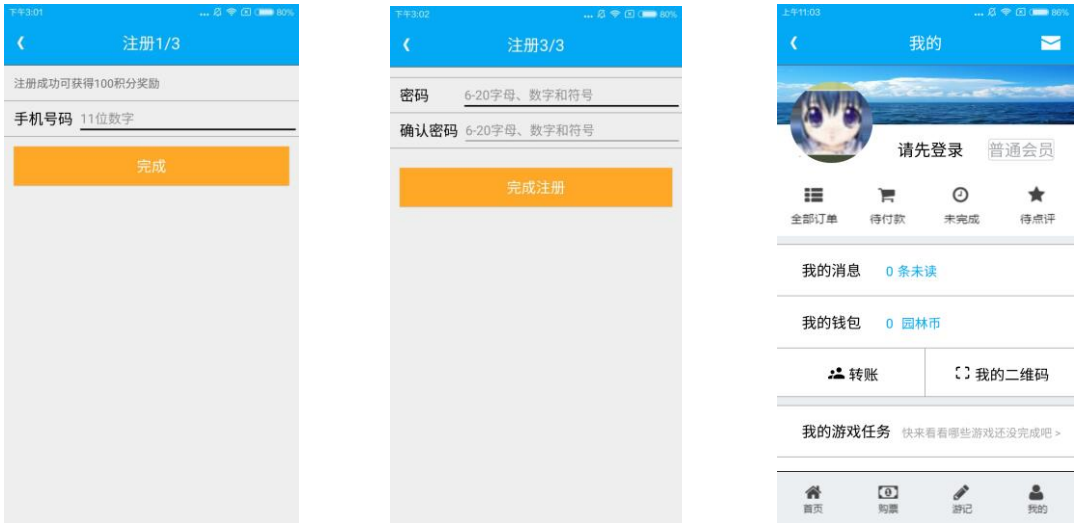
4.4G网络情况下，页面刷新响应时间低于1秒。

5.输入法没有遮盖输入框，输入法切换正常。

6.在网络不稳定的条件下，页面没有意外退出。

7.安装时可覆盖旧版本更新。

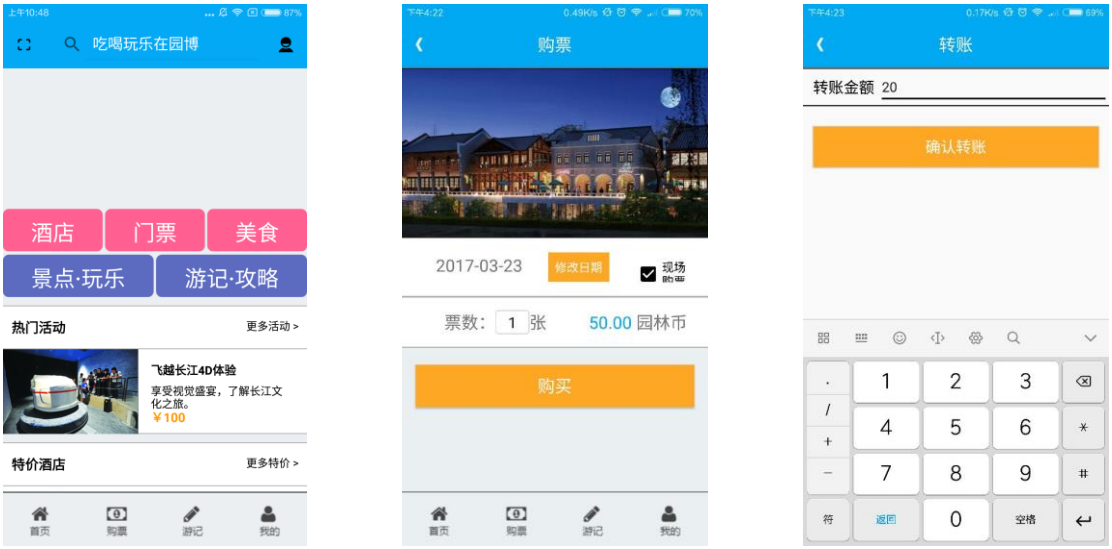
如图4.1(a) (b)为注册界面，注册后界面会跳转到用户信息界面，如图4.1(c)用户信息界面上有订单信息，用户的钱包余额，用户的地址信息在注册时本地生成并以二维码的形式显示在客户端中，点击用户信息界面的我的二维码按钮就可查看用户钱包地址。



(a)注册页面1　　(b)注册页面2　　(c)用户信息界面

图4.1 注册和用户信息界面测试

如图4.2(a)主界面留有美食、游记等扩展接口，如图4.2(b)购票界面可以勾选现场购票组件，如图4.2(c)转账界面是在扫描对方二维码之后跳转到输入转账金额。

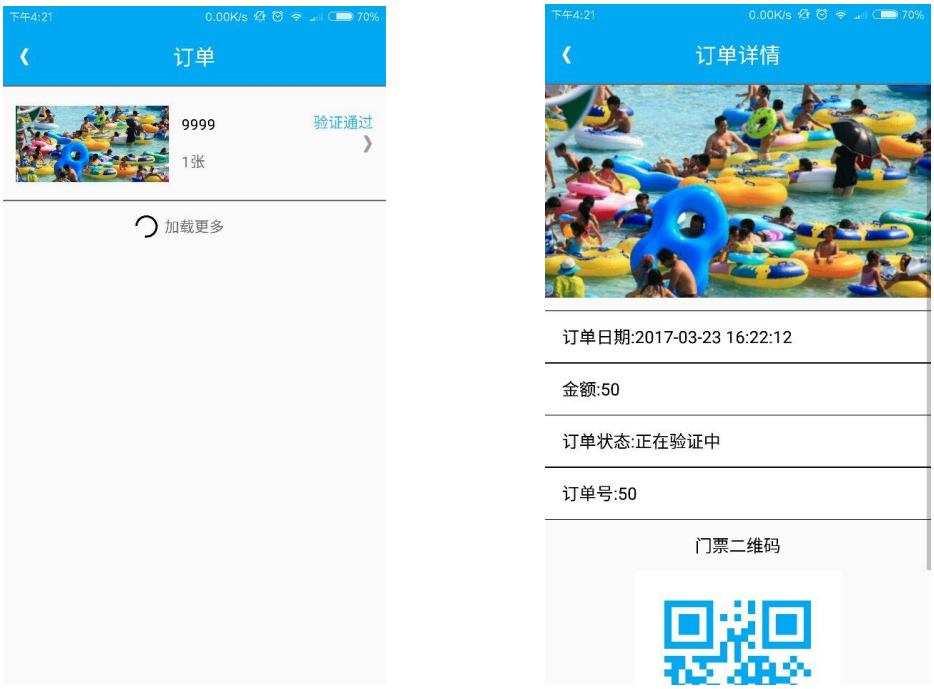


(a)首页　　(b)购票　　(c)转账

图4.2 首页、购票和转账界面测试

如图4.3订单信息中可以查询到用户的购票信息，显示门票二维码，使用门票二

维码在规定时间内进入景区。



(a)订单　　(b)订单详情

图4.3 订单和订单详情界面测试

4.2.3 功能测试

1.注册登录：用户通过注册界面进行注册，需要考虑到用户名和密码的格式和字符要求，为此编写的测试用例和结果如表4.2所示。测试用例1和2属于边界值分析，取上点，用例3、4、5、6属于边界值分析，取离点，用例7、8、9测试错误的信息

2.转账：用户转账时需要输入转账金额，对转账金额的测试用例设计如表4.3所示。用例1、2、3、4属于边值分析，用例5属于测试错误的信息。

3.在线购票：用户选择在线购票，等待二十分钟以内的确认时间。在交易量较少的情况下，测试的等待时间为1分钟左右。

4.现场购票：用户选择现场购票，立即能够查询到订单信息。在一周内入园，入园时扣除虚拟币，超过时间则返还余额。

表4.2 注册用例

软件名称：基于区块链的区块链的额安全支付平台		软件版本：V1.0	
需求描述	用户第一次使用客户端时使用手机号注册，测试注册的结果		
用例ID	输入/动作	预期输出/响应	允许偏差
1	填写符合要求的数据，密码为最大长度12	显示注册成功，并跳转个人信息页面	不允许
2	填写符合要求的数据，密码为最小长度6	显示注册成功，并跳转个人信息页面	不允许
3	填写手机号为12位数字	显示不是有效手机号	不允许
4	填写手机号为10位数字	显示不是有效手机号	不允许
5	填写密码长度为13位字符	显示密码长度不在6-12位以内	不允许
6	填写密码长度为5位字符	显示密码长度不在6-12位以内	不允许
7	填写手机号为非数字	显示不是有效手机号	不允许
8	填写密码为非数字和字母	显示存在无效字符	不允许
9	填写已经注册过的手机号	显示账号已经注册	不允许

表4.3 转账用例

软件名称：基于区块链的区块链的额安全支付平台		软件版本：V1.0	
需求描述	转账时输入转账金额		
用例ID	输入/动作	预期输出/响应	允许偏差
1	输入整数0	显示不是有效金额	不允许
2	输入整数10000	显示转账成功	不允许
3	输入小数，小数点后1位	显示转账成功	不允许
4	输入小数，小数点后2位	显示不是有效金额	不允许
5	输入10001	显示金额超过限额	不允许

4.2.4 性能测试

对比现有的第三方软件购票系统的购票时间，如图4.4所示，在线购票方式中，本系统的在线购票只需要提前二十分钟购买，等待六个确认的时间，比特币系统和本系统同样都需要等待六个确认时间，而目前第三方购票软件都需要提前一天购买门票；现场购票方式中，本系统使用零确认机制，基本不需要等待时间，而比特币

的现场购票和在线购票一样需要六个确认的时间，需要等待二十分钟完成交易，第三方软件中不具备完成现场购票的功能，无法比较。

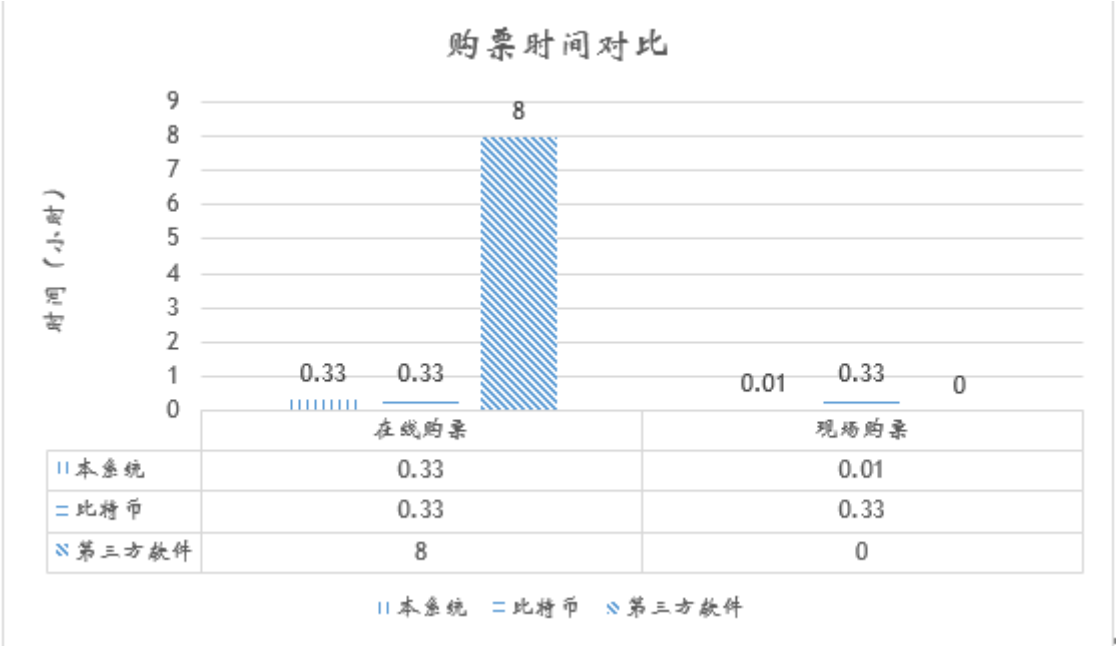


图4.4 购票时间对比

对比现有的银行转账系统，如表4.4所示，本系统大金额转账时间在20分钟之内，而普通银行跨行大金额转账需要1天的清算时间，跨境支付时需要1-3天的时间清算。

表4.4 转账时间对比

	银行	本系统
跨行大金额转账	1天	20分钟以内
跨境支付	1-3天	20分钟以内

由此测试可以得出结论：基于区块链的安全支付系统在门票购买时，对比传统的比特币和第三方购票软件，可以为用户节省大量时间；在转账时，对比银行可以为用户节省1至3天的时间。

4.3 本章小结

本章对本系统进行了安全分析和系统测试，首先从可信性和透明性两个角度分析了系统可以有效抵御的安全问题，接着对系统进行全面的测试，包括界面测试、功能测试和性能测试，在界面测试中使用AndroidUITestRunner测试了每个页面的可用性与美观性，在功能测试中编写了测试样例测试了系统中的输入与输出，在性能

测试中与其他同类对象对比了完成购票时间和转账的时间。在经过了安全分析与系统测试之后，得出了基于区块链的安全支付平台是一个透明、可信并且节省交易时间的交易系统。

5 总结与展望

5.1 总结

随着信息技术的飞速发展，互联网已经渗透到生活中的方方面面，包括金融、医疗、管理工作都离不开信息技术的支撑。在这样的快速发展下，网络环境也变得日益复杂，数据泄露的事件时有发生，其中有一些造成了巨大的经济损失，大量的用隐私数据存储在与互联网联通的设备中，为了保全这些数据的安全，许多公司开始探索全新的存储方式，致力于建立一个可信的网络生态圈，而区块链就是其中一大热点。

用区块链技术解决信任问题已成为构建可信互联网的发展趋势，因此，本系统设计了一种基于区块链的安全支付平台，其设计目的是为旅游公司提供一个透明、可信的门票订单存储系统，编写智能合约完成管理工作和交互业务，为旅游公司提供一个可追溯的账本存储模式，节省清算账目的时间，为游客提供一个无欺诈的娱乐、交易环境。

论文主要设计与实现了一种基于区块链的安全支付平台，首先介绍了平台的整体架构，分为交互功能层、平台内核层和区块链层，交互功能层主要负责与游客进行交互，平台内核层负责处理账户的安全问题，区块链层负责存储数据、制定智能合约和完成共识。本系统使用了基于MVC安卓构架改良的MVP构架，提高了系统的可维护性和可扩展性。为了解决数据在不同平台间传输的不匹配问题，使用了ProtoBuf序列化数据结构，在序列化过程中发现了序列化Map类型顺序不固定的问题，使用变换函数将message结构体变换成不含Map类型的数据结构，解决了传输中的问题。介绍了数据在不同平台间交互的处理过程。为了保障用户账户的安全性，根据用户的手机号为每个账户生成了唯一密钥和账户地址，并设计了密钥加密备份和安全恢复的功能。使用零确认的智能合约机制，减少了用户购票的交易时间。

本论文设计与实现的功能主要包括以下几部分：

1. 为了向旅游景区提供安全的支付系统，本文提出了基于区块链的数据存储方式，将交易数据写入可信、透明的区块链中，保证了门票交易数据的可靠性。

2. 在进行交易的过程中，为了保障用户身份的可靠性，为每个用户生成密钥来验证身份，并将私钥信息存储在手机客户端本地。

（1）使用比特币地址生成算法，即椭圆曲线算法，生成用户的私钥和公钥，根

据公钥生成用户的交易地址，保证交易中每个消息的可靠性。

（2）为了避免用户丢失私钥而造成虚拟币的丢失，设计了密钥备份与密钥恢复的机制，让用户选择性的将私钥使用md5加密后备份到云端。

3. 设计了一套权限管理方法，将特殊身份的公钥和权限范围存储在智能合约中，使用智能合约中的公钥来验证用户身份，判断不同角色的不同权限，减小高权限操作被不合法利用的可能性。

4. 设计了两种购票方式，包括在线购票和现场购票，并将交易数据存储在区块链上，保证数据的可靠性。

（1）在线购票执行普通的区块链交易流程，即游客先购买门票，等待区块链上节点验证过程，验证游客地址里存在足够的虚拟币。

（2）现场购票设计了零确认合约机制，利用分配临时地址进行购票，省去了区块链上节点确认过程，让用户买到票后可以立即入园，缩短了交易时间。

5. 进行了转账和购票实验，展示并分析了本设计的转账和购票方式与其他应用的时间对比结果。

5.2 展望

目前本系统的数据量较小，确认时间较短，系统在运行过程中，数据量会不断增加，确认时间也会变长。在今后拥有大量客户和数据同时访问时，系统确认时间能否控制在较短的时间内还有待进一步的探索和测试。

目前本系统面向游客的实用功能仅限于购票和游戏，结合系统底层区块链结构特点，还可能为用户提供更多可信与透明的功能业务。现在已有一款AR游戏计划加入本平台成为首款第三方应用，使用虚拟币作为游戏币，将游戏信息存储在区块链上，营造一款无造假数据、对玩家透明的可信AR游戏。除了游戏之后，区块链的智能合约特质使得该平台可以容纳各种不同的业务，包括订酒店、购买电影票、爱心捐款等，针对第三方业务，平台计划实现智能合约模板，让不同的应用设计师都能快速的掌握智能合约的使用方式，设计出适合自己的智能合约。对于旅游公司来说，本系统可以将区块链数据查询接口开放给公司审计人员，设计合适的统计数据接口，为公司提供便捷的清算账目服务接口。利用本系统的可信、透明的特质，设计实现更多的接口和实用功能，有望在未来构造一个可信的旅游景区生态圈。

致 谢

时光荏苒，研究生的两年学习生活转眼就快结束了，在这两年的学习生活中，不仅学习到了很多技术知识，更重要的是掌握了学习的方法，养成了良好的学习习惯。两年前本科毕业时，我还只是个只学习过基础知识只会写简单项目的本科毕业生，来到实验室之后，跟着老师同学一起科研一起做实验一起想创新点，才发现学海无涯，学习的道路上还有着无数值得钻研的地方。感谢在这两年的学习中陪伴和指导我的老师和同学们，帮助我克服难题，帮助我找到前进的方向，感谢实验室提供了良好的学习环境，让我能专心学习知识。这两年的研究生学习不仅让我收获了技术知识、增强了动手能力，还让我收获了友谊和为人处世的道理。

首先要感谢我的导师金海老师，金海老师严谨治学，把握着实验室科研的主要方向，对实验室大小事务尽心尽力。在研究生的培养上您严格把关，从开题到论文都给予我们诸多指导，还会定期邀请国内外知名学者来实验室进行学术讲座，为实验室创造了良好的学术氛围。在平时的交流中您亲切幽默，坚持举办一年一度的元宵晚会，让我们体会到实验室的欢乐和温馨。感谢金海老师为实验室和学生们付出的一切，愿您工作顺利，家庭美满。

感谢我的课题组老师邹德清老师，您作为安全组的组长老师，参与到安全组的每一个项目当中，把握着最新的科研创新点，为学生的项目规划发展方向。尽管十分忙碌也坚持每周与学生开会，关注学生的科研进度，还请国外知名教授指导学生的科研工作与论文。在生活中您平易近人、和蔼可亲，对待我们很有耐心，从来不对我们发脾气，并时刻关注着我们的心理健康。感谢您为安全组和学生付出的一切，祝您身体健康，阖家幸福。

感谢本论文项目的负责老师代炜琦老师，您在项目开发过程中严格督促我们，在大小事务上亲力亲为，和我们一起讨论每一个细节，在我们科研工作遇到瓶颈和不顺时，用心开导我们，关注每一个学生的情绪。正因为您的认真与努力，才确保我们项目的每项工作都如期完成。在此我表示非常感谢。

感谢实验室的其他老师。感谢章勤老师和王多强老师对我们开题和论文工作的指导，感谢刘英书老师为我们创造良好的实验室环境。祝福实验室的所有老师。

感谢项目组里的张玮炜、王晨龙、肖德山、陈庆祥、陈天阳、包庆华在项目开发和论文撰写过程中给予我很大的帮助。感谢安全组里的李珍师姐、齐汉超师兄、

杨巨师兄、赵健、邓之君在研究生学习期间对我的科研与实验任务的帮助，非常感谢与大家一起奋斗的时光，希望大家在以后的科研和工作中能取得优异的成绩。

感谢我的室友常拴霞和吴思尧以及硕士1505班的所有同学，感谢你们在日常生活中对我的照顾，让我度过了美好而欢乐的两年。

感谢研究生期间的饭友杜佳雨、杜铁、黄波、陈维志、邓奎，你们是我在实验室最好的朋友，也是知己，是我在研究生期间收获的宝贵财富。

感谢我的父母、家人和男友，你们是我坚实的后盾，也是我奋斗与前进的动力！

最后感谢答辩委员会的老师们认真审阅我的毕业论文并提出宝贵的意见，谢谢！

参考文献

[1] 佚名. 中国互联网 20 年发展报告. 网络传播, 2015, 12: 26-28

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008, [Online]. Available: http://www. bitcoin. org/bitcoin. pdf

[3] 刘晓韬, 廉小伟, 类延春. Verizon 2011 年度数据泄露调查报告摘述. 保密科学技术, 2011, 12: 17-22

[4] Swan Melanie. Blockchain thinking: The brain as a Decentralized Autonomous Corporation. in: Proceedings of the 2015 IEEE Technology and Society Magazine, 2015. 27-29

[5] Dwork Cynthia. Differential privacy: A survey of results. in: Proceedings of the 2006 International Conference on Theory and Applications of Models of Computation, 2006. 1-12

[6] Sweeney Latanya. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(05): 557-570

[7] Li Ninghui, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. in: Proceedings of the 2007 IEEE International Conference on Data Engineering, 2007. 106-115

[8] McKusick Kirk, Sean Quinlan. GFS: evolution on fast-forward. Communications of the ACM, 2010, 53(3): 42-49

[9] Chang Fay, Dean Jeffrey, Ghemawat Sanjay, et al. Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 2008, 26(2): 4

[10] Schaub Alexander, Bazin Remi, Hasan Omar, et al. A trustless privacy-preserving reputation system. In: Proceedings of the 2016 IFIP International Information Security and Privacy Conference, 2016. 398-411

[11] Yang Beverly, Garcia-Molina Hector. PPay: micropayments for peer-to-peer systems. in: Proceedings of the 2003 ACM conference on Computer and communications security, 2003. 300-310

[12] Luu Loi, Teutsch Jason, Kulkarni Raghav, et al. Demystifying incentives in the consensus computer. in: Proceedings of the 2015 ACM SIGSAC Conference on Computer and Communications Security, 2015. 706-719

[13] Xu Xiwei, Pautasso Cesare, Zhu Liming, et al. The Blockchain as a Software Connector. in: Proceedings of the 2016 Working IEEE/IFIP Conference on Software Architecture, 2016. 182-191

[14] Zyskind Guy, Nathan Oz, Pentland Alex. Decentralizing privacy: Using blockchain to protect personal data. in: Proceedings of the 2015 Security and Privacy Workshops, 2015. 180-184

[15] Swan Melanie. The quantified self: Fundamental disruption in big data science and biological discovery. Big Data, 2013, 1(2): 85-99

[16] Bonneau Joseph, Miller Andrew, Clark Jeremy, et al. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. in: Proceedings of the 2015 IEEE Symposium on Security and Privacy, 2015. 104-121

[17] Meiklejohn Sarah, Pomarole Marjori, Jordan Grant, et al. A fistful of bitcoins: characterizing payments among men with no names. in: Proceedings of the 2013 ACM conference on Internet measurement, 2013. 127-140

[18] Watanabe Hiroki, Fujimura Shigeru, Nakadaira Atsushi, et al. Blockchain contract: Securing a blockchain applied to smart contracts. in: Proceedings of the 2016 IEEE International Conference on Consumer Electronics, 2016. 467-468

[19] Castro Miguel, Barbara Liskov. Practical Byzantine fault tolerance. in: Proceedings of the 1999 Symposium on Operating Systems Design and Implementation, 1999.173-186

[20] Kosba Ahmed, Miller Andrew, Shi Elaine, et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. in: Proceedings of the 2016 IEEE Symposium on Security and Privacy, 2016. 839-858

[21] Miers Ian, Garman Christina, Green Matthew, et al. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. in: Proceedings of the 2013 IEEE Symposium on Security and Privacy, 2013. 397-411

[22] Dwork Cynthia, Moni Naor. Pricing via processing or combatting junk mail. in: Proceedings of the 1992 Annual International Cryptology Conference, 1992.

139-147

[23] Gifford David. Weighted voting for replicated data. in: Proceedings of the 1979 ACM Symposium on Operating Systems Principles, 1979. 150-162

[24] Chaum David. Blind signatures for untraceable payments. in: Proceedings of the 1983 Advances in cryptology, 1983. 199-203

[25] Kleinberg Jon, Prabhakar Raghavan. Query Incentive Networks. in: Proceedings of the 2005 IEEE Symposium on Foundations of Computer Science, 2005. 132-141

[26] Kiayias Aggelos, Koutsoupias Elias, Kyropoulou Maria, et al. Blockchain mining games. in: Proceedings of the 2016 ACM Conference on Economics and Computation, 2016. 365-382

[27] Reid Fergal, Martin Harrigan. An analysis of anonymity in the bitcoin system. in: Proceedings of the 2013 Security and privacy in social networks, 2013. 197-223

[28] Androulaki Elli, Karame Ghassan, Roeschlin Marc, et al. Evaluating user privacy in bitcoin. in: Proceedings of the 2013 International Conference on Financial Cryptography and Data Security, 2013. 34-51

[29] Bamert Tobias, Decker Christian, Elsen Lennart, et al. Have a snack, pay with Bitcoins. in: Proceedings of the 2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing, 2013. 1-5

[30] Decker Christian, Roger Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. in: Proceedings of the 2015 Symposium on Self-Stabilizing Systems, 2015. 3-18

[31] Decker Christian, Roger Wattenhofer. Information propagation in the bitcoin network. in: Proceedings of the 2013 IEEE International Conference on Peer-to-Peer Computing, 2013. 1-10

[32] Eyal Ittay, Gencer Adem Efe, Sirer Emin Gun, et al. Bitcoin-ng: A scalable blockchain protocol. in: Proceedings of the 2016 USENIX Symposium on Networked Systems Design and Implementation, 2016. 45-59

[33] Morris David. Bitcoin is Not Just Digital Currency, it's Napster for Finance. CNN Money, 2014, 12: 12-13

[34] Luu Loi, Chu Duc-Hiep, Olickel Hrishi, et al. Making Smart Contracts Smarter. in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and

Communications Security, 2016. 254-269

[35] Chen Ting, Li Xiaoqi, Luo Xiapu, et al. Under-optimized smart contracts devour your money. in: Proceedings of the 2016 IEEE International Conference on Software Analysis, Evolution and Reengineering, 2017. 442-446

[36] Omohundro Steve. Cryptocurrencies, smart contracts, and artificial intelligence. AI matters, 2014, 1(2): 19-21

[37] 薛健. 区块链或引领颠覆式创新浪潮. 中国战略新兴产业, 2016, 16: 67-69.

[38] 佚名. 兴业银行区块链防伪平台对外公开. 金融科技时代, 2017, 3: 93-93

[39] Castro Miguel, Barbara Liskov. Practical Byzantine Fault Tolerance. in: Proceedings of the 2002 Symposium on Operating Systems Design and Implementation, 2002. 398-461

[40] 何广锋, 黄未晞. 区块链技术本质以及对金融业的影响. 清华金融评论, 2016, 4. 102-106

[41] Johnson Don, Alfred Menezes, Scott Vanstone. The elliptic curve digital signature algorithm (ECDSA). International journal of information security, 2001, 1(1): 36-63

[42] Zhang Fan, Cecchetti **Ethan,** Croman **Kyle,** et al. Town Crier: An Authenticated Data Feed for Smart Contracts. in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016. 270-282

[43] Eyal Ittay, Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. in: Proceedings of the 2014 International Conference on Financial Cryptography and Data Security, 2014. 436-454

[44] Szabo Nick. Formalizing and securing relationships on public networks. First Monday, 1997, 2(9): 1-9

[45] Ojeda-Guerra Carmen. A simple software development methodology based on MVP for Android applications in a classroom context. in: Proceedings of the 2015 IEEE International Conference on Computer and Information Technology, 2015. 1429-1434

[46] Zhang Yang, Luo Yanjing. An architecture and implement model for Model-View-Presenter pattern. in: Proceedings of the 2010 IEEE International Conference on Computer Science and Information Technology, 2010. 532-536

[47] 李旭伟. Protocol Buffers——比 XML 快近 100 倍. 电脑与信息技术, 2009, 17(1): 65-68

[48] Müller Jürgen, Lorenz Martin, Geller Felix, et al. Assessment of Communication Protocols in the EPC Network-Replacing Textual SOAP and XML with Binary Google Protocol Buffers Encoding. in: Proceedings of the 2010 IEEE International Conference on Industrial Engineering and Engineering Management, 2010. 404-409

附录 1　攻读硕士期间申请的国家发明专利

[1] 金海，代炜琦，李峰，邹德清，张舒；一种基于区块链的旅游景区售票方法和系统。专利申请号 CN:201710320678.1，申请日：2017 年 5 月 9 日，申请单位：武汉凤链科技有限公司

附录 2　攻读硕士期间参与的项目

[1] 基于代码相似性的漏洞检测研究项目，2015.9-2016.10

[2] 与武汉园林绿化建设发展公司、凤链科技有限公司合作项目，基于区块链的安全支付平台，2016.10-2017.4