学校代码：10286
分类号：TN918.91
密　级：公开
U D C：621.3
学　号：150762

东南大學
SOUTHEAST UNIVERSITY

硕士学位论文

基于区块链的物流用户隐私数据保护

研究生姓名：**张克落**
导 师 姓 名：**宋宇波**

申请学位类别 **工学硕士**　学位授予单位 **东 南 大 学**

一级学科名称 **信息与通信工程**　论文答辩日期 **20　年　月　日**

二级学科名称 **信息安全**　学位授予日期 **20　年　月　日**

答辩委员会主席＿＿＿＿＿＿　评　阅　人＿＿＿＿＿＿

＿＿＿＿＿＿

20　年　月　日

東南大學

硕士学位论文

基于区块链的物流用户隐私数据保护

专 业 名 称： 信息与通信工程（信息安全）

研究生姓名： 张克落

导 师 姓 名： 宋宇波

BLOCKCHAIN-BASED LOGISTICS USER PRIVACY DATA PROTECTION

A Thesis Submitted to

Southeast University

For the Academic Degree of Master of Engineering

BY

ZHANG Ke-luo

Supervised by

Associate Professor Song Yubo

School of Information Science and Engineering

Southeast University

May 2018

东南大学学位论文独创性声明

■

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，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。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作了明确的说明并表示了谢意。

研究生签名：_____日期：_____

东南大学学位论文使用授权声明

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括以电子信息形式刊登）论文的全部内容或中、英文摘要等部分内容。论文的公布（包括以电子信息形式刊登）授权东南大学研究生院办理。

研究生签名：_____导师签名：_____日期：_____

摘要

随着物流业在现代互联网应用中扮演越来越重要的角色，其用户隐私数据泄露问题也日益受到关注。目前，现有物流行业用户隐私泄露问题主要由物流企业提供安全保障，虽在一定程度上可以解决部分隐私泄露问题，但依然存在很多不足：1) 用户隐私数据通常由物流企业类的第三方进行保管。这类第三方缺乏可信性，也很难杜绝隐私数据的泄露和窃取；2) 用户缺乏对隐私数据访问权限的管理，也无法决定隐私数据的访问范围。针对上述问题，本文提出了一种基于区块链的物流用户隐私数据保护方案，将隐私数据分层加密云存储和去中心化的访问权限控制管理相结合。在此基础上设计和实现了一物流下单原型系统，验证了该隐私数据保护方案的有效性。本文主要工作内容如下：

1. 在研究和分析现有隐私数据加密存储安全机制，以及数据访问权限控制管理方案的基础上，结合分层加密和区块链技术提出一种新型的物流用户隐私数据保护方案。该方案将隐私数据进行分层加密处理上传至云端存储，以解决隐私数据在不可信第三方平台上的安全存储问题；同时提供去中心化隐私数据访问权限管理，实现用户对隐私数据访问权限的控制管理。
2. 针对用户隐私数据通常存放在不可信第三方数据保管平台的现状，本文提出了一种基于分层加密的隐私数据保护机制。该机制采用基于属性的分层加密方案，通过安全属性表述数据访问方的身份并确定可读取的数据内容范围，将密文上传至云存储平台保管。该机制可有效防止物流用户隐私数据在不可信第三方平台的泄露。
3. 针对物流派发时用户隐私数据访问方身份无法预先获知从而难以授权的问题，本文提出了一种改进的 CP-ABE 算法实现基于属性的分层加密方案。该加密方案采用嵌套访问控制树结构，用户可根据数据访问方属性确定其数据访问权限。拥有较高权限级别的数据访问方可以直接访问低级别的数据，无需重复解密，数据读取效率高。
4. 针对现有隐私数据访问权限管理方案中对隐私数据访问权限的管理难以做到用户可自主控制、授权过程可追溯以及访问记录可审计的问题，本文提出了一种基于区块链和 DAA 匿名认证相结合的访问权限管理机制。基于 DAA 匿名认证实现成员身份管理，为区块链上的实体提供匿名可验证的身份；通过维护一个交易公钥列表 TPL 来实现对区块链节点的访问权限控制，解决当前区块链技术存在的隐私泄露风险；同时，该机制通过分布式账本保存用户对隐私数据的访问权限，使用链码（即智能合约）来封装物流中各角色与隐私数据之间存在的业务逻辑，实现了用户对隐私数据的访问权限控制管理。
5. 在所提方案的基础上，本文设计并实现了一个物流下单原型验证系统，该系统由数据拥有方、数据访问方及云存储平台构成。其主要功能包括：物流订单生成、物流状态查询、隐私数据权限管理以及用户信息访问。该系统保证了物流用户隐私数据的安全性，提供了数据拥有方对物流数据的访问权限控制管理，实现了数据访问方对物流隐私数据的分层访问。经测试表明，该系统在保证隐私数据安全的同时，拥

有较高的计算效率，用户与区块链模块的交互延时保持在 80-100ms 区间内，有一定使用价值。

关键词：物流隐私保护，CP-ABE，区块链，访问权限管理

Abstract

With the logistics industry playing an increasingly important role in modern Internet applications, the leakage of privacy data of logistics user has drawn considerable attention. At present, the security of privacy leakage depends mainly on the protection measures carried out by logistics company in the logistics industry. Although it can solve some privacy leakage problems to a certain extent, there are still many deficiencies: 1) User's privacy data is usually kept by a third-party, such as Logistics company. Such institutions lack credibility and it is difficult to prevent the disclosure and theft of privacy data. 2) User lacks the management of access to privacy data and cannot determine the scope of it. Aiming at these problems, this paper proposes a blockchain-based logistics user privacy data protection scheme that combines the privacy data hierarchical encryption cloud storage with decentralized control of access rights. In addition, a logistics order prototype system has been designed and implemented to verify the effectiveness of the privacy data protection scheme. The novelty and contribution of this paper can be summarized as follows:

1. Based on the research and analyses of existing privacy data encrypted storage security mechanisms and data access rights control management schemes, a novel privacy data protection scheme is proposed by combining hierarchical encryption and blockchain technology. The scheme utilizes a hierarchical encryption algorithm to encrypt privacy data and then upload it to the cloud storage to guarantee the secure storage of privacy data on unauthentic third-party platforms. MeanWhile, it provides decentralized control of privacy data access rights and enables logistics user to manage private data access rights.
2. Given the current situation where user's privacy data is usually stored on an unauthentic third-party data storage platform, we propose a privacy data protection mechanism based on hierarchical encryption. This mechanism adopts a attributes-based hierarchical encryption scheme, then describes the identity of data visitor and determines the range of data readable with the security attributes, finally uploads the ciphertext to the cloud storage platform. This mechanism can effectively prevent the leakage of user's privacy data on untrusted third-party platforms.
3. The identity of data visitor cannot be obtained in advance so that it is difficult to be authorized while dispatching delivery. Therefore, the paper proposes an improved CP-ABE algorithm to implement attributes-based hierarchical encryption scheme. The hierarchical encryption scheme adopts a nested access control tree structure, and user can determine his data access rights according to the attributes of data visitor. Data visitor with a higher privilege level can access low-level data directly without repeated decryption, and the efficiency of data reading is high.
4. An access rights management mechanism based on blockchain and DAA is proposed to solve the problem user is unable to manage privacy data unautonomously in the existing privacy data access rights management scheme with untraceable authorization process and unauditable access records. Implement membership management based on DAA anonymous authentication, providing anonymous but verifiable identity for entities on the blockchain. Furthermore, it obtains access rights to control blockchain nodes to prevent the leakage of privacy data by maintaining a transactional public key list. And this mechanism saves user's access rights of privacy data through distributed ledger, and

utilizes chaincode to encapsulate the business logical relation between roles in logistics and privacy data, enables user to control access rights of privacy data finally.

5. Based on the scheme mentioned in this paper, a logistics order prototype system has been designed and implemented. The system consists of data owner, data visitor, and cloud storage platforms. Its main functions include generating logistics orders, querying logistics status, managing access rights to privacy data, and accessing to user privacy data. The system ensures the security of logistics user privacy data, enables data owner to control access rights to privacy data, and achieves data visitor's hierarchical access to logistics privacy data.The test shows that this practical system can achieve high computational efficiency while ensuring the security of privacy data, and the interaction delay between user and the blockchain module can be kept within the interval of 80-100ms.

Keywords: logistics privacy protection, CP-ABE, blockchain, access rights management

目录

插图目录

表格目录

第一章 绪论

1.1 研究背景

近些年来，伴随着互联网的迅速发展以及普及程度的加深，电子商务这种以信息网络为手段进行商务活动的新兴产业也得到了飞速的发展，在生活中应用得越来越广泛。电子商务在随着信息化深入的过程中，一方面为广大的人民群众提供了低成本、高效率、信息化的商务模式等诸多的便捷，另一方面也促进了物流行业的快速发展。据资料统计显示，2013 年中国快递业务量已经达到 92 亿件，业务量同比增加 60%，人均 6.8 件，位居世界第二，仅次于美国；2014 年我国快递业业务量达到 140 亿件，同比增长了 50%，位居世界第一[1]。京东商城为了提高自身在行业中的竞争力提出了“夜间配”、“极速达”、“次日达”等配送业务，由此可见物流配送和电子商务之间的关系已经变得越来越密不可分了。

在现在的物流信息系统中，为了方便物流企业实现收件、配送、发件等过程，发件人需要在快递单上填写很多的物流隐私数据，如表 1.1 列出了大部分物流快递公司快递单上需要发件人填写的物流隐私数据。在整个的物流过程中，这些物流隐私数据均处于明文可见的状态，势必会造成物流隐私数据泄露的问题。据报道，快递单信息在一些网站上被明码标价，并且提供“生成底单”等服务[2]。不仅如此，信息的泄露同时也带来了很多隐性但是非常严重的问题。例如破解密码，针对物流隐私数据采取的字典式攻击的成功率高达 50%，在这中间特别是年纪偏大人群设定的密码被成功破解的概率更是高达 70%~80%。除此之外，发件人在使用物流服务的过程中防范物流隐私数据泄露也是很被动的，虽然发件人可以通过填写假名来防止姓名信息泄露，但是其他的物流隐私数据（例如：电话信息、地址信息）依旧没有办法得到保护。

表 1.1 快递中包含的物流用户隐私数据

物流公司	发件人信息				收件人信息			
	单位	姓名	电话	地址	单位	姓名	电话	地址
顺丰快递	√	√	√	√	√	√	√	√
京东快递	×	√	√	√	×	√	√	√
圆通快递	√	√	√	√	√	√	√	√
德邦物流	√	√	√	√	√	√	√	√

目前，许多的物流公司采用的服务模式是分点集中、统一派送。图 1-1 显示了快递的派件流程。快递一般从发件人的手中集中到就近的物流中转站点，然后通过不同的物流中转站最终到达终点区域，最后将由该终点区域的派送员派送快递到收件人的手中。

物流过程中物流隐私数据泄露的三个环节为：

1. 新收快递信息的全公开。发件人在寄送快递的时候，需要在快递单上填写相关

的物流隐私数据。这些敏感的隐私数据均以明文的形式呈现在快递单上，收件人员以及快递服务点可以很容易地获得这些敏感的隐私数据。

2. 快递中转过程的全公开。在物流的中转过程中，物流企业的工作人员可以轻易地通过二维码扫描仪获取大量的物流用户隐私数据。
3. 快递终点派送的全公开。快递在最后的派送过程中，派送员可以轻易的通过二维码扫描仪获取大量的物流用户隐私数据。

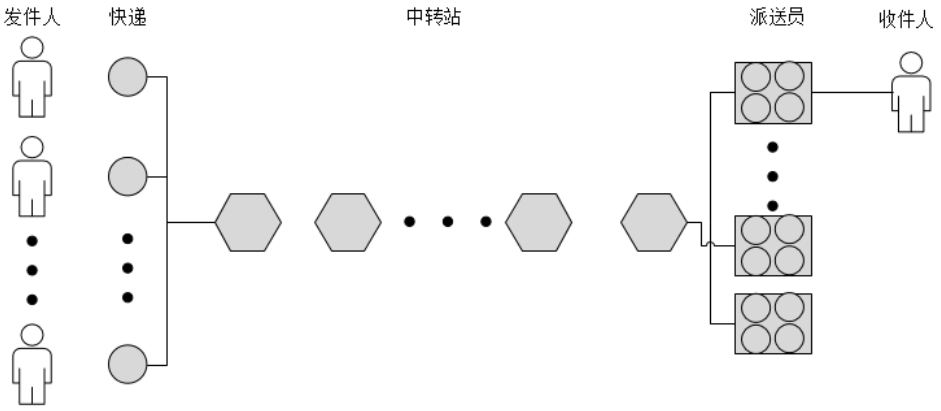


图 1-1 快递派件流程

现阶段针对物流行业中存在的物流隐私数据泄露的问题，国内的对策有：一方面从管理、制度、法规层面上着手[3]，一方面通过技术来保护物流用户隐私数据，例如京东快递、韵达快递已经尝试将二维码技术使用在信息分装中，物流企业的工作人员在物流的中转过程中通过专门的二维码扫描仪扫描二维码来获取物流隐私数据[4]。但是扫描获得的数据依旧是快递单上的所有的明文信息，物流隐私数据泄露的问题依旧没有得到完全的解决。主要的原因有两点：1) 物流企业并没有重视用户的物流隐私数据的保护；2) 在物流的中转过程中，依旧是靠人力来完成中转的过程，并且物流企业的工作人员依旧能够获得所有的明文信息。因此，对物流中隐私数据保护技术的研究也变得愈发的重要。

1.2 国内外研究现状

本文所设计的物流用户隐私数据保护方案主要涉及了物流隐私数据保护、区块链访问控制等领域，因此在国内外研究现状部分，将分别对物流隐私数据保护和区块链访问控制的国内外研究现状进行介绍。

1.2.1 物流隐私数据保护

随着物流行业的快速发展，物流信息系统(LIS，logistics information systems)[5]已经成为了物流行业中必不可少的工具。目前针对 LIS，国内外主要通过结合快速发展的 IT 解决方案，在应用软件互操作性标准化（EAI 技术）、产品配置、RFID 技术融合等方面上展开研究[6]。Rai 等人[7]通过将 LIS 应用在物流企业中提高管理模式并且取得更高的

市场效益。Kahraman 等人[8]基于从业人员的选择标准和物流信息技术评价，提出“层次模糊 TOPSIS 法”解决了 LIS 存在的评价与选择的问题，从而提高了 LIS 的服务质量和效益。Hazen 等人[9]通过研究 EDI 以及采用 RFID 技术来整合企业资源，从而提高企业竞争力。Wood 等人[10]利用 RFID 技术识别和跟踪库存、处理交易以及支持改进决策等技术，使 LIS 的效率得到提高。张彤等人[11]通过采用 SOABPM 组合架构解决系统松散耦合、物流企业间信息系统的异构问题。徐斌等人[12]提出了基于 OPC 技术实现物流自动化系统的继承，提高了整体运作效率和信息化水平。

以上是针对 LIS 的研究，对于 LIS 的物流隐私保护，国外一些发达国家主要是通过个人、政府和法律对物流行业进行规范管理和监督。美国规定所有的快递员在就业的时候都必须提供社会安全号[13]。英国邮政管理委员会[14]将所有与邮件安全相关的条款都写入到颁发给物流企业的许可证当中，通过法律的形式强制执行。这些方法在一定的程度上能够起到保护用户物流隐私的作用，但是隐私泄露的风险依然存在。

国内的一些研究人员对 LIS 中存在的物流隐私泄露问题进行了研究。韦茜等人[15]针对物流隐私泄露的问题，提出了一种 K-匿名模型对物流隐私进行匿名处理，但是姓名、联系方式和所在省市等个人信息依然保留在快递单上以方便物流企业员工实现物流中转。文献[16]提出了一种物流过程中不使用快递单的形式，而是设想将派件员当天需要派送包裹的收件人信息加密存储在 Android 手机中，并且使用家庭地址作为加密信息的文件名以配送至收件人地址，但是该方案在实际的应用中存在着一些局限性：1) 放弃使用快递单，无法实现物流的中转功能；2) 收件人的信息加密存储在手机中，依旧没有完全实现对物流隐私的隐藏加密。Zhang 等人[17]提出了一个基于二维码技术的个人信息隐私保护物流系统(LIPPS)，该系统采用了分段加密技术对物流用户隐私数据进行加密，并将加密之后的密文存储到二维码中，并设计了不同等级授权机制解密相应的信息，从而完成物流业务操作，但是该系统并没有给出具体的实现以及分级加密技术实现的细节。

从上述研究可知，现有的 LIS 研究的主要目标是针对物流平台服务的高效化、信息化、提高服务质量和增加企业效益，在物流用户隐私保护方面还存在很多的不足：1) 用户隐私数据依旧保存在非完全可信的第三方，用户隐私依旧得不到保障；2) 用户无法对物流隐私数据的访问权限进行控制管理，其控制过程缺乏透明性和追溯性的问题，并且无法控制第三方对隐私数据的部分读取。针对存在的问题，本文首先提出一种改进的 CP-ABE 算法实现基于属性的分层加密方案，然后结合区块链公开透明、无法篡改、方便追溯等特点，提出一种访问权限控制管理机制，下面对区块链访问控制研究进行介绍。

1.2.2 区块链访问控制研究

通常，系统中的访问控制是按用户身份及其所归属的某项定义组来限制用户对某些信息项的访问[18]。在传统的访问控制系统中，系统管理员决定了用户能够访问哪些特定的信息，即决定了用户的访问权限，这也就导致了传统的系统更容易受到黑客攻击以及匿名入侵[19]。区块链技术提供了安全的加密技术来识别和验证用户，从而能够以可扩展、

分布式和安全的方式设计访问控制结构。下面介绍区块链在访问控制方面的研究现状。

Guy Zyskind 等人[20]解决了在使用第三方应用服务时存在的隐私问题，提出了一个去中心的个人数据管理系统，确保用户拥有并控制他们自己的隐私数据。一般情况下，用户首次注册使用移动应用程序时，需要为应用程序的正常运行授予一组权限（如位置，联系人列表，相机等权限）。该文献提出的平台只允许用户根据存储在区块链中的访问控制策略来更改权限。该系统由三个实体组成：移动手机用户、服务提供商（应用程序）和维护区块链的节点，在区块链网络中，只接受两种类型的交易：1) T_{access}：用于访问控制管理；2) T_{data}：用于数据检索和存储。对于每个应用程序的用户都会生成一个和身份标识相关联的权限，并通过T_{access}发送到区块链中。从用户手机中收集到的数据被加密保存在区块链之外，只有数据的哈希值存储在区块链之中，用户和服务提供商可以通过T_{data}来查询相应的用户数据。

肖月等人[21]提出了一个医疗数据共享系统来解决医疗数据共享过程中存在的访问控制管理问题。在这个系统中，他们提出了一个基于区块链的应用程序框架，能够使用户轻松安全的拥有、控制和共享自己的医疗数据的同时，保证自己的隐私不会被侵犯，从而维护了用户医疗数据的隐私性。文章提出的以目的为中心的访问模式将用户分为两个类型：1) r-用户：表示试图读取原始数据的用户；2) p-用户：表示试图读取数据和检索结果的用户。对于每一个数据请求，一个交易由一个想要在有限时间内访问一个特定类别数据的请求者生成。

Azaria 等人[22]提出了一个基于以太坊智能合约的电子病历管理系统。该系统为患者提供了一个全面的、不可变的日志，并且用户可以轻松地访问自己的电子病历，同时该系统可以管理用户的身份、在共享电子病历的同时保障了电子病历的安全性并且提供了可追责的机制。该系统鼓励医疗利益相关者（研究人员、公共卫生机构等）以区块链“矿工”的身份参与到区块链网络中，然后将访问部分电子病历作为挖掘奖励给予他们。

Ouaddah 等人[23, 24]提出了一个基于区块链的物联网访问控制框架 FairAccess。该框架使用区块链的智能合约来分配访问令牌。他们提出的 FairAccess 存在着一些缺陷，例如方案仅支持基于令牌的访问授权、每次有新的访问请求或者访问令牌过期时都必须与资源所有者进行联系、获取访问令牌需要很高的时间成本等。

上面的几篇研究都是通过区块链来解决不同领域中的访问权限控制管理问题，但是依旧存在着一些问题：1) 用户身份隐私存在泄漏的风险；2) 基于以太坊或者比特币网络来进行权限管理，存在着吞吐量小以及交易验证较慢的缺点。本文研究区块链技术，提出了一种基于区块链和 DAA 匿名认证相结合的访问权限管理机制，具体的内容在文章后面介绍。

1.3 论文研究内容及意义

本文针对物流用户隐私数据的保护进行了研究。现有物流行业用户隐私泄露问题主要通过法律法规和物流企业提供安全保障，虽在一定程度上可以解决部分隐私泄露问

题，但依然存在很多不足：1) 用户隐私数据通常由物流企业类的第三方进行保管，但即无法保证第三方的可信性，也很难杜绝隐私数据的泄露和窃取。2) 用户即缺乏对隐私数据访问权限的管理，也无法决定隐私数据的访问范围。针对上述问题，本文提出了一种基于区块链技术的物流用户隐私数据保护方案，将隐私数据分层加密云存储和去中心化的访问权限控制管理相结合。在此基础上设计和实现了一物流下单原型系统，以验证该隐私数据保护方案的有效性。本文主要工作内容如下：

1. 在研究和分析现有隐私数据加密存储安全机制，以及数据访问权限控制管理方案的基础上，结合分层加密和区块链技术提出一种将隐私数据分层加密云存储和去中心化的访问权限管理相结合的物流用户隐私数据保护方案。该方案将隐私数据进行分层加密处理，将密文存储在云端，以防止隐私数据在不可信第三方平台上的泄露问题；同时提供去中心化隐私数据访问权限管理，由用户控制隐私数据的访问权限。
2. 针对用户隐私数据通常存放在不可信第三方数据保管平台的现状，本文提出了一种基于分层加密的隐私数据保护机制。该机制采用基于属性的分层加密方案，根据安全等级划分用户隐私数据的读取范围，通过安全属性表述数据访问方的身份并确定可读取的数据内容范围，在云存储平台上存储隐私数据密文。该方案可有效防止物流用户隐私数据的泄露。
3. 针对物流派发时用户隐私数据访问方身份无法预先获知从而无法授权的问题，本文提出了一种改进的 CP-ABE 算法实现基于属性的分层加密方案。该加密方案采用三层嵌套访问控制树架构，用户可根据数据访问方属性确定其数据访问权限，拥有较高权限级别的数据访问方可以访问低级别的数据。经实验表明，改进的 CP-ABE 算法在提供分层访问的同时，还拥有更高的解密效率。
4. 针对现有隐私数据访问权限管理方案中用户无法对物流隐私数据的访问权限进行控制管理，其控制过程缺乏透明性和追溯性的问题，本文提出了一种基于区块链和 DAA 匿名认证相结合的访问权限管理机制。该机制基于 DAA 匿名认证实现了成员身份管理，为区块链上的实体提供匿名可验证的身份，通过维护的一个交易公钥列表 TPL 来实现对区块链节点的访问控制，解决了当前区块链技术存在的隐私泄露的风险；同时，该方案通过分布式账本保存用户对隐私数据的访问权限，使用链码（即智能合约）来封装物流中各角色与隐私数据之间存在的业务逻辑，实现了用户对隐私数据的访问权限控制管理。
5. 在所提方案的基础上，本文设计并实现了一个物流下单原型验证系统，该系统由数据拥有方、数据访问方及云存储平台构成。其主要功能包括：物流订单生成、物流状态查询、隐私数据权限管理以及用户信息访问。该系统保证了物流用户隐私数据的安全性，提供了数据拥有方对物流数据的访问权限控制管理，实现了数据访问方对物流隐私数据的分层访问。经测试表明，该系统在保证隐私数据安全的同时，拥有较高的计算效率，用户与区块链模块的交互延时保持在 80-100ms 区间内，有一定的使用价值。

1.4 论文组织结构

本文分为了六章，每一章的主要内容如下：

第一章为绪论。首先介绍了物流行业的发展状况，并对物流行业中存在的物流隐私数据泄露的问题进行了分析；然后介绍了现阶段物流隐私保护以及区块链访问控制的研究现状以及研究成果；最后提出了本文的主要研究内容和章节安排。

第二章为相关技术研究。首先对本文中涉及到的密码学技术进行介绍，包括对称密码算法、HMAC 算法以及基于属性的加密算法；然后对区块链技术进行了相关的研究，包括区块链的基本原理、相关特点、区块链种类、区块链隐私定义及风险以及区块链技术中的智能合约；最后介绍了匿名认证技术，主要包括群签名、零知识证明以及 DAA 匿名认证方案。

第三章介绍了基于分层加密的隐私数据保护机制。本章首先对该机制的总体架构进行详细的描述；接着介绍了本章提出的一种改进的 CP-ABE 算法实现基于属性的分层加密方案；然后对基于分层加密的隐私数据保护机制的具体实现步骤进行描述，其中包括了用户注册、客户端加密和客户端解密；最后对机制的安全性和性能进行了分析。

第四章介绍了本文提出的基于区块链和 DAA 匿名认证相结合的访问权限管理机制。本章首先对许可链进行了简单的介绍；然后给出该机制总体架构的描述；最后对成员身份管理模块以及区块链模块进行了详细的说明。

第五章介绍了本文提出的物流下单原型系统的设计与实现。本章首先对系统的整体架构进行详细的描述，系统由数据拥有方、数据访问方以及云存储平台构成；然后分别介绍了系统功能模块中的分层加密模块以及访问权限管理模块的设计与实现；最后对系统的功能以及性能进行了相关的测试。

第六章为总结与展望。本章对本文所做的工作进行了总结，并对未来更进一步的研究工作进行展望和规划。

第二章 相关技术研究

2.1 密码学技术

本节介绍了文章研究过程中使用到的密码学技术，其中包括了对称加密算法、HMAC算法以及基于属性的加密算法。本文的研究中将使用对称加密算法对不同安全等级的物流隐私数据进行加密，并且使用 HMAC 算法为加密的物流隐私数据提供完整性认证，最后使用改进的 CP-ABE 实现的基于属性的分层加密方案对对称加密算法使用的对称密钥进行加密，这样做的原因是对称加密算法的运算效率比基于属性的加密方案高很多，既可以提供数据的安全性保证、完整性认证，又可以保证高效地完成对数据的加密。

2.1.1 对称加密算法

加密算法分为：对称加密算法和非对称加密算法。非对称加密算法不在本文的研究中，因此不做介绍。对称加密算法也称传统加密或者单钥加密，具有算法简单、运算效率高而且安全性较好等优点[25]。对称加密算法有 5 个基本成分：

1. 明文：原始可理解的数据或者消息，是加密算法的输入。
2. 加密算法：对明文进行各种代替和变换，生成密文。
3. 密钥：密钥是加解密算法的输入。密钥独立于算法和明文。加解密算法根据所输入的密钥而产生不同的输出。
4. 密文：作为加密算法的输出，看起来杂乱而随机的消息，依赖于明文和密钥。对于给定的明文，不同的密钥加密明文产生不同的密文。
5. 解密算法：本质上是加密算法的逆运算。输入密钥和密文，能够解密出原本的明文。

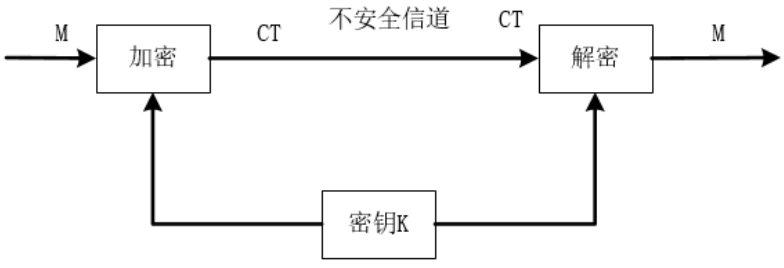


图 2-1 对称加密工作过程示意图

对称加密的工作过程如图 2-1 所示。对称加密算法是一对一的通信方式，在加密开始之前，通信双方需要提前商量好一个密钥，然后使用该密钥进行加解密操作。这种加密体制带来的缺点是当有 n 个用户需要互相通信的时候，如果想要实现任意两方进行保密通信，则需要 $n(n-1)/2$ 个密钥，每个用户都需要安全存储和管理大量的密钥，这是不太现实的。常见的对称加密算法有：DES、AES、SEAL 等。对称加密算法因其具有计

算效率高、安全性高和硬件开销小的优点，非常适合加密大量的数据。在本文中，就使用 AES-128 对称加密算法来加密不同安全等级的物流用户隐私数据。

2.1.2 HMAC 算法

HMAC 算法最早是在 1996 年由 H.Krawezyk，M.Bellare，R.Canetti 提出的一种结合密钥和 Hash 运算来进行消息认证的方法[26]。HMAC 能提供的消息认证包括以下的两个方面内容：

1. 消息完整性认证。可以证明数据在非安全通道传递的过程中有没有被修改。
2. 数据发送者身份认证。由于通信双方使用相同的一个密钥，接受方能够验证数据发送方的身份是否真实有效。

HMAC 算法将一个密钥和一个可行的 hash 算法结合，计算出消息认证码。设 H 是嵌入的 hash 算法，M 是输入 HMAC 的明文消息，将 M 分成长度是 b 的多个分组，k 是 HMAC 算法使用的密钥，如果 k 的长度小于 b 则在 k 的末尾用 0 填充到 b 比特长，记为 K；如果 k 的长度大于 b 比特，则 $K=\mathrm{H}(k)$，$opad$ 和 $ipad$ 是 HMAC 两个固定的长度为 b 比特的参数，HMAC 算法的描述如公式(1.1)：

$$\mathrm{HMAC}_K(M)=\mathrm{H}(IV,(K\oplus opad)\,\|\,\mathrm{H}(IV,(K\oplus ipad)\,\|\,M)) \tag{1.1}$$

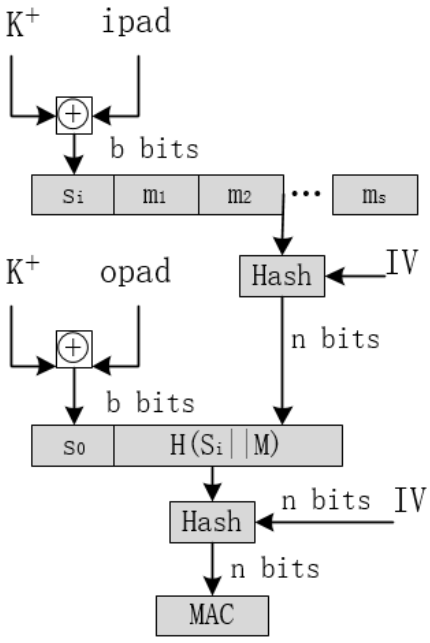


图 2-2 HMAC 的结构

HMAC 步骤描述如图 2-2 所示：

1. 在密钥 k 后面添加 0 生成 K。
2. 将步骤 1 生成的密钥 K 与 $ipad$ 做异或运算。
3. 将消息 M 的数据流填充至步骤 2 生成的字符串中。
4. 用 hash 函数 H 作用于步骤 3 生成的数据流。
5. 将步骤 1 生成的密钥 K 与 $opad$ 做异或运算。

6. 将步骤 4 生成的结果填充进步骤 5 生成的结果中。
7. 用 hash 函数 H 作用于步骤 6 生成的数据流，作为最终的结果输出。

2.1.3 基于属性的加密方案

2.1.3.1 简单介绍

基于属性加密（ABE，Attribute-Based Encryption）的思想最先是在 2005 年由 Sahai 和 Waters[27]在 EUROCRYPT 上提出，他们以基于身份的加密方案为原型[28]。ABE 算法可以实现一对多的通信模式，即数据拥有方根据访问控制策略使用 ABE 来加密数据，只要数据访问方拥有的属性集满足数据拥有方加密使用的访问控制策略的时候就能够对密文进行解密，因此 ABE 算法通常被用于解决数据共享的问题。

2006 年，Goyal 等人将 ABE 划分为密文策略的属性加密[29]（CP-ABE，Ciphertext-Policy Attribute-Based Encryption）和密钥策略的属性加密[30]（KP-ABE，Key-Policy Attribute-Based Encryption）。KP-ABE 将属性集与密文相关联，访问策略与密钥相关联；CP-ABE 则将访问策略与密文相关联，属性集与密钥相关联，与 KP-ABE 方案相比，CP-ABE 方案的数据拥有方在加密数据的时候可以设定一个任意的访问控制策略，自行控制共享的范围，数据加密之后，就已经确定了具有访问控制策略中规定的某些相关属性的用户才能够获得解密密文的权利。本文中也就是基于 CP-ABE 这一特点提出了一种改进的 CP-ABE 算法实现基于属性的分层加密方案，使得数据拥有方在将数据分为低、中、高三个不同的安全等级之后，根据数据访问方的属性集授予不同的隐私数据访问权限，使得拥有高访问权限的访问方能够解密高、中、低三个安全等级数据的密文得到相应的明文数据。

2.1.3.2 相关定义

1. 属性集

设 P_x 代表一个属性，则 $P=\{P_1,P_2,\cdots,P_n\}$ 是所有属性的集合，一个属性集 A 是 P 的一个非空子集，即 $A\subseteq 2^{\{P_1,P_2,\cdots,P_n\}}$。如果对于任意的属性集 B 和 C，$B\in A$ 且 $B\subseteq C$，则 $C\in A$，我们就称属性集 A 是单调的。用户定义的访问控制结构即是属性集合 P 的一个非空子集 A，如果一个属性集是 A 的子集，则这个属性集被称为授权集合，反之就被称为非授权集合。

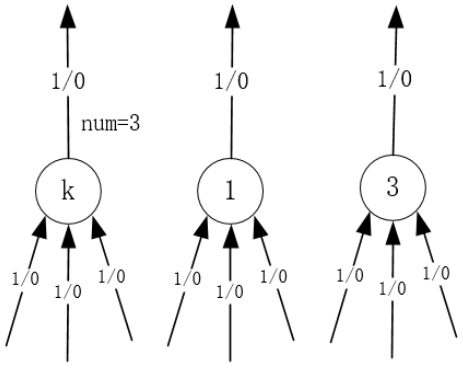


图 2-3 门限的三种情况

2. 门限

一个门限就是一个逻辑的运算单元，门限具有 $num(num \geq 1)$ 个输入和 1 个阈值 $k(0 < k \leq num)$，每个输入都只有 $1/0$ 两种状态，称为 (k,n) 门限[31]。当状态为 1 的输入的个数大于等于阈值 k 的时候，门限输出就为 1，否则门限输出为 0。特别的，当 $k=1$ 的时候，该门限表示一个或门(OR)，即只要输入中有一个状态为 1，门限输出就为 1；当 $k=n$ 的时候，该门限表示一个与门(AND)，即只有说有的输入的状态为 1，门限输出才为 1。图 2-3 分别表示了输入为 3 的门限、输入为 3 的或门和输入为 3 的与门三种情况。

3. 访问控制树

ABE 算法的主要特性是用户能够自己设定一定的权限来对解密数据的用户群体进行限制，而完成这个特性的主要技术就是访问结构，用户可以通过设置一个特定的访问结构来控制谁能够成功地完成对密文的解密动作，而访问结构的验证就是通过验证与用户绑定的属性集是否满足用户设定的访问结构的要求。目前访问结构实现的方法有很多，常见的方法有：访问控制树、LSSS 矩阵访问结构[32]、“与”门访问结构[33]等，访问控制树相比于另外两种访问结构更加直观易懂，最主要的是能够灵活的表示多种复杂的访问结构，因此，在本文提出的基于属性的分层加密方案利用访问控制树来实现访问控制。

访问控制树通过一个访问树来表示一个访问控制策略。其具体的表现形式是通过树形的结构来完成的，其中树中的每一个叶子节点都与一个属性绑定，并使用 $att(x)$ 表示与叶子节点 x 对应的属性元素，而树中的每一个非叶子节点都表示一个门限。一般使用 T 来表示一棵访问控制树，使用 r 来表示树中的根节点，用 x 表示 T 中的任意一个非叶子节点，定义 num_x 表示叶子节点 x 的子节点的个数，用 k_x 表示非叶子节点 x 的门限值，且 k_x 需要满足 $0 < k_x \leq num_x$。非叶子节点使用逻辑单元“OR”表示的时候，该非叶子节点的门限值为 $k_x = 1$；非叶子节点使用逻辑单元“AND”表示的时候，该非叶子节点的门限值为 $k_x = num_x$。当 x 作为叶子节点的时候，可以认为它的门限值为 $k_x = 1$。树节点 x 的父节点使用 $parent(x)$ 表示。访问控制树 T 对每一个非叶子节点的子节点都从 1 到 num_x 进行编号，并且使用 $index(x)$ 表示节点 x 在兄弟节点中的编号。

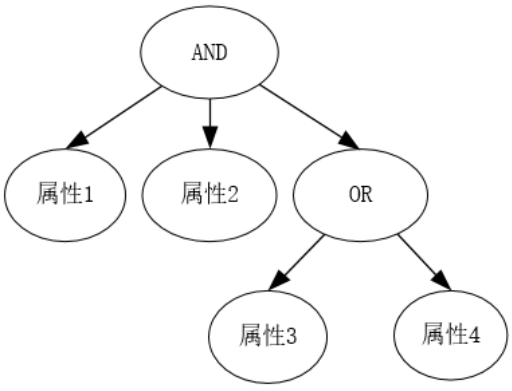


图 2-4 访问控制树示例图

访问控制树的一个很大的优点就是可以将用户设定的访问结构使用图形直观地表示出来，这样用户就能够通过图形中的结构快速地判断出来一个用户的属性集是否符合自己设定的访问结构。图 2-4 中给出了一个访问控制树的实例。

图 2-4 中的访问控制树对应的访问结构为{属性 1AND 属性 2AND（属性 3OR 属性 4）}。从访问控制树中，可以直观判别出满足该访问结构的属性集有：{属性 1、属性 2、属性 3}、{属性 1、属性 2、属性 4}以及{属性 1、属性 2、属性 3、属性 4}，除了这 3 个属性集之外，其他的属性集都不会满足该访问结构。

4. 满足访问控制树

当用户访问使用 CP-ABE 算法加密的数据时，需要判断该用户的属性集γ是否满足数据加密方设定的访问结构。符号T表示一棵根节点为r的访问控制树，使用T_x表示T子节点为x的子树，这时T也可以表示为T_r。如果属性集γ满足访问控制树T_x，就记为$T_x(\gamma)=1$。$T_x(\gamma)$可以按照下面的递归方法计算：当x是访问控制树的叶子节点时，如果用户的属性集γ中包含了x节点代表的属性，那么$T_x(\gamma)=1$；当x是访问控制树的非叶子节点时，就需要先计算出节点x的所有子节点的$T_x(\gamma)$值，如果$T_x(\gamma)=1$的节点个数大于节点x处设置的阈值k_x时，那么$T_x(\gamma)=1$。通过这样的计算，一直递归计算到访问控制树 T 的根节点r，求得$T_r(\gamma)$的值。如果$T_r(\gamma)=1$，就说明该属性集γ满足该访问控制树设定的权限要求，该用户能够成功进行解密操作；如果$T_r(\gamma)=0$，就说明该属性集γ不满足该访问控制树设定的权限要求，解密运算将被终止，解密失败。

2.1.3.3 CP-ABE 算法流程

本小节对本文使用的 CP-ABE 算法的基本流程进行简单的介绍。CP-ABE 算法包括四个基本步骤：初始化、密钥生成、加密和解密。其各个步骤的描述如下：

1. 初始化

该步骤由授权机构执行，输入为安全参数，输出 CP-ABE 算法的公钥PK和系统主密钥MK，其中系统主密钥MK由授权机构安全保存，公钥PK对所有用户公开。

2. 密钥生成

该步骤同样由授权机构执行，输入系统主密钥MK以及和用户的属性集γ，输出该用户对应的解密私钥SK，并通过安全的通道发送给用户。

3. 加密

该步骤由数据拥有方执行，算法使用如下三个输入参数：公钥PK、将要被加密的明文数据M以及数据拥有方定义的访问控制结构T。该算法将明文数据M加密生成密文CM，只有当其他用户绑定的属性集满足数据加密方设定的访问控制结构T的时候，才能够解密CM得到明文数据M。我们可以认为访问控制结构T是隐含在密文CM中的。

4. 解密

解密算法有以下三个输入参数：数据访问方的解密私钥SK、系统公钥PK、密文

CM。如果数据访问方关联的属性集γ满足密文CM中隐含的访问控制结构T时，那么该数据访问方就能够使用他的解密私钥SK成功解密密文CM，得出明文数据M，否则不予解密。

2.2 区块链技术

中本村所撰写的区块链白皮书[34]是公认的最早介绍区块链的描述性文献。但是在该文献中，重点是讨论比特币系统，实际上并没有明确的提出区块链的概念和定义。该文献指出了，区块链是用来记录比特币交易账目历史的数据结构。

Wikipedia 上给出的定义较为详细："区块链是用分布式数据库识别、传播和记载信息的智能化对等网络，起源自比特币。区块链是一串使用密码学方法相关联产生的数据块，每一个数据块中包含了若干次比特币网络交易的信息，用于验证其信息的有效性（防伪）和生成下一个区块"[35]。

传统的记账方式是中心化的，而区块链技术采用一种去中心化的分布式方式来记录交易数据[36]，如图 2-5 所示。

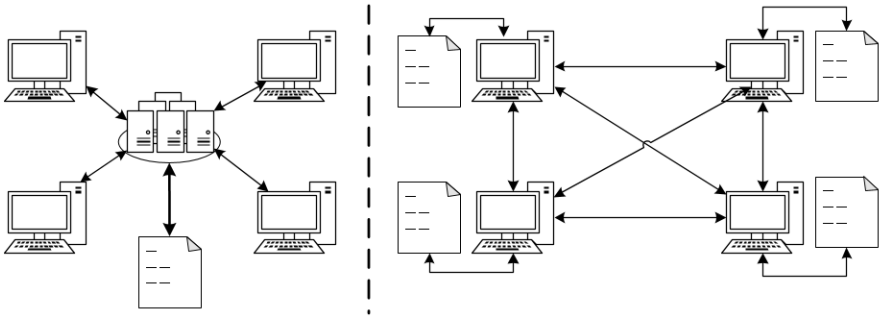


图 2-5 中心化记账方式和去中心化记账方式对比

区块链技术虽因数字货币而生，但因其实现了去中心化的共识、高度安全等特性，正广泛应用在各个领域之中[37]，如表 2.1 所示。国外众多公司（例如 IBM、德勤、德意志银行等）纷纷发布了自己的区块链设计方案。Google 和 SONY 也分别启动了基于区块链的合作项目。2016 年 10 月 18 日，中国电子技术标准化研究院在工信部信软司和国标委工业标准二部的指导下联合了国内很多家从事区块链研发的骨干企业共同成立了"中国区块链技术和产业发展论坛"，并同时发表了首个《中国区块链技术和应用发展白皮书》。

表 2.1 区块链应用场景

分类	实例
金融保险	股票交易、股权管理、债券、集资、基金管理、保险证明等
产权证明	实物产权（例如房产证明、车辆登记等）、无形资产（例如专利、商标等）
社会生活	公正证明、遗嘱、彩票、投票等
其他	物联网、医疗数据共享、物品溯源、身份证明、电子签名等

2.2.1 基本原理

区块链作为一个分布式的账本数据库，所有参与到区块链网络中的节点都是独立的，并不属于同一个组织，节点之间也并不需要任何的信任。区块链网络中的数据是由网络中的所有节点共同维护的，网络中的每一个节点都有一份完整的交易记录。

区块链包含了三个基本概念：

1. 交易（transaction）

一次对账本进行的操作，这个操作将会导致账本状态发生一次改变，例如添加一条转账记录。

表 2.2 区块结构

大小	字段	描述
4 字节	区块大小	利用字节表示该字段之后的区块大小
80 字节	区块头部	组成区块头部的字段
1-9（可变的整数）	交易计数器	区块中包含的交易数量
可变大小	交易数据	记录在区块内的交易数据

2. 区块（block）

区块是将一段时间内发生过得所有的交易数据记录下来，区块主体可以说就是交易数据的集合。不同类别的区块链的区块结构可能会有不同，但是区块整体的结构一般包含区块头（Header）和区块体（Body）两个部分，如表 2.2 所示。区块头的作用是和之前的区块进行一个链接，并且为区块链数据提供一个完整性保证，区块头的数据结构如表 2.3 所示。区块体中则主要包含了交易计数和所有交易的详情。

3. 链（chain）

区块按照产生的先后顺序串联形成的一条链，是整个账本状态变化的记录[38]。

表 2.3 区块头部结构

大小	字段	描述
4 字节	版本号	用于表示当前系统的版本
32 字节	父区块哈希	引用前一个区块的哈希值
32 字节	Merkle 根	该区块中交易的 merkle 树根的哈希值
4 字节	时间戳	产生该区块的时间
4 字节	难度目标	该区块工作量证明算法的难度目标
4 字节	Nonce	随机数

以比特币的交易为例，在区块链中的交易并不是实际中的一手交钱一手交货，而是交易双方转账的行为。在比特币的交易中，为了使得价值易于分割和组合，交易被设计为可以有多个输入和输出，即一笔交易可以转账给多个人。交易从生成到在区块链网络中传播，然后通过共识机制（在这里是工作量证明）、整个网络节点验证，最后记录到

区块链中，就是区块链交易的整个生命周期。整个区块链的交易流程如下所述：

1. 交易的生成。交易发起方 A 利用他的私钥对前一次交易和交易的接收方 B 签署一个数字签名，并且将这个签名附在此次交易的货币的末尾，形成交易单。
2. 交易的传播。交易发起方 A 将交易单广播至区块链全网中，网络中的每一个节点都会将收到的交易信息纳入到一个区块中。
3. 工作量证明。网络中的每个节点通过相当于解一道数学难题的工作量证明机制，获得创建一个新区块的权利，并且得到一定量的数字货币的奖励。
4. 整个网络节点验证。当网络中的某个节点率先找到解时，就会将它创建满足难度条件的新区块在网络中广播出去，并由其他的节点进行验证。
5. 区块加入区块链。新区块得到网络中所有节点的验证之后，就会被链接到区块链中，网络中的节点重新开始竞争创建下个新区块的能力，这样就形成一个合法记账的区块链。

上述的比特币区块链交易流程的步骤 3 和步骤 4 可以统称为共识节点执行共识机制来生成并确认区块是否正确，如果正确就可以被链接到区块链中。

2.2.2 区块链特点

区块链具备去中心化、公开透明、匿名、无法篡改、方便追溯等特点。

1. 去中心化是指区块链数据的存储、验证和传输等过程都是基于分布式的系统结构，整个区块链网络中没有中心化的硬件或者管理机构。作为区块链的一种部署模式，许可链网络中所有参与的节点都具有相同的权利和义务。

2. 公开透明是指区块链网络中的每一个节点都保存了账本的一个备份，并且实时地获取最新的数据。区块链数据可以被任意地查询和检索，它本身是没有被加密的。

3. 匿名是指由于区块链技术解决了节点之间信任的问题，因此在交易的时候，交易的双方节点可以在匿名的方式下进行交易。

4. 无法篡改是指区块链上的数据是无法被更改的。因为区块之间是根据时间的先后顺序链式存储，并实现被数学证明可靠的验证功能，确保无法篡改，所以要想修改区块链上的数据，那就必须要通过验证，要想通过验证，就必须修改整个的区块链，这其中需要付出的代价是非常巨大的。

5. 方便追溯是指区块链中每一笔交易都是通过了密码学的方法与前后的两个区块之间串联起来的，可以从任意一个区块向前追踪，并且能够通过区块链查找到与之关联的所有的交易。

2.2.3 区块链分类

区块链以参与方来分类可以分为：非许可链（Permissionless Blockchain）、许可链（Permissioned Blockchain）。下面简单介绍一下两种类型的区块链。

1. 非许可链

非许可链对外公开，用户不需要注册就能够匿名参与，并且无需授权就能够访问区块链网络和区块链数据。区块链网络中的节点可以自由的进出网络。非许可链上的区块

数据可以被任何人查看，是真正意义上的完全去中心化的区块链，它通过密码学保证交易是不可篡改的，并且利用密码学验证以及经济上的激励，在陌生的网络环境中建立出节点间的共识，从而形成去中心化的信用机制。非许可链上的共识机制一般是工作量证明（POW）或者权益证明（POS）。

比特币和以太坊[39]等都是非许可链。非许可链一般适合应用于虚拟的货币、面向大众的互联网金融、电子商务等 C2B、B2C 或 C2C 等应用场景。

2. 许可链

许可链则是并不对外公开，用户需要注册才能够参与到区块链网络中，也就是许可链是仅限于已经验证身份的成员参与。超级账本（Hyperledger）[40]就属于许可链的架构。

许可链的共识过程是有预先选定好的节点来完成的。许可链一般适用于机构之间进行交易、清算或者结算的 B2B 场景。例如在银行之间进行清算、结算、支付的系统就可以使用许可链，将每家银行的网关节点作为结账节点，当区块链网络中有超过 2/3 的节点确认了一个区块的时候，该区块中记录的交易就会得到全网的认可。因为许可链中参与到共识过程的节点比较少，许可链采用的共识机制一般是权益证明或者 PBFT（Practical Byzantine Fault Tolerant）、RAFT 等共识算法。许可链对交易每秒交易数、确认时间都和非许可链有着很大的区别，对安全和性能的要求比许可链要高。

2.2.4 区块链隐私定义及风险

在通常的信息系统中，隐私是指数据拥有方不希望被其他人获得的敏感数据或者数据所表征的特性[41]。

在区块链技术中，为了在网络中各节点之间维持数据同步并对交易达成一定的共识，各节点必须公开一些信息，例如交易的内容和建议使用的区块链交易地址。通过分析区块链的特点，在本文中将区块链中存在的隐私分为了两类：

1. 身份隐私

身份隐私是指节点的身份信息与交易所使用的区块链交易地址之间的关联关系。区块链交易地址是节点在区块链系统中使用的假名，节点在交易的时候使用该区块链交易地址作为输入账号或者输出账号。两种常见的区块链交易地址如下所示：

1） 比特币地址：“1DAY1DupbBdGLkkFYj32J5g4h9X2zsxDv5”

2） 以太坊地址：“02B51B20185C04D1CbDA2996dFA02AF2D308EeEa”

区块链交易地址由节点自行生成，与节点的身份信息无关，节点创建和使用地址不需要第三方的参与。因此，相对于传统的交易账号（例如：银行卡账号等），区块链交易地址具有较好的匿名性。

2. 交易隐私

交易隐私是指存储在区块链中的交易记录以及交易记录背后的知识。在之前的区块链应用中，交易记录通常都是公开的，并不需要额外的保护措施，但是随着区块链技术被应用到银行等金融领域中，交易记录属于高度敏感的数据，需要采取额外的措施限制非授权用户的使用。

身份隐私和交易隐私是用户在使用区块链技术时需要重点保护的数据，如果这些内容泄露将会对用户造成一定的危害。

区块链技术在保护隐私方面存在着一定的安全风险。区块链网络中的节点通过区块链地址来进行节点间的交易，虽然区块链地址并非与节点的身份信息有关，但是由于区块链数据是完全公开透明的，随着各类反匿名身份识别技术和大数据技术的飞速发展，攻击者通过分析交易中存在的关联关系，攻击者能够逐步的降低区块链交易地址的匿名性，甚至能够将区块链交易地址与节点的真实身份相对应起来。

2.2.5 智能合约

智能合约就是执行合约条款的程序代码，在区块链技术中得到了长足的发展，以 Hyperledger 和以太坊为代表的区块链将智能合约的应用推向了更高的水平。最早由密码学家尼克萨博（Nick Szabo）[42]在 1994 年提出，用智能合约替代传统的纸质合约能够很大程度上减少在合约制定、执行效能和控制协议上的人工花费与计算成本。

对于区块链技术中的智能合约可以从以下 5 点进行理解：1) 通过一段代码或者脚本来实现业务逻辑；2) 具有图灵完备性[43]；3) 能够注入到区块链的执行环境中执行；4) 是由时间驱动；5) 具有状态。从安全的角度来说，首先智能合约同区块链数据一样拥有不可篡改、分布式、一致完整性等特点；其次，智能合约也作为一种保障区块链安全的技术手段。在智能合约中规定了参与节点的权利和义务、触发合约执行的条件以及执行的对应结果，当智能合约被注入到区块链中之后，就不会收到任何一方的影响，智能合约能够客观准确地执行，得到相应的结果。

2.3 匿名认证技术

在许可链中，节点产生的交易被记录在区块之中，如果区块链网络中存在着有竞争关系的节点，节点的一些隐私信息就会暴露出来，为了防止出现这样的隐私问题，可以使用 Brickell 等人[44]提出的 DAA 匿名身份认证方案，该方案基于群签名和知识证明，区块链网络中的节点发起交易的时候就不会泄露自己的身份等隐私信息。本节对本文提出的匿名身份认证方案所用到的相关概念与技术进行介绍。

2.3.1 群签名

Chaum 和 Heyst 在 1993 年首次提出了群签名的概念以及相应的协议方案[45]，一个基本的群签名系统一般由一个群管理员和若干个群成员组成，这些成员构成一个集合称为群。群签名的基本思想是任何合法的群成员都能够以群的名义来进行签名，同时对于其他的成员来说只能使用群公钥确认该签名是不是由群中某一个合法的成员所产生，却不能获得产生签名的成员的真实身份信息，只有在需要的时候由群管理员来打开签名，找到真正的签名成员。群签名一般由以下三个基本的属性要求：1) 对于一个消息，群中只能有一个成员对其进行签名，而不能同时有多个群成员对这个消息进行签名；2) 成员收到一个签名的时候，只能够对该签名的有效性进行验证，而不能获得具体签名成员

的身份信息；3) 当产生纠纷的时候，只有群管理员能够揭露具体产生签名的是哪一个成员。群签名的流程如图 2-6 所示。具体的步骤描述如下：

1. 系统初始化：群管理员创建群系统，一个多项式时间概率算法输入安全参数，输出群公开参数、群公钥和群私钥。
2. 成员加入：该步骤是在群管理员和用户之间执行的，一个想要加入群的用户需要与群管理员交互，然后群管理员生成该用户的密钥，并将生成的群证书发送给用户，用户就成为群的成员之一了。
3. 成员签名：输入一个消息 m 、一个群成员的群证书和签名密钥后，生成一个群签名。
4. 成员验证：输入一个群签名、群公开参数、群公钥和消息 m ，验证该群签名是否有效。
5. 成员揭露：该算法由群管理员来完成，输入一个群签名、群私钥、群公开参数、消息 m ，利用一个有效的签名揭露算法输出签名用户的身份信息。
6. 成员撤销：该算法由群管理员来完成，群管理员可以撤销群中的某个用户，之后该用户就不能再生成有效的群签名。

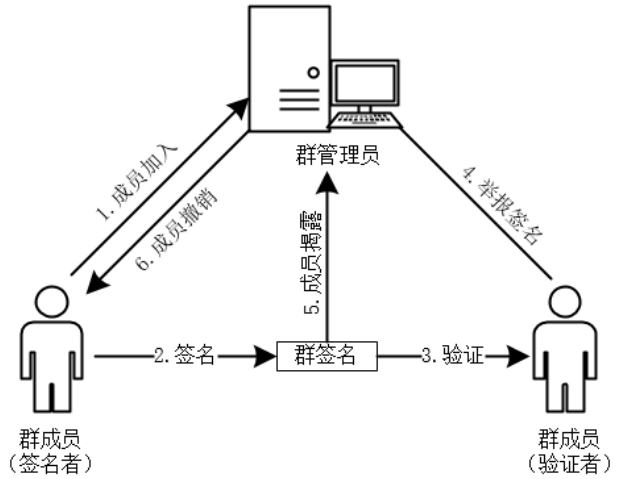


图 2-6 群签名流程图

一个可靠的群签名方案不仅需要有灵活、安全的成员加入和撤销过程，而且还需要满足如下的安全性：

1. 正确性：一个合法的群成员按照正确的签名算法产生群签名，那么该群签名就一定能够被验证者通过验证，并且在必要的时候，群管理员能够通过群签名得到签名者的真实身份。
2. 匿名性：一个群成员产生的群签名对于其他的任何成员来说，他们只能验证该签名是不是由群中合法的成员生成，而无法得到该签名者的具体身份信息。
3. 防伪造攻击性：群签名只能由合法的群成员产生，并且一个群成员想要得到其他成员的密钥在计算上是行不通的，所以无法伪造出其他群成员的签名。
4. 防联合攻击性：任意数目的合法群成员想要联合起来产生一个其他成员的签名，

并且做到无法让群管理员追踪到真实的身份是不行的。

5. 可追踪性：在特殊情况下，如果一个群签名被举报，群管理员是能够打开签名获得签名者真实的身份信息，并且签名者不能够否认该签名是自己产生的。
6. 非关联性：只有群管理员能够判断多个签名是不是由同一个签名者产生，其他的群成员并不能判断。

2.3.2 零知识证明

零知识证明[46]（ZKP，Zero Knowledge Proof）这个概念最早是由 S.Goldwasser、S.Micali 及 C.Rackoff 提出的。在密码学中，零知识证明是各种密码协议重要组成部分之一，在密码协议的分析和设计当中有着重要的地位，它的主要功能是如果存在一件真实的事，协议的一方证明者（Prover）能够在脱离事实却又不向协议另一方验证者（Verifier）提供任何关于有用信息的情况下，使验证者相信该事的称述是真实可信的。零知识证明可以分为交互式零知识证明和非交互式零知识证明，S.Goldwasser 等人提出的零知识证明就是交互式零知识证明，证明者和验证者之间必须进行交互。非交互式零知识证明是使用一个短随机串来替代交互过程，并且实现零知识证明。下面通过一个开门的例子简单说明零知识证明，如图 2-7 所示。

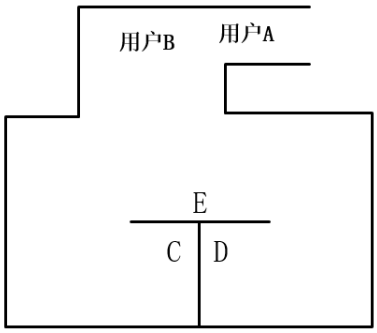


图 2-7 零知识证明例子

在图 2-7 中，C 和 D 之间是一扇门，用户 A 和用户 B 必须通过钥匙才能够打开该门，用户 B 声称自己有门的钥匙，并通过下面的方法向用户 A 证明自己有钥匙：

第一步：用户 A 在入口处等待，用户 B 来到 C 点或者 D 点。

第二步：用户 A 来到 E 点，然后 A 让用户 B 从 C 点方向或者 D 点方向出来。

第三步：如果用户 B 有钥匙，那他就能够按照 A 的要求从正确的方向出来，即他在 C 点方向，用户 A 要求他从 D 点方向出来，B 也能使用钥匙打开门从 D 点方向出去。

第四步：多次重复上面的三个步骤，如果每次用户 B 都能够从用户 A 要求的方向出来的话，那么用户 A 就能够相信用户 B 具有门的钥匙。

从这个例子可以看出，用户 A 虽然没有亲自看到用户 B 拿出钥匙开门的情况下，通过反复的让用户 B 从自己要求的方向出来从而确定用户 B 有没有钥匙，该过程就称为零知识证明，钥匙就代表了秘密知识。根据投硬币的原理，用户 B 从一个方向出来有两种可能，一个是 B 本来就在那个方向，一个是 B 通过打开门从另一个方向过来，因此

如果用户 B 没有钥匙还能够每次都成功的从用户 A 要求的方向出现的概率是$1/2^n$，n 是用户 A 让 B 反复试验的次数。

2.3.3 DAA 认证方案

当前计算机安全技术的一个主要趋势是可信计算技术。为了实现可信计算平台的远程认证，可信计算组织（TCG，Trusted Computing Group）先后提出了基于可信第三方的 privacy CA 和直接匿名认证（DAA，Direct Anonymous Attestation）两项技术，这两项技术确立了远程证明的基本原则。

privacy CA 方案中将一个可信的第三方 privacy-ca 作为证书发布中心，TPM（Trusted Platform Modules）可信计算模块得从 privacy-ca 处获得自己的身份证书。当 TPM 向一个验证方证明他的身份时，TPM 将身份证书发送给验证方，然后验证方将 TPM 的身份证书发送给 privacy-ca，privacy-ca 证书发布中心对验证方发来的 TPM 的身份证书进行验证，最后将验证结果发送给验证方，到这验证方才能在不知道 TPM 的身份信息情况下完成对 TPM 的身份认证。privacy-ca 参与到了身份认证的每一步，这也是 privacy CA 方案的瓶颈，可能会出现 privacy-ca 和验证方勾结的情况。

相比较与 privacy CA 而言，DAA 方案只需要颁发一次证书，效率较高。另外，DAA 方案采用了基于知识的离散对数算法以及 Camenisch-Lysyanskaya 群签名方案[47]。在本质上 DAA 认证方案可以看做是一种不能公开群成员身份的群签名方案，该方案证书发布中心并不是每次都参与，用户只要向证书发布中心申请一次 DAA 证书，就能够多次使用，从而降低了对发布中心的依赖。DAA 方案主要包括了三个子协议：初始化协议（Setup）、加入协议（Join）和签名/验证协议（Sign/Verify）。图 2-8 表示的是 DAA 方案的模型图，方案涉及到了三个实体：DAA 证书发布中心（Issuer）、用户（TPM 及相应的 host 组成）和验证方（Verifier）。用户选取一个秘密的随机数，通过一个安全的协议与 Issuer 进行交互；Issuer 给用户颁发 DAA 证书，即 Issuer 在秘密的随机数上的 CL 签名，当用户向 Verifier 证明自己的身份时，用户就利用秘密随机数和 DAA 证书对消息进行 DAA 签名；Verifier 对消息进行验证，验证通过的话就证明用户通过知识证明得到了匿名签名，整个身份验证过程中，Verifier 并不会得到用户任何的身份信息。

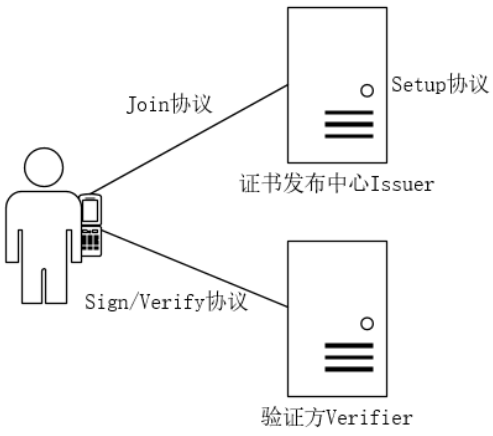


图 2-8 DAA 方案模型图

2.4 本章小结

本章主要对基于区块链的物流用户隐私保护中涉及到的相关技术进行介绍。首先介绍了密码学相关技术，包括对称加密算法、HMAC 算法以及基于属性的加密算法；其次对区块链的基本原理、特点、种类、智能合约以及存在的隐私风险进行介绍；最后介绍了匿名认证相关技术，包括群签名、零知识证明以及直接匿名认证方案。

第三章 基于分层加密的隐私数据保护机制

3.1 引言

本文 1.1 节中提到目前许多的物流企业的采用的服务模式是分点集中、统一派送，该模式中快递首先从发件人手中集中到就近的中转站点，然后通过不同的物流中转站最终到达终点区域，最后由该区域的快递员将快递派送到收件人的手中。通常物流过程中涉及到的角色有：发件人、收件人、物流中转人员以及快递员。一个物流过程一般由发件人首先发起，即发件人通过物流企业寄送一件物品，快递员将快递集中到就近的物流中转站点，然后通过不同的物流中转站最终到达快递收件人所在的终点区域，最后将由该终点区域的快递员将快递派送到收件人的手中。

该物流过程中存在着物流用户隐私数据泄露的风险。存在的风险主要有：1) 用户隐私数据通常由物流企业类的第三方进行保管，但即无法保证第三方的可信性，也很难杜绝隐私数据的泄露和窃取；2) 物流过程中各角色获得多余用户隐私数据的情况依旧存在；3) 物流过程中用户的隐私数据很多依旧是以明文的方式保留在快递单上。物流过程中用户的隐私数据通常可以分为以下几类：发件人及收件人隐私信息、寄送物品的描述信息以及物品的物流路径信息。通过表 3.1 可以得知各角色在物流过程中对应的业务操作、理想情况接触到的用户隐私数据（即获得最少的用户隐私数据）、实际接触到的以及是否获得多余的隐私数据。因为物流过程中各个角色并不是完全可信的，所以当一个角色获得了多余隐私数据时，就存在着隐私数据泄露的风险。

表 3.1 物流过程中涉及角色、业务操作及获得的用户隐私数据

角色	业务操作	实际得到的数据	理想情况得到的数据	是否得到多余数据
发件人	寄出包裹，物流跟踪，提供快递单号信息	所有数据	所有数据	否
快递员	接收包裹、派送包裹	所有数据	收件人信息、发件人信息	是
物流中转人员	分拣、中转包裹	所有数据	物流路径信息	是
收件人	物流跟踪，接收包裹	所有数据	所有数据	否

对于以上物流过程中存在的隐私数据泄露的风险，本章提出了一种基于分层加密的隐私数据保护机制，该机制采用基于属性的分层加密方案，根据安全等级划分用户隐私数据的读取范围，通过安全属性表述数据访问方的身份并确定可读取的数据内容范围，

在云存储平台上存储隐私数据密文。该机制可有效的防止物流用户隐私数据的泄露，实现了数据访问方对隐私数据的分层访问，从而不会获得多余的隐私数据。

基于属性的分层加密方案是通过改进的 CP-ABE 算法实现的。该加密方案采用三层嵌套访问控制树结构，用户可根据数据访问方关联的属性集确定其隐私数据的访问权限，拥有较高级别访问权限的数据访问方可以访问低级别的数据。该分层加密方案具体的描述将在 3.3 节中给出，下面对基于分层加密的隐私数据保护机制的整体架构进行介绍。

3.2 整体架构

在本章提出的基于分层加密的隐私数据保护机制中将物流用户隐私数据划分为三个安全等级：1（low）、2（mid）和 3（high），使用对称加密算法对于每一个安全等级的隐私数据进行加密，在保证数据机密性的同时具有较高的加密效率；然后采用 HMAC 算法来保证隐私数据的完整性，即通过 HMAC 算法计算出隐私数据密文的 *MAC* 码，数据访问方在解密隐私数据密文之前，可以再次计算 *MAC* 码，通过比较两者是否相等来判断用户隐私数据密文是否完整；接着通过基于属性的分层加密方案对对称加密使用的三个对称密钥以及 HMAC 算法密钥进行分层加密，实现了数据访问方对物流隐私数据的分层访问；最后将得到的密钥密文文件、隐私数据密文以及隐私数据密文的 *MAC* 码一起上传至云存储平台。

基于分层加密的隐私数据保护机制的整体架构如图 3-1 所示，主要有四个参与方，分别为：可信任的授权中心（TA）、客户端、云存储平台和用户，各部分详细的介绍如下：

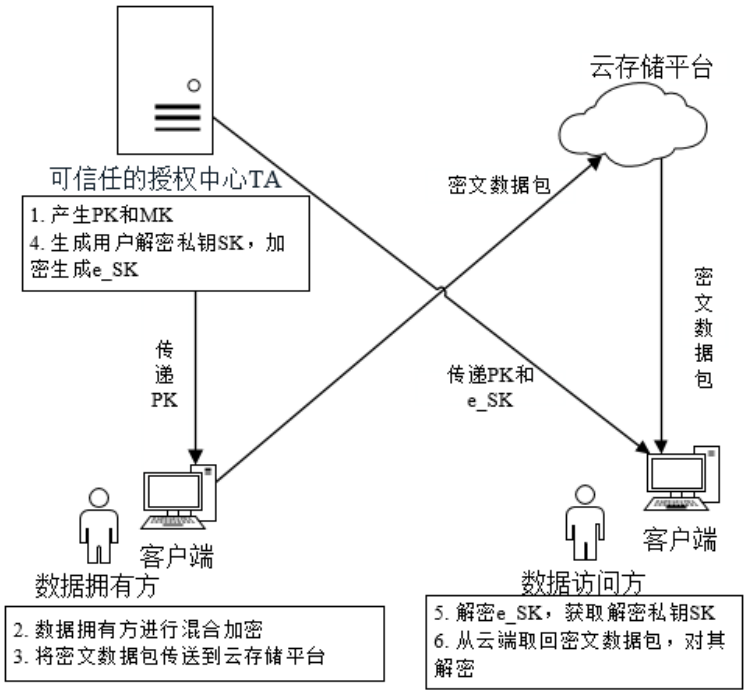


图 3-1 基于分层加密的隐私数据保护机制整体架构图

1. 可信任的授权中心（TA）

TA 由属性授权、密钥生成、公开数据库以及密钥管理四个模块组成。属性授权模块负责数据访问方属性集的注册；密钥生成模块负责执行基于属性的分层加密方案的初始化操作生成公钥 PK 和系统主密钥 MK，并根据数据访问方的属性集生成他的解密私钥 SK；公开数据库用来存储公钥 PK；密钥管理模块负责对系统主密钥 MK 的保存和更新操作，以及保存和更新加密后的用户解密私钥 e_SK。

2. 客户端

客户端包括密钥生成、数据加解密、格式化三个模块。密钥生成模块为方案中使用的对称加密算法以及 HMAC 算法生成密钥；数据加解密模块负责对物流用户隐私数据进行对称加解密、HMAC 计算以及分层加解密操作；格式化模块负责将隐私数据按照安全等级进行分割并且对加密之后产生的密文数据进行格式化处理，产生密文数据包，最后发送到云存储平台上存储。

3. 云存储平台

客户端对隐私数据加密之后，将生成的密文数据包上传到云存储平台中，同时云存储平台对每一个存储的密文数据包提供一个 URL 供用户下载该密文数据包。

4. 用户

用户可以分为数据拥有方和数据访问方。数据拥有方（即发件人）通过客户端生成一个物流隐私数据，然后将隐私数据加密生成密文数据包，最后将密文数据包上传到云存储平台。数据访问方在物流的过程中代表了所有可能访问隐私数据的用户，数据访问方需要从 TA 请求得到自己对应的 e_SK，将 e_SK 解密得到解密私钥 SK，同时下载密文数据包到本地，进而通过解密私钥 SK 来尝试运行基于属性的分层加密方案的解密步骤得到对称密钥，如果数据访问方的属性集满足数据拥有方设定的访问结构，那么数据访问方就能够成功解密得到对称密钥，不同角色解密得到的对称密钥是不同的，从而解密得到不同部分的隐私数据；如果不满足，则该数据访问方就无权访问该隐私数据。

图 3-2 为本章提出的基于分层加密的隐私数据保护机制的简单工作流程图。首先可信任的授权中心 TA 执行初始化的操作，生成系统公钥 PK 和系统主密钥 MK，TA 将公钥 PK 公开，系统中所有的用户都能够获得，主密钥 MK 由 TA 秘密保存；然后数据拥有方生成一个隐私数据之后，将隐私数据按照三个安全级别分为三部分，并通过对称加密算法加密数据、HMAC 算法保证数据的完整性，然后使用基于属性的分层加密方案加密三个对称密钥以及 HMAC 密钥，格式化生成密文数据包并将其存储到云存储平台中；TA 为数据访问方授予一个属性集，并根据该属性集生成用户的解密私钥 SK；数据访问方从 TA 处请求得到 e_SK，解密得到 SK，然后从云存储平台取回密文数据包，然后尝试解密获得相应部分的物流用户隐私数据。

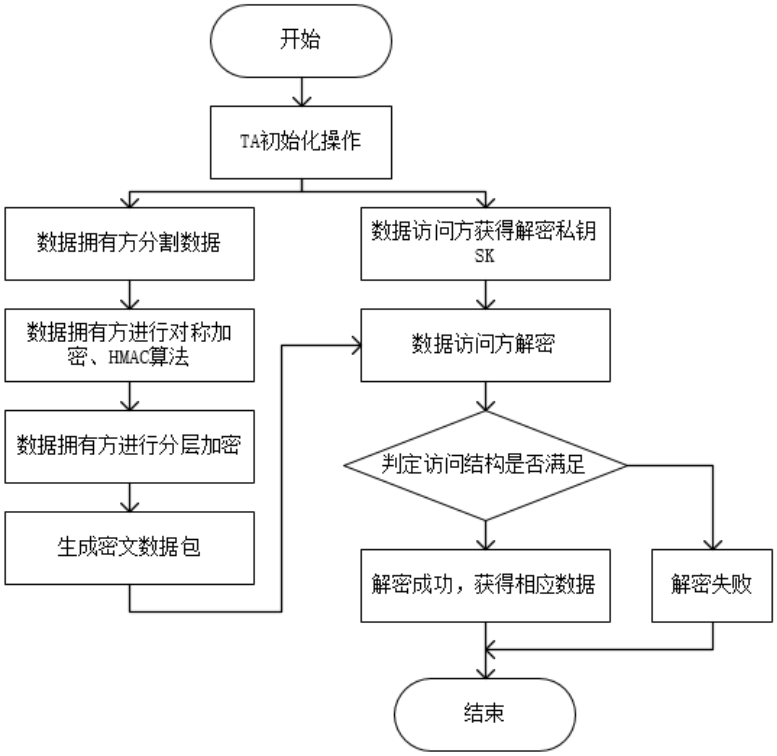


图 3-2 简单工作流程图

3.3 基于属性的分层加密方案

本章提出了一种改进的 CP-ABE 算法实现基于属性的分层加密方案，下面对该分层加密方案进行详细的介绍。基于属性的分层加密方案同样由初始化、密钥生成、加密和解密四个步骤组成。对传统的 CP-ABE 算法的改进之处在于：传统的 CP-ABE 算法对多个数据进行加密时，使用到的多个访问控制树 T 相互没有关联的，本文提出改进的 CP-ABE 算法在加密多个安全等级的隐私数据时采用了一个多层嵌套的访问控制树 T_i，从而加密隐私数据设定的访问结构也是嵌套的，实现了物流过程中不同的角色对隐私数据的分层访问，拥有较高解密权限的角色可以访问其权限以下的物流隐私数据。下面以本文使用基于属性的分层加密方案加密三个不同安全级别数据的对称密钥 key_1、key_2 和 key_3 为例来介绍该分层加密方案的使用。

相关符号及定义如下表 3.2 所示：

表 3.2 第三章使用的符号说明表

符号	说明
M	物流用户隐私数据明文
T	访问控制树
security_level	数据安全等级，设定的可选值为：1/2/3（1 代表安全级别最低，3 代表安全级别最高）
key	对称加密密钥

γ	用户属性集
G_1、G_2	乘法循环群
g	生成元
e	双线性对 $e: G_1 \times G_1 \to G_2$

本文将物流用户隐私数据分成三个不同的安全级别，安全级别由低到高分别为 $security\_level = 1, 2, 3$，对应的隐私数据明文为 m_1、m_2、m_3。在分层加密方案中，分别使用 T_1、T_2、T_3 三棵嵌套的访问控制树来分别加密 m_1、m_2、m_3 对称加密时使用的对称密钥 key_1、key_2 和 key_3。T_1、T_2、T_3 三棵访问控制树具有嵌套关系，具体的关系如下图 3-3 所示。

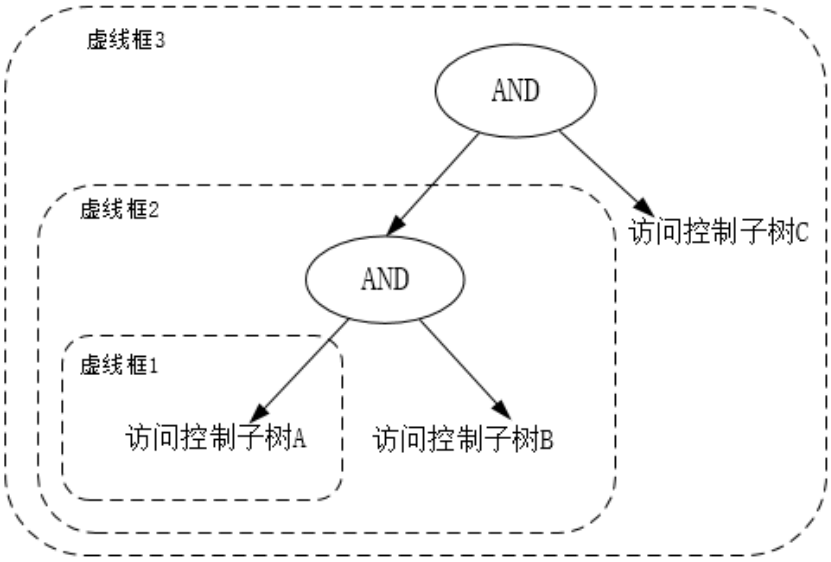


图 3-3 分层加密方案中访问控制树的嵌套结构

图 3-3 中访问控制树的根节点是一个“AND”门限，它的两个子节点分别是一个“AND”门限以及访问控制子树 C。这个左子节点“AND”门限的两个子节点为访问控制子树 A 以及访问控制子树 B。数据拥有方在使用分层加密方案对对称密钥 key_1、key_2 和 key_3 进行加密的时候，选用的 T_1、T_2、T_3 三棵访问控制树分别是：1) 选用虚线框 1 中的访问控制树（即访问控制子树 A）作为 T_1；2) 选用虚线框 2 中的访问控制树作为 T_2；3) 选用虚线框 3 中的访问控制树（即图中整棵访问控制树）作为 T_3。这样选择的具有嵌套关系的访问控制树 T_1、T_2、T_3 就能实现：当一个数据访问方执行解密步骤的时候，如果他的属性集 γ 满足访问控制树 T_1 设定的访问结构，那么他就能够解密得到 key_1；如果他的属性集 γ 满足访问控制树 T_2 设定的访问结构，那么他就能解密得到 key_1 和 key_2；如果他的属性集 γ 满足访问控制树 T_3 设定的访问结构，那么他就能解密得到全部的对称密钥，即 key_1、key_2 和 key_3。

下面将对本章提出的基于属性的分层加密方案四个步骤：初始化、解密私钥生成、加密和解密进行详细的介绍：

1. 初始化(setup)

该步骤由 TA 执行，用来生成一个系统公钥 PK 和系统主密钥 MK。首先，随机选择两个随机数 $\alpha,\beta \in \mathrm{Z_p}$，然后根据公式(3.1)计算出公钥 PK：

$$PK = G_1, g, h = g^{\beta}, e(\mathrm{g},\mathrm{g})^{\alpha} \tag{3.1}$$

然后根据公式(3.2)计算出系统的主密钥 MK：

$$MK = (\beta, g^{\alpha}) \tag{3.2}$$

系统的主密钥 MK 由 TA 安全保存，公钥 PK 对所有的用户公开。

2. 解密私钥生成(skgen)

该步骤由 TA 执行，用来生成数据访问方对应的解密私钥 SK。该步骤需要输入的参数为：系统主密钥 MK 以及数据访问方的属性集 γ，通过结合属性集 γ 与系统的主密钥 MK 生成与该属性集 γ 密切相关的解密私钥 SK。该步骤首先选择一个随机数 $r \in \mathrm{Z_p}$，同时为数据访问方的属性集 γ 中的每个属性选择一个对应的随机数 $r_j \in \mathrm{Z_p}$。然后通过公式(3.3)计算出该数据访问方对应的解密私钥：

$$SK = (D = g^{(\alpha+r)/\beta}, \forall j \in \gamma : D_j = g^{r} \cdot \mathrm{H}(j)^{r_j}, D'_j = g^{r_j}) \tag{3.3}$$

3. 加密(encrypt)

该步骤由数据拥有方执行，数据拥有方的隐私数据 M 被按照安全等级分为了 m_1、m_2 和 m_3，在这里需要对 m_1、m_2 和 m_3 使用的对称密钥 key_1、key_2 和 key_3 分别进行加密。该步骤需要输入的参数为：系统主密钥 MK、对称密钥 key_1、key_2 和 key_3 以及数据拥有方设定的访问控制树 T_1、T_2、T_3（三棵访问控制树需要满足图 3-3 所示的嵌套关系）。

加密步骤首先需要从访问控制树 T_3 的根节点 R_3 开始从上到下的为每一个节点 x 随机产生一个多项式 q_x，并且多项式 q_x 的次数 d_x 需要满足：$d_x \le k_x - 1$（k_x 为该节点的阈值）。然后选择一个随机数 $s_3 \in \mathrm{Z_p}$，对于根节点 R_3，需要满足 $q_{R_3}(0) = s_3$，然后对于多项式 q_{R_3}，随机选择其他的多项式次数 d_{R_3} 来完成多项式的定义。对于其他的节点 x，需要满足 $q_x(0) = q_{parent(x)}(\mathrm{index}(x))$，同样选择其他的多项式次数 d_x 完成该节点多项式的定义。

访问控制树 T_1、T_2 都是 T_3 的子树，它们的根节点也是 T_3 的子节点，记为 R_1 和 R_2，同时 s_1 和 s_2 都由随机数 s_3 计算得到。按照下式(3.4)计算出密文 CT_1、CT_2 和 CT_3，其中公式中的 Y_i 是访问控制树 T_i 的叶子节点的集合，$\mathrm{H}(x)$ 表示一个哈希函数 $\mathrm{H}:\{0,1\}^{*} \to \mathrm{G_0}$，$CT$ 代表最后生成的包含访问结构的密文：

$$CT_i = (T_i, \tilde{C}_i = key_i \cdot \mathrm{e}(\mathrm{g},\mathrm{g})^{\alpha s_i}, \mathrm{C}_i = h^{s_i}, \forall y_i \in Y_i : C_{y_i} = g^{q_{y_i}(0)}, C'_{y_i} = \mathrm{H}(att(y_i))^{q_{y_i}(0)}) \tag{3.4}$$

输出的密文为 $CT = < CT_1, CT_2, CT_3 >$。

4. 解密(decrypt)

该步骤由数据访问方执行，需要输入的参数为：需要解密的密文$CT =< CT_1, CT_2, CT_3 >$以及数据访问方的解密私钥SK，如果解密私钥SK所关联的属性集γ（即数据访问方的属性集）满足数据拥有方加密隐私数据使用的访问控制树T表示的访问结构，则该数据访问方就能成功解密获得相应的对称密钥$key_i (i \in \{1,2,3\})$。

解密步骤的第一步是判断属性集γ是否满足访问控制树T_1设定的访问结构，本文将判定访问结构的递归算法定义为$decryptNode(CT_1, SK, x)$，算法的输入为密文数据$CT_1 = (T_1, \tilde{C_1}, \mathrm{C}_1, \forall y_1 \in Y_1 : \mathrm{C}_{y_1}, \mathrm{C}'_{y_1})$、数据访问方的解密私钥$SK$以及访问控制树$T_1$中的每一个节点$x$。节点$x$可能使叶子节点和非叶子节点，需要进行不同的计算：

- x是叶子节点时

令$j = att(x)$，如果i代表的属性元素包含于数据访问方的属性集γ中，则根据公式(3-5)计算：

$$decryptNode(CT, SK, x) = \frac{e(D_j, C_x)}{e(D'_j, C'_x)} = \frac{e(g^r \cdot \mathrm{H(i)}^{r_j}, g^{q_x(0)})}{e(g^{r_j}, \mathrm{H(i)}^{q_x(0)})} = e(\mathrm{g,g})^{rq_x(0)} \tag{3.5}$$

如果i代表的属性元素不包含于数据访问方的属性集γ中，则令$decryptNode(CT_1, SK, x) = \perp$。

- x是非叶子节点时

当需要计算的x是一个非叶子节点时，需要对该x节点的所有的子节点z进行公式(3.5)的计算，并将结果记为$F_z = decryptNode(CT_1, SK, z)$，令$S_x$为子节点$z$的任意一个大小为$k_x$且$F_z \neq \perp$的子集和，如果不存在这样的集合就说明数据访问方的属性集γ不满足访问结构，返回“$\perp$”，递归运算被终止。

如果存在这样的集合就说明数据访问方的属性集γ满足访问结构，令$j = index(z)$，$S'_x = \{index(z) : z \in S_x\}$，通过公式(3.6)计算$F_x$：

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{j, S'_x}(0)} \\ &= \prod_{z \in S_x} (\mathrm{e(g,g)}^{r \cdot q_z(0)})^{\Delta_{j, S'_x}(0)} \\ &= \prod_{z \in S_x} (\mathrm{e(g,g)}^{r \cdot q_{parent(z)}(index(z))})^{\Delta_{j, S'_x}(0)} \\ &= \prod_{z \in S_x} \mathrm{e(g,g)}^{r \cdot q_x(j) \Delta_{j, S'_x}(0)} \\ &= e(\mathrm{g,g})^{r \cdot q_x(0)} \end{aligned} \tag{3.6}$$

公式(3-6)中的$\Delta_{j,S}(x)$为拉格朗日系数，通过公式(3.7)计算，公式中的S是属于$\mathrm{Z_p}$成员集。

$$\Delta_{j,S}(x) = \prod_{l \in S, l \neq i} \frac{x - l}{j - l}, j \in \mathrm{Z_p} \tag{3.7}$$

上面的第一步是判断数据访问方的属性集γ是否满足加密设定的访问结构，如果不满足，解密步骤到此结束；如果满足的话，上述步骤最终输出的值应该是访问控制树T_1的根节点R_1处计算得到的值，记为$A_1 = decrptNode(CT_1, SK, R_1) = \mathrm{e}(\mathrm{g},\mathrm{g})^{rq_{R_1}(0)} = e(\mathrm{g},\mathrm{g})^{rs_1}$。最后根据下面的公式(3.8)解密出对称密钥$key_1$：

$$\tilde{C}_1 / (\mathrm{e}(C_1, D) / A_1) = \tilde{C}_1 / (\mathrm{e}(h^{s_1}, \mathrm{g}^{(\alpha+\mathrm{r})/\beta}) / \mathrm{e}(\mathrm{g},\mathrm{g})^{rs_1}) = key_1 \tag{3.8}$$

在计算出对称密钥key_1之后，继续执行解密步骤，按照上面计算key_1相同的步骤计算key_2和key_3。在计算key_2的时候可以利用$A_1 = e(\mathrm{g},\mathrm{g})^{rs_1}$这个结果继续执行判定访问结构的递归算法$decryptNode(CT_2, SK, x)$来尝试计算$A_2$。如果数据访问方的属性集$\gamma$不满足$T_2$设定的访问结构则得不到$A_2$并且解密步骤停止；如果属性集$\gamma$满足$T_2$设定的访问结构，则得到$A_2 = decrptNode(CT_2, SK, R_2) = \mathrm{e}(\mathrm{g},\mathrm{g})^{rq_{R_2}(0)} = e(\mathrm{g},\mathrm{g})^{rs_2}$，从而根据公式(3.9)解密出对称密钥$key_2$：

$$\tilde{C}_2 / (\mathrm{e}(C_2, D) / A_2) = \tilde{C}_2 / (\mathrm{e}(h^{s_2}, \mathrm{g}^{(\alpha+\mathrm{r})/\beta}) / \mathrm{e}(\mathrm{g},\mathrm{g})^{rs_2}) = key_2 \tag{3.9}$$

同理，通过得到的A_2继续判断属性集γ是否满足T_3设定的访问结构，不满足则停止解密，满足则得到$A_3 = decrptNode(CT_3, SK, R_3) = \mathrm{e}(\mathrm{g},\mathrm{g})^{rq_{R_3}(0)} = e(\mathrm{g},\mathrm{g})^{rs_3}$，进而根据公式(3.10)解密出对称密钥$key_3$：

$$\tilde{C}_3 / (\mathrm{e}(C_3, D) / A_3) = \tilde{C}_3 / (\mathrm{e}(h^{s_3}, \mathrm{g}^{(\alpha+\mathrm{r})/\beta}) / \mathrm{e}(\mathrm{g},\mathrm{g})^{rs_3}) = key_3 \tag{3.10}$$

3.4 具体实现步骤

基于分层加密的隐私数据保护机制的具体实现步骤主要分为：用户注册、客户端加密以及客户端解密。下面对每个步骤进行详细的介绍。

3.4.1 用户注册

本章提出一种改进的 CP-ABE 算法实现基于属性的分层加密方案，该分层加密方案会为物流过程中的各个角色赋予相关联的属性集，用户属性注册是分层加密方案的基础，每一个用户在进入系统之前都需要根据自身的身份信息、证件等到可信任的授权中心 TA 处注册，TA 返回用户的属性集以及相应的注册结果。详细的过程如图 3-4 所示：

1. 系统初始化。

在开始阶段，TA 需要执行分层加密方案的 setup 步骤，该步骤由 TA 的密钥生成模块执行，生成系统的主密钥*MK*和公钥*PK*，其中系统的主密钥*MK*是秘密保存在密钥管理模块中，*PK*是在整个系统中公开的，提供给系统中所有的用户在加密数据的时候使用，保存在公开数据库模块中。

2. 用户属性注册。

用户通过客户端将自己的身份信息、证件信息以及客户端生成的 RSA 公钥*u_pub*发送给 TA（RSA 私钥*u_pri*保存在客户端，用于之后解密*e_SK*获得解密私钥*SK*，TA 中

的属性授权模块会根据用户发送来的相关信息生成用户的属性集γ。

3. 用户解密私钥 SK 的生成。

TA 的密钥生成模块执行分层加密算法的 skgen 模块，根据用户的属性集γ生成用户的解密私钥SK，然后 TA 使用用户发送来的 RSA 公钥u_pub加密SK生成e_SK，e_SK保存在 TA 的密钥管理模块中，用户可以向 TA 发送请求获得e_SK。

4. 返回用户注册结果。

最后 TA 返回用户的属性集以及注册的结果。

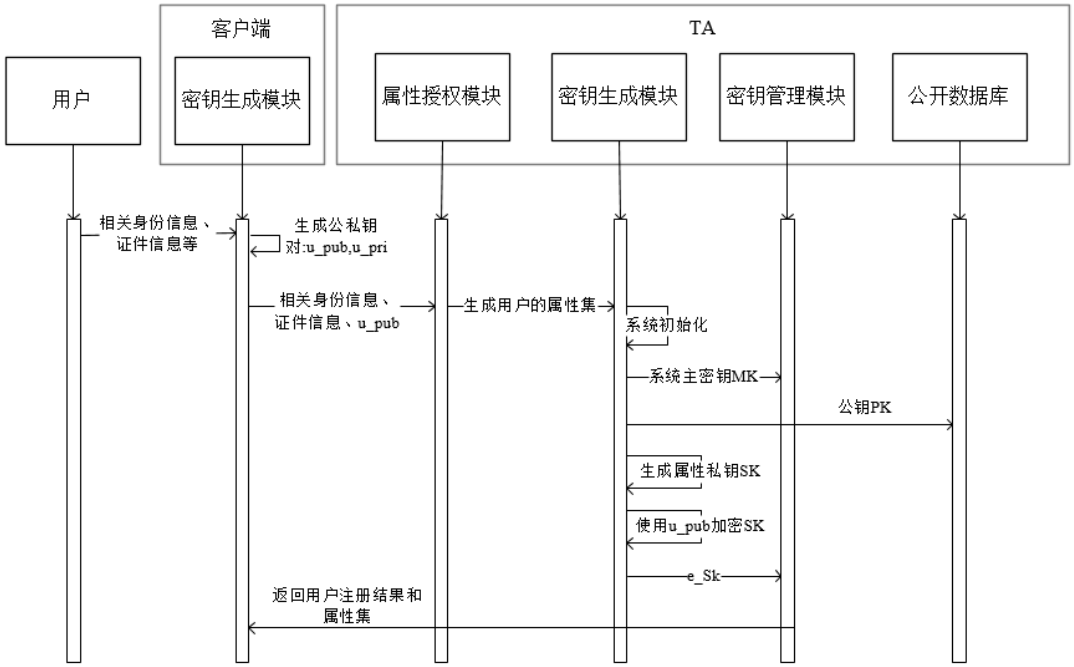


图 3-4 用户注册过程

3.4.2 客户端加密

目前物流用户隐私数据是由第三方（如：物流企业、电商平台等）保存在自己的数据库中，即使第三方宣称不会窃取和泄露用户隐私数据，用户隐私依旧得不到保证。本章提出的基于分层加密的隐私数据保护机制将物流用户隐私数据保存到云存储平台上，但是云存储平台也不是完全可信的，为了保证用户隐私数据的安全，在上传之前对隐私数据进行加密处理是一项必不可少的措施。

在该步骤中，数据拥有方在客户端对隐私数据进行加密处理，加密过程采用的算法是对称加密算法 AES-128、基于属性的分层加密方案以及 HMAC 算法。考虑到分层加密方案加密大数据时效率比较低，因此采用了 Shoup 等人提出的混合加密机制(KEM-DEM，Key Encapsulation Mechanism, and Data Encapsulation Mechanism)[48]。KEM-DEM 机制即先使用对称加密算法对隐私数据明文进行加密，再使用基于属性的分层加密方案对对称加密算法中的对称密钥key进行分层加密，同时使用 HMAC 算法来保证隐私数据密文的完整性。

在该步骤中使用的对称加密算法为 AES-128，记为 AES，该机制将物流用户隐私数

据明文 M 按照安全等级分为了 $security\_level=1,2,3$，三部分，分别为 m_1、m_2、m_3，对应使用的 AES 算法的密钥为 key_1、key_2、key_3，长度均为 128 位。HMAC 算法中嵌入的 Hash 函数为 SHA-1，密钥为 h_key，长度设置为 512 位，产生的消息认证码为 160 位。加密的详细过程如图 3-5 所示，具体描述如下：

1. 物流用户隐私数据分割。为了实现在物流中转过程中，不同的角色根据自己关联的属性集在获得必要的用户隐私数据的同时不会获得多余的隐私数据，本文将物流用户隐私数据 M 分成三个不同的安全级别，分割成的数据为 m_1、m_2、m_3。
2. 对称密钥 key 以及 HMAC 密钥 h_key 的生成。随机选择 $f\in Z_p$，$u\in G_0$，计算 u^f，将结果用二进制序列表示为 $bits=(bits_1,bits_2,bits_3,bits_4)$，其中四个子二进制序列的长度相同，令 $key_1=\mathrm{MD5}(bits_1)$，$key_2=\mathrm{MD5}(bits_2)$，$key_3=\mathrm{MD5}(bits_3)$，$h\_key=H_1(bits_4)$，其中 H_1 为 SHA-512 算法。

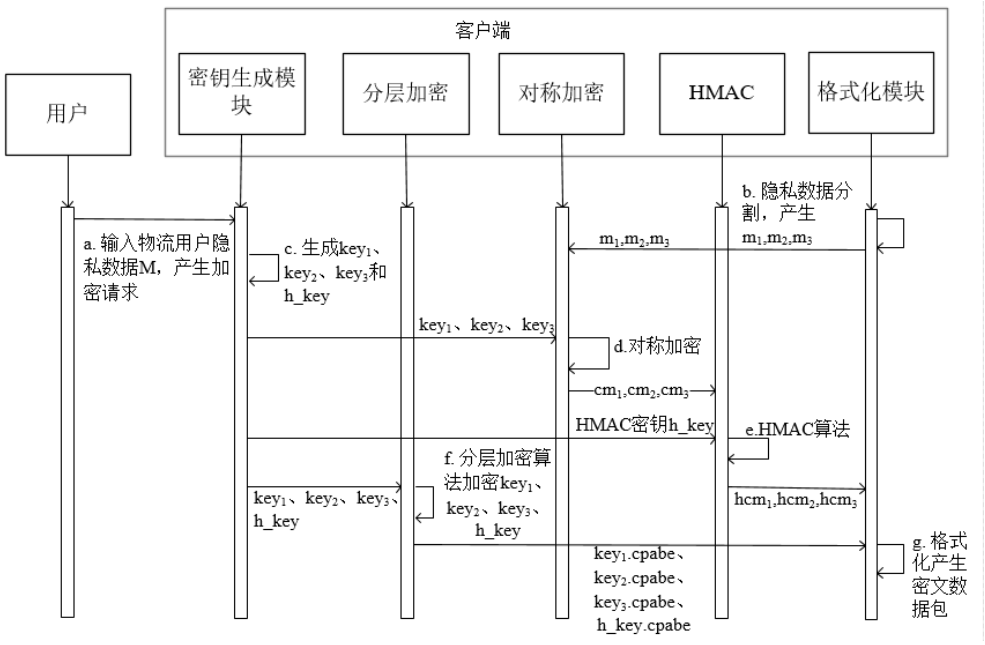


图 3-5 客户端加密过程

3. 物流用户隐私数据加密。在步骤 1 中，用户已将待上传的隐私数据 M 按照安全等级划分为 m_1、m_2、m_3，用户在使用分层加密方案时需要设定一个访问控制树 T，T 的结构需要满足图 3-3 表示的访问控制树嵌套结构，即可以从 T 中选择出三层嵌套的访问控制树 T_1、T_2、T_3。具体的加密的步骤如下：首先进行对称加密，分别使用 AES 算法加密 m_1、m_2、m_3 得到相应的密文 cm_1、cm_2、cm_3，使用的对称密钥分别为 key_1、key_2、key_3；然后使用 HMAC 算法分别求出 cm_1、cm_2、cm_3 的消息认证码 hcm_1、hcm_2、hcm_3；最后使用基于属性的分层加密方案分别对 key_1、key_2、key_3 进行分层加密，使用的访问控制树分别为 T_1、T_2、T_3，生成密钥密文文件分别为 $key_1.cpabe$、$key_2.cpabe$、$key_3.cpabe$，同时使用分层加

密方案对 h_key 进行加密，使用的访问控制树为 T_1，生成密钥密文文件为 $h\_key.cpabe$，这样拥有解密权限的数据访问方都能够获得 HMAC 的密钥，从而验证下载得到的隐私数据密文 cm_1、cm_2、cm_3 是不是完整的。

4. 格式化生成密文数据包。首先对于用户的物流隐私数据 M 生成一个 id，用 $dataid$ 表示，$dataid = \mathrm{MD5}(M)$；然后按照图 3-6 所示的数据包结构将对称密钥的密文文件、HMAC 算法密钥的密文文件以及三个安全等级数据的密文数据存储到云存储平台上。hcm_1、hcm_2、hcm_3 三个长度相同并且为 160 位，占 20 字节，cm_1、cm_2 和 cm_3 表示 $security\_level = 1, 2, 3$ 数据的密文信息，长度不定。

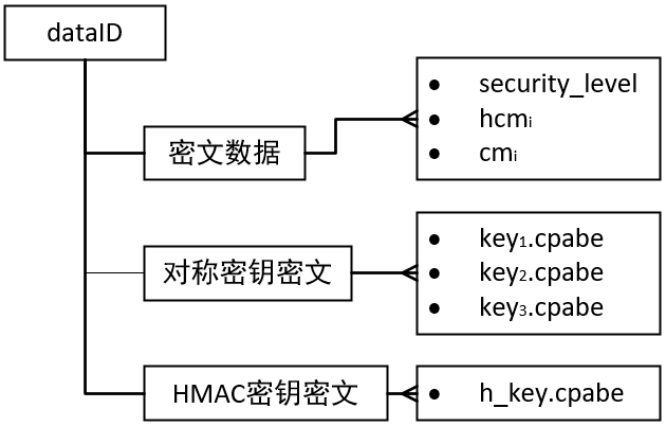


图 3-6 云存储平台存储的密文数据包结构

3.4.3 客户端解密

数据访问方通过客户端从云存储平台上根据 $dataid$ 将对应的隐私数据密文数据包下载到本地，并且从 TA 的密钥管理模块请求获得自己对应的 e_SK，通过客户端保存的 u_pub 解密得到自己的解密私钥 SK；然后使用解密私钥 SK 尝试解密出对称密钥，进而才能解密得到 $dataid$ 相应部分的隐私数据，同时通过解密过程的成功与否可以判断该数据访问方是否拥有对 $dataid$ 对应的物流用户隐私数据的访问权限。详细的解密过程如图 3-7 所示（在这里为了更加清楚的说明解密过程，我们假设数据访问方的属性集满足访问控制树 T_2）。

1. 密文数据包格式化处理。

数据访问方将 $dataid$ 对应的密文数据包去格式化，得到密文数据部分、对称密钥密文部分以及 HMAC 密钥密文部分。

2. 解密获得对称密钥和 HMAC 算法密钥。

通过分层加密方案的 decrypt 步骤，使用数据访问方的解密私钥 SK 分别对对称密钥密文 $key_1.cpabe$、$key_2.cpabe$、$key_3.cpabe$ 和 HMAC 密钥密文 $h\_key.cpabe$ 进行解密操作。因为数据访问方的属性集满足 T_2 设定的访问结构，并且 $h\_key.cpabe$ 是使用访问控制树 T_1 加密产生的，所以数据访问方能够成功解密出 $security\_level$ 为 1 和 2 的物流用户隐私

数据对称加密使用的对称密钥 key_1、key_2 以及 h_key，在这里数据访问方并不能成功的对 $key_3.cpabe$ 执行 decrypt 操作，因为数据访问方的属性集并不满足 T_3 设定的访问结构。

3. 判断隐私数据密文的完整性。

数据加密方通过解密获得的密钥 h_key 对下载得到的 cm_1 和 cm_2 运行 HMAC 算法，计算出 hcm_1' 和 hcm_2'，然后比较 hcm_1' 和 hcm_2' 是否与下载得到的 hcm_1、hcm_2 相同。如果两对都相同的话，就说明隐私数据密文没有被篡改过，是完整的，可以接着进行解密操作；如果两对有一对或者都不相同的话，就说明隐私数据密文已经被篡改了，直接终止解密过程并删除从云存储平台下载到本地的 *dataid* 的密文数据包。

4. 解密获得数据访问方对应部分的物流用户隐私数据。

验证了隐私数据密文的完整性之后，数据访问方通过分层加密方案解密得到的对称密钥 key_1 和 key_2 运行 AES 算法对 *security_level* 为 1 的密文数据 cm_1 以及 *security_level* 为 2 的密文数据 cm_2 进行解密，获得明文数据 m_1 和 m_2。

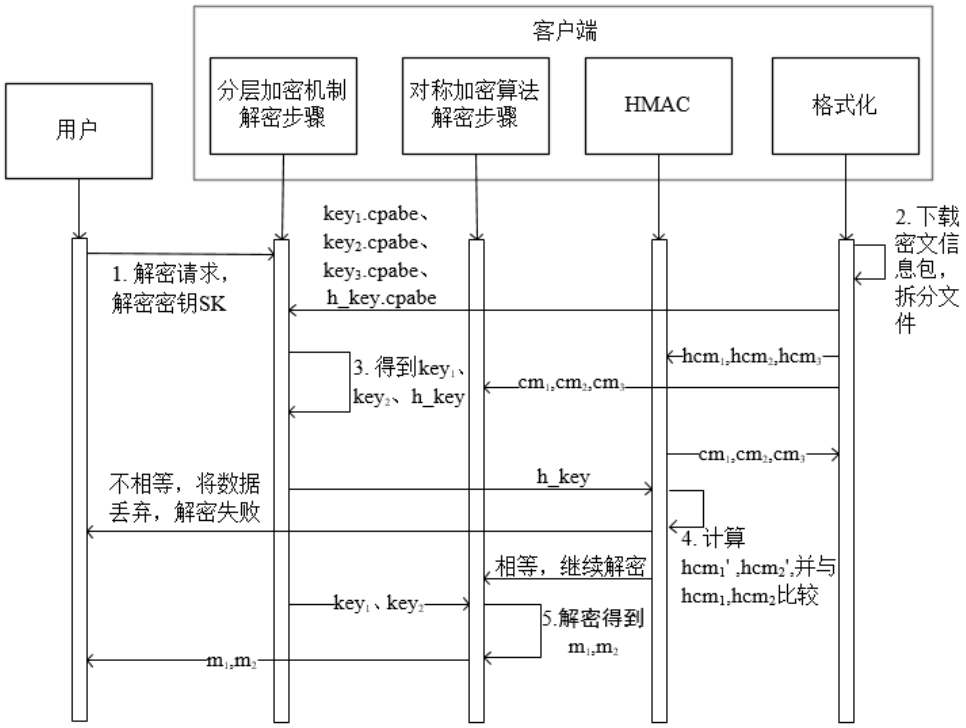


图 3-7 客户端数据解密过程

3.5 机制安全性分析

3.5.1 隐私数据安全性保证

本章提出了一种改进的 CP-ABE 算法实现基于属性的分层加密方案，然后使用该分层加密方案和 AES-128 算法相结合的混合加密机制 KEM-DEM 来实现物流用户隐私数据的加密过程，在保证了物流用户隐私数据安全性的同时，实现了物流过程中不同角色根据自己关联的属性集能够解密获得不同安全级别的物流用户隐私数据。KEM-DEM 加

密模式中，KEM 模块使用的是基于属性的分层加密方案，DEM 模块使用的是 AES-128 对称加密算法。

1. 对称算法安全性

AES 是一个对称分组密码算法，旨在取代 DES 算法成为广泛使用的加密标准。AES 算法加密的数据块长度必须为 128 比特即 16 字节，密钥长度可以为 16、24、32 字节（即 128、192 和 256 比特）。根据密钥的长度，算法分为 AES-128、AES-192、AES-256。AES 算法比 DES 算法具有更高的安全性，能够有效的抵抗差分攻击、线性攻击、穷举攻击、内插攻击和平方攻击等多种已知的攻击方法[49]。因此使用 AES 算法能够保证物流用户隐私数据的安全性。

2. 基于属性的分层加密方案的安全性

本章使用改进的 CP-ABE 算法实现了基于属性的分层加密方案来保证不同安全等级物流隐私数据对称密钥的安全性。该基于属性的分层加密方案在加密单个对称密钥的时候可以看作传统的 CP-ABE 算法，CP-ABE 算法在 DBDH 模型下被证明是 IND-sAtt-CPA 安全的，能够达到选择明文安全级别（简称 CPA 安全），符合公钥加密机制的基本的安全要求[50]，因此在加密单个对称密钥的时候能够保证对称密钥的安全性；分层加密方案在加密多个对称密钥时，当一个数据访问方仅能解密出 key_1，而他想要解密得到 key_2、key_3 时，需要得到加密步骤使用的访问控制树 T_2、T_3 以及缺少的属性，想要获得这些缺少的数据是不可能，因为用户的属性集是由 TA 根据用户的身份授予，加密使用的访问控制树包含在密文当中。所以可以认为基于属性的分层加密方案能够保证对称密钥的安全性。

3. KEM-DEM 安全性

本方案采用的 KEM-DEM 加密模式中，KEM 模块使用的是改进的 CP-ABE 算法实现的基于属性的分层加密方案，DEM 模块使用的是 AES-128 对称加密算法。由于 KEM-DEM 加密模式的安全性主要取决于 KEM 模块，前面已经分析了该分层加密方案的安全性，因而本章提出的基于分层加密的隐私数据保护机制够达到 IND-CPA 级别。

3.5.2 隐私数据完整性保证

本方案采用 HMAC 算法保证物流用户隐私数据的完整性。HMAC 算法是一种基于密钥和 Hash 函数来进行消息认证的方法，因此 HMAC 算法能够保证数据在云存储平台以及传输的过程中不被恶意的修改或攻击。HMAC 算法的安全性主要由选用的 Hash 函数以及密钥的安全性决定[51]。本文中使用的 HMAC 算法选用的 Hash 函数是 SHA-1，拥有较高的安全性。HMAC 算法的密钥是通过基于随机种子的伪随机生成方法生成的，长度为 512 位。本方案采用的 HMAC 算法有很高的安全性。

3.5.3 性能分析

本章提出的基于属性的分层加密方案是通过改进的 CP-ABE 算法实现的，该分层加密方案在加密单个数据时，可以看作一个传统的 CP-ABE 算法，但是在加密多个数据时，

分层加密方案在解密的步骤中拥有比传统的 CP-ABE 算法更高的效率。因为在解密的步骤中，第一阶段对 $security\_level=1$ 的数据使用的解密算法仍然类似于传统的 CP-ABE 算法中的解密算法，但是当在第二阶段对 $security\_level>1$ 的数据解密时将使用第一阶段解密步骤中计算得到的 $A_1=decrptNode(CT_1,SK,R_1)=e(g,g)^{rq_{R_1}(0)}=e(g,g)^{rs_1}$ 来计算 A_2、A_3，这将减少很多的双线性运算，从而提高了解密的效率。

3.6 本章小结

本章针对现有方案中第三方保管物流用户隐私数据，依旧存在隐私泄露的问题，提出了一种基于分层加密的隐私数据保护机制。在该机制中，隐私数据由用户分层加密之后上传到云端，杜绝了非完全可信第三方存储隐私数据存在隐私泄露的问题。该机制首先将用户隐私数据划分为不同的安全等级；然后通过 AES 算法加密不同安全等级的隐私数据，HMAC 保证隐私数据密文的完整性，同时该机制使用一种由改进的 CP-ABE 算法实现的基于属性的分层加密方案加密对称密钥和 HMAC 算法使用的密钥，实现物流过程中不同角色拥有不同解密权限，能够解密获得不同部分的物流用户隐私数据；最后通过云存储平台安全来提供隐私数据密文的安全存储和访问。该机制保证了物流用户隐私数据的安全性和完整性，确保物流过程中各角色不获得多余部分用户的隐私数据，同时基于属性的分层加密方案在提供分层访问的同时，还拥有更高的解密效率。

第四章 基于区块链和 DAA 匿名认证的访问权限管理机制

4.1 引言

物流过程中涉及到的角色有：发件人、收件员、物流中转人员、快递员，表 3.1 介绍了在物流过程中各角色的业务操作，以及为了顺利完成相应的业务操作必须获得的物流用户隐私数据。按照第三章提出的基于分层加密的隐私数据保护机制，各角色通过与自身相关联的属性集获得一个解密私钥 *SK*，然后通过该解密私钥 *SK* 解密得到角色对应部分的物流用户隐私数据，从而帮助完成自己的业务操作。

第三章提出的机制能够保证物流用户隐私数据的安全性、完整性以及实现物流中各角色对隐私数据的分层访问，但是该机制存在着以下的不足：1. 发件人作为物流隐私数据的拥有方缺乏对隐私数据的访问权限控制管理的能力；2. 在整个物流过程中，发件人并不知道自己的隐私数据在什么时候被哪一个数据访问方使用，整个过程中缺少对物流用户隐私数据的使用记录。

针对以上存在的不足，结合区块链公开透明、无法篡改、方便追溯等特点，本章在许可链基础上提出了基于区块链和 DAA 匿名认证的访问权限管理机制，该机制提供的访问权限管理主要包括两方面：

1. 对区块链节点的访问权限管理。基于 DAA 匿名认证提出了一种成员身份管理机制，该机制为区块链上的实体提供匿名可验证的身份，通过维护的一个交易公钥列表（TPL，Transaction Public Key List）来实现对区块链节点的访问权限管理，只有当用户将自己的交易公钥注册进 TPL 之后，该用户才能生成区块链网络上有效的交易，杜绝了当前区块链技术存在的隐私泄露的风险；

2. 数据拥有方对隐私数据的访问权限管理。数据拥有方根据自己的意愿在分布式账本中为每个物流用户隐私数据生成一个对应的数据权限状态，该数据权限状态决定了每个数据访问方的访问权限，数据访问方必须通过应用程序构造一个交易访问该分布式账本来判断自己是否有权访问对应的物流用户隐私数据，如果有权访问，就能通过分布式账本获得对应的 e_SK 存储路径以及隐私数据密文对应的云存储路径，进而解密获得相应部分的隐私数据，同时在区块链上记录下数据访问方访问该物流用户隐私数据的记录。

下面先简单介绍该访问权限管理机制中涉及到的许可链，然后再给出该机制具体的介绍。

4.2 许可链

本章提出的基于区块链和 DAA 匿名认证的访问权限管理机制中使用到的区块链属于许可链，在 2.2.3 节中已经简单介绍了许可链是一种不对外公开的区块链，并且该区块链网络中的节点都是要经过注册才能参与到区块链网络的交易中。超级账本

Hyperledger 是由 Linux 基金会在 2015 年发起的推进区块链数字技术和交易验证的开源项目，Fabric 是其中的一个子项目，Hyperledger Fabric 的架构属于许可链。本章提出的访问权限管理机制主要通过 Fabric 架构中的账本和链码（chaincode）实现了数据拥有方根据自己的意愿对物流隐私数据进行访问权限的控制管理。数据拥有方和数据访问方都是区块链网络中的成员，数据拥有方在生成一个新的物流隐私数据时，在分布式账本的 State 数据库添加一个对应的以键值对（key-value）形式表示的数据权限状态，该数据权限状态中包含了物流隐私数据的基本信息（数据拥有方编号、数据存储路径、上传时间）以及数据访问方的权限数据集，具体的将在下面章节中介绍。下面简单介绍下 Hyperledger Fabric。

1. Fabric 整体架构

超级账本 Fabric 的整体架构如图 4-1 所示。

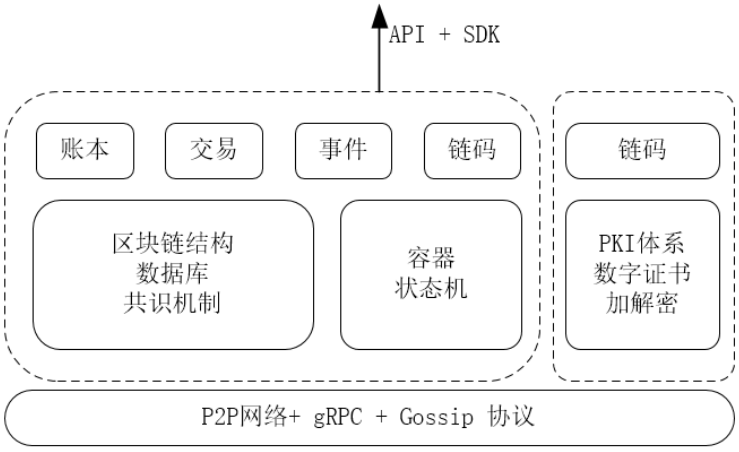


图 4-1 Hyperledger Fabric 架构图

Fabric 为应用提供了 gRPC API，以及封装 API 的 SDK 供应用调用。应用可以通过 SDK 访问 Fabric 网络中的多种资源，包括分布式账本、交易、链码（chaincode，也称智能合约）、事件和权限管理等。应用开发者只需要跟这些资源打交道即可，无需关心底层是如何实现的。其中，分布式账本是最核心的结构，负责记录应用信息，应用通过发起一个交易来向账本中记录、更改数据。链码封装了交易执行的逻辑。整个 Fabric 网络运行中发生的时间都可以被应用访问。权限管理负责了整个过程中的访问控制。

底层则是由多个节点组成 P2P 网络，节点都是经过认证的，各节点之间通过 gRPC 通道进行交互，利用 Gossip 协议进行同步。

2. 核心概念

1) 交易

交易意味着通过调用链码实现对账本状态进行一次改变。客户端可以通过发送交易请求来让分布式账本记录信息。通常来说，当一个合法的交易请求构造出来之后会发给 Fabric 网络中的排序节点进行排序，生成一个区块，然后广播到 Fabric 网络中的各个节点进行确认。如果节点对区块中的交易验证通过的话，则接受该交易指定的账本状态的

变更，最终将该区块加入到区块链中，并更新本地账本。

2) 区块

区块意味着一组进行排序后的交易的合集。区块链以区块为单位对多个交易的历史进行链接，通过调整区块大小可以在吞吐性能和确认时间之间进行平衡。典型的，区块结构包括区块头（Header）、数据（Data）和元数据（Metadata）三部分结构。区块头用于构建区块链结构，包括 Number、PreviousHash、DataHash 等三个值。Number 表示区块的序号；PreviousHash 表示前一个区块的头部域的 Hash 值；DataHash 则为本区块 Data 域内容 Hash 值。Data 域记录了区块内的多个交易信息，这些交易采用了单层的 Merkle 树结构。Metadata 域中则记录了一些辅助信息。

图 4-2 表示 Fabric 的区块链结构。区块链是由各个区块依次串联而成的线性结构。

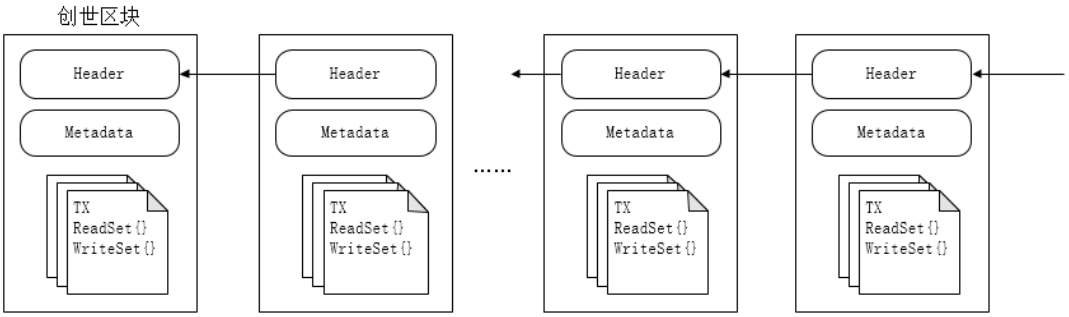


图 4-2 链结构

3) 链码

链码直接与分布式账本结构打交道，一般封装了与分布式账本直接交互的相关过程。区块链网络中的成员商定好业务逻辑之后，可以将业务逻辑编程到链码中，之后的交易都遵循链码执行。链码会对 Fabric 应用程序发送的交易做出响应，执行链码中相应的代码逻辑，与分布式账本进行交互，即可对分布式账本中的状态进行更新操作。

链码会创建一些状态（state）并写入到分布式账本中。状态带有绑定到链码的命名空间，仅限于创建它的链码使用，不能被其他链码直接访问。不过，在合适的许可范围下，一个链码也可以调用另一个链码，间接访问其状态。

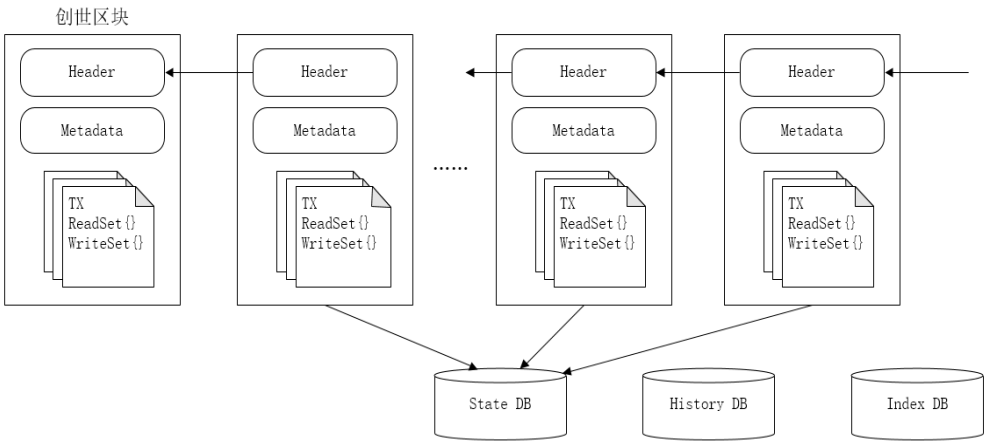


图 4-3 分布式账本结构

4) 分布式账本

分布式账本主要负责记录发生在网络中的交易信息。Fabric 应用开发人员通过编写和执行链码发起交易，实际上是对账本中记录的状态进行更改。

一个典型的分布式账本结构如图 4-3 所示。

从结构上看，分布式账本包括了区块链结构，以及多个数据库结构。

1) State Database：状态数据库，由区块链结构中交易执行推演而成，记录最新的数据状态。
2) History Database：历史数据库，存放各个状态的历史变化记录。
3) Index Database：索引数据库，存放索引信息，例如从 Hash、编号索引到区块，从 ID 索引到交易等。

从数据库角度看，区块链结构记录的是使状态变更的历史，State 数据库记录的是状态变更的最终结果。每一次对账本状态的变更通过交易导致的读写集合表达。

4.3 整体架构

本章提出的基于区块链和 DAA 匿名认证的访问权限管理机制整体架构如图 4-4 所示。该机制由用户模块、成员身份管理模块以及区块链模块组成。各部分详细的介绍如下：

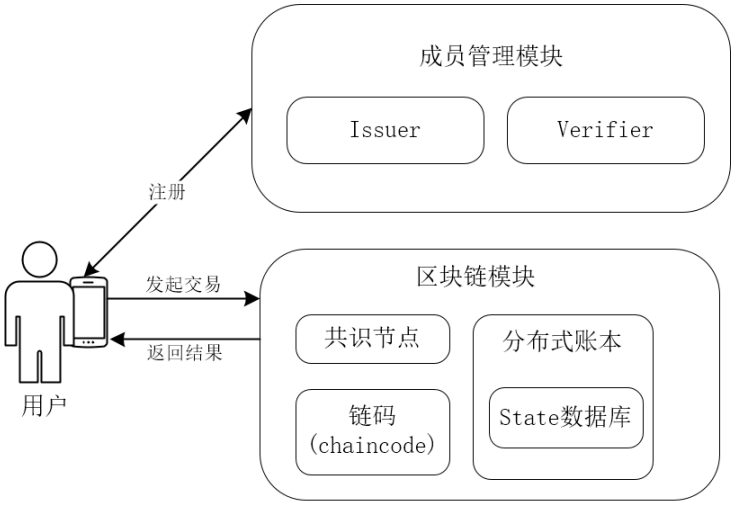


图 4-4 基于区块链和 DAA 匿名认证的访问权限管理机制整体架构

1. 用户模块

用户模块由不同类别的用户组成，例如有：研究人员、商业人员、政府机构以及物流使用方等类别的用户。在物流场景中，用户包括了物流过程中涉及到的各种角色，有发件人、收件员、物流中转人员、快递员。发件人作为物流隐私数据的拥有方可以创建物流隐私数据、访问自己或别人的隐私数据以及对自己的隐私数据进行访问权限的控制管理；物流过程中的其他角色只能够访问发件人的物流隐私数据，并不能对发件人的物流隐私数据做任何的修改操作。

在本方案中，将各种类别的用户按照物流隐私数据的所属权分为了数据拥有方以及

数据访问方。数据拥有方在生成一个新的隐私数据时，会主动通过客户端应用程序发起一个交易，在分布式账本的 State 数据库中生成对应物流隐私数据的数据权限状态。数据拥有方通过将数据访问方的加密后的解密私钥 *e_SK* 存储路径以及其他的访问限制包含在该数据权限状态中，达到对该物流隐私数据访问权限控制管理的目的。数据访问方如果想要访问数据拥有方的物流隐私数据，则必须通过客户端应用程序发起一个交易从分布式账本的 State 数据库中读取该隐私数据对应的数据权限状态，如果数据拥有方有该隐私数据的访问权限，则他能够读取到自己的 *e_SK* 存储路径以及隐私数据密文的存储路径，进而成功执行 3.4.3 节的客户端解密操作获得相应部分的物流用户隐私数据。

2. 成员身份管理模块

用户模块中的数据拥有方和数据访问方都是区块链网络中的实体节点，都必须通过成员身份管理模块验证身份之后才能加入到区块链网络中。成员身份管理模块基于 DAA 匿名认证方案提出了一种成员身份管理机制，该机制为区块链上的实体提供了匿名可验证的身份，同时通过建立的交易公钥列表 TPL 来实现对区块链节点的访问权限控制。

在该机制中主要参与者是证书发布方 Issuer、用户（数据拥有方和数据访问方）、验证方（Verifier）。当一个用户希望加入到区块链网络中时，Issuer 会根据用户的身份信息判断用户是否有资格加入到区块链网络中，如果判断用户有资格，则 Issuer 会为用户颁发一个身份证书，用户使用自己的身份证书向 Verifier 证明自己是合法的用户，然后在 Verifier 处将自己的交易公钥注册进 Verifier 维护的交易公钥列表 TPL，区块链模块中的共识节点在检查一个交易是否合法时，会判断该交易对应的交易公钥是否存在于 TPL 中，只有 TPL 中存在对应的交易公钥，该交易才能被共识节点接受，进而加入到区块链结构中，更改或读取 State 数据库中的数据。

3. 区块链模块

区块链模块是基于 Hyperledger Fabric 建立的。模块由共识节点、分布式账本、链码组成。共识节点负责验证一批未确认的交易的发生顺序、合法性以及对分布式账本状态的更新结果达成一致的观点，共识节点会根据成员管理模块中的 Verifier 维护的 TPL 来实现对区块链节点的访问权限控制；分布式账本中记录了区块链网络中的所有交易的信息，其中 State 数据库中保存了最新的账本数据状态；链码封装了数据拥有方、数据访问方与物流用户隐私数据之间存在的业务逻辑，链码会在 State 数据库中添加新的状态信息或者更改已经存在的状态信息，数据拥有方在生成一个新的物流隐私数据时，会通过链码在 State 数据库中添加一个对应的数据权限状态。当数据访问方想要访问数据拥有方的一个物流隐私数据时，他必须通过客户端程序构造一个交易，调用链码访问 State 数据库中隐私数据对应的数据权限状态，进而判断自己是否具有该隐私数据的访问权限。

图 4-5 通过数据访问方访问数据拥有方的 *dataid* 标识的隐私数据为例介绍该访问权限管理机制的简单工作流程。

1. 首先数据访问方需要加入到区块链网络中，成为网络中的一个节点。数据访问方将自己的身份信息发送给 Issuer，Issuer 验证数据访问方身份，为其颁发身份

证书；

2. 数据访问方通过将自己的身份证书发送给 Verifier 来验证自己是区块链网络中合法的节点。在验证之后，数据访问方生成一对交易公私钥（*tran_pub*，*tran_pri*），并将交易公钥*tran_pub*传送给 Verifier，Verifier 将*tran_pub*添加到交易公钥列表 TPL 中。
3. 数据访问方访问 *dataid* 标识的物流用户隐私数据，通过客户端应用程序构造交易，尝试着从分布式账本中获取自己*e_SK* 的存储路径，该交易使用*tran_pri* 进行签名。
4. 区块链网络中的共识节点检查交易的合法性，保证交易对应的*tran_pub*存在于 TPL 中，如果交易对应的*tran_pub*不在 TPL 中，说明发起交易的节点不是区块链网络中的合法节点，则直接丢弃该交易。
5. 合法的交易通过调用链码中相关的代码从 State 数据库中访问 key 为 *dataid* 对应的数据权限状态，从中取得对应数据访问方的*e_SK* 以及隐私数据密文存储路径。如果成功取得*e_SK* 存储路径，则说明该数据访问方对 *dataid* 标识的物流隐私数据具有访问权限，否则没有访问权限。
6. 当取得了*e_SK* 和密文数据包的存储路径之后，客户端会下载*e_SK* 和密文数据包到本地并进行 3.4.3 节中的解密操作，最后获得相应部分的物流隐私数据。

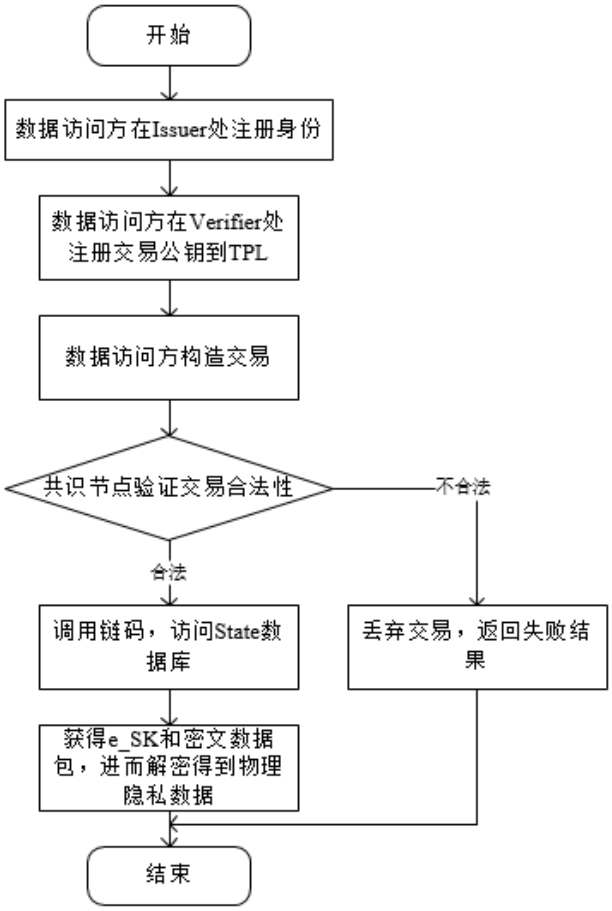


图 4-5 基于区块链和 DAA 匿名认证的访问权限管理机制工作流程图

4.4 成员身份管理模块

成员身份管理模块根据 DAA 匿名认证的方案为想要加入区块链网络的用户提供匿名可验证的身份。成员身份管理模块中使用到的参数及长度见表 4.1。

表 4.1 成员身份管理模块中使用的参数及其长度

安全参数	解释	长度(bit)
l_n	RSA 模数 n 的长度	2048
l_f	TPM 密钥 ID 的长度	104
l_e	指数 e 的长度	368
l_v	随机数 v 的长度	2536
$l_\varnothing$	零知识证明协议的安全长度	80
l_{H}	Hash 函数的输出长度	160
l_Γ	Γ 的长度	1632
l_ρ	ρ 的长度	208
l_r	一个与 Γ，ρ 相关的安全系数	80

DAA 方案中用户是由 TPM 和相应的 host 组成。用户首先在 Issuer 获得身份证书，然后通过该证书向 Verifier 证明自己是经过 Issuer 认证过的合法的用户，最后用户将自己用于对区块链网络中交易签名的交易公钥 $tran\_pub$ 注册进 Verifier 维护的交易公钥列表 TPL 中。在整个过程中，Verifier 不会接触到任何关于用户的身份信息，用户对 Verifier 具有匿名性，所以 Verifier 并不知道 TPL 中的各个交易公钥 $tran\_pub$ 对应的真实用户是谁。并且一个用户可以通过多次与 Verifier 进行匿名认证过程，注册多个 $tran\_pub$ 进 TPL 中，这些 $tran\_pub$ 之间并没有任何的关联性，这能够解决区块链技术存在的隐私泄露的风险，即攻击者不可能根据通过区块链中交易中存在的关联关系分析出区块链交易地址与节点身份的关联。图 4-6 是成员身份管理模块的工作流程图。

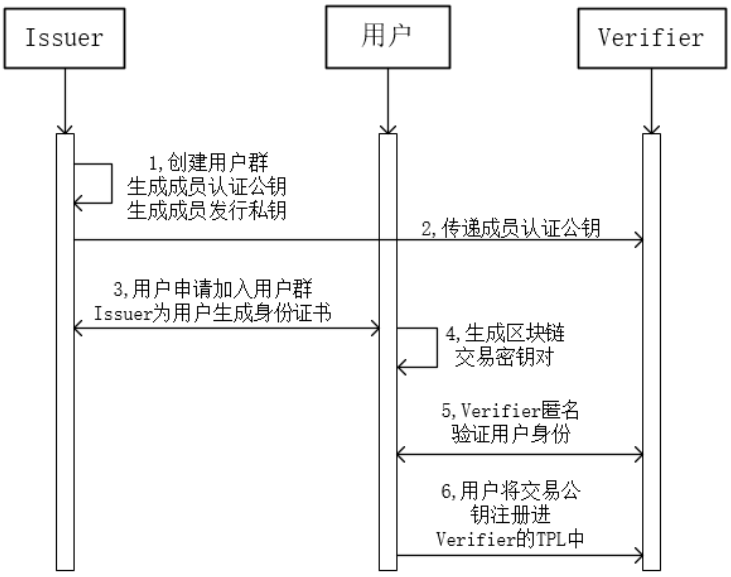


图 4-6 成员身份管理模块工作流程图

1. Issuer 创建一个用户群。这个群中的成员也是区块链网络中的实体节点，该步骤会生成成员认证公钥 K_{PG} 和成员发行私钥 K_{MIPK}。具体的步骤如下：
   a) Issuer 选取一个 RSA 模数 $n=pq$，其中有 $p=2p'+1$，$q=2q'+1$，p、q、p' 和 q' 都是素数，并且 p 和 q 是长度相同的素数，n 的长度是 l_n。
   b) 然后 Issuer 选择 QR_N 的一个随机生成员 g'。
   c) Issuer 选择随机的整数：$x_0, x_1, x_z, x_s, x_h, x_g \in [1, p'q']$，并利用下列随机整数计算：
   $$g := g'^{x_g} \bmod n,\quad h := g'^{x_h} \bmod n,\quad S := h^{x_s} \bmod n$$
   $$Z := h^{x_z} \bmod n,\quad R_0 := S^{x_0} \bmod n,\quad R_1 := S^{x_1} \bmod n$$
   d) 提供一个非交互式的零知识证明发布方产生的 g、h、S、Z、R_0 以及 R_1 都是正确计算出来的。具体的证明方法可以参考[44]。
   e) Issuer 选择长度为 l_Γ 的素数 Γ 和长度为 l_ρ 的素数 ρ，Γ 和 ρ 满足 $\Gamma = r\rho + 1$。随机选择一个数 $\gamma' \in_R Z_\Gamma^*$ 使得 $\gamma'^{(\Gamma-1)/\rho} \neq 1 \bmod \Gamma$，记 $\gamma := \gamma'^{(\Gamma-1)/\rho} \bmod \Gamma$。
   f) 最后 Issuer 将 $(n, g', g, h, S, Z, R_0, R, \gamma, \Gamma, \rho)$ 作为成员认证公钥 K_{PG}，并且将 $p'q'$ 作为成员发行私钥 K_{MIPK} 秘密的保存起来。
2. Issuer 将成员认证公钥 K_{PG} 发送给验证方（Verifier）。Verifier 将在后续的步骤中使用 K_{PG} 对匿名用户的身份进行认证。
3. 用户加入 Issuer 创建的用户群。Issuer 创建的群中的成员也是区块链网络中的实体节点，用户为了加入区块链网络，必须在 Issuer 处注册获得身份证书。为了加入 Issuer 创建的用户群，用户首先发送一个请求给 Issuer，请求中包含了用户的一些身份信息，如果 Issuer 验证用户信息之后确定用户可以加入到用户群中，才会执行下面的步骤为用户颁发身份证书。
   a) 首先用户中 host 计算 $\varsigma_1 = (\mathrm{H}(1 \| bsn_1))^{(\Gamma-1)/\rho} \bmod \Gamma$（其中 bsn_1 为 Issuer 的基名），用户中 TPM 模块检查是否 $\varsigma_1^{\rho} = 1 \bmod \Gamma$，记 Issuer 对 K_{PG} 签名使用的公钥是 K'_{PG}，根据公式 $f = \mathrm{H}(\mathrm{H}(DAAseed \| \mathrm{H}(PK_1')) \| cnt \| 1) \bmod \rho$ 计算得到 f，使 $f_0 = \mathrm{LSB}_{l_f}(f)$，$f_1 = \mathrm{CAR}_{l_f}(f)$，选择随机数 $v_1' \in_R \{0,1\}^{l_s}$ 和 $v_2' \in_R \{0,1\}^{l_n + l_\varnothing - l_s}$ 计算 $U = R_0^{f_0} R_1^{f_1} S^{v_1'} S^{rv_2'} \bmod n$，$N_I = \varsigma_I^{f_0 + f_1 2^{l_f}} \bmod \Gamma$，将 (U, N_I) 发送给 host，host 转发给 Issuer。
   b) 用户 TPM 模块通过零知识证明协议向 Issuer 证明用户拥有 f_0、f_1 和 v'（其中 $v' = v_1' + 2^{l_s} v_2'$），协议具体的步骤如下所示：
      i. TPM 模块选择随机整数 $r_{f_0}, r_{f_1} \in_R \{0,1\}^{l_f + l_\varnothing + l_\mathrm{H}}$，$r_{v_1'} \in_R \{0,1\}^{l_s}$ 和 $r_{v_2'} \in_R \{0,1\}^{l_n + 2l_\varnothing + l_\mathrm{H} - l_s}$，根据公式 $\tilde{U} = R_0^{rf_0} R_1^{rf_1} S^{r_{v_1'}} S'^{r_{v_2'}} \bmod n$，$\tilde{N}_I = \varsigma_I^{r_{f0} + r_{f1} 2^{l_f}} \bmod \Gamma$ 计算得到 $\tilde{U}$ 和 $\tilde{N}_I$，将 $\tilde{U}$ 和 $\tilde{N}_I$ 发送给 host。

ii. Issuer 选择一个随机的比特串 $n_i \in \{0,1\}^{l_H}$，并将 n_i 发送给 host。

iii. host 将通过公式 $c_h = \mathrm{H}(n \| R_0 \| R_1 \| S \| U \| N_I \| \tilde{U} \| \tilde{N}_I \| n_i)$ 计算得到的 c_h 发送给 TPM。

iv. TPM 选择一个随机的比特串 $n_t \in \{0,1\}^{l_\varnothing}$，根据公式 $c = \mathrm{H}(c_h \| n_t)$ 计算得到 c，然后再根据公式 $s_{f_0} = r_{f_0} + cf_0$，$s_{f_0} = r_{f_0} + cf_0$，$s_{v_1'} = \mathrm{LSB}_{l_s}(r_{v_1'} + cv_1')$，$s_{v_1'}' = \mathrm{CAR}_{l_s}(r_{v_1'} + cv_1')$，$s_{v_2'} = (r_{v_2'} + cv_2' + s_{v_1'}')$ 计算得到 $(s_{f_0}, s_{f_1}, s_{v_1'}', s_{v_1'}, s_{v_2'})$，然后 TPM 将 $(c, n_t, s_{f_0}, s_{f_1}, s_{v_1'}, s_{v_2'})$ 发送给 host。

v. host 收到 TPM 传来的 $(c, n_t, s_{f_0}, s_{f_1}, s_{v_1'}, s_{v_2'})$ 之后，计算 $s_{v'} = s_{v_1'} + 2^{l_s} s_{v_2'}$，然后将 $(c, n_t, s_{f_0}, s_{f_1}, s_{v'})$ 发送给 Issuer。

vi. Issuer 通过公式 $\hat{U} = U^{-c} R_0^{sf_0} R_1^{sf_1} S^{s_{v'}} \bmod n$ 和 $\hat{N}_I = N_I^{-c} \varsigma_I^{sf_0 + 2^{l_f} s_{f_1}} \bmod \Gamma$ 计算得到 $\hat{U}$ 和 $\hat{N}_I$，然后验证 $c \stackrel{?}{=} \mathrm{H}(\mathrm{H}(n \| R_0 \| R_1 \| S \| U \| N_I \| \hat{U} \| \hat{N}_I \| n_i) \| n_t)$、$s_{f_0}, s_{f_1} \stackrel{?}{\in} \{0,1\}^{l_f + l_\varnothing + l_H + 1}$ 以及 $s_{v'} \stackrel{?}{\in} \{0,1\}^{l_n + 2l_\varnothing + l_H + 1}$。

c) Issuer 选择一个随机数 $\hat{v} \in_R \{0,1\}^{l_v - 1}$ 和一个素数 $e \in_R [2^{l_e - 1}, 2^{l_e - 1} + 2^{l_e' - 1}]$，然后根据公式 $v'' = \hat{v} + 2^{l_v - 1}$ 和 $A = (\frac{Z}{US^{v'}})^{1/e} \bmod n$ 计算得到 v'' 和 A。然后将 (A, e, v'') 发送给 host，(A, e, v'') 就是 Issuer 对用户的身份证书。

d) host 收到 (A, e, v'') 后，计算 $v_1'' = LSB_{l_s}(v'')$ 和 $v_2'' = CAR_{l_s}(v'')$，然后将 v_1'' 和 v_2'' 发送给 TPM。

e) TPM 收到 v_1'' 和 v_2'' 之后，令 $v_1 = LSB_{l_s}(v_1'' + v_1')$，$\bar{v}_2 = CAR_{l_s}(v_1'' + v_1')$，$v_2 = v_2'' + v_2' + \bar{v}_2$，最后将 (f_0, f_1, v_1, v_2) 秘密保存起来。

4. 用户获得 DAA 证书后，即加入到了区块链网络中，然后用户生成用户的区块链交易密钥对 $(tran\_pub, tran\_pri)$。

5. 用户向 Verifier 匿名证明自己的成员身份。用户在上面的步骤中已经加入到了 Issuer 创建的群中，即加入到区块链网络中，成为区块链网络中的一个节点。接下来用户需要向 Verifier 证明自己是群中合法的成员，并且在认证过程中对 Verifier 保持匿名。具体的步骤如下：

a) 首先用户向 Verifier 发起一个匿名认证的请求。

b) Verifier 收到用户发来的匿名认证请求之后，返回一个响应给用户，响应中包含了 Verifier 的基名 bsn_V 和一个消息 m。

c) 用户使用第四步获得的身份证书 (A, e, v'') 对消息 m 生成知识签名 σ。具体的步骤如下：

i. host 根据 Verifier 提供的基名 bsn_V，根据公式 $\varsigma = (\mathrm{H}_\Gamma(1 \| bsn_V))^{(\Gamma-1)/\rho} \bmod \Gamma$ 计

算ς。

ii. host 选择随机整数 $w,r\in\{0,1\}^{l_n+l_\varnothing}$，然后计算 $T_1=Ah^w \bmod n$ 和 $T_2=g^w h^e (g')^r \bmod n$。TPM 计算 $N_V=\varsigma^{f_0+f_1 2^{l_f}} \bmod \Gamma$，TPM 将计算得到的 N_V 发送给 host。

iii. host 和 TPM 联合生成一个零知识证明，证明 T_1 和 T_2 来自 Issuer 办法的身份证书，并且计算 N_V 时使用到的 f 为该身份证书的秘密值。

1. TPM 随机选择 $r_v\in_R\{0,1\}^{l_v+l_\varnothing+l_H}$，计算 $\tilde{T}_{1t}=R_0^{r_{f0}}R_1^{r_{f1}}S^{r_v} \bmod n$，$\tilde{r}_f=r_{f_0}+r_{f_1}^{2^{l_f}} \bmod \rho$，$\tilde{N}_V=\varsigma^{\tilde{r}_f} \bmod \Gamma$。TPM 将 $\tilde{T}_{1t}$ 和 $\tilde{N}_V$ 发送给 host。
2. host 随机选择整数 $r_e\in_R\{0,1\}^{l_e'+l_\varnothing+l_H}$，$r_{ee}\in_R\{0,1\}^{2l_e+l_\varnothing+l_H+1}$，$r_w,r_r\in_R\{0,1\}^{l_n+2l_\varnothing+l_H}$，$r_{ew},r_{er}\in_R\{0,1\}^{l_e+l_n+2l_\varnothing+l_H+1}$；然后根据公式 $\tilde{T}_1=\tilde{T}_{1t}T_1^{r_e}h^{-r_{ew}} \bmod n$，$\tilde{T}_2=g^{r_w}h^{r_e}g'^{r_r} \bmod n$，$\tilde{T}_2'=T_2^{-r_e}g^{r_{ew}}h^{r_{ee}}g'^{r_{er}} \bmod n$ 计算得到$\tilde{T}_1$、$\tilde{T}_2$和$\tilde{T}_2'$。
3. host 计算$c_h=H((n\|g\|g'\|h\|R_0\|R_1\|S\|Z\|\gamma\|\Gamma\|\rho)\|\varsigma\|(T_1\|T_2)\|N_V\|(\tilde{T}_1\|\tilde{T}_2\|\tilde{T}_2')\|\tilde{N}_V)\|n_v)$，并将结果发送给 TPM，TPM 选择一个随机数$n_t\in\{0,1\}^{l_\varnothing}$，然后根据公式 $c=\mathrm{H}(\mathrm{H}(c_h\|n_t)\|b\|m)$，并将 n_t 和 c 返回给 host。
4. TPM 计算 $s_v=r_v+cv$，$s_{f_0}=r_{f_0}+cf_0$，$s_{f_1}=r_{f_1}+cf_1$，然后将 s_v、s_{f_0} 和 s_{f_1} 发送给 host。
5. 平台计算 $s_e=r_e+c(e-2^{l_e-1})$，$s_{ee}=r_{ee}+ce^2$，$s_w=r_w+cw$，$s_{ew}=r_{ew}+cwe$，$s_r=r_r+cr$，$s_{er}=r_{er}+cer$ 和 $s_v=s_{v_1}+2^{l_s}s_{v_2}$，最后输出对消息 m 的签名为：$\sigma=(\varsigma,(T_1,T_2),N_V,c,n_t,(s_v,s_{f_0},s_{f_1},s_e,s_{ee},s_w,s_{ew},s_r,s_{er}))$。

d) Verifier 验证签名 σ 是否有效，如果 Verifier 证明了 σ 是有效的，则将生成一个与该用户共享的对称密钥 *PSK*，并通过安全的方式将 *PSK* 传送给发起匿名认证请求的用户。具体的 Verifier 验证 σ 的步骤如下所示：

i. 首先 Verifier 根据下面的公式

$\hat{T}_1=Z^{-c}T_1^{s_e+c2^{l_e-1}}R_0^{s_{f0}}R_1^{s_{f1}}S^{s_v}h^{-s_{ew}} \bmod n$，$\hat{T}_2=T_2^{-c}g^{s_w}h^{s_e+c2^{l_e-1}}g'^{s_r} \bmod n$

$\hat{T}_2'=T_2^{-(s_e+c2^{l_e-1})}g^{s_{ew}}h^{s_{ee}}g'^{s_{er}} \bmod n$，$\hat{N}_V=N_V^{-c}\varsigma^{s_{f0}+s_{f1}2^{l_f}} \bmod \Gamma$

计算得到$\hat{T}_1$、$\hat{T}_2$、$\hat{T}_2'$和$\hat{N}_V$。

ii. 验证 $c\overset{?}{=}\mathrm{H}(\mathrm{H}(\mathrm{H}(n\|g\|g'\|h\|R_0\|R_1\|S\|Z\|\gamma\|\Gamma\|\rho)\|\varsigma\|(T_1\|T_2)\|N_V\|(\tilde{T}_1\|\tilde{T}_2\|\tilde{T}_2')\|\tilde{N}_V)\|n_v)\|n_t)\|b\|m)$ 以及 $N_{V,\varsigma}\overset{?}{\in}\langle\gamma\rangle$，$s_{f_0},s_{f_1}\overset{?}{\in}\{0,1\}^{l_f+l_\varnothing+l_H+1}$，$\varsigma\overset{?}{\equiv}(\mathrm{H}_\Gamma(1\|bsn_V))^{(\Gamma-1)/\rho} \bmod \Gamma$ 以及

$s_e \overset{?}{\in} \{0,1\}^{l_e'+l_\varnothing+l_H+1}$。

6. 用户将交易公钥$tran_pub$注册交易公钥列表 TPL 中。用户在向 Verifier 匿名证明了自己是群中有效的成员之后，为了在区块链网络中生成一个合法的交易来，必须将自己的交易公钥$tran_pub$添加到 Verifier 维护的交易公钥列表 TPL 中，只有这样，共识节点在处理到该用户生成的交易时才能认为该交易是合法的，进而用户才能通过该交易完成对分布式账本的读取或更新操作。用户通过步骤 5 中从 Verifier 处获得的共享对称密钥 *PSK* 对$tran_pub$加密处理后传送给 Verifier，然后 Verifier 解密得到$tran_pub$，并将$tran_pub$添加到 TPL 中。表 4.2 表示 TPL 的结构。

表 4.2 交易公钥列表 TPL 的结构

交易公钥$tran_pub$	添加时间	有效时间

4.5 区块链模块

区块链模块处理用户在区块链网络中产生的交易，共识节点会判断未处理的交易是否是合法的，以及判断该未处理交易对应的交易公钥$tran_pub$是否存在于 Verifier 维护的交易公钥列表 TPL 中；如果交易合法，那么该交易会调用链码中相应的方法与分布式账本进行交互，例如数据拥有方向分布式账本 State 数据库中添加一个物流隐私数据对应的数据权限状态或者数据访问方从分布式账本 State 数据库中读取e_SK存储路径信息等。下面将介绍区块链模块中分布式账本和链码的设计方法。

4.5.1 相关数据结构设计

本小节内容主要是介绍如何对分布式账本中 State 数据库中的数据结构进行设计。State 数据库中存储的是物流用户隐私数据对应的数据权限状态，当数据拥有方产生一个新的物流隐私数据时，通过构造一个交易来调用链码中相应的方法向 State 数据库中添加该隐私数据对应的数据权限状态。State 数据库具体的数据结构如图 4-7 所示。

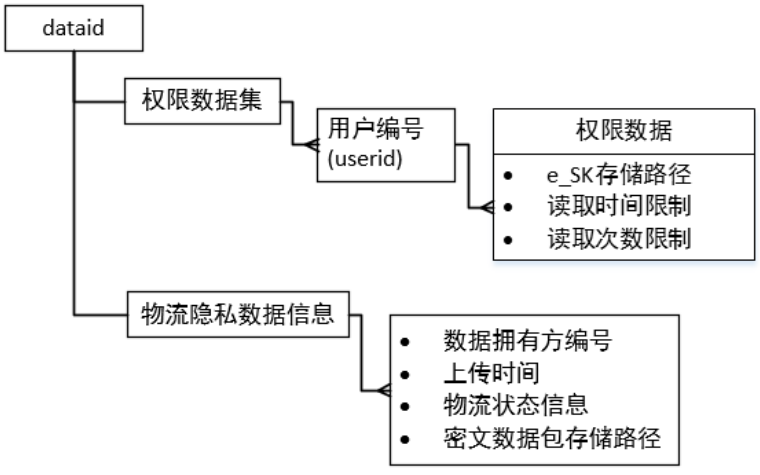


图 4-7 State 数据库数据权限状态的数据结构

如上图所示，State 数据库中数据权限状态是以物流用户隐私数据的 id 标识的，即每一个物流用户隐私数据都在 State 数据中对应一个 key 为 *dataid* 的数据权限状态。数据权限状态包括两类：权限数据集和物理隐私数据信息。权限数据集是根据数据拥有方根据自己的意愿设置的，用来对物流隐私数据的访问权限进行控制管理，数据拥有方可以在权限集中更改一个用户的权限数据或者添加一个新用户的权限数据，用户对应的权限数据中包含了：*e_SK* 存储路径、读取时间的限制以及读取次数的限制，用户可以根据是否能够读取出 *dataid* 下自己的 *userid* 对应的权限数据，来判断自己是否有权限访问编号为 *dataid* 的物流隐私数据。物流隐私数据信息包括了：数据拥有方的编号、隐私数据的上传时间以及密文数据包的存储路径。数据拥有方的编号是标识了 *dataid* 对应的隐私数据数据拥有方；隐私数据的上传时间则是该隐私数据对应的数据权限状态添加到分布式账本 State 数据库中的时间；密文数据包存储路径是访问云存储平台上该物流隐私数据密文的 URL，用于下载该隐私数据的密文数据包；物流状态信息记录了物流从发件人寄出到收件人签收整个过程的实时状态信息。权限数据集和物流隐私数据信息相应的结构体表示如下（go 语言格式）：

```
type DataPermissionsInfo struct {
    DataID      string          `json:"dataid"`
    Pms         []Permission    `json:"permissions"`
    Lstatus     []string        `json:"lstatus"`
    OwnerID     string          `json:"ownerid"`
    StoragePath string          `json:"storagepath"`
}
type Permission struct {
    UserID      string  `json:"userid"`
    EncryptedSK int     `json:"e_SK"`
    Timelimit   int     `json:"timelimit"`
    Times       int     `json:"times"`
}
```

4.5.2 链码设计

在 4.2 节中已经介绍到 Fabric 的链码就是智能合约，目前超级账本 Fabric 项目中提供了系统链码和用户链码。系统链码通过硬编码嵌入系统中，提供对系统进行配置、管理的支持；一般所谈的链码为用户链码，通过提供可编程能力提供了对上层应用的支持，用户通过链码相关的 API 编写用户链码，将用户与分布式账本直接交互的相关逻辑封装在用户链码中，用户即可以通过区块链应用程序向区块链网络发送交易来调用相应的链码方法来与分布式账本进行交互，本文就是对用户链码进行设计。

目前，超级账本 Fabric 项目支持使用 Go 语言开发链码，每个链码都需要实现以下 Chaincode 接口：

```
type Chaincode interface{
    Init(stub ChaincodeStubInterface) pb.Response
    Invoke(stub ChaincodeStubInterface) pb.Response}
```

其中当链码收到实例化（instantiate）或升级（upgrade）类型的交易时，Init 方法或被调用，通常只调用一次；当链码收到调用（invoke）或查询（query）类型的交易时，Invoke 方法将被调用，并且可以被调用多次。本文针对物流场景中的业务需求进行链码的设计，设计的方法如表 4.3 所示。

表 4.3 链码设计中的方法汇总

方法	分支方法	功能	参数实例
Init	无	无	[]
Invoke	addDataPermissionsInfo	创建一个数据权限状态数据	[“addDataPermissionsInfo”,”dataid”,”userid”,”encryptedSK”,”timelimit”,”times”,””,”storagepath”]
Invoke	addAUserPermission	添加一个用户的权限数据	[“addAUserPermission”,”dataid”,”userid”,”encryptedSK”,”timelimit”,”times”]
Invoke	deleteAUserPermission	删除一个用户的权限数据	[“deleteAUserPermission”,”dataid”,” userid”]
Invoke	readUserESK	读取一个用户的 e_SK 存储路径	[“readUserESK”,”dataid”,”userid”]
Invoke	updateUserTimes	更新一个用户的读取次数限制	[“updateUserTimes”,”dataid”,”userid”,”times”]
Invoke	updateUserTimelimit	更新一个用户的读取时间限制	[“updateUserTimelimit”,”dataid”,” userid”,”timelimit”]
Invoke	addLstatus	添加新的物流状态	[“addLstatus”,”dataid”,”newLstatus”]
Invoke	readLstatus	读取物流状态信息	[“readLstatus”,”dataid”]
Invoke	readStoragePath	读取物流隐私数据密文存储路径	[“readStoragePath”,”dataid”]

表 4.3 中总共有 9 个方法，数据拥有方和数据访问法通过其中的 8 个 Invoke 方法来与分布式账本进行交互，下面对这 8 个 Invoke 方法进行介绍：

1. addDataPermissionsInfo：该方法负责在 State 数据库中添加新产生隐私数据对应的数据权限状态，由数据拥有方产生一个新的物流隐私数据时调用，需要提供的参数有：新产生物流隐私数据的 *dataid* 、权限数据集（包含多个用户的权限

数据）、空的物流状态信息（代表还没有状态信息）、数据拥有方的*userid* 以及物流隐私数据密文存储路径 URL。

2. addAUserPermission：该方法负责向 State 数据库中某个数据权限状态添加一个用户的权限数据，由数据拥有方执行，需要提供的参数是：物流隐私数据的*dataid* 以及待添加的用户权限数据。
3. deleteAUserPermission：该方法负责从 State 数据库中某个数据权限状态删除一个用户的权限数据，由数据拥有方执行，需要提供的参数是：物流隐私数据的*dataid* 以及数据访问方的*userid* 。
4. readUserESK：该方法负责从 State 数据库中某个数据权限状态中读取出一个用户 e_SK 的存储路径，由数据访问方在解密数据拥有方物流隐私数据之前执行，需要提供的参数是：物流隐私数据的*dataid* 和数据访问方的*userid* 。
5. updateUserTimes：该方法负责更新 State 数据库中某个数据权限状态某个用户的访问次数，由数据拥有方执行，需要提供的参数是：物流隐私数据的*dataid* 、数据访问方的*userid* 以及更新的访问次数值。
6. updateUserTimelimit：该方法负责更新 State 数据库中某个数据权限状态某个用户的访问时间限制，由数据拥有方执行，需要提供的参数是：物流隐私数据的*dataid* 、数据访问方的*userid* 以及更新的访问时间限制。
7. addLstatus：该方法负责向 State 数据库中添加某个数据权限状态新的物流状态，由物流中转人员和快递员调用，需要提供的参数是：物流隐私数据的*dataid* 以及待添加的物流状态。
8. readLstatus：该方法负责从 State 数据库中读取出某个物流隐私数据对应的物流状态信息，用于跟踪物流的状态，数据拥有方和数据访问方都可以调用该方法，需要提供的参数是：物流隐私数据的*dataid* 。
9. readStoragePath：该方法负责从 State 数据中读取该物流隐私数据密文的云存储路径，需要提供的参数是：物流隐私数据的*dataid* 。

4.6 本章小结

本章针对现有方案中用户无法对物流隐私数据的访问权限进行控制管理，其控制过程缺乏透明性和追溯性的问题，提出了一种基于区块链和 DAA 匿名认证相结合的访问权限管理机制。该机制基于 DAA 匿名认证提出了一种成员身份管理机制，为区块链上的实体提供匿名可验证的身份，通过维护的一个交易公钥列表 TPL 来实现对区块链节点的访问权限控制，解决了当前区块链技术存在的隐私泄露的风险；同时，该机制通过分布式账本保存用户对隐私数据的访问权限，使用链码（即智能合约）来封装物流中各角色与隐私数据之间存在的业务逻辑，实现了用户对隐私数据的访问权限控制管理，并且通过区块链提供了一个不可更改的访问记录。

第五章 物流下单原型系统的设计与实现

本章提出的物流下单原型系统是在前两章提出的基于分层加密的隐私数据保护机制和访问权限管理机制的基础上设计和实现的。该物流下单原型系统能够保证物流用户隐私数据的安全存储，同时用户能够对物流隐私数据的访问权限进行控制管理。本章首先介绍了系统的整体架构，该系统由数据拥有方、数据访问方及云存储平台构成，描述了整个系统的工作流程；然后介绍了系统功能模块中的分层加密模块和访问权限管理模块的具体实现；最后对本章提出的物流下单原型系统进行功能测试和性能测试，并就测试结果进一步分析。

5.1 系统整体架构

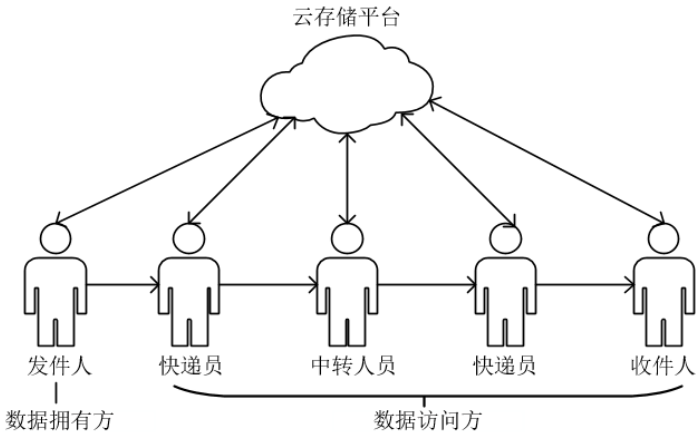


图 5-1 系统架构图

本章提出的物流下单原型系统由数据拥有方、数据访问方以及云存储平台构成，系统的架构如图 5-1 所示，各个部分的介绍如下：

1. 数据拥有方：

在物流的过程中，发件人就可以视为一个物流隐私数据的数据拥有方。数据拥有方可以生成一个物流订单，从而产生一个物流隐私数据，该隐私数据由数据拥有方加密处理后存储在云端，数据拥有方可以查询该订单的状态，并且对物流隐私数据的访问权限进行控制管理。

2. 数据访问方

在物流的过程中，快递员、中转人员以及收件人等都可以视为数据访问方。数据访问方可以根据数据拥有方设定的访问权限访问数据拥有方的物流隐私数据，如果数据访问方某个隐私数据的访问权限，则他可以从云存储平台将对应隐私数据的密文数据包下载下来，并解密得到对应访问权限的部分隐私数据。

3. 云存储平台

该平台用来存储物流用户隐私数据加密生成的密文数据包，数据拥有方和数据访问

方可以通过该平台提供的密文数据包 URL 来访问云存储平台上对应的密文数据包。在本章提出的物流下单原型系统中使用的云存储平台是阿里巴巴的开放云存储平台（OSS，Open Storage Service）。

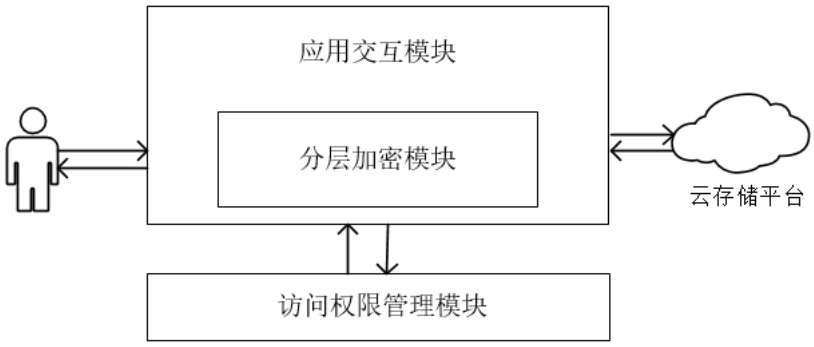


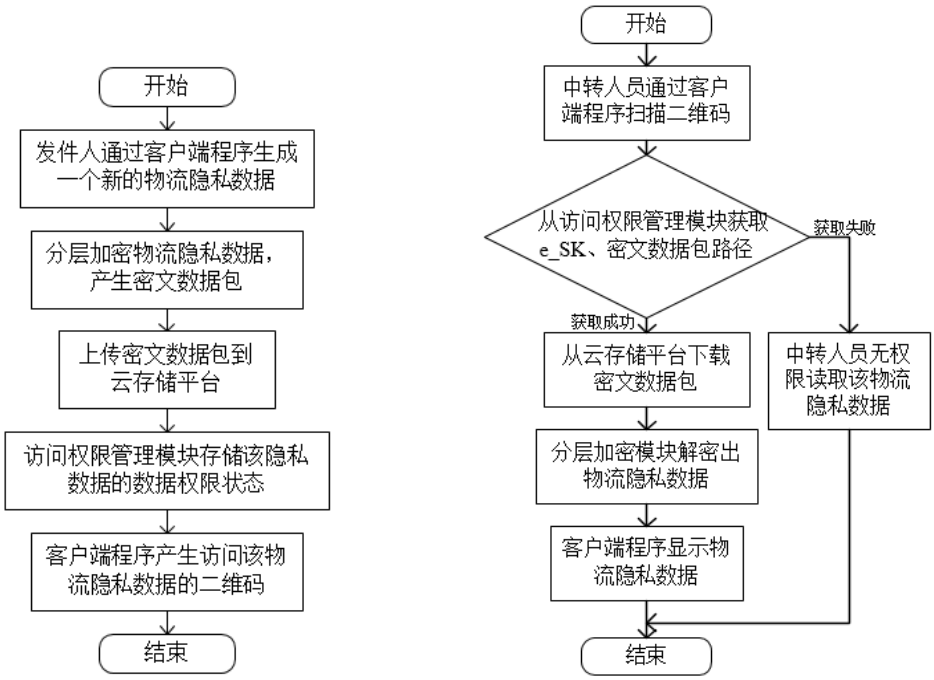
图 5-2 系统功能模块

图 5-2 是该物流下单原型系统的功能模块图，主要包括应用交互模块、分层加密模块以及访问权限管理模块，各个模块的主要功能如下：

1. 应用交互模块：该模块即是客户端程序，数据拥有方通过客户端程序生成一个物流订单，进而生成一个新的物流隐私数据，同时数据拥有方可以访问自己的物流隐私数据，并且对隐私数据的访问权限进行控制管理。
2. 分层加密模块：该模块是基于第三章的分层加密隐私数据云存储机制设计的，主要功能是将数据拥有方的物流隐私数据分层加密生成密文数据包以及解密从云存储平台下载下来的密文数据包来获得数据拥有方的物流隐私数据。
3. 访问权限管理模块：该模块是基于第四章提出访问权限管理机制设计的，主要功能是通过区块链保存和分发数据访问方解密密文数据包所需要的 e_SK 存储路径和密文数据包的存储路径。当数据拥有方产生一个物流隐私数据时，会生成该隐私数据对应的数据权限状态，并将其保存到分布式账本的 State 数据库中，当数据访问方想要访问一个物流用户隐私数据时，必须通过访问 State 数据库中的该隐私数据对应的数据权限状态获得该隐私数据的云存储路径以及数据访问方的 e_SK 存储路径。

图 5-3 通过介绍发件人（数据拥有方）、中转人员（数据访问方）两个不同的角色的工作流程来介绍整个系统的工作流程。图 5-3 (a)是发件人通过应用交互模块生成一个新的物流隐私数据的过程，首先发件人通过客户端程序填写物流隐私数据，确认生成该隐私数据之后，客户端程序中的分层加密模块对该物流隐私数据进行分层加密，产生密文数据包；然后客户端程序将密文数据包存储在云存储平台上；接着客户端程序会生成该物流隐私数据的数据权限状态，将数据权限状态存储在访问权限管理模块之中；最后客户端程序会生成一个访问该物流隐私数据的二维码，数据访问方可以扫描该二维码来获取该物流隐私数据。图 5-3 (b)是中转人员通过应用交互模块访问一个已存在的物流隐私数据的过程。首先中转人员通过客户端程序扫描二维码会向访问权限管理模块发送一个

包含物流隐私数据 *dataid* 、中转人员 *userid* 以及时间戳 *timestamps* 的请求；当访问权限管理模块收到该请求之后，会判断该中转人员是否有访问 *dataid* 的权限，如果有，则返回该中转人员的 *e_SK* 存储路径以及物流隐私数据的云存储的路径 *storagepath*，如果没有则返回失败的结果；然后客户端程序根据 *storagepath* 从云存储平台上将 *dataid* 对应的密文数据包下载到本地，并且根据 *e_SK* 存储路径下载 *e_SK*；之后分层加密模块通过 *e_SK* 解密密文数据包，获得中转人员对应部分的物流隐私数据；最后客户端程序将获得的物流隐私数据展示出来。



(a) 发件人生成一个物流隐私数据的流程　　(b) 中转人员访问一个物流隐私数据的流程

图 5-3 系统的工作流程

5.2 分层加密模块

```
发件人姓名：张三
发件人地址：AA省AA市AA区AA街道AA号
发件人电话号码：18888888888
发件人单位名称：张三有限公司

收件人姓名：李四
收件人地址：BB省BB市BB区BB街道BB号
收件人电话号码：19999999999
收件人单位名称：李四有限公司

类别：书籍
体积：300mm*400mm
数量：1
重量：4kg

付款方式：现金
运费：20RMB

物流路径信息：AA省AA市AA区AA街道AA号-->起点快递站-->1号物流中转站-->2号物流中转站-->3号物流中转站-->4号物流中转站-->终点快递站-->BB省BB市BB区BB街道BB号
```

图 5-4 物流用户隐私数据示例

分层加密模块是基于第三章的分层加密隐私数据云存储机制设计的，主要负责将用户的物流隐私数据分层加密生成密文数据包，以及解密从云存储平台下载下来的密文数据包获得物流用户隐私数据。图 5-4 是一个具体物流用户隐私数据示例，主要包括的数据类型有：发件人信息、收件人信息、物品信息、费用信息以及物流路径信息（物流路径信息是系统根据物流隐私数据的起点地址和终点地址自动生成的）

接下来主要介绍密文数据包的产生和密文数据包的解密过程。

5.2.1 密文数据包的产生

当数据拥有方通过客户端程序产生一个新的物流隐私数据之后，分层加密模块负责将物流隐私数据加密生成密文数据包，图 5-5 显示的是密文数据包的产生过程。具体的步骤如下所述：

1. 首先将数据拥有方的物流隐私数据根据安全级别（$security\_level = 3,2,1$）分割为三部分 m_1、m_2、m_3，如表 5.1 所示。$security_level = 3$的部分有物品和费用信息；$security_level = 2$的部分有发件人信息和收件人信息； $security_level = 1$的部分是物流路径信息。
2. 使用对称加密算法 AES-128 算法加密 m_1、m_2、m_3，生成物流隐私数据密文 cm_1、cm_2、 cm_3。

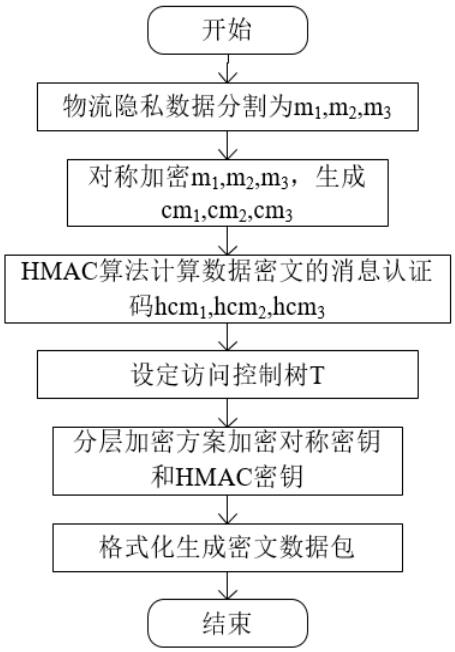


图 5-5 密文数据包的产生过程

3. 使用 HMAC 算法计算出 cm_1、 cm_2 和 cm_3 的消息认证码 hcm_1、 hcm_2、 hcm_3。
4. 为了实现在物流场景下各个角色（发件人、收件人、快递员以及中转人员）只能获得完成角色对应业务操作所需要的用户物流隐私数据，在使用基于属性的分层加密方案加密 key_1、 key_2、 key_3 以及 h_key 的时候，需要使用如图 5-6 所示的访问控制树（用户的属性由字符串表示，用户的属性集就是由多个属性组成

的字符串数组，在这里本文为了更加清楚的说明，将各个角色的属性集设为角色的名字字符串）。

表 5.1 物流用户隐私数据安全分割策略

类别	安全级别 (security_level)	可访问该数据的角色
物品信息	3	发件人、收件人
费用信息	3	发件人、收件人
发件人信息	2	发件人、收件人、快递员
收件人信息	2	发件人、收件人、快递员
物流路径信息	1	发件人、收件人、快递员、中转人员

5. cpabeEncrypt 是分层加密算法的加密函数，在加密 key_1、key_2、key_3 时，分别使用图 5-6 中 T_1、T_2、T_3 表示的访问控制树，在加密 h_key 时，使用 T_1 表示的访问控制树，在这里访问控制树通过字符串来描述，例如 T_3 用字符串“（收件人 or 发件人）and（1 of （收件人，发件人，快递员）and 1 of（收件人，发件人，快递员，中转人员））”表示，加密产生密钥密文文件分别为 $key_1.cpabe$、$key_2.cpabe$、$key_3.cpabe$ 和 $h\_key.cpabe$。

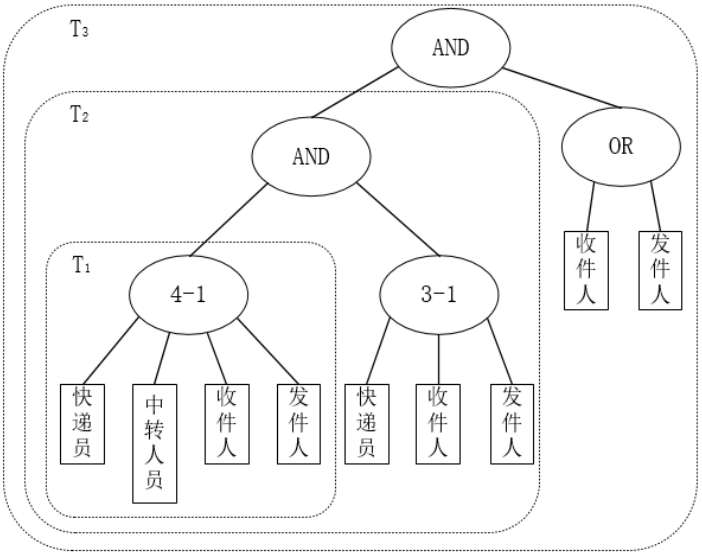


图 5-6 物流场景下访问控制树的嵌套结构

6. 通过 cipherformat 函数将产生的密文格式化生成密文数据包，形成的数据包格式如图 5-7 所示。其中 ciphertextdata 文件中保存的是 cm_1、cm_2、cm_3 以及消息认证码 hcm_1、hcm_2、hcm_3。”*.cpabe*”后缀的文件是分层加密函数 cpabeEncrypt 加密密钥产生的密钥密文文件。

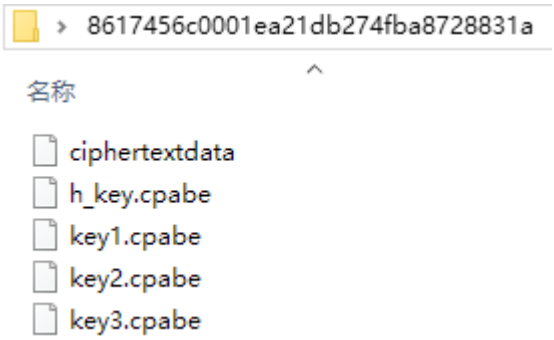


图 5-7 密文数据包

密文数据包产生的代码实现如下所示(java)，其中 aesEncrypt 是 AES-128 加密函数，hmacsha1 是 HMAC 算法函数，cpabeEncrypt 是分层加密算法的加密函数，md5 是 MD5 算法函数，cipherformat 是密文数据包格式化函数。

```
public static String hierarchicalEncrypt (String[] content, String[] k ,String[] T){
    String[] plaintext = content;//待加密的物流隐私数据 m1,m2,m3
    String[] keys = k;//加密 plaintext 需要使用的密钥 key1,key2,key3,h_key
    String[] actree = T;//分层加密算法使用的访问控制树 T1，T2，T3
    ArrayList<String> ciphertext =new ArrayList<>();//加密之后的密文数据包
    //对称加密 m1,m2,m3 产生 cm1,cm2,cm3
    String cm;
    for (int i =0;i<3;i++){
        cm = aesEncrypt(plaintext[i],keys[i]);
        ciphertext.add(cm);}
    //分别求出 cm1,cm2,cm3 的 HMAC 码 hcm1,hcm2,hcm3
    for (int i =0;i<3;i++){
        cm = hmacsha1(ciphertext.get(i),keys[3]);
        ciphertext.add(cm);}
    //使用分层加密算法对 key1,key2,key3,h_key 进行加密
    // 产生 key1.cpabe,key2.cpabe,key3.cpabe 以及 h_key.cpabe 文件
    String path;
    for (int i =0;i<3;i++){
        path = cpabeEncrypt(actree[i],keys[i]); // 返回密钥密文文件的路径
        ciphertext.add(path);}
    ciphertext.add(cpabeEncrypt(actree[1],keys[4]));  //添加 h_key.cpabe 路径
    //格式化生成物流密文数据包
    String dataid = md5(plaintext);  //生成物流隐私数据的 dataid 值
    //对密文数据包进行格式化，返回物流隐私数据密文数据包的路径
    path = cipherformat(dataid, ciphertext);
```

```
    return path;
}
```

5.2.2 密文数据包的解密

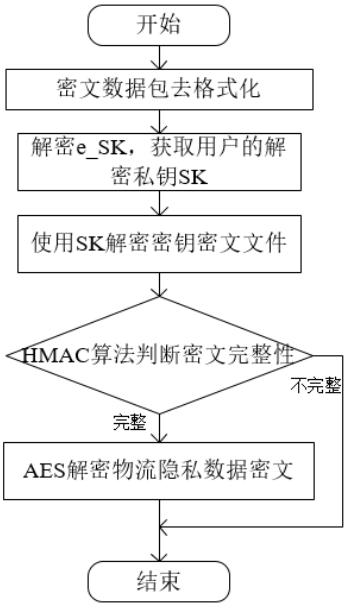


图 5-8 密文数据包解密流程

图 5-8 是密文数据包的解密流程图。密文数据包的解密过程是密文数据包加密的逆过程，该过程的前提是客户端程序从访问权限管理模块成功请求得到数据访问方的 e_SK 存储路径以及密文数据包的云存储路径 *storagepath*，然后客户端将 e_SK 以及密文数据包下载到本地。解密具体的过程如下所示（具体的代码实现在这里就不给出了）：

1. 使用 cipherdeformat 函数将密文数据包去格式化，得到一个字符串数组 ciphertext，里面的内容是 cm_1、cm_2、cm_3、hcm_1、hcm_2、hcm_3 以及”*.cpabe*”后缀的密钥密文文件的路径信息。
2. 通过客户端保存的数据访问方的公钥 *u_pub* 运行 RSA 算法对 e_SK 进行解密，得到用户解密私钥 *SK* 文件。
3. 通过解密私钥 *SK* 运行分层加密算法的解密函数 cpabeDecrypt 对”*.cpabe*”后缀的密钥密文文件进行解密，得到 h_key 以及数据访问方在物流过程中对应角色的对称密钥 $key_i, \mathrm{i} \in \{3,2,1\}$，例如收件人能够解密获得 key_1、key_2 和 key_3。
4. 通过解密得到的 h_key 运行 hmacsha1 函数，判断 cm_1、cm_2 和 cm_3 的完整性，如果不完整，则密文数据包解密失败，反之继续解密。
5. 通过解密得到的 key_i 运行 aesDecrypt 函数，对 $cm_i, i \in \{3,2,1\}$ 进行解密，得到数据访问方访问权限对应部分的物流隐私数据。表 5.2 表示了物流过程中各个角色对应能够解密得到的物流用户隐私数据。

表 5.2 物流中各角色能够解密得到的物流用户隐私数据

角色	解密得到的物流用户隐私数据
发件人	全部隐私数据
收件人	全部隐私数据
快递员	发件人信息、收件人信息、物流路径信息
物流中转人员	物流路径信息

5.3 访问权限管理模块

访问权限管理模块是基于第四章提出访问权限管理机制设计的，该模块包括两个子模块：成员身份管理模块与区块链模块。成员身份管理模块负责为区块链上的实体节点提供匿名可验证的身份，并通过维护的一个交易公钥列表 TPL 来实现对区块链节点的访问权限管理；区块链模块负责实现数据拥有方对物流隐私数据的访问权限进行控制管理。下面分别对两个模块的实现进行介绍。

5.3.1 成员身份管理模块

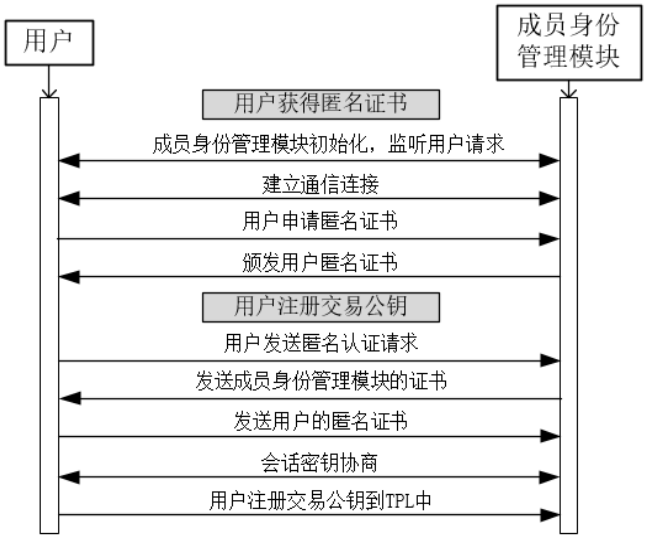


图 5-9 成员身份管理工作流程

图 5-9 表示的是成员身份管理模块的工作流程图。图中用户是物流过程中参与的各个角色，用户通过客户端与成员身份管理模块通信，成员身份管理模块同时扮演了 4.4 节中 Issuer 与 Verifier 的角色。整个工作可以分成两个部分，第一个部分是用户获的自己的匿名身份证书，第二个部分是用户注册交易公钥到 TPL 中。在第一部分中，成员身份管理模块会进行初始化，并监听是否有来自用户的请求，用户在刚进入到系统中的时候会向成员身份管理模块请求自己的匿名证书，成员身份管理模块在验证了用户的身份之后，会为用户颁发匿名证书；在第二部分中，用户需要将自己的交易公钥注册到 TPL 中才能创建一个合法的区块链交易，首先用户需要向成员管理模块发送匿名认证请求，

然后用户和成员身份管理模块交换证书完成用户的匿名身份认证，协商出彼此之间的会话密钥，最后用户将交易公钥注册到 TPL 中。

在用户与成员身份管理模块交互的过程中，需要传递各种类型的消息。一个消息数据结构主要由三个部分组成，分别是：消息类型、消息长度以及消息内容，Java 的定义消息数据结构如下所示：

```
public class info{
        private InfoType    infotype;     //消息的类型
        private int infolength;   //消息的长度
        private ArrayList<Byte> infobody;   //消息的内容
        其他的方法……
}
enum InfoType{
       userHello(0x70),memberHello(0x71),memberCredential(0x72),
       userAnonyCredential(0x73),memberKeyNegotiate(0x74),
       userKeyNegotiate(0x75),userSKCompleted(0x76),
       memberSKCompleted(0x77),errorInfo(0x78),encryptedData(0x79);
       其他的方法……
}
```

其中 infobody 是一个不定长度的字节数组，代表了消息的具体内容，消息的长度由 infolength 表示，infolength 是一个 2 字节长度的非负的整数，InfoType 是一个枚举类，代表了消息的类型，长度为一个字节。消息主要有 10 种，可以分为三类，分别为：握手协议消息、错误消息和加密数据消息。介绍如下：

1. 握手协议消息
   a) userHello 消息：该消息由用户发送给成员身份管理模块，包含了用户生成的随机数 user_random 以及扩展字段 user_extensions 来说明使用的加解密算法，默认的加解密算法是 RSA 算法。
   b) memberHello 消息：该消息由成员身份管理模块生成，包含了成员身份管理模块生成的随机数以及 member_extensions 来说明使用的加解密算法，默认的是 RSA。
   c) memberCredential 消息：该消息由成员身份管理模块生成，包含了成员身份管理模块基于 X.509 的公钥证书，证书中包含了成员身份管理模块的 RSA 公钥。
   d) userAnonyCredential 消息：该消息由用户生成，包含了用户的匿名证书。
   e) memberKeyNegotiate 消息：该消息包含了 DH 密钥协商协议的公开参数和成员身份管理模块生成的密钥协商的公钥，同时消息中还包含了成员身份管理模块对公开参数、user_random 以及 member_random 的签名。
   f) userKeyNegotiate 消息：该消息中包含了用户根据成员身份管理模块的 DH 公开参数生成自己的密钥协商公钥。

g) userSKCompleted 消息：成员身份管理模块通过该消息通知用户已对其身份进行验证并且会话密钥协商成功。

h) memberSKCompleted 消息：用户通过该消息通知成员身份管理模块会话密钥协商成功。

2. 错误消息：包括了 errorInfo 消息。在用户和成员身份管理模块交互的过程中可能会发生错误后者双方的身份认证不成功，需要通过发送该消息来提示对方错误的存在。
3. 加密数据消息：包括了 encryptedData。当用户和成员身份管理模块成功协商了会话密钥之后，双方通过该会话密钥进行安全通信。用户就是通过该消息使用将自己的交易公钥发送给成员身份管理模块，然后成员身份管理模块将用户的交易公钥存储在 TPL 中。

5.3.2 区块链模块

图 5-10 表示的是客户端程序与区块链网络交互的过程。Hyperledger Fabric 提供了多种语言的 SDK，包括了 Node、JS、Python、Java 和 Go 等，本文使用 Java 版本的 SDK（fabric-sdk-java）来完成客户端 API 的开发，数据访问方和数据拥有方通过客户端程序调用不同的 API 来构造出不同功能的交易，交易会进入到区块链网络中，只有当共识节点判断该交易是合法的之后（即交易对应的交易公钥已经注册到 TPL 中），才会调用链码相应的代码对分布式账本执行对应的操作；区块链网络会将此次交易操作分布式账本的相关结果返回给客户端。

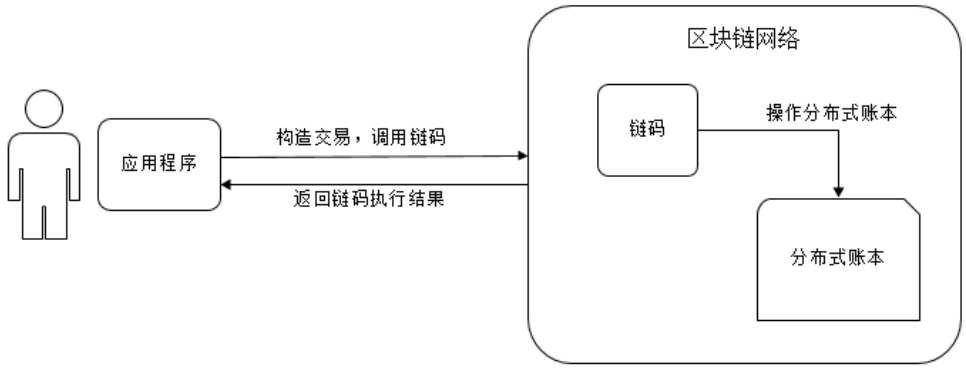


图 5-10 客户端程序与区块链网络的交互过程

表 5.3 列出了物流场景中创建的 API、对应可调用该 API 的角色以及在区块链网络中调用的链码方法。

发件人（即数据拥有方）在生成一个新的物流隐私数据后调用 addDataPerStatus 方法创建一个交易，该交易调用链码中的 addDataPermissionsInfo 和 addAUserPermission 方法将该物流隐私数据对应的数据权限状态存储到分布式账本的 State 数据库中（数据权限状态的数据结构已在 4.5.1 节中给出）。同时发件人可以通过调用 updateTimes、updateTimeLimit 和 deleteUserPer 方法来对数据访问方进行访问权限的控制管理，控制他们读取的时间限制以及次数限制。默认情况下在一个物流过程还未结束时，为了让物流过程顺利的完成，发件人无法通过调用这些方法对数据访问方进行访问权限的控制管

理。

快递员以及中转人员在物流的过程中会调用 addLstatus 方法实时的更新该物流的状态信息。快递员、中转人员以及其他人员都可以通过调用 getLstatus 方法获取到实时的物流状态信息。

对于物流过程中的所有角色，当他们想要获得发件人的物流隐私数据时，都必须调用 getStoragePath 方法获得密文数据包的云存储路径，将密文数据包下载到本地；以及调用 getESK 获取到自己的 *e_SK* 存储路径，用来获得 *e_SK* 解密下载得到的密文数据包，如果 *e_SK* 存储路径获取失败，则代表该用户无权限访问该物流用户隐私数据。

表 5.3 不同角色对应的 API 及调用的链码方法

API	作用	可调用的角色	调用的链码方法
addDataPerStatus	向 State 数据库中添加数据权限状态	发件人	addDataPermissionsInfo、addAUserPermission
updateTimes	更新访问次数	发件人	updateUserTimes
updateTimeLimit	更新访问时间限制	发件人	updateUserTimeLimit
deleteUserPer	删除权限数据	发件人	deleteAUserPermission
addLstatus	添加新的物流状态	快递员、物流中转人员	addLstatus
getESK	获取 e_SK 存储路径	所有角色	readUserESK
getStoragePath	获取密文数据包存储路径	所有角色	readStoragePath
getLstatus	获取当前物流状态	所有角色	readLstatus

链码中的 readUserESK 方法是每个数据访问方在解密获得密文数据包时必须调用的链码方法，下面给出了链码中 readUserESK 对应的代码（go 语言）。在 readUserESK 执行的时候首先通过 GetState 方法从 State 数据库中将物流隐私数据 *dataid* 对应的数据权限状态提取出来；然后提取出该数据访问方的时间限制 *Timelimit* 和访问次数限制 *Times*，如果 *Timelimit* 比访问的时间早或者 *Times* 等于 0，则代表该数据访问方已经无权访问 *dataid* 标识的物流隐私数据；反之，如果 *Timelimit* 和 *Times* 都满足访问的条件，则会从 *dataid* 标识的数据权限状态中取出数据访问方 *viewerid* 对应的 *e_SK* 存储路径，同时将该数据访问方对应的 *Times* 减一。

```
func (t *SimpleChaincode) readUserESK(stub shim.ChaincodeStubInterface, args []string) pb.Response {
    if len(args) != 2 {     //判断参数个数是否正确
        return shim.Error("Incorrect number of arguments. Expecting 2") }
    dataid := strings.ToLower(args[0])      //物流隐私数据的 id 值
    viewerid := strings.ToLower(args[1])    //数据访问方的 id 值
    //根据 dataid 从 State 数据库中取出相应的数据权限状态
```

```
    dataPermissionsInfoAsBytes, _ := stub.GetState(dataid)
    var datapermissionsinfo DataPermissionsInfo
    json.Unmarshal(dataPermissionsInfoAsBytes, &datapermissionsinfo)   //序列化
    var i int
    for a := 0; a < len(datapermissionsinfo.Pms); a++ {
        if datapermissionsinfo.Pms[a].ViewerID == viewerid {
            i = a
        // 判断该数据访问方的时间限制和次数限制
        var nowtime int
        nowtime, _ = strconv.Atoi(strconv.FormatInt(time.Now().Unix(), 10))
        if nowtime > datapermissionsinfo.Pms[i].Timelimit {
            break}   //可访问时间 Timelimit 已经无效了
        if datapermissionsinfo.Pms[i].Times == 0 {
            break}   //访问次数已经为 0
        //读取 e_SK，并将访问次数 Times 减一
        datapermissionsinfo.Pms[i].Times = datapermissionsinfo.Pms[i].Times - 1 //Times-1
        databytes := []byte(datapermissionsinfo.Pms[i].EncryptedSK) //读取 e_SK

        data, _ := json.Marshal(datapermissionsinfo) //序列化为 JSON 对象
        err = stub.PutState(dataid, data)   //将更新的数据权限状态存进 State 数据库
            return shim.Success(databytes)   //返回数据访问方的 e_SK 存储路径
      }
    }
    return shim.Error("readUserESK failed!")
}
```

图 5-11 表示在终端下构造客户端请求调用链码中 readUserESK 方法成功得到数据访问方的 *e_SK* 存储路径。调用 readUserESK 方法使用的参数是：{“Args”:[“readUserESK”,” 8617456c0001ea21db274fba8728831a”,”testuser1”]}，其中第一个 readUserESK 代表的是调用链码中的方法，8617456c0001ea21db274fba8728831a 是物流隐私数据的 *dataid*，testuser1 是数据访问方的 *userid*。

```
2018-03-23 12:52:04.600 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 00a Chaincode inv
oke successful. result: status:200 payload:"https://encryptedlogisticsdata.oss-cn-beijing.a
liyuncs.com/e_sk.txt"
2018-03-23 12:52:04.600 UTC [main] main -> INFO 00b Exiting.....
```

图 5-11 readUserESK 终端返回结果

图 5-12 表示在终端下构造客户端请求调用链码中 readStoragePath 方法获取物流隐私数据密文数据包的存储路径。调用 readStoragePath 方法使用的参数是：{“Args”:[“readStoragePath”,” 8617456c0001ea21db274fba8728831a”]}，在这里参数中不需

要包含构造请求数据访问方的*userid* 。

图 5-12 readStoragePath 终端返回结果

图 5-13 表示在终端下构造客户端请求调用链码中 readLstatus 方法获取物流状态信息。调用 readLstatus 方法使用的参数是：{“Args”:[“readLstatus”,” 8617456c0001ea21db274fba8728831a”]}，在这里参数中同样不需要包含构造请求用户的*userid* 。

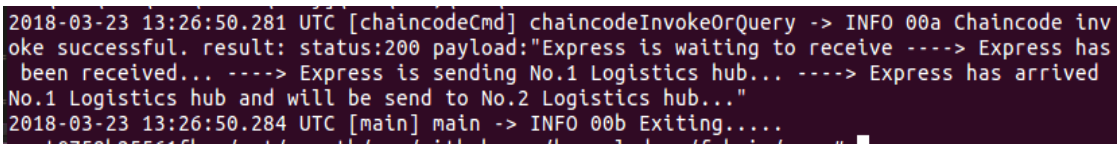


图 5-13 readLstatus 终端返回结果

5.4 系统测试与分析

本小节的主要内容是对整个系统的功能进行测试，并且对系统中分层加密模块以及访问权限管理模块的性能进行分析与测试。

5.4.1 测试环境

本文的测试环境如表 5.4 所示。

表 5.4 测试环境

名称	参数
CPU	Intel(R) Core(TM) i7-2600K CPU @3.40GHz
内存	8.0GB
操作系统	Ubuntu 16.04 64bit
系统开发工具	Eclipse Juno 23.0.2、Android Studio 2.2.2、Sublime Text 3、MySQL WorkBench 6.3.8
Android 版本	Android 6.0

5.4.2 系统功能测试

本文基于 Android 智能手机进行物流下单原型系统的应用开发。物流下单原型系统 APP 分为了两种：一种是针对物流业务使用方（即发件人、收件人）开发的，一种是针对物流工作人员（即快递员、物流中转人员）开发的。前者主要包含的功能是：发件人寄件、发件人查看寄件的历史记录、发件人和收件人查看物流状态以及发件人对物流隐私数据进行访问权限管理；后者包含的功能是：物流工作人员查看物流用户隐私数据。

接下来通过系统的界面操作来简要的介绍物流下单原型系统的工作流程。

(a) 系统登录界面

(b) 发件人寄件界面

(c) 查看寄件记录界面

(d) 发件人收件人查看物流状态界面

(e) 发件人对物流隐私数据进行访问权限管理界面

图 5-14 发件人收件人操作界面

物流下单原型系统的登录界面如图 5-14(a)所示。当用户想要加入到系统中时，用户需要通过提供自己的个人身份信息，包括真实姓名、证件号等。已注册的用户直接通过用户名和密码登录进入系统。当用户注册进入系统之后，该用户就有权利发起一个新的物流过程以及对物流隐私数据的访问权限进行控制管理。

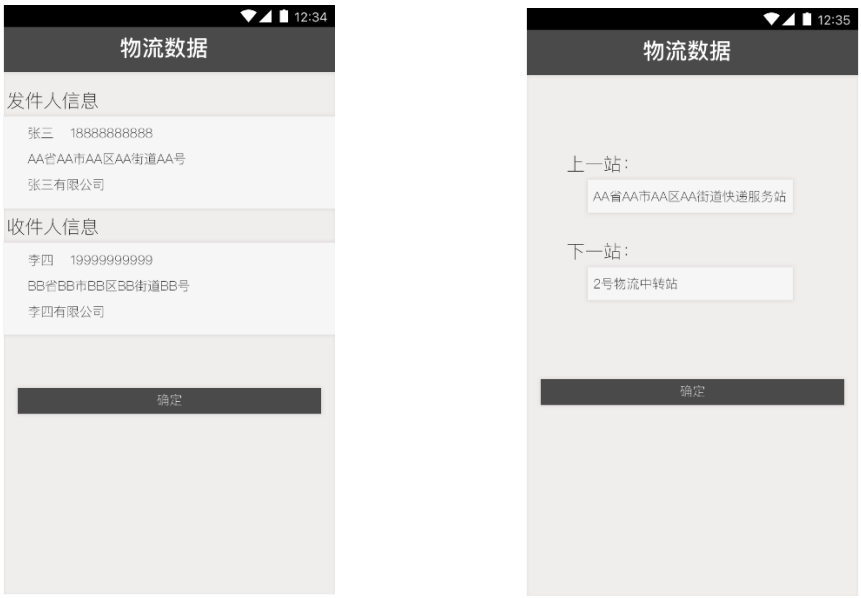
图 5-14 (b)是发件人发起一个新物流过程的界面，发件人在该界面中填写好相关的物流隐私数据之后，点击“提交订单”按钮，发起一个新的物流过程，最后会生成该物流的编号和二维码供其他人访问该物流隐私数据。在生成一个新的物流过程时，物流隐私数据会被分层加密存储到云存储平台，并且该物流隐私数据对应的数据权限状态也会被

添加到区块链的分布式账本中。

图 5-14 (c)是查看寄件记录界面，发件人和收件人可以在该页面中查看自己的寄件历史。每一个物流编号下都有“点击查询物流状态”、“访问权限管理”按钮，提供了查看物流状态和对物流隐私数据进行访问权限管理的功能。

图 5-14 (d)是查看物流状态界面，发件人和收件人可以通过扫描二维码或者点击“点击查询物流状态”按钮来查看物流状态。物流状态数据是从区块链分布式账本中的 State 数据库中取得的。

图 5-14 (e)是发件人对物流隐私数据进行访问权限管理的界面，发件人在该界面中可以对收件人、快递员和中转人员的访问权限进行管理。访问权限主要包括了：访问状态、访问日期限制和访问次数限制，当发件人更改这些访问权限的时候，会将相应的权限变化更新到区块链分布式账本的 State 数据库中。



(a) 快递员获得的物流数据　　(b) 中转人员获得的物流数据

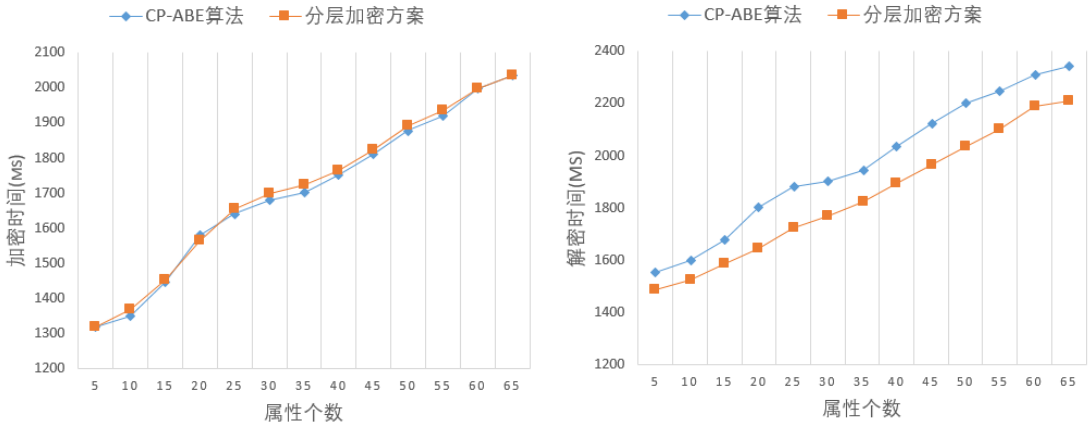
图 5-15 物流工作人员操作界面

图 5-15(a)是快递员通过扫码获得的物流隐私数据。快递员能够获得的物流隐私数据包括：发件人和收件人的信息以及物流路径信息（在这里界面中没有显示处物流路径信息）。当快递员扫码的时候，首先会调用链码 readUserESK 和 readStoragePath 方法从区块链分布式账本的 State 数据库中取得物流隐私数据密文数据包存储的路径和e_SK的存储路径；然后将密文数据包和e_SK下载到本地，接着进行密文数据包的解密操作，解密出快递员对应部分的物流隐私数据。快递员点击“确定”按钮会将快递员的业务操作结果通过调用链码 addLstatus 更新到区块链分布式账本的 State 数据库中。

图 5-15 (b)中转人员通过扫码获得的物流隐私数据。中转人员获得的隐私数据只有物流路径信息，界面中会根据中转人员所在的中转站显示出上一站和下一站信息。物流路径信息的获取过程和快递员获取物流隐私数据的过程是一样的，当中转人员点击“确定”按钮时，会将中转人员业务操作的结果通过调用链码 addLstatus 更新到 State 数据库中。

5.4.3 系统性能测试

在物流下单原型系统中，物流用户隐私数据的加解密时间是一个重要的性能指标。物流用户隐私数据的加解密时间又由分层加密模块加解密数据的时间以及区块链模块请求响应的时间决定，下面分别对分层加密模块加解密时间和区块链模块请求响应的时间进行测试。



(a) 加密数据时间比较　　(b) 解密数据时间比较

图 5-16 基于属性的分层加密方案与 CP-ABE 算法加解密时间的比较

在本文中，我们通过改进的 CP-ABE 算法实现一种基于属性的分层加密方案；该分层加密方案的具体描述在本文的 3.3 节中已经给出。当使用分层加密方案加密单个文件的时候，该算法可以看作一个传统的 CP-ABE 算法；但是当使用该分层加密方案加密多个文件时，该分层加密方案在数据解密的步骤中拥有比 CP-ABE 算法更高的效率，图 5-16 显示了在加解密三个文件时，本文提出的基于属性的分层加密方案与 CP-ABE 算法的加解密时间的比较，横坐标是属性个数。从图中可以得出结论，本文提出的分层加密方案在加密多个文件时的效率与 CP-ABE 算法的效率相当，但是在解密多个文件时，基于属性的分层加密方案的效率要比 CP-ABE 算法的效率更优。

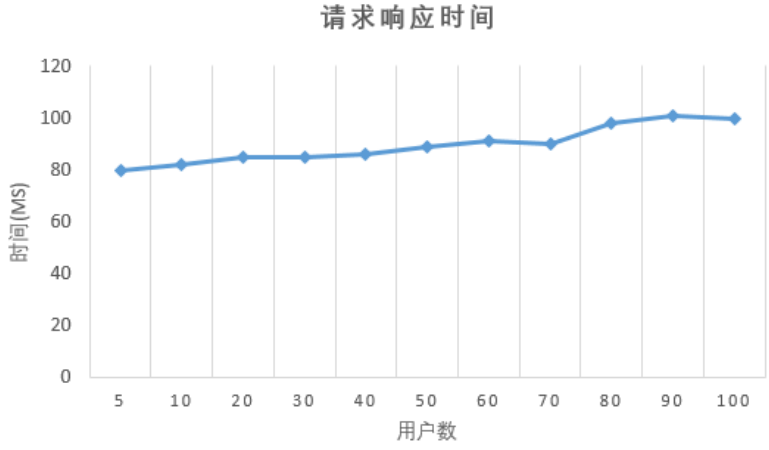


图 5-17 区块链模块请求响应时间

用户通过客户端程序向区块链模块发送一个交易，调用相应的链码方法，链码执行之后返回相应的结果，我们在这里检测整个过程随着用户数量的变化所用时间的变化，结果如图 5-17 所示。由图可以看出，在用户数从 5 到 100 变化的过程中，区块链模块的响应时间也随着变化，但一直保持在 80ms 到 100ms 区间中，这样的响应时间是可以满足在系统中应用的。

5.5 本章小结

本章主要介绍了物流下单原型系统的设计和实现。系统由数据拥有方、数据访问方以及云存储平台组成。该系统保障了物流用户隐私数据的安全性，提供了数据拥有方对物流隐私数据访问权限的控制管理，实现了数据访问方对隐私数据的分层访问。经测试表明，该系统在保证隐私数据安全的同时，有较高的计算效率，用户与区块链模块的交互延时保持在 80-100ms 区间内，有较高的使用价值。

第六章 总结与展望

6.1 总结

随着互联网的发展，物流行业在生活中扮演的角色越来越重要，物流用户隐私保护有着十分重大的意义。本文在分析了现有物流信息系统以及物流用户隐私保护研究成果及其存在问题的基础上，提出了一种基于区块链的物流用户隐私数据保护方案，将隐私数据分层加密云存储和去中心化的访问权限控制管理相结合。在此基础上设计和实现了一物流下单原型系统，验证了该隐私数据保护方案的有效性。本文主要工作内容可以总结为如下几点：

1. 在研究和分析现有隐私数据加密存储安全机制，以及数据访问权限控制管理方案的基础上，结合分层加密和区块链技术提出一种新型的物流用户隐私数据保护方案。该方案将隐私数据进行分层加密处理上传至云端存储，以解决隐私数据在不可信第三方平台上的安全存储问题；同时提供去中心化隐私数据访问权限管理，实现用户对隐私数据访问权限的控制管理。
2. 提出了一种基于分层加密的隐私数据保护机制。该机制采用基于属性的分层加密方案，通过安全属性表述数据访问方的身份并确定可读取的数据内容范围，将密文上传至云存储平台保管。该机制可有效防止物流用户隐私数据在不可信第三方平台的泄露。
3. 提出了一种改进的 CP-ABE 算法实现基于属性的分层加密方案。该加密方案采用嵌套访问控制树结构，用户可根据数据访问方属性确定其数据访问权限。拥有较高权限级别的数据访问方可以直接访问低级别的数据，无需重复解密，数据读取效率高。该方案有效的解决了物流派发时用户隐私数据访问方身份无法预先获知从而难以授权的问题。
4. 提出了一种基于区块链和 DAA 匿名认证相结合的访问权限管理机制。基于 DAA 匿名认证实现成员身份管理，为区块链上的实体提供匿名可验证的身份；通过维护一个交易公钥列表 TPL 来实现对区块链节点的访问权限控制，解决当前区块链技术存在的隐私泄露风险；同时，该机制通过分布式账本保存用户对隐私数据的访问权限，使用链码（即智能合约）来封装物流中各角色与隐私数据之间存在的业务逻辑，实现了用户对隐私数据的访问权限控制管理。该机制有效的解决了现有隐私数据访问权限管理方案中对隐私数据访问权限的管理难以做到用户可自主控制、授权过程可追溯以及访问记录可审计的问题。
5. 设计并实现了一个物流下单原型验证系统，该系统由数据拥有方、数据访问方及云存储平台组成，保障了物流用户隐私数据的安全性，提供了数据拥有方对物流隐私数据的访问权限控制管理，实现了数据访问方对隐私数据的分层访问。经测试表明，该系统在保证隐私数据安全的同时，有较高的计算效率，用户与区块链模块的交互

延时保持在 80-100ms 区间内，有较高的使用价值。

6.2 展望

本文对物流用户隐私保护进行了研究，提出了基于分层加密的隐私数据保护机制和基于区块链和 DAA 匿名认证相结合的访问权限管理机制，并基于此设计并实现了一个物流下单原型验证系统，测试证明了该系统能够保证物流隐私数据的安全，并提供了用户对隐私数据访问权限的控制管理。回顾本文的主要研究内容，仍然存在改进的空间，现将一些反思和改进思路列出，以为后续的拓展研究提供参考：

1. 对基于属性的分层加密方案作进一步的研究。本文通过改进的 CP-ABE 算法实现了基于属性的分层加密方案，实现了物流中各角色对隐私数据的分层访问，并且拥有更高的解密效率，接下来，我们将继续研究如何引入其他的技术，利用 CP-ABE 的天然优势，在保证物流用户隐私数据安全的前提下，进一步的提高分层加密方案的效率，使之更加适用于物流场景之中。
2. 对访问权限管理机制作进一步研究。本文提出的访问权限管理机制保证用户可以通过访问时间、访问次数、解密私钥存储路径以及密文数据包存储路径的分配来控制数据访问方对隐私数据的访问，虽然实现了基本的访问权限管理，但是缺乏更加完善的访问策略，下一步的目标是制定一个更加完善的访问策略帮助用户更方便的对物流隐私数据的访问权限进行控制管理。
3. 建立物流下单原型系统通用的性能评估标准。本文从分层加密模块加解密数据的时间以及区块链模块请求响应的时间对物流下单原型系统的性能进行一定的评估，但是该评估标准依旧不全面，并缺乏一定的理论基础，更全面、更具科学性的性能评估标准有待建立。
4. 完成其他平台的物流下单原型系统应用程序开发。本文在 Android 平台实现了物流下单原型系统应用程序的开发，并没有在其他平台，例如 iOS、Windows Phone、Windows 等，实现相应应用程序的开发。因此下一步的目标是在其他平台上完成物流下单原型系统的研究与开发工作。

致谢

三年的研究生学习生涯马上就要结束了，总的来说自己在这两年半的时间内收获和成长了很多，也认识了自身存在的一些缺点。在论文即将完成之际，我想对所有关心、帮助、批评过我的老师、同学和朋友表达最诚挚的感谢之情。

首先我要衷心的感谢我的导师宋宇波老师。宋老师对学术严瑾的态度、工作的热情以及对生活积极向上的态度深深的激励和感染着我们。每当在项目或者论文遇到瓶颈的时候，宋老师总会不断的鼓励我们，帮助我们开阔学术的视野，给我们提出极具启发性的指导意见。能在宋老师的指导下度过研究生生涯让我感到特别的幸运，在此，我要向宋老师致以最真挚的敬意和感谢！

然后，我要感谢我最亲爱的父母，感谢他们为我营造了一个幸福的家庭，谢谢你们二十多年来对我无微不至的爱，在今后的生活里，我会一直陪伴在你们的身边。感谢我的女友，谢谢你一直在我身边为我排忧解难。

感谢我可爱的室友纪策、李逸之、陈迪以及冯连森，谢谢你们这几年的关心和照顾，愿大家在以后的生活里快快乐乐，少点烦恼！

感谢所有的同学以及朋友们，两年半的时间，一起科研，一起玩乐，一起进步，有你们真好。希望今后的日子里，大家一切顺利，幸福快乐！

感谢师弟黄强、武威、魏一鸣、石伟、李轩、宋睿，谢谢你们的陪伴，愿你们心想事成！

感谢东大。百载文枢江左，东南辈出英豪，祝福您！

最后，非常感谢各位评审老师在百忙之中抽出时间审阅我的论文，祝您们工作顺利。

参考文献

[1] 王志玲. 从大数据时代看电子商务物流配送发展趋势[J]. 品牌研究, 2015(4):27-28.

[2] 赵宇飞. 快递单信息买卖形成“灰色产业链”[J]. 法制与经济旬刊, 2013(10).

[3] 段沛佑, 曲延旭, 马晓宁. 我国快递行业安全体系建设研究探讨[J]. 物流科技, 2012(6):110-112.

[4] 潘尤兴. 电子商务O2O中条码技术的应用——以京东商城为例[J]. 中国物流与采购, 2015(13):74-75.

[5] 牛东来. 现代物流信息系统[J]. 2011.

[6] Helo P, Szekely B. Logistics information systems: An analysis of software solutions for supply chain co-ordination[A]. In, 2005[C].

[7] Rai A, Pavlou P A, Im G, et al. Interfirm IT capability profiles and communications for cocreating relational value: evidence from the logistics industry[J]. Mis Quarterly, 2012, 36(1):233-262.

[8] Kahraman C, Ateş N Y, Çevik S, et al. Hierarchical fuzzy TOPSIS model for selection among logistics information technologies[J]. Journal of Enterprise Information Management, 2007, 20(2):143-168.

[9] Hazen B T, Byrd T A. Toward creating competitive advantage with logistics information technology[J]. International Journal of Physical Distribution & Logistics Management, 2012, 42(1):8-35.

[10] Wood L C, Reiners T, Pahl J. Manufacturing and Logistics Information Systems[J]. Journal of Computational Neuroscience, 2015, 16(3):251-256.

[11] 张彤. 基于SOA-BPM组合架构的第三方物流信息系统研究[J]. 物流技术, 2012, 31(23):419-422.

[12] 徐斌, 潘瑞林, 暴伟. 基于OPC XML的物流自动化系统集成研究[J]. 制造业自动化, 2015(8):37-40.

[13] Berghel H. Identity theft, social security numbers, and the Web[J]. Communications of the Acm, 2000, 43(2):17-21.

[14] 郭宗杰. 邮政普遍服务法律问题研究[J]. 暨南学报(哲学社会科学版), 2012, 34(10):95-100.

[15] 韦茜, 李星毅. 基于K-匿名的快递信息隐私保护应用[J]. 计算机应用研究, 2014, 31(2):555-557.

[16] 韦茜, 王晨, 李星毅. 基于RSA算法的快递信息隐私保护应用[J]. 电子技术应用, 2014, 40(7):58-60.

[17] Zhang X, Li H, Yang Y, et al. LIPPS: Logistics Information Privacy Protection System Based on Encrypted QR Code[A]. In: Trustcom/bigdatase/ispa, 2017[C].

[18] 百度百科. 访问控制[EB/OL]. https://baike.baidu.com/item/%E8%AE%BF%E9%97%AE%E6%8E%A7%E5%88%B6/8545517?fr=aladdin, 2017/2018-3-20.

[19] Ausanka-Crues R. Methods for Access Control: Advances and Limitations[EB/OL]. https://www.cs.hmc.edu/~mike/public_html/courses/security/s06/projects/ryan.pdf, 2017/2018-3-10.

[20] Zyskind G, Nathan O, Pentland A S. Decentralizing Privacy: Using Blockchain to Protect Personal Data[A]. In: IEEE Security and Privacy Workshops, 2015[C].

[21] Xiao Y, Wang H, Jin D, et al. Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control[J]. Journal of Medical Systems, 2016, 40(10):218.

[22] Azaria A, Ekblaw A, Vieira T, et al. MedRec: Using Blockchain for Medical Data Access and Permission Management[A]. In: International Conference on Open and Big Data, 2016[C].

[23] Ouaddah A, Abou Elkalam A, Ait Ouahman A. FairAccess: a new Blockchain-based access control framework for the Internet of Things[J]. Security & Communication Networks, 2016, 9.

[24] Ouaddah A, Elkalam A A, Ouahman A A. Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT[J]. 2017.

[25] 杨波. 现代密码学[M].清华大学出版社, 2015: 65-69.

[26] Bellare M, Ran C, Krawczyk H. Keying Hash Functions for Message Authentication[A]. In: International Cryptology Conference on Advances in Cryptology, 1996[C].

[27] Sahai A, Waters B. Fuzzy Identity-Based Encryption[A]. In: International Conference on Theory and Applications of Cryptographic Techniques, 2005[C].

[28] Shamir, Adi. Identity-based cryptosystems and signature schemes[J]. Lect.notes Comput.sci, 1985, 196(2):47-53.

[29] Bethencourt J, Sahai A, Waters B. Ciphertext-Policy Attribute-Based Encryption[A]. In: IEEE Symposium on Security and Privacy, 2007[C].

[30] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data[A]. In: ACM Conference on Computer and Communications Security, 2006[C].

[31] Sahai A, Waters B. Fuzzy Identity-Based Encryption[A]. In: International Conference on Theory and Applications of Cryptographic Techniques, 2005[C].

[32] Goyal V, Jain A. Bounded Ciphertext Policy Attribute Based Encryption[A]. In: International Colloquium on Automata, Languages, and Programming, 2016[C].

[33] Beimel A. Secure Schemes for Secret Sharing and Key Distribution[J]. International Journal of Pure & Applied Mathematics, 1996.

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

[35] Wikipedia. Blockchain[EB/OL]. http://en.wikipedia.org/wiki/Blockchain, 2015/2018-3-19.

[36] 蒋润祥, 魏长江. 区块链的应用进展与价值探讨[J]. 甘肃金融, 2016(2):19-21.

[37] Pilkington M. Blockchain Technology: Principles and Applications[J]. Social Science Electronic Publishing, 2016.

[38] Papadopoulos G. Chapter 7 – Blockchain and Digital Payments : An Institutionalist Analysis of Cryptocurrencies[J]. Handbook of Digital Currency, 2015:153-172.

[39] Paper W. Ethereum: A next-generation smart contract and decentralized application platform[EB/OL]. https://github.com/ethereum/wiki/wiki/White-Paper, 2017/2018-3-20.

[40] White H. Hyperledger Project[EB/OL]. https://github.com/hyperledger/, 2017/2018-3-19.

[41] Zhou S G, Feng L I, Tao Y F, et al. Privacy Preservation in Database Applications: A Survey: Privacy Preservation in Database Applications: A Survey[J]. Chinese Journal of Computers, 2009, 32(5):847-861.

[42] Szabo N. Formalizing and Securing Relationships on Public Networks[J]. 1997, 2(9).

[43] Wikipedia. 图灵完备性[EB/OL]. https://zh.wikipedia.org/zh-cn/圖靈完備性, 2015/2018-3-11.

[44] Brickell E. Direct anonymous attestation[A]. In: ACM Conference on Computer and Communications Security, 2004[C].

[45] Chaum D, Van Heyst E. Group signatures[A]. In: The Workshop on the Theory and Application of of Cryptographic Techniques, 1991[C].

[46] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof-systems[A]. In, 1985[C].

[47]Camenisch J, Groth J. Group Signatures: Better Efficiency and New Theoretical Aspects[J]. Scn, 2004, 10(1):55-63.

[48] Shoup V. A proposal for the ISO standard for public-key encryption (version 2.1)[J]. Iacr E, 2001.

[49] 王红珍, 张根耀, 李竹林. AES算法及安全性研究[J]. 信息技术, 2011(9):20-22.

[50] 陈原. 公钥加密与混合加密的可证明安全性研究[D]. 西安电子科技大学, 2006:

[51] Bellare M, Canetti R, Krawczyk H. Keying hash function for message authentication[J]. Advances in Cryptology-CRYPTO'96, 1996, 1109:1-15.

作者简介

张克落（1993—），男，汉族，安徽滁州人，现为东南大学信息安全研究中心硕士研究生，研究方向为用户隐私数据保护。

- **攻读硕士学位期间发表的论文**

[1] 张克落. 比特币钱包取证技术的研究[C]. 东南大学校庆研究生学术报告会，2017.12

- **参与的科研项目**

[1] 2016.07-2016.11，南京 841 研究所项目，“比特币钱包取证技术的研究”