基于 Hyperledger Fabric 的区块链农产品溯源方案

卢瑞元　王子恒　陶嘉淳　李领治　王进
（苏州大学计算机科学与技术学院　江苏 苏州 215006）

摘　要　传统的农产品溯源系统的实现依赖于中心化存储，数据不透明，易被篡改。而现有的区块链溯源系统与农产品溯源的需求结合不紧密，且对大规模溯源信息的管理和查询较为低效。提出并实现一种基于 Hyperledger Fabric 构架的农产品联盟区块链溯源方案，在保证溯源可信的基础上，针对农产品溯源流程和联盟区块链技术的特点，设计系统架构及链操作方法，提升大规模数据上链和查询的速度，使其满足农产品溯源应用的需求。结果表明，基于该方案实现的区块链溯源系统优化了大规模溯源信息的存储与查询流程，覆盖农产品的全生命周期，具有一定的实用性。

关键词　联盟区块链　农产品溯源　超级账本　智能合约

中图分类号　TP309; TP311　**文献标志码**　A　　DOI: 10.3969/j.issn.1000-386x.2018.01.001

A Scheme about Agricultural Produce Traceability using Blockchain based on Hyperledger Fabric

Ruiyuan Lu　Ziheng Wang　Jiachun Tao　Lingzhi Li　Jin Wang
(*School of Computer Science and Technology, Soochow University, Suzhou 215006, Jiangsu, China*)

Abstract　The implementation of the current agricultural produce traceability system based on centralized storage shows many disadvantages. Its data is not transparent, easy been tampered with, and incredible. The existing blockchain traceability system is not closely combined with the requirements of agricultural produce traceability, and the management and query of large-scale traceability data are inefficient. Therefore, this paper proposes a consortium blockchain scheme based on the Hyperledger Fabric framework which ensures traceability credible and promotes transaction efficiency by designing the system architecture and the chain operation method in the view of the agricultural produce traceability and consortium blockchain to meet the needs of agricultural produce traceability. The results show that the blockchain traceability system based on our scheme optimizes the transaction process of large-scale traceability data, covering the whole life cycle of agricultural products, and has certain practicability.

Keywords　consortium blockchain　agricultural produce traceability　Hyperledger Fabric　smart contract

0　引　言

农产品溯源系统对农产品从种植到销售的全生命周期进行跟踪，基于溯源数据对相关人员进行追责是保障农产品食品质量安全的重要手段。目前，许多地区已经使用信息系统实现了农产品的溯源跟踪，但相关数据易篡改，溯源信息难可信，事故责任难落实。溯源系统的信用危机严重影响了优质企业的发展。

区块链作为一种去中心化的数据存储技术，其维护的数据具备公开透明、不可篡改等特性。利用区块链技术为相关企业建立信任中心，可以有效地解决农产品溯源系统不安全不可信的问题。

本文总结区块链和溯源技术的发展现状，分析农产品溯源的需求特点，基于 Hyperledger Fabric 构架[1]，设计一种改进的联盟区块链农产品溯源方案。方案的实施可以进一步巩固相关企业在农产品产销领域的领先地位，对提升全社会农产品安全具有重要意义。

1　相关研究

近年来，许多科研人员对农产品溯源技术进行了

收稿日期：xxxx-xx-xx。苏州大学“大学生创新创业训练计划”(201810285032Z)。赛尔网络“下一代互联网技术创新项目”(NGII20190314)。**卢瑞元**，本科生，主研领域：区块链；**王子恒**，本科生；**陶嘉淳**，本科生；**李领治**，副教授；**王进**，教授。

深入研究，RFID、电子标签、物联网等技术在农产品溯源跟踪中得到了广泛应用。文献[2]中实现了基于USB Key的水产品企业监管溯源系统，解决了水产品企业的溯源身份认问题；文献[3]利用猪肉生产和HACCP体系相结合的方法筛选出溯源信息，实现了基于HACCP体系的绿色猪肉生产质量监管与溯源系统；文献[4]对我国种子质量可追溯系统进行了深入的分析和研究。上述溯源系统将数据存储于常见的集中式数据库内，容易遭到破坏或是企业自行篡改：掌握了集中式数据库的企业，可以为了自己的利益而随意修改数据；数据也可能被黑客窃取，或是因数据库单点故障而受损。这些问题都会导致消费者无法正常溯源到真正的原始信息，使得溯源系统失去可信度。

自比特币诞生起，区块链技术不断发展演进并逐渐成熟。目前，区块链生态形成了基于分布式账本、共识信任、非对称加密、智能合约等主要特征的应用范式[5]。最为核心的特征是，区块链可在没有中心权威机构的情况下，可使互相协作的参与方建立起基于数学模型的相互信任，实现去中心化[6]。区块链的上链过程，就是不断地将新数据转化为链上的历史数据的过程；而数据溯源的过程，就是将链上的历史数据按照时序还原为溯源数据，并按照一定的形态进行呈现的过程。Ramachandran等基于区块链实现了科学数据溯源管理模型框架SmartProvenance，该框架利用智能合约、开放溯源模型记录溯源信息，而区块链在其中承担数据的收集、验证和管理任务，避免任何针对数据的恶意篡改行为[7]。

区块链系统利用共识机制来维护多地数据一致。目前成功应用于公有链、联盟链和私有链中较为典型的共识机制有：工作量证明PoW机制、实用拜占庭PBFT机制、Paxos共识及其各类改进机制等[8]。区块链天然具有的诸多特性，例如去中心化、流程透明、不可篡改，使得它成为数据溯源领域一大重要的技术趋势。不少研究人员注意到了区块链的特性，将研究方向转移到了基于区块链技术的溯源系统开发上。[9]

文献[10]在区块链的语义下对当前物联网设备中常见的RFID标签数据定义了物联网大数据溯源安全模型。该模型采用了去中心化的思想分析RFID射频识别标签数据及其溯源信息，实现高信任度的权限验证和分布式管理。京东所开发的“智臻链BaaS平台”可以让用户通过简单、灵活的配置方式，快速搭建安全可靠的区块链网络，将商品相关的交易信息加密存储于区块链当中[11]。天猫国际利用区块链技术、药监码技术以及大数据跟踪进口商品全链路，给进口商品打上唯一的身份证码，将商品生产、检测、运输、通关等环节的信息完整展现在用户面前[12]。区块链企业VeChain发布了全球第一款基于区块链技术的NFC防伪芯片，并开发了一站式区块链BaaS(Blockchain as a Service)服务平台ToolChain[13]。

区块链溯源虽然已经被诸多企业应用到生产环境、电子商务信息系统当中，但这些方案并没有很好地结合区块链和溯源需求的特点对方案流程及系统架构以进行整体改良。使用区块链技术虽然提高了溯源数据的可信度，但系统本身的可信管理以及弹性部署能力依旧有较大的提升空间。现有的区块链实现中，其数据查询、数据分析功能较为简单，随着区块链平台上应用与数据规模的增长，如何更高效地管理和查询溯源信息将是区块链溯源所要解决的重要问题[14]。本文提出的基于Hyperledger Fabric构架的区块链溯源系统，充分结合农产品溯源的需求特点，在保证溯源可信的前提下，对系统构架及其查询方法进行了设计，有效提升了大规模数据上链和查询的速度。

2 方案系统构架

本文实现的基于区块链的农产品质量安全溯源设计方案的系统构架如图1所示。该构架包括：区块链网络层、中间应用服务层、前端应用层。

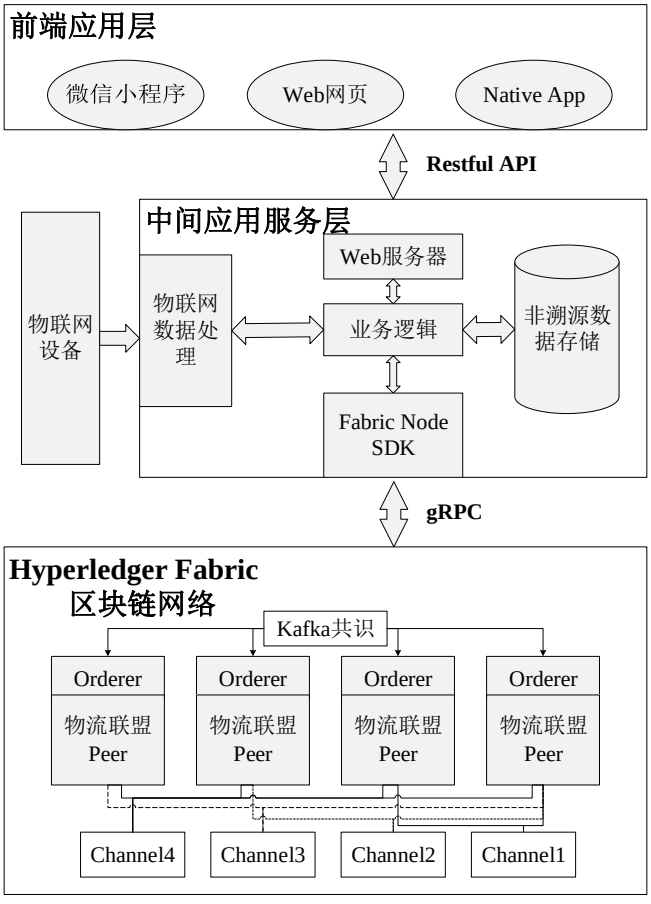

图1 基于联盟区块链的农产品溯源系统构架

2.1 联盟区块链

参与农产品溯源链条的各方成员是权限角色地位不等的机关和企业，每个成员都是得到准入许可的可信参与方，这些成员组成拥有共同目的的企业联盟。联盟区块链的共识过程受若干个主要参与方管理，这种形式既能够保证系统运行的效率，又可以兼顾系统安全性和成员共同维护的特性。因此，联盟区块链也被称为“多中心”区块链。选用联盟链的具体原因有：1) 降低去中心化程度，减少共识过程中的资源浪费。2) 维护农产品安全溯源系统是联盟链中各参与节点的责任，区块链系统的运行无需激励机制。使用联盟区块链可以不依赖数字货币，简化记录账簿，降低系统运行所需算力与带宽。Hyperledger Fabric 是开源的联盟区块链，提供完善的准入与安全机制。本文的联盟区块链构架基于 Hyperledger Fabric。

本文的系统将农业基地、物流、销售、监管机构等参与方规划进入 Hyperledger Fabric 联盟区块链的相应 Channel，对相应信息进行查询和管理。Channel 的结构如图 2 所示。每一个 Channel 都是一个虚拟的区块链网络，Channel 和 Channel 之间相互隔离；Peer 是参与区块链网络的服务节点。对于每一个 Channel 里的所有 Peer，都会为该 Channel 维护相同的账本数据。此外，Channel 中还运行着链码。链码是运行在区块链上的不可篡改的代码，是一种智能合约。Channel 的账本数据分为两部分，一部分数据是链码调用记录，这部分数据以区块链的形式存储；另一部分数据是账本的当前状态，这部分数据以 key-value 数据库的形式存储。

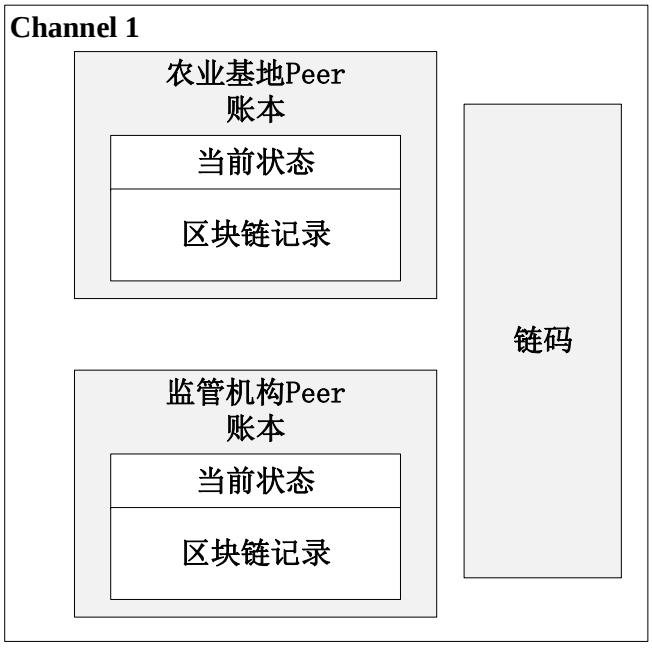

图 2 Channel 结构示意图

对于消费者，第三方监管机构是数据可信的基础，因此，在联盟链中引入第三方监管机构的成员角色，任何数据都应该与监管机构共享。农业基地、物流、销售三方组织都和监管机构单独建立 Channel，以分享需要被监管的信息。为维护农产品的声誉，农业基地也将大部分信息公开给所有用户。根据上述特征，本文的方案建立了四个 Channel。

Channel1: 农业基地、监管机构；

Channel2: 物流、监管机构；

Channel3: 零售、监管机构；

Channel4: 农业基地、物流、零售、监管机构。

所有新加入系统的参与方都需要通过 Fabric-CA 证书管理服务器为自己的节点和管理员用户创建证书。证书管理服务器基于非对称秘钥签名技术。服务器向每个参与方分发一对公钥和私钥，参与方在数据上传过程中使用私钥签名，以此来保证数据的可验证性。其他成员将可以用其公钥验证该参与者的身份与信息的来源是否真实。上传的每块数据均采用哈希算法给出校验，来判断数据内容是否被篡改。

所有溯源信息均存储在区块链中，并且支持被授权的节点对其进行访问。参与方在供应链中所拥有的职能决定其在区块链中所拥有的权限。每一次对区块链数据的操作都必须得到有权进行该操作的参与方的背书签名，才能够通过判断，分发到其他参与成员所在的区块链分布式存储模块中，因此数据的存储公开、透明。此外，区块链的协议规则由代码定义并存储在区块链中，无法被区块链中的某一参与者所篡改，从而保证数据的真实性与有效性。

每一方组织各自维护自己的 Orderer 和 Peer 节点。Orderer 节点是用于对事务进行排序的节点。事务主要是对链码的调用记录。事务记录被 Orderer 节点排序后打包成区块，分发给对应的 Channel 上链。Kafka 是开源的分布式消息队列。所有组织的 Orderer 节点都在同一个 Kafka 集群中，以此确保在复杂的网络环境下所有的 Orderer 节点都能得到一致的事务记录排序、分发相同的区块。Peer 节点加入对应的 Channel，作为区块链网络的成员维护账本数据，并对外提供服务。

数据的上链和查询通过 Channel 中的链码进行。在本文的设计中，联盟链中的每一个成员方根据自身的业务需求，选择需要进行上链的数据，编写链码。区块链存储相较于传统的数据库，操作耗时更长。因此，仅选择最关键的农产品溯源数据存储其中，其他溯源无关的业务数据则由对应的应用层来维护。

2.2 中间层与前端应用

本文所提出的方案中，任何一个参与方都可以在联盟区块链存储的基础上，维护自己的 B/S 架构的应

用服务。gRPC 是一种高性能的开源的 RPC 框架[15]，应用服务通过 gRPC 来向底层的区块链网络来请求服务。

在本文的样例应用服务中，通过 Hyperledger Fabric 提供的 SDK 来向底层的区块链网络发送 gRPC 请求，调用智能合约，以实现业务逻辑。业务逻辑位于中心位置，和所有的数据源交互，满足实际的业务需求。

系统中的数据有多个来源，并通过不同方式采集和存储。基础数据来源于农产品种植养殖、生产加工、包装、运输、销售和消费的完整生命周期。对于这些溯源相关的基础数据，其一部分是来自生产加工的数据，该部分数据可以通过智能传感器和物联网设备采集，经由 udp 通信发送到中间应用服务层中。中间应用服务层对数据进行结构化处理之后，通过 gRPC 请求将该部分数据发送给联盟区块链网络进行上链。而另一部分数据是难以通过物联网设备采集的数据，该类数据由相关企业的员工使用客户端扫描产品二维码，输入产品溯源信息。另外，对于现存于其他系统的信息，例如物流信息，则通过物流联盟提供的统一接口，定期获取，自动上链。

Web 服务器对外提供 Restful API 接口。Restful API 是一种通用性强的 Web API 规范，易于开发人员理解，方便调用。提供给终端用户使用的应用，如小程序、安卓应用、Web 网页等，都可以通过 Web 服务器开放的接口，调用业务逻辑。

3 方案实现方法

3.1 区块链集群部署

按照上节给出的系统构架，本文的方案遵循以下步骤部署联盟链网络：

命令 1 使用 cryptogen 工具为每一个 Orderer 和 Peer 节点和它们的管理员用户创建必须的证书。

```
cryptogen generate --config=/path/to/
cryptogen_config --output /path/to/output/dir
```

这些证书包括各个组织的根证书、节点或管理员的身份证书与私钥等。它们存储在固定结构的文件树里。通过这些证书，系统可以使用 tls 来加密请求，并对所有的对链操作进行签名。

命令 2 使用 configtxgen 工具为 Orderer 节点生成初始块文件。

```
configtxgen -configPath /path/to/config -profile
OrdererGenesisProfile -outputBlock /path/to/output/
block
```

初始块是区块链中的一个特殊区块，它作为最开始的区块，没有指向前一个区块的指针。Orderer 集群会共同维护相同的区块链，这个区块链中会存储对整个区块链网络的操作记录，包括建立 Channel，在 Channel 上安装与实例化链码等。

命令 3 对于每一个 Channel，都先使用 configtxgen 工具，生成用于建立 Channel 的事务文件。

```
configtxgen -configPath /path/to/config -profile
AllChannelProfile -outputCreateChannelTx
/path/to/AllChannel.tx -channelID allchannel
```

这个文件内结构化地描述了 Channel 的信息，尤其是包括了 Channel 所属的区块链网络的成员方的信息。

命令 4 对于每一个组织，都使用 configtxgen 工具生成用于更新 Anchor Peer 信息的事务文件。

```
configtxgen -configPath /path/to/config -profile
AllChannel -outputAnchorPeersUpdate
/path/to/anchors.tx -channelID allchannel -asOrg
OrgName
```

在同一个组织里可能有多个 Peer，这些 Peer 中会有且只有一个 Anchor Peer。Anchor Peer 接受其他组织的节点发来的网络信息，并广播给自己组织的 Peer，以确保不同组织的 Peer 之间知道互相的位置。

命令 5 在部署任何一个 Orderer 节点之前，为将部署该节点的主机准备 Kafka 与 ZooKeeper 环境。

```
docker-compose -f docker-compose-zk.yml up -d
docker-compose -f docker-compose-kafka.yml up -d
```

Kafka 是一个高吞吐的分布式的消息队列，它的运行依赖 ZooKeeper。ZooKeeper 是一个分布式协调服务，可以为分布式系统提供一致性服务。

命令 6 完成了上述所有的准备工作后，分别使用如下所示的命令，启动所有的 Orderer 和 Peer 节点。

```
orderer start
peer node start
```

Orderer 节点在启动时，会读取初始块文件，并以此开始维护该区块链。

命令 7 在所有的 Orderer 节点和 Peer 节点启动后，Peer 节点消费先前生成的事务文件，以建立 Channel 和更新 Anchor Peer 信息。

```
configtxgen -configPath /path/to/config -profile
OrdererGenesisProfile -outputBlock /path/to/
output/block
```

命令 8 Channel 建立完成之后，则在 Channel 上安装和实例化链码。

```
configtxgen -configPath /path/to/config -profile
OrdererGenesisProfile -outputBlock /path/to/
output/block
```

实例化的链码是一种智能合约，运行在 Docker 容器中。链码被用于对所属的 Channel 的账本进行读写

操作。

至此，就完成了整个联盟链网络的部署。对于测试和开发环境，使用 Docker 作为应用程序引擎，打包所有的应用程序，并编写脚本完成上述步骤，实现一键启动开发环境。对于生产环境，每个 Peer、Orderer 和其他应用服务都直接运行在单独的主机上，根据各自配置文件中指定的地址来尝试连接其他节点网络。

3.2 中间层 egg.js

egg.js 是一个高可定制的 Web 应用层框架[16]。在本文的样例应用服务中，egg.js 被用于设计一个应用层和区块链网络之间的中间层。其向上提供 Web 应用服务，向下与区块链网络进行通信，以调用智能合约，修改账本。

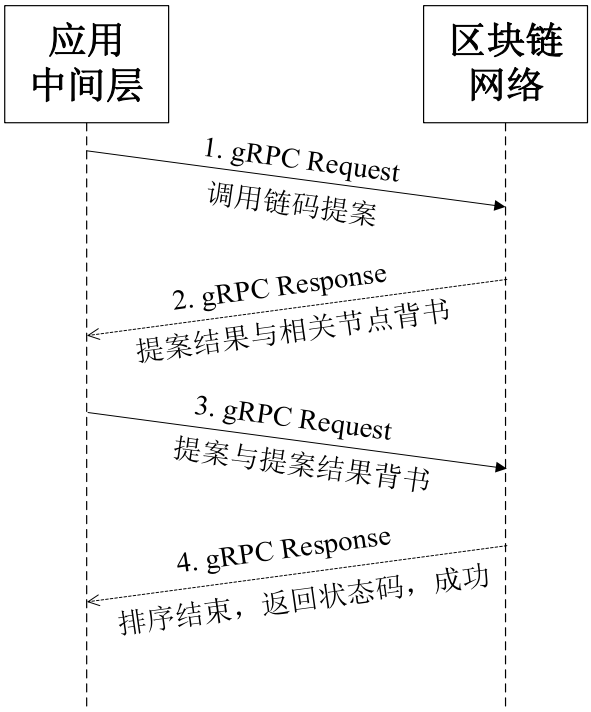

图 3 中间层与区块链网络的通信

如图 3 所示，当应用服务需要与账本中的数据进行交互时，该中间层会向区块链网络发送多次 gRPC 请求进行通信。在第一次通信请求中，请求体的内容包括：待调用的链码所属的 Channel、待调用的链码名称及具体的函数名、提供给这次链码调用的参数、临时生成的事务 ID。并且，这次请求会使用当前执行操作的用户的私钥进行签名。这次请求仅向区块链网络提交了一个提案。如果这个提案通过，则意味着请求方的身份符合该链码的调用策略，且该链码的模拟调用过程中也没有发生其他错误。

提案通过后，区块链网络会返回一个响应，该响应是相关节点对提案的背书。响应体的内容包括：该提案的模拟运行结果、区块链网络中与该链码相关的节点对该结果的签名。需要注意的是，区块链网络虽然会根据这次提案模拟地调用链码，但并不会因此实际更新账本。

该中间层在拿到这份背书响应后，会以该背书作为请求体，向区块链网络发起第二次 gRPC 请求。这次请求会发起一次正式的事务。区块链网络的 Orderer 节点对该事务完成排序后，会返回成功的状态信息。成功的状态信息表明该事务被网络一致排序，即将打入区块并分发给维护该账本的所有节点。从请求回执中拿到状态信息之后，该中间层会认为这次对账本数据的交互已经结束，向上层应用返回响应。上述中间层与区块链网络通信的伪代码如下所示。

代码 1　中间层与区块链网络通信

```
async function invokeChaincode(ca, channelMeta,
chaincodeMeta, invokeArgs){
    const txId = generateTxId();
    const proposalRequest =
combineProposalRequest(ca, channelMeta,
chaincodeMeta, invokeArgs, txId);
    const proposalResponse = await
sendTransactionProposal(proposalRequest);
    const transactionRequest =
generateTransactionRequest(proposalResponse);
    const transactionResponse = await
sendTransaction(transactionRequest);
    return transactionResponse;
}
```

每当该中间层试图与账本数据进行交互，都必须进行上述流程。然而，如果一个业务逻辑希望与账本数据进行多次交互，很可能会因为频繁进行上述流程而拖长应用服务的响应时间。在为该中间层设计与编写业务逻辑时，应当针对这一点进行优化。

此外，该中间层还负责解决一些横切关注点。例如，在本文的样例应用服务中，在该中间层编写了一些额外的中间件，用于鉴权和记录系统运行状况。

3.3 链码设计

基于实践经验，本文的方案建议将业务逻辑从区块链网络层分离，降低开发成本。但如上一节所述，当一个业务逻辑需要中间层与账本数据进行频繁交互时，可能会严重地拖长响应时间。对该问题的一个解决方案是将这种多次交互的逻辑移入链码中。

例如，当业务需要批量查询或添加一批数据时，可以将该逻辑从中间层移入链码。如此一来，中间层只需要进行一次链码调用，仅进行一次上一节所述的通信流程，以避开损耗时间的线性增长。该逻辑的伪代码如下所示。

代码 2　合并批量请求

```
function putBulkKeyValue(asset){
    kvlist := unmarshal(asset)
    for index := 0; index < len(kvlist); index++ {
        stub.PutState(kvlist[index].key,
                kvlist[index].value)
    }
}
function getBulkValue(asset){
    klist := unmarshal(asset)
    kvlist := new KVList()
    for index := 0; index < len(kvlist); index++ {
        value := stub.GetState(klist[index].key)
        kv := combineKeyValue(klist[index].key,
                              value)
        kvlist.push(kv)
    }
    return kvlist
}
```

4　区块链苹果溯源系统

基于上文提出的方案，本文以苹果溯源为例实现了一个系统原型。

4.1　流程分析

图 4 展示了该原型系统的基本运作流程，下文对该流程进行分析。

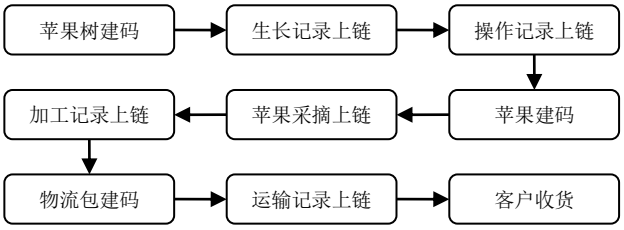

图 4 苹果溯源记录流程图

在苹果成熟采摘前，以苹果树为单位进行维护。果树的发芽、花开、结果等生长观测信息需要追溯，应当上链；果农对果树所进行施肥、修剪、浇水、除虫、采摘等操作信息需要追溯，也应当上链。

苹果在果树上生长成熟后，即可作为基础单位进入系统维护。苹果的来源信息，即该苹果从具体某棵果树上采摘的信息上链；对该苹果所进行的除梗、包装、打蜡等粗加工操作也上链溯源。

在粗加工完成后，苹果被打入物流包裹中，运输至客户仓储。负责运输的物流公司、物流单号、运输状态等信息上链；此外，运输包中所包含的苹果信息也上链。

上述过程涉及三种主体：苹果树、苹果、物流包。在逻辑和物理上，每一个苹果树、苹果、物流包都有自己的唯一标识。

本应用使用二维码作为物理标识。在果园中，苹果的产量十分巨大，相对于传统的昂贵的 RFID 卡，二维码更适应每一个苹果都需要单独标识的情景。

在溯源过程中的任何一环，都可以对某个具体的溯源单位进行查询。如图 5 所示，如果用户扫描苹果树或物流包的二维码，即可获取目标的溯源记录；如果用户扫描苹果的二维码，除获得苹果自身的溯源记录外，也获取该苹果对应的果树和物流包的溯源记录。

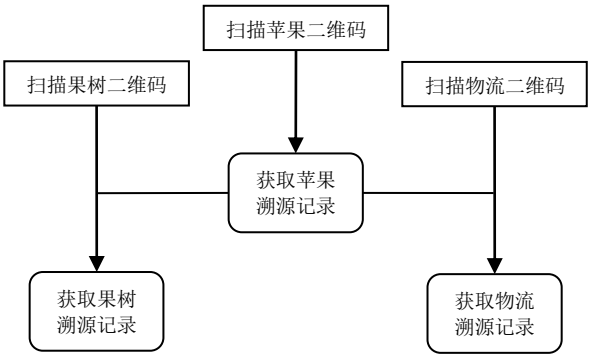

图 5 溯源查询流程图

4.2　数据模型分析

代码 3　溯源主体数据模型

```
Apple: {
  Meta: {
    product_id: "id of the product",
    product_name: "name of the product",
    description: "description of the product",
    org: "organization the product belong to",
    previous: ["product id of apple tree", "product
id of package"]
  },
  Transactions: [
    {
      tx_id: "hashed transaction id",
      block_id: "hashed block id the transaction
belong to",
      transaction_time: "when transaction committed
into the block",
      operation_type: "type of the operation",
      operation_user: "who make the operation",
      operation_time: "when the operation happened",
      description: "description of the operation",
    },
  ]
}
```

苹果树、苹果、物流包三种主体的数据都需要保存。如上所示，它们的数据形式类似，可以大致分为两部分：一部分是 Meta 字段，它描述主体本身的属性，例如品名、所属公司等；另一部分是 Transactions

字段，它描述主体被进行的操作，包括操作类型，操作描述，操作执行人，操作时间，上链时间等。

此外，苹果树和物流包只需要维护自身的溯源信息，而每个苹果除了自身维护的信息外，溯源时还需要得到所属苹果树和物流包的溯源信息。因此，苹果主体的 Meta 字段中，有一个额外的 previous 属性。该属性是该苹果的所有额外信息源的编号。在对苹果进行溯源查询时，递归地查询该列表中的所有编号，将额外信息源的溯源信息与苹果的溯源信息合并。

存在两个可能改变该 previous 属性的时间点。果树采摘操作时，在所采摘苹果的 previous 属性中添加该果树的唯一编号；苹果打包成物流包裹时，在这些苹果的 previous 属性中添加该物流包的唯一编号。

4.3 实现结果

在本应用系统上为苹果树建立数据时，会为该苹果树生成唯一编号和对应的二维码。果园员工将该二维码贴在果树上，每当对果树进行操作或观察到果树进入新的生长阶段，扫描二维码，进入如图 6 所示的果树数据录入表单页，将过程和结果录入进去。

图 6 溯源信息录入界面(部分)

在二维码中，编码了溯源查询网站的链接，并把产品的唯一标识码加密地编码在内。任何用户只要扫描二维码，就可看到对应实体的溯源信息；任何登录系统的操作员，只要通过系统提供的页面扫描二维码，就可进入对应的数据录入页面。对于气温、空气湿度等易于智能监控的生长数据，通过物联网设备进行数据采集，发送给系统自动上链。

当果树上的苹果成熟，扫描果树二维码，可生成对应数量的苹果唯一编号与二维码。

在采摘到的苹果的简易包装上贴上苹果的二维码，每当对该苹果进行加工操作后，扫描二维码，将结果录入进系统内。

当苹果准备打包装箱时，操作员进入系统，生成物流包所对应的唯一编号与二维码。将二维码贴到包裹上，扫描二维码，进入物流包溯源页，扫描每个要被打入包中的苹果的二维码，为这些苹果的 previous 属性添加该物流包的编号。物流包打包完成后，中间层会定时调用物流公司的物流查询接口，如有物流进度更新，自动将物流数据录入。

消费者收到苹果后，可以扫描其上的二维码，看到如图7所示的从果树到苹果到物流的完整溯源记录。

图 7 溯源查询手机界面(部分)

4.4 性能测试与分析

本节进行性能测试。将系统部署在一个具有 4 核 1.5Ghz CPU、16GB 内存的主机上，在中间层模拟用户行为，持续向区块链网络发送上链事务请求。完成固定上链事务请求数的平均所用时间和单个请求的平均延迟如表 1 所示。从表 1 可以看出，该系统在该环境下的吞吐量能够稳定在 100TPS 以上，具有承载溯源应用的能力。

表 1 系统完成固定上链请求所用时间统计表

完成上链请求/次	所用时间/s	平均响应时间/s
10000	96	3.5
20000	187	3.4
30000	294	3.6
40000	388	3.6
50000	491	3.7

在本文的系统中，单个功能可能需要顺序地进行多个请求，响应时间随请求数线性增长。基于 3.3 节的思路，本文对该系统的业务代码和链码设计进行了优化。优化前后部分功能的区块链事务请求数如表 2 所示。从表 2 可以看出，需要多次请求的功能都被降低为仅需 1 或 2 次请求，有效提升了大规模数据上链和查询的速度。

表 2 系统功能优化前后上链请求数对比

功能名称	优化前请求数/次	优化后请求数/次
果树信息上链	1	1
苹果采摘上链	2	1
为 n 个苹果树建码	n	1
为 n 个苹果打包装箱	2n	2

4.5 安全性分析

本节结合场景特点，从业务层面和数据层面展开分析，从安全角度评估本文所提出的方案的可行性。

业务逻辑集中于中间应用层，中间应用层是传统的 Web 应用。在传统 Web 应用中，存在许多常见的安全威胁，如 XSS 攻击、CSRF 攻击、流量劫持、钓鱼攻击等。对于这些威胁，都可以由传统 Web 应用已有的防范措施进行应对。

溯源数据存储在区块链网络中，这些数据的真实性、完整性和隐私性是溯源可信的基石。对于真实性和完整性，中间层作为传统的 Web 应用，以 RBAC 确认用户的操作权限；所有向联盟链网络请求都由发起请求的用户私钥进行签名，并且请求用户和请求内容都符合链码的背书策略，保证了相关信息来源的真实性；在此之外，由 Hyperledger Fabric 提供的 CA 和 MSP 等机制可以确保链上数据无法被伪造或篡改。对于隐私性，依照不同的保密要求，利用多 Channel 实现不同种数据的隔离，联盟内的成员只能访问有权访问的数据，保证成员的隐私信息。综上，在数据存储层面，可以认为存在足够的安全保障，使得溯源数据高可信。

5 结束语

本文设计并实现了一种联盟区块链农产品溯源系统，充分结合农产品溯源的需求特点，有效提升了大规模数据上链和查询的速度，形成多方参与的多层次、多中心农产品安全解决方案。该系统的区块链存储网络基于超级账本，未来将跟随相关技术的发展状况进行维护。下一步拟将该方案进一步在农产品种植基地推广，并结合实际业务需求加以优化。

参 考 文 献

[1] A Blockchain Platform for the Enterprise [EB/OL]. [2019-12-20].https://hyperledger-fabric.readthedocs.io/en/latest/index.html.

[2] 杨信廷, 吴滔, 孙传恒等. 基于 USB Key 的水产品企业监管溯源系统设计与应用 [J]. 农业机械学报, 2012, 43 (08): 128-133.

[3] 任守纲, 徐焕良, 黎安等. 基于 RFID/GIS 物联网的肉品跟踪及追溯系统设计与实现 [J]. 农业工程学报, 2010, 26 (10): 229-235.

[4] 张艳娜. 基于HACCP体系的猪肉安全生产溯源系统的设计与实现[D]. 安徽农业大学, 2015.

[5] 张亮, 刘百祥, 张如意, 江斌鑫, 刘一江. 区块链技术综述 [J]. 计算机工程, 2019, 45(5): 1-12.

[6] 袁勇, 王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016, 42(4): 481-494.

[7] Ramachandran A, Kantarcioglu M. Smart Provenance: a distributed, blockchain based data provenance system[C]// CODASPY′18 Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy. Tempe, Arizona, USA, 2018: 35-42.

[8] Zheng Z, Xie S,Dai H,et al. An overview of blockchain technology: architecture, consensus, and future trends[C] //Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress). Honolulu, Hawaii, USA, 2017: 557-564.

[9] 仵冀颖, 杜聪, 马志远, et al. 应用于食品追溯体系的区块链架构设计[J]. 计算机应用与软件, 2019(12): 46-50.

[10] 刘耀宗, 刘云恒. 基于区块链的 RFID 大数据安全溯源模型 [J]. 计算机科学, 2018, 45(S2): 367-368.

[11] BaaS平台-京东智臻链[EB/OL].(2019-04-03)/[2019-12-20]. https://baas.jd.com/doc/overview/overview.html.

[12] 区块链在天猫国际商品溯源中的应用[EB/OL]. (2018-01-11)/[2019-12-20].https://yq.aliyun.com/articles/348 7 87.

[13] Vechain[EB/OL].[2019-12-20]. https://www.vechain.com.

[14] 钱卫宁, 邵奇峰, 朱艳超。区块链与可信数据管理：问题与方法 [J]. 软件学报，2018,29 (1):150-159.

[15] gRPC-A high performance, open-source universal RPC framework[EB/OL]. [2019-12-20]. https://grpc.io/.

[16] egg-为企业级框架和应用而生[EB/OL]. [2019-12-20]. https://eggjs.org/en/.

请在文章最后另页提供本文负责人、电话、手机和 E-mail,以便联系。并附上所有作者简介，内容为:姓名、职称 / 学位、主研领域、身份证号或军官证号、手机号或电话、单位、通信地址、邮政编码、E-mail 地址。如果在发表之前其中任何一项有变化,请及时通知编辑部。

本文受苏州大学“大学生创新创业训练计划”（编号 201810285032Z）和赛尔网络“下一代互联网技术创新项目”（编号：NGII20190314）资助

作者简介：

卢瑞元，本科生，主研方向为区块链；王子恒，本科生；陶嘉淳，本科生；李领治，博士，副教授；王进，博士，教授。

联系人：

卢瑞元：18915422865 rylu@qq.com

李领治：15995848982 lilingzhi@suda.edu.cn