基于区块链的物流信息隐私保护*

赵灵奇[1]，宋宇波[1]，张克落[1]，胡爱群[1]，罗坚[2,3]

[1](东南大学 网络空间安全学院学院,江苏 南京 211111)

[2](南京网络空间安全技术研究院,江苏 南京 211111)

[2](浙江工贸职业技术学院,浙江 温州 325003)

通讯作者：宋宇波, E-mail: songyubo@seu.edu.cn

摘 要:在当代物流业中，用户往往缺乏对物流信息数据的控制权，用户隐私数据通常由第三方保管，这极大增加了用户隐私数据泄露的可能。针对上述问题，本文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。该方案提出利用基于区块链和 DAA 匿名认证相结合的访问权限管理机制，实现用户对隐私数据访问权限的控制管理，以及隐私数据访问记录的可追溯性。此外该方案对隐私数据采用改进的 CP-ABE 算法，实现基于属性的分层加密，通过嵌套访问控制树结构实现用户根据数据访问方属性确定其隐私数据的访问权限。最后，本文设计并实现了一个物流下单原型验证系统，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，可以满足实际物流系统的需求。

关键词： 物流隐私保护，CP-ABE，区块链，访问权限管理

中图法分类号：TP311

Logistics Information Privacy Protection Based on Blockchain

ZHAO Ling-Qi[1], SONG Yu-Bo [1], ZHANG Ke-Luo[1], HU Ai-Qun[1], LUO Jian[2,3]

[1](School of Cyberspace Security, Southeast University, Nanjing, Jiangsu 211111, China)

[2](Nanjing Institute of Cyberspace Security Technology, Nanjing, Jiangsu 211111, China)

[3](Zhejiang Industry and Trade Vocational and Technical College, Wenzhou, Zhejiang 325003, China)

Abstract:In the contemporary logistics industry, users often lack control over logistics information data. User privacy data is usually kept by third parties, which greatly increases the possibility of user privacy data leakage. Aiming at the above problems, this paper proposes a new type of logistics user privacy data protection scheme based on layered encryption and blockchain technology. The scheme proposes to use the access rights management mechanism based on blockchain and DAA anonymous authentication to realize the user's control and management of privacy data access rights and the traceability of private data access records. In addition, the scheme adopts the improved CP-ABE algorithm for privacy data to implement attribute-based hierarchical encryption. The nested access control tree structure enables users to determine the access rights of their private data according to the attributes of the data accessor. Finally, this paper designs and implements a logistics order proofing verification system. The test shows that the system has high computational efficiency while ensuring the security of private data. The interaction delay between the user and the blockchain module is kept at 80- Within the 100ms interval, it can meet the needs of the actual logistics system.

Key words: Logistics privacy protection, CP-ABE, blockchain, access rights managemen

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。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[1]。

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

基金项目：国家自然科学基金（61601113）；

Fundation item: National Natural Science Foundation of China (61601113);

法得到保护。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[2]，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[3]。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。

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，设计并实现了一个物流下单原型验证系统，并对该系统的性能和安全性进行了测试和评估。本文具体的贡献如下：

1. 针对用户缺乏对物流隐私数据控制权的问题，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。
2. 针对物流用户隐私数据易被泄露和窃取的问题，本文提出了一种基于分层加密的隐私数据访问机制，该机制采用改进的 CP-ABE 算法实现基于属性的分层加密方案，可通过嵌套访问控制树结构实现用户根据数据访问方属性确定其隐私数据的访问权限。同时将隐私数据密文上传至云存储平台保管，有效防止物流用户隐私数据在不可信第三方平台的泄露。

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。该系统实现了数据访问方和数据拥有方的身份管理，提供了数据拥有方对物流隐私数据的访问权限控制，保证了物流隐私数据安全性、完整性以及数据访问方对隐私数据的分层访问。并对系统性能以及隐私数据保护安全性进行测试和评估。

1 相关工作

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。对这两个领域，国内外也对其有相应的研究。

在物流隐私数据保护领域，在国内韦茜等人[4]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。文献[5]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[6]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，现有的研究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。

在区块链访问控制研究领域，Guy Zyskind 等人[7]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。肖月等人[8]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。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。

2 许可链

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[9]。根据分类，区块链可分为许可链和非许可链。本文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，许可链并不对外公开，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。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。

超级账本（Hyperledger）[10]就属于许可链的架构。它是由 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。本文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。

3 用户隐私数据保护方案

通过上文对现今物流系统的分析，我们可以发现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) 用户既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。2) 用户隐私数据通常由物流企业类的第三方进行保管，但既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。

我们知道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，各个角色所需要的物流用户隐私数据也都不同。因此本文考虑到了使用基于分层加密的隐私数据访问机制，采用改进的 CP-ABE 算法实现的基于属性的分层加密方案，通过安全属性表述数据访问方身份，可通过嵌套访问控制树结构实现用户根据数据访问方属性确定其隐私数据的访问权限，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，确定了数据访问方可读取隐私数据的范围，同时将隐私数据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,实现了数据访问方对物流用户隐私数据的分级访问，从而不会获得多余的隐私数据。但是该方法并不能对隐私数据的访问权限进行管理，现有的访问权限管理方案中存在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。

针对现有隐私数据访问权限管理方案存在的问题，本文结合区块链公开透明、无法篡改、方便追溯等特点，以及相较于非许可链来说，许可链吞吐量大以及交易验证较快的优点，提出了基于区块链和 DAA 匿名认证的访问权限管理机制。

具体方案的整体架构图如图 1 所示。主要可以分为成员身份管理、访问权限管理以及隐私数据访问三个部分。成员身份管理部分是基于 DAA 匿名认证实现的，为数据访问方和数据拥有方提供匿名可验证的身份，并通过交易公钥列表 TPL 实现对区块链节点的访问控制；访问权限管理部分是基于区块链实现的，实现了数据拥有方对物流隐私数据访问权限的控制，并记录下数据访问方的访问记录；隐私数据访问是基于分层加密实现的，为物流用户隐私数据提供安全性、完整性保证，并实现数据访问方对隐私数据的分层访问。

3.1 成员身份管理

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。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。

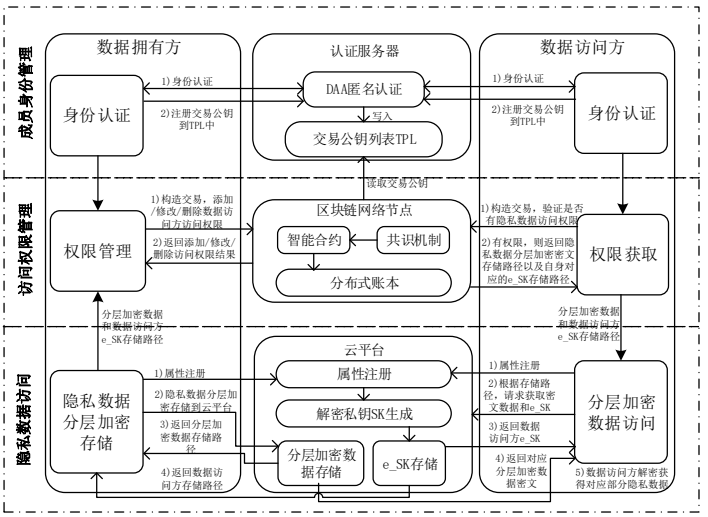


图 1 基于区块链的物流用户隐私数据保护方案整体架构图

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[11]。

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 $v'_1 \in_R \{0,1\}^{l_v}$ 和 $v'_2 \in_R \{0,1\}^{l_n + l_\varnothing - l_v}$ 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 $r_{f_0}, r_{f_1} \in_R \{0,1\}^{l_f + l_\varnothing + l_H}$，$r_{v_1'} \in_R \{0,1\}^{l_s}$ 和 $r_{v_2'} \in_R \{0,1\}^{l_n + 2l_\varnothing + l_H - l_s}$，根据公式 $\tilde{U} = R_0^{r_{f_0}} R_1^{r_{f_1}} S^{r_{v_1'}} S^{r_{v_2'}} \bmod n$，$\tilde{N}_I = \varsigma_I^{r_{f_0} + r_{f_1} 2^{l_f}} \bmod \Gamma$ 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。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。

IV. 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。

V. Issuer 通过公式 $\hat{U} = U^{-c} R_0^{s_{f_0}} R_1^{s_{f_1}} S^{s_{v'}} \bmod n$ 和 $\hat{N}_I = N_I^{-c} \varsigma_I^{s_{f_0} + 2^{l_f} s_{f_1}} \bmod \Gamma$ 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 $v \in_R \{0,1\}^{l_v - 1}$ 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 $v'' = v + 2^{l_{v'} - 1}$ 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 $v_1 = LSB_{l_s}(v_1'' + v_1')$，$\tilde{v}_2 = CAR_{l_s}(v_1'' + v_1')$，$v_2 = v_2'' + v_2' + \tilde{v}_2$，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 $\tilde{T}_{1t} = R_0^{r_{f_0}} R_1^{r_{f_1}} S^{r_v} \bmod n$，$\tilde{r}_f = r_{f_0} + r_{f_1}^{2^{l_f}} \bmod \rho$，$\tilde{N}_V = \varsigma^{\tilde{r}_f} \bmod \Gamma$。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 $\tilde{T}_1 = \tilde{T}_{1t} T_1^{r_e} h^{-r_{ew}} \bmod n$，$\tilde{T}_2 = g^{r_w} h^{r_e} g'^{r_r} \bmod n$，$\tilde{T}_2' = T_2^{-r_e} g^{r_{ew}} h^{r_{ew}} g'^{r_{er}} \bmod n$ 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

$\hat{T}_1 = Z^{-c} T_1^{s_e + c2^{l_e-1}} R_0^{s_{f0}} R_1^{s_{f1}} S^{s_r} h^{-s_{ew}} \bmod n$ ， $\hat{T}_2 = T_2^{-c} g^{s_w} h^{s_e + c2^{l_e-1}} g'^{s_r} \bmod n$ $\hat{T}_2' = T_2^{-(s_e + c2^{l_e-1})} g^{s_{ew}} h^{s_{er}} g'^{s_{er}} \bmod n$ ，$\hat{N}_V = N_V^c \varsigma^{s_{f0} + s_{f1} 2^{l_f}} \bmod \Gamma$ 计算得到 $\hat{T}_1$ 、 $\hat{T}_2$ 、 $\hat{T}_2'$ 和 $\hat{N}_V$ 。

II. 验证 $c = \mathrm{H}(\mathrm{H}(\mathrm{H}(n \| g \| g' \| h \| R_0 \| R_1 \| S \| Z \| \gamma \| \Gamma \| \rho) \| \varsigma \| (T_1 \| T_2) \| N_V \| (\tilde{T}_1 \| \tilde{T}_2 \| \tilde{T}_2') \| \tilde{N}_V) \| n_v) \| n_t) \| b \| m)$ 以及 $N_{V,\varsigma} \in \langle \gamma \rangle$， $s_{f0}, s_{f1} \in \{0,1\}^{l_f + l_\varnothing + l_H + 1}$， $\varsigma \equiv (\mathrm{H}_\Gamma(1 \| bsn_V))^{(\Gamma-1)/\rho} \bmod \Gamma$ 以及 $s_e \in \{0,1\}^{l_e + l_\varnothing + l_H + 1}$ 。

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，共识机制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。

3.2 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，同时通过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，实现隐私数据访问记录的可追溯性。具体的框架如图 2 所示

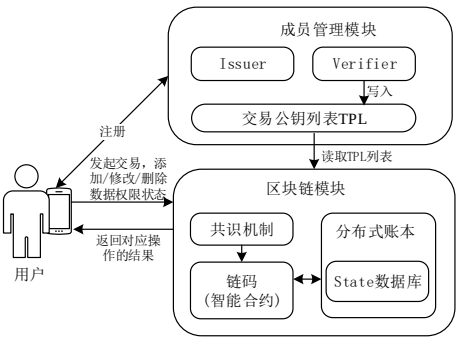


图 2 基于区块链和 DAA 匿名认证的访问权限管理机制整体架构

其中区块链模块是基于 Hyperledger Fabric 建立的。模块由共识机制、分布式账本、链码组成。共识机制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，共识机制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。

区块链模块处理用户在区块链网络中产生的交易，节点中的共识机制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的设计方法。

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 3 所示。

如图 3 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

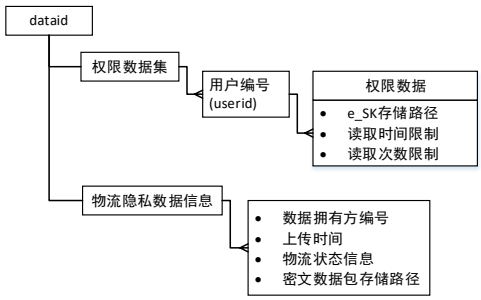


图 3 State 数据库数据权限状态的数据结构

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。

3.3 隐私数据访问

基于分层加密的隐私数据访问机制的架构如图 4 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。

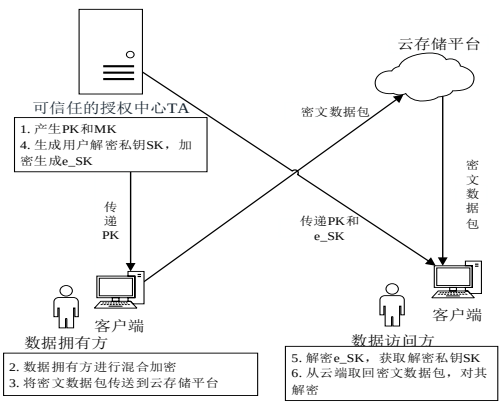


图 4 基于分层加密的隐私数据访问机制整体架构图

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。

在这个机制中的关键就是基于属性的加密，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。它由初始化、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。

本文将物流用户隐私数据按安全级别由低到高分为 *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 5 所示。

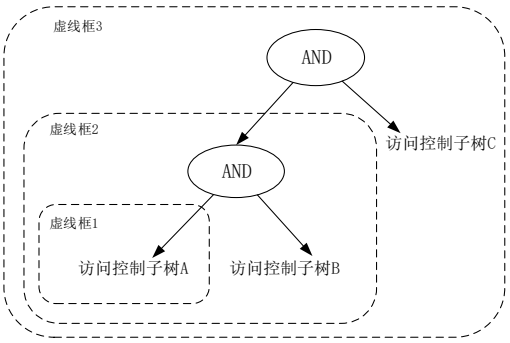


图 5 分层加密方案中访问控制树的嵌套结构

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 $\alpha,\beta \in Z_p$，利用乘法循环群 G_1、双线性对 e、生成元 g，然后根据公式(3.1)计算出公钥 PK：

$$PK = G_1, g, h = g^{\beta}, e(g,g)^{\alpha} \tag{2.1}$$

然后根据公式(3.2)计算出系统的主密钥 MK：

$$MK = (\beta, g^{\alpha}) \tag{2.2}$$

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 $r \in Z_p$，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 $r_j \in Z_p$。然后通过公式(3.3)计算出该数据访问方对应的解密私钥：

$$SK = (D = g^{(\alpha+r)/\beta}, \forall j \in \gamma : D_j = g^{r} \cdot H(j)^{r_j}, D_j^{'} = g^{r_j}) \tag{2.3}$$

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、m_3，在这里需要对 m_1、m_2、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 $s_3 \in Z_p$，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，$H(x)$ 表示一个哈希函数 $H:\{0,1\}^* \to G_0$，CT 代表最后生成的包含访问结构的密文：

$$CT_i = (T_i, \tilde{C}_i = key_i \cdot e(g,g)^{\alpha s_i}, C_i = h^{s_i}, \forall y_i \in Y_i : C_{y_i} = g^{q_{y_i}(0)}, C_{y_i}^{'} = H(att(y_i))^{q_{y_i}(0)}) \tag{2.4}$$

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 $CT_1 = (T_1, \tilde{C}_1, C_1, \forall y_1 \in Y_1 : C_{y_1}, C'_{y_1})$ 、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：

$$decryptNode(CT, SK, x) = \frac{e(D_j, C_x)}{e(D'_j, C'_x)} = \frac{e(g^r \cdot \mathrm{H(i)}^{r_j}, g^{q_x(0)})}{e(g^{r_j}, \mathrm{H(i)}^{q_x(0)})} = e(g,g)^{rq_x(0)} \tag{2.5}$$

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(2.6)计算 F_x：

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{j,S'_x}(0)} \\ &= \prod_{z \in S_x} (e(g,g)^{r \cdot q_z(0)})^{\Delta_{j,S'_x}(0)} \\ &= \prod_{z \in S_x} (e(g,g)^{r \cdot q_{parent(z)}(index(z))})^{\Delta_{j,S'_x}(0)} \\ &= \prod_{z \in S_x} e(g,g)^{r \cdot q_x(j) \Delta_{j,S'_x}(0)} \\ &= e(g,g)^{r \cdot q_x(0)} \end{aligned} \tag{2.6}$$

公式(2-6)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 Z_p 成员集。

$$\Delta_{j,S}(x) = \prod_{l \in S, l \neq i} \frac{x-l}{j-l}, j \in Z_p \tag{2.7}$$

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 $A_1 = decrptNode(CT_1, SK, R_1) = e(g,g)^{rq_{R_1}(0)} = e(g,g)^{rs_1}$。最后根据下面的公式(3.8)解密出对称密钥 key_1：

$$\tilde{C}_1 / (e(C_1, D) / A_1) = \tilde{C}_1 / (e(h^{s_1}, g^{(\alpha+r)/\beta}) / e(g,g)^{rs_1}) = key_1 \tag{2.8}$$

在计算出对称密钥 key_1 之后，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 $A_1 = e(g,g)^{rs_1}$ 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。如果数据访问方的属性集 γ 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 γ 满足 T_2 设定的访问结构，则得到 $A_2 = decrptNode(CT_2, SK, R_2) = e(g,g)^{rq_{R_2}(0)} = e(g,g)^{rs_2}$，从而根据公式(3.9)解密出对称密钥 key_2：

$$\tilde{C}_2 / (e(C_2, D) / A_2) = \tilde{C}_2 / (e(h^{s_2}, g^{(\alpha+r)/\beta}) / e(g,g)^{rs_2}) = key_2 \tag{2.9}$$

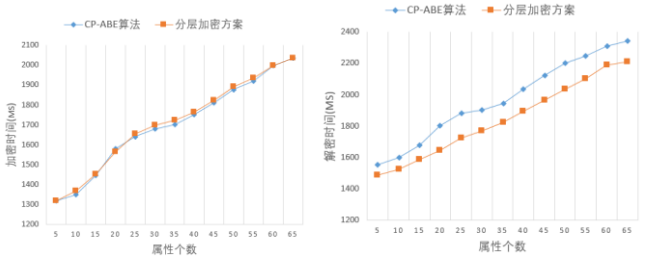
同理，通过得到的 A_2 继续判断属性集 γ 是否满足 T_3 设定的访问结构，不满足则停止解密，满足则进而根据公式(2.10)解密出对称密钥：

$$\tilde{C}_3 / (e(C_3, D) / A_3) = \tilde{C}_3 / (e(h^{s_3}, g^{(\alpha+r)/\beta}) / e(g,g)^{rs_3}) = key_3 \tag{2.10}$$

4 测试与评估

4.1 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 6 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。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 9 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。

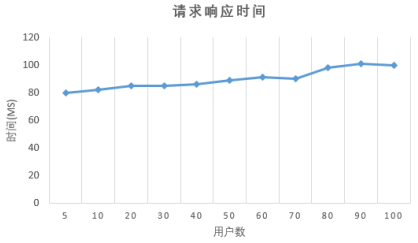


图 7 区块链模块请求响应时间

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 10 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。

4.2 安全性分析

1）成员身份管理安全性分析

在成员身份管理模块中，我们主要使用的是 DAA 匿名认证，它的安全性主要包括：隐私密钥的保密性、匿名性和签名不可伪造性。

隐私密钥的保密性：隐私密钥的安全性至关重要，如果隐私密钥发生了泄漏，整个 DAA 协议失去了可信性。考虑到 TPM 所在的平台主机不一 定可信，协议将可能会暴露隐私密钥的所有运算均由 TPM 完成。在整个方案中，TPM 始终以零知识证明的方式对外证明 TPM 拥有隐私密钥，确保了隐私密钥的绝对保密性。

匿名性：在签名协议阶段，Host 先将 Issuer 发送的信任证书 $(A,e,v^{"})$ 盲化，然后利用它将消息 m 生成知识签名 σ 发送给 Verifier，使得 Verifier 和 Issuer 之间不存在相同的消息，即便 Verifier 和 Issuer 合谋也无法识别出具体的 TPM，防止了 Verifier 和 Issuer 的合谋攻击，实现了平台的匿名性，保证了平台的隐私性。

签名不可伪造性：方案中签名的不可伪造性包含 2 层含义：一是加入协议中，发布者不能伪造身份证书；二是签名协议中，可信计算平台不能伪造签名。发布者在生成身份证书时使用了发布者的私钥，可信计算平

台收到发布者的信任证书后，用对应的公钥验证，进而确定发布者是否伪造了信任证书。可信计算平台收到发布者的信任证书后，对信任证书进行盲化并生成自己的签名，但签名验证协议需要向验证者零知识证明可信计算平台的确拥有发布者颁发的信任证书。因此，可信计算平台在安全假设下无法伪造一个合法的签名

2) 访问权限管理安全性分析：

在访问权限管理模块中主要就是区块链技术，而区块链的安全性主要依靠非对称密钥算法和哈希算法的安全性保证。另外区块链模块处理用户在区块链网络中产生的交易，节点中的共识机制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，因此在保证 DAA 匿名认证的安全性也就能保证交易公钥 *tran_pub* 的正确性。

3）隐私数据访问安全性分析

在隐私数据访问中，本文使用改进的 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[12]，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。

5 总结

本文提出了一种基于区块链的物流隐私数据保护方案。该方案首先利用改进的 CP-ABE 算法实现基于属性的分层加密，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，确定了数据访问方可读取隐私数据的范围，同时将隐私数据密文上传至云存储平台保管。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。其次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。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。最后设计并实现了一个物流下单原型验证系统，并对其进行了测试和评估。

参考文献:

[1] 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.

[2] 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.

[3] 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].

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

[5] 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.

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

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

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

[12] 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.