云辅助工业物联网中基于区块链的用户友好型数据检索和共享方案

张波[1,2]，李哲成[1]，徐兴帅[1]
（1.济南大学信息科学与工程学院，山东 济南 250022；2.山东省泛在智能计算重点实验室（筹），山东 济南 250022）

摘　要：为解决工业物联网中产生的数据安全检索和共享的挑战，以及在使用公钥加密索引时无法抵御离线关键词猜测攻击的问题，提出了一种适合工业物联网的基于区块链的用户友好型数据检索和共享方案。通过区块链和代理重加密设计了灵活、便捷的动态数据申请机制，允许数据用户对那些没有权限的数据请求授权，以合法获取数据，通过云服务器对数据进行预解密降低用户端计算开销，实现了具有细粒度访问控制和用户定义搜索精度的安全多关键词搜索。通过智能合约为用户生成临时查询ID，以及返回查询结果时随机选择用户转发结果，使攻击者无法将关键词与用户相关联，因此可以抵抗离线关键词猜测攻击。实验结果和安全性分析表明，所提方案具有更高的效率，并可以抵抗选择明文攻击、选择关键词攻击和离线关键词猜测攻击。

关键词：云辅助工业物联网；加密数据检索；代理重加密；属性基加密；数据共享；区块链
中图分类号：TP309
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025237

Blockchain-based user-friendly data retrieval and sharing scheme in cloud-assisted industrial Internet of things

ZHANG Bo[1,2], LI Zhecheng[1], XU Xingshuai[1]
1. School of Information Science and Engineering, University of Jinan, Jinan 250022, China
2. Shandong Key Laboratory of Ubiquitous Intelligent Computing, University of Jinan, Jinan 250022, China

Abstract: To address the challenges of secure data retrieval and sharing in the industrial Internet of things (IIoT), as well as the vulnerability to offline keyword guessing attacks when using public-key encrypted indexes, a blockchain-based user-friendly data retrieval and sharing scheme was proposed suitable for IIoT. The scheme leveraged blockchain and proxy re-encryption to design a flexible and convenient dynamic data request mechanism, allowing data users to request authorization for data that they do not have access to, thereby obtaining it legally. Cloud servers performed pre-decryption of data, reducing the computational overhead on user devices. The scheme achieved secure multi-keyword search with fine-grained access control and user-defined search precision. A smart contract was used to generate temporary query ID for users, and when returning searched results, additional users were randomly selected to forward the results, preventing attackers from associating keywords with users. As a result, the scheme can resist offline keyword guessing attacks. Experimental results and security analysis show that the proposed scheme offers higher efficiency and is resistant to chosen-plaintext attacks, chosen-keyword attacks, and offline keyword guessing attacks.

Keywords: cloud-assisted IIoT, encrypted data retrieval, proxy re-encryption, attribute-based encryption, data sharing, blockchain

收稿日期：2025–08–22；**修回日期：**2025–09–15
通信作者：张波，ise_zhangb@ujn.edu.cn
基金项目：山东省自然科学基金资助项目（No.ZR2022MF264）
Foundation Item：The Natural Science Foundation of Shandong Province (No.ZR2022MF264)

0　引言

随着云计算[1]和云辅助工业物联网的迅速发展，云服务器通常被视为部分可信的分布式基础设施。当数据所有者直接将未加密的工业传感器数据外包给云服务器时，敏感信息可能因云服务提供商的内部威胁或外部攻击而面临泄露风险[2]。为此，加密作为隐私保护的核心技术在学术界得到了广泛应用。

在各类加密方案中，具有细粒度访问控制特性的密文策略属性基加密（CP-ABE, ciphertext-policy attribute-based encryption）成为工业物联网数据共享的首选解决方案。具体而言，工厂管理员可以根据数据敏感性定义动态访问策略，并通过CP-ABE对设备生成的时间序列工业数据进行加密，然后上传至云端。该机制在确保数据机密性的同时，保证只有满足属性条件的授权实体才能解密并访问数据。

然而，现有解决方案在动态共享场景中面临可扩展性瓶颈：当新的协作者需要访问历史加密数据时，传统方法要求数据所有者对数据进行重新加密，导致计算开销随用户规模成线性增长。尽管一些研究尝试将解密密钥委托给云服务器以实现代理重加密，但该模型存在密钥泄露的风险。对此，代理重加密（PRE, proxy re-encryption）技术通过引入半可信的第三方代理，允许数据所有者生成针对特定用户的重加密密钥，使云服务器能够在不解密原始数据的情况下将密文转换为可由目标用户私钥解密的形式。这在降低本地计算负载的同时，保持了端到端的安全性。

在数据共享中，数据检索是一个关键步骤。明文搜索简单但不安全，在加密数据上进行搜索时，检索过程变得更加复杂，需要在检索效率与安全性、隐私性之间取得平衡。公钥可搜索加密为这一问题提供了解决方案。然而，文献[3]指出，公钥可搜索加密的传统方案面临基本的安全挑战：在使用公钥生成索引时，如果关键词空间未达到足够规模，系统容易受到离线关键词猜测攻击。区块链为此问题提供了解决方案。近期的研究结合了基于属性的加密、可搜索加密和区块链，以应对工业物联网中的数据共享问题[4]。

为解决工业物联网环境中传统公钥可搜索加密方案面临的离线关键词猜测攻击、动态权限管理效率低下和用户差异性需求等挑战，本文进行了以下工作。

1) 本文提出了一种基于区块链的、用户友好的云辅助工业物联网数据检索和共享方案。用户友好性体现在允许自定义搜索精度，利用云端密文预解密以减少用户解密负载，并提供一种便捷的请求数据访问权限的有效机制。该方案支持多关键词匹配，同时用户能够在本地解密过程中验证数据的完整性。

2) 本文提出了一种智能合约机制，该机制生成唯一的请求ID并将查询结果发送给另外两个随机用户以隐藏用户身份。这种方法防止攻击者将关键词猜测结果与特定用户关联，可以实现抵抗离线关键词猜测攻击。

3) 本文对所提方案进行了安全分析和性能评估。结果表明，该方案可以有效抵御选择明文攻击、选择关键词攻击和离线关键词猜测攻击，确保了数据的保密性、完整性和可验证性。此外，该方案在处理速度和资源消耗方面表现出色，满足实际应用需求。

1　相关工作

为增强安全和高效的细粒度数据访问控制，Sahai等[5]提出了基于属性的加密（ABE, attribute-based encryption），结合Shamir秘密共享和双线性映射，使用基于属性集的加密操作取代了传统的单钥加密。这项基础工作通过后续创新分为两个主要分支：Goyal等[6]提出了密钥策略ABE（KP-ABE, key-policy attribute-based encryption），实现了带有属性标签的密文绑定策略的秘密密钥，突破了传统意义上单一身份密钥的限制，但该方案中数据拥有者无法控制访问策略，缺乏灵活性；Bethencourt等[7]提出了基于密文策略的属性加密，使只有符合属性条件的密钥才能解密嵌入策略的密文。但在访问结构复杂度增加的情况下，密文规模和解密计算开销提升较大，严重影响效率。这两种方案针对不同的应用场景和安全需求提供了相应的解决方案和安全保证。在此基础上，一些工作[8-9]专注于提高基于属性的加密效率，而其他一些工作通过在完全安全模型下进行严格的安全性证明，增强了基于属性加密方案的安全性[10-11]。

现有方案中的数据共享面临一个挑战：当所有

者与新用户共享数据时，他们必须解密并重新加密密文，这非常烦琐。研究人员探索了数据共享和代理重加密技术以解决这一问题。Liang等[12]提出了一种结合基于属性的加密与代理重加密的方案。Sun等[13]引入了一种多接收者广播代理重加密方案，该方案允许为不同的接收者转换密文，同时保护接收者的身份免受恶意对手的攻击，但该方案未考虑到属性撤销与密钥更新的问题。Wang等[14]提出了一种跨域代理重加密方案，使委托人能够授权半可信的云服务器将基于身份的加密密文转换为嵌入访问结构基于属性的加密密文，并且方案通过线性熵扩展使属性数量增长时计算开销始终保持常量级。此外，一些研究通过跨策略的密文形式转换增强了方案的灵活性[15]，另一些研究扩展了基于属性加密中访问策略的表现[16]。

研究人员还专注于可搜索加密方案，以在保护关键词隐私的同时实现对加密数据的高效搜索。Zheng等[17]将基于属性的加密与加密关键词搜索相结合，使用布隆过滤器来实现高效的关键词成员检索且数据用户可以在不需要与数据拥有者交互的情况下生成陷门。但其要求代理对原始密文进行解密和重新加密以实现数据共享，引入了安全风险。Yang等[3]提出了一种基于云的数据检索解决方案，分析了用户兴趣和服务器行为。Cheng等[18]提出了一种服务器辅助的公钥验证加密与关键词搜索方案，通过涉及发送方和接收方服务器来降低运算成本，并且通过设计常量级大小的密文和陷门提高了可扩展性，但未考虑到多用户场景的需求。此外，一些研究进一步探索了可搜索加密对量子计算的抵抗能力[19-20]以及多关键词搜索的相关场景[21-22]。

鉴于区块链的去中心化和透明性特征，研究人员探索了基于区块链的数据搜索与共享。为降低用户端成本，Jiang等[23]设计了一种基于代理的协同搜索方案，通过区块链和智能合约实现了数据的存储和远程检索，并引入协同检索机制，即允许多用户合并查询请求来降低整体操作成本。Zhou等[24]提出了一种面向车载社交网络的轻量级区块链可搜索数据共享方案。Agyekum等[25]提出了一种基于身份的代理重加密方法，并将其与区块链结合，用于物联网（IoT）中的数据共享，但未考虑到用户友好性的问题。Zhang等[26]在数据共享方案中结合了基于属性的加密与关键词搜索，引入了一种时间锁定的智能合约支付和一种去中心化的验证机制，确保了可验证性和公平性。

2 系统模型和定义

2.1 系统架构

本文方案的系统架构如图1所示。方案主要包含6类实体，具体介绍如下。

1) 权威机构（AU, authority）：完全可信实体，负责系统主密钥与公共参数生成，并根据用户属性分发用户密钥。

2) 云查询服务器（CQS, cloud query server）：半可信实体，负责处理用户请求、预解密密文和重新加密数据。与通常将这些服务结合在一起的模型不同，本文方案将数据存储与查询服务分离，以减少云服务器的工作负载并提高处理速度。

3) 云存储服务器（CSS, cloud storage server）：半可信实体，数量相对云查询服务器来说较少，负责数据检索和存储加密数据。

4) 数据所有者（DO, data owner）：完全可信实体，管理云中加密数据的访问控制策略。物联网设备数据经工厂服务器执行数据加密与索引构建后上传云端，工厂管理员作为数据所有者管理这些数据。

5) 数据用户（DU, data user）：如监管机构、供应商或合作研究实验室等实体，通过向云查询服务器提交陷门和预解密密钥来访问存储在云中的密文。在访问授权数据时，数据用户被视为完全可信，即他们不会将明文数据泄露给未经授权的第三方，从而在授权使用范围内保持数据的完整性和保密性。

6) 联盟区块链（CBC, consortium blockchain）：负责通过智能合约转发数据用户的各类请求，确保半可信的云服务器无法获取数据用户的信息。

2.2 方案定义

定义1 方案中的算法详细定义如下。

$\mathrm{Setup}(\kappa,U)\rightarrow(\mathrm{Para},\mathrm{MK})$：由AU执行，输入安全参数$\kappa$和属性集$U$，输出系统的公共参数Para和主密钥MK。

$\mathrm{SKGen}(\mathrm{MK},\varphi)\rightarrow(\mathrm{sk})$：由AU执行，输入系统的主密钥MK和用户的属性集$\varphi$，输出用户的属性密钥sk。

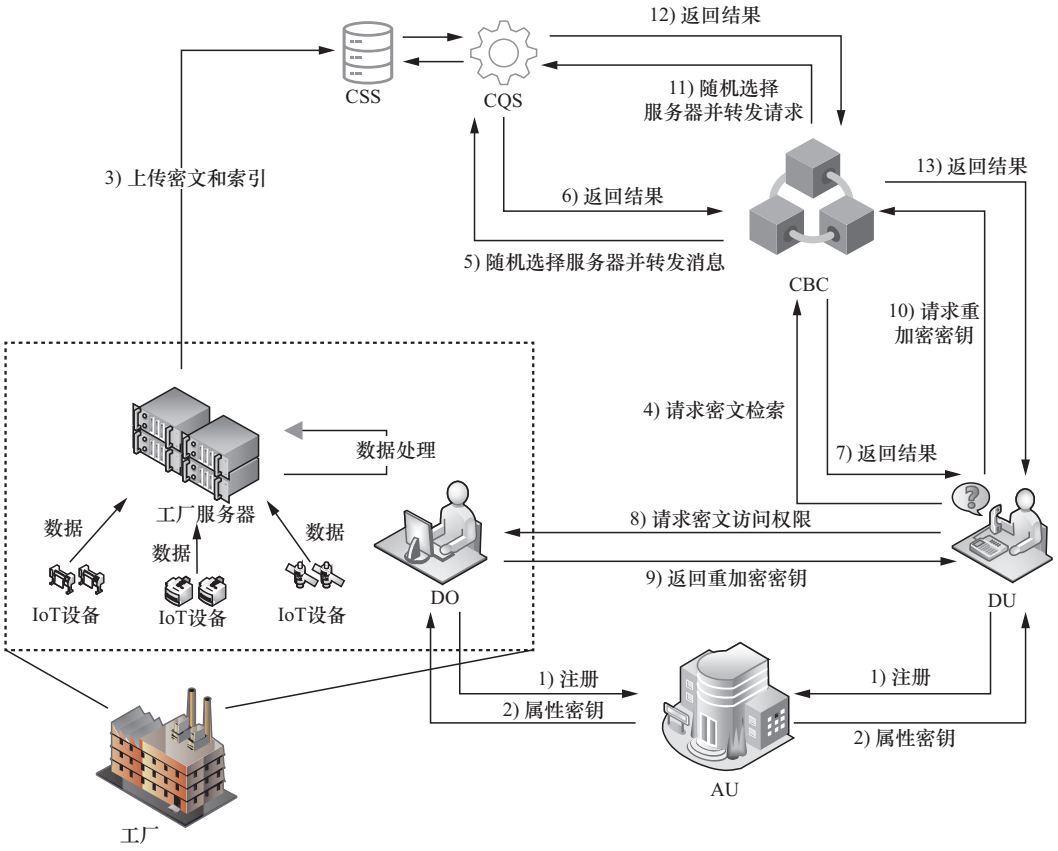

图1　系统架构

IndexGen$(\omega_n)\rightarrow(I,v_t,D)$：由数据所有者执行，输入与数据关联的一组关键词$\omega_n$，输出索引$I$、随机值$v_t$和$D$。

Encrypt$(m,\mathbb{R},v_t,D)\rightarrow$CT：由数据所有者执行，输入明文$m$、访问控制策略$\mathbb{R}=(M,\pi)$、随机值$v_t$和$D$，输出密文CT。

PreKeyGen$(\mathrm{sk})\rightarrow(\mathrm{PreK},\mathrm{DecK})$：由数据用户执行，输入用户的属性密钥sk，输出预解密密钥PreK和解密密钥DecK。

TDGen$(\mathrm{sk},\mathrm{DecK},\omega_{n_1})\rightarrow$TD：由数据用户执行，输入用户的属性密钥sk、解密密钥DecK和搜索关键词ω_{n_1}，输出对应的搜索陷门TD。

TDTest$(\mathrm{TD},I,t_u)\rightarrow(E,D)$或⊥：由云查询服务器执行，输入搜索陷门TD、数据索引I和匹配阈值t_u，输出匹配的数据(E,D)或⊥。

$\mathrm{PreDec}_{\mathrm{Or}}(\mathrm{PreK},\mathrm{CT},D,E)\rightarrow\mathrm{CT}_{\mathrm{OPreDec}}$或⊥：由云查询服务器执行，输入预解密密钥PreK、密文CT和匹配的数据(D,E)，输出预解密密文$\mathrm{CT}_{\mathrm{OPreDec}}$或⊥。

RKGen$(\mathrm{sk},\mathbb{R}')\rightarrow$rk：由数据所有者执行，输入所有者的属性密钥sk和修改后的访问策略$\mathbb{R}'=(M',\pi')$，输出重加密密钥rk。

ReEnc$(\mathrm{CT},\mathrm{rk})\rightarrow\mathrm{CT}'$：由云查询服务器执行，输入原始密文CT和重加密密钥rk，输出可被属性密钥满足访问策略$\mathbb{R}'=(M',\pi')$的用户解密的重加密密文CT′。

$\mathrm{PreDec}_{\mathrm{Re}}(\mathrm{PreK},\mathrm{CT}',\mathrm{rk})\rightarrow\mathrm{CT}_{\mathrm{RPreDec}}$或⊥：由云查询服务器执行，输入预解密密钥PreK、重加密密文CT′和重加密密钥rk，输出预解密密文$\mathrm{CT}_{\mathrm{RPreDec}}$或⊥。

$\mathrm{Dec}_1(\mathrm{CT}_{\mathrm{OPreDec}},\mathrm{DecK})\rightarrow m$：由数据用户或数据所有者执行，输入预解密后的密文$\mathrm{CT}_{\mathrm{OPreDec}}$和解密密钥DecK，输出明文$m$或⊥。

$\mathrm{Dec}_2(\mathrm{CT}_{\mathrm{RPreDec}'},\mathrm{DecK})\rightarrow m$：由数据用户执行，输入预解密后的重加密密文$\mathrm{CT}_{\mathrm{RPreDec}'}$和解密密钥DecK，输出明文$m$或⊥。

注：该方案省略了重加密密文的直接解密过程，数据用户和数据所有者可以通过$\mathrm{Dec}_1()$算法解密预解密的密文来获取明文。

为了评估该方案的安全性，给出如下定义。

定义2 如果满足定义3~定义5，则称该方案是安全的。

定义3 如果没有任何概率多项式时间敌手$\mathcal{A}$能够在游戏$\mathrm{GameC_O}$中以不可忽略的优势获胜，那么此方案中的原始密文满足选择明文攻击下的不可区分性。

1) $\mathrm{GameC_O}$

初始化 敌手$\mathcal{A}$首先选择挑战访问策略$\mathbb{R}^*=(M^*,\pi^*)$和挑战关键词ω^*，并发送给挑战者$\mathcal{C}$。

挑战者$\mathcal{C}$通过运行算法$\mathrm{Setup}(\kappa,U)$生成公共参数Para和主密钥MK，并将公共参数Para发送给敌手$\mathcal{A}$，主密钥MK保密。

查询阶段1 敌手$\mathcal{A}$将进行以下查询，挑战者$\mathcal{C}$将作出如下响应。

$\mathcal{O}_{\mathrm{sk}}(\varphi)$：敌手$\mathcal{A}$提供属性$\varphi$。若$\varphi$满足访问策略$\mathbb{R}^*$，则返回⊥（表示拒绝或无效）；否则，挑战者$\mathcal{C}$将运行算法$\mathrm{KeyGen}(\mathrm{Para},\varphi)$生成密钥sk并发送给$\mathcal{A}$。

$\mathcal{O}_{\mathrm{rk}}(\varphi,\hat{\mathbb{R}})$：敌手$\mathcal{A}$提供属性$\varphi$和访问策略$\hat{\mathbb{R}}=(\hat{M},\hat{\pi})$。若$\varphi$满足$\hat{\mathbb{R}}$，则返回⊥；否则，挑战者$\mathcal{C}$运行算法$\mathrm{KeyGen}(\mathrm{Para},\varphi)$生成sk，并运行算法$\mathrm{RKGen}(\mathrm{sk},\hat{\mathbb{R}})$生成rk，然后将rk发送给$\mathcal{A}$。

$\mathcal{O}_{\mathrm{PreK}}(\varphi)$：敌手$\mathcal{A}$提供属性$\varphi$。若$\varphi$符合访问策略$\mathbb{R}^*$，则返回⊥；否则，挑战者$\mathcal{C}$运行算法$\mathrm{KeyGen}(\mathrm{Para},\varphi)$生成sk，运行算法$\mathrm{PreKeyGen}(\mathrm{sk})$生成预解密密钥PreK和解密密钥DecK，然后将PreK发送给$\mathcal{A}$。

$\mathcal{O}_{\mathrm{DecK}}(\varphi)$：敌手$\mathcal{A}$提供属性$\varphi$。若$\varphi$满足访问策略$\mathbb{R}^*$，则返回⊥；否则，挑战者$\mathcal{C}$运行算法$\mathrm{KeyGen}(\mathrm{Para},\varphi)$生成sk，运行算法$\mathrm{PreKeyGen}(\mathrm{sk})$生成预解密密钥PreK和解密密钥DecK，然后将DecK发送给敌手$\mathcal{A}$。

$\mathcal{O}_{\mathrm{ReEnc}}(\varphi,\mathrm{CT},\hat{\mathbb{R}})$：敌手$\mathcal{A}$提供属性$\varphi$、密文CT以及访问策略$\hat{\mathbb{R}}=(\hat{M},\hat{\pi})$。挑战者$\mathcal{C}$首先运行算法$\mathrm{KeyGen}(\mathrm{Para},\varphi)$生成密钥sk，然后运行算法$\mathrm{RKGen}(\mathrm{sk},\hat{\mathbb{R}})$生成重加密密钥rk，最后运行算法$\mathrm{ReEnc}(\mathrm{rk},\mathrm{CT})$生成重加密密文CT′，并将CT′发送给$\mathcal{A}$。

此阶段的明确限制如下。

1) 敌手$\mathcal{A}$不能执行$\mathcal{O}_{\mathrm{sk}}(\varphi)$查询，其中属性$\varphi$满足挑战访问策略$\mathbb{R}^*=(M^*,\pi^*)$。

2) 敌手$\mathcal{A}$不能执行$\mathcal{O}_{\mathrm{rk}}(\varphi,\hat{\mathbb{R}})$查询，其中属性$\varphi$满足挑战访问策略$\mathbb{R}^*=(M^*,\pi^*)$，并且$\mathcal{A}$已经查询过满足访问策略$(\hat{M},\hat{\pi})$的密钥$sk$。

挑战阶段 敌手$\mathcal{A}$选择两个等长的明文m_0,m_1，其均与关键词ω^*不同，将它们发送给挑战者$\mathcal{C}$。挑战者$\mathcal{C}$随机选择$b\in\{0,1\}$，运行算法$\mathrm{Encrypt}(m_b,(M^*,\pi^*))$计算挑战密文$\mathrm{CT}^*$，并运行算法$\mathrm{IndexGen}(\omega^*)$计算挑战索引$I^*$，然后将$\mathrm{CT}^*$和$I^*$发送给敌手$\mathcal{A}$。

查询阶段2 敌手$\mathcal{A}$可以继续像在查询阶段1那样进行查询，但需遵守以下限制。

1) 敌手$\mathcal{A}$不能执行$\mathcal{O}_{\mathrm{sk}}(\varphi)$查询，其中属性$\varphi$满足挑战访问策略$\mathbb{R}^*=(M^*,\pi^*)$。

2) 敌手$\mathcal{A}$不能执行$\mathcal{O}_{\mathrm{rk}}(\varphi,\hat{\mathbb{R}})$和$\mathcal{O}_{sk}(\hat{\varphi})$查询，其中$\varphi$满足$\mathbb{R}^*$，$\hat{\varphi}$满足$\hat{\mathbb{R}}$。

3) 敌手$\mathcal{A}$不能执行$\mathcal{O}_{\mathrm{ReEnc}}(\varphi,\mathrm{CT}^*,\hat{\mathbb{R}})$和$\mathcal{O}_{\mathrm{sk}}(\hat{\varphi})$查询，其中$\varphi$满足$\mathbb{R}^*$，$\hat{\varphi}$满足$\hat{\mathbb{R}}$。

猜测阶段 敌手$\mathcal{A}$输出一个猜测b'，如果$b'=b$，则敌手$\mathcal{A}$赢得游戏。

敌手$\mathcal{A}$在这场游戏中的优势定义为

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{GameC_O}}=\left|\Pr[b'=b]-\frac{1}{2}\right| \tag{1}$$

定义4 如果没有任何概率多项式时间敌手$\mathcal{A}$能够在游戏$\mathrm{GameC_R}$中以不可忽略的优势获胜，那么此方案中的重加密密文满足选择明文攻击下的不可区分性。

2) $\mathrm{GameC_R}$

初始化 敌手$\mathcal{A}$选择一个挑战访问策略$\mathbb{R}^*=(M^*,\pi^*)$和一个挑战关键词ω^*，然后将它们发送给挑战者$\mathcal{C}$。

挑战者$\mathcal{C}$通过运行算法$\mathrm{Setup}(\kappa,U)$生成公共参数Para和主密钥MK，然后将公共参数Para发送给敌手$\mathcal{A}$。

查询阶段1 敌手$\mathcal{A}$将进行以下查询，挑战者$\mathcal{C}$将作出如下响应。

$\mathcal{O}_{\mathrm{sk}}(\varphi)$：敌手$\mathcal{A}$查询对应于属性$\varphi$的属性密钥。若$\varphi$满足访问策略$\mathbb{R}^*$，则$\mathcal{C}$返回⊥；否则，$\mathcal{C}$通过运行算法$\mathrm{KeyGen}(\mathrm{Para},\varphi)$生成sk并将其发送给敌手$\mathcal{A}$。

$\mathcal{O}_{\mathrm{rk}}(\varphi,\hat{\mathbb{R}})$：敌手$\mathcal{A}$提供属性$\varphi$和访问策略$\hat{\mathbb{R}}=$

$(\hat{M},\hat{\pi})$ 查询重加密密钥。若 φ 满足 $\hat{\mathbb{R}}$，则 $\mathcal{C}$ 返回 $\perp$；否则，$\mathcal{C}$ 运行算法 KeyGen(Para,φ) 生成 sk，运行算法 ReKeyGen(sk,$\hat{\mathbb{R}}$) 生成 rk，然后将 rk 发送给敌手 $\mathcal{A}$。

$\mathcal{O}_{\mathrm{PreK}}(\varphi)$：敌手 $\mathcal{A}$ 查询给定属性 φ 的预解密密钥。若 φ 符合访问策略 $\mathbb{R}^*$，则 $\mathcal{C}$ 返回 $\perp$；否则，$\mathcal{C}$ 运行算法 KeyGen(Para,φ) 生成 sk，运行算法 PreKeyGen(sk) 生成预解密密钥 PreK 和解密密钥 DecK，并将 PreK 发送给敌手 $\mathcal{A}$。

$\mathcal{O}_{\mathrm{DecK}}(\varphi)$：敌手 $\mathcal{A}$ 查询给定属性 φ 与预解密密钥一同生成的解密密钥。若 φ 满足访问策略 $\mathbb{R}^*$，则 $\mathcal{C}$ 返回 $\perp$；否则，$\mathcal{C}$ 运行算法 KeyGen(Para,φ) 生成 sk，随后运行算法 PreKeyGen(sk) 生成预解密密钥 PreK 和解密密钥 DecK，并将 DecK 发送给敌手 $\mathcal{A}$。

此阶段唯一的限制是，敌手 $\mathcal{A}$ 不能执行 $\mathcal{O}_{\mathrm{sk}}(\varphi)$ 查询，其中属性 φ 满足挑战访问策略 $\mathbb{R}^* = (M^*,\pi^*)$。

挑战阶段　敌手 $\mathcal{A}$ 结束查询阶段 1，则选择两个等长的明文 m_0,m_1，其均与关键词 ω^* 不同，并将 m_0,m_1 发送给挑战者 $\mathcal{C}$。$\mathcal{C}$ 随机选择 $b\in\{0,1\}$，运行 Encrypt(m_b,(M,π)) 获得密文 CT、运行算法 KeyGen(φ) 生成 sk、运行 ReKeyGen(sk,(M^*,π^*)) 生成 rk、运行 ReEnc(rk,CT) 生成 CT′、运行 IndexGen(ω^*) 生成挑战索引 I^*，最后将 CT^* 和 I^* 发送给敌手 $\mathcal{A}$。

查询阶段 2　敌手 $\mathcal{A}$ 可以继续像查询阶段 1 那样进行查询，并遵守与查询阶段 1 相同的限制。

敌手 $\mathcal{A}$ 在此游戏中的优势定义为

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{GameC_R}} = |\Pr[b'=b]-\frac{1}{2}| \tag{2}$$

注：在此游戏中，由于敌手可以随意查询重加密密钥，并能够独立执行重加密过程，因此没有设置专门的重加密查询。

定义 5　如果没有任何概率多项式时间敌手 $\mathcal{A}$ 能够以不可忽略的优势赢得游戏 $\mathrm{GameC_I}$，那么方案满足选择关键词攻击下不可区分性（IND-CKA, indistinguishability under chosen keyword attack）。

3) $\mathrm{GameC_I}$

初始化　挑战者 $\mathcal{C}$ 执行 Setup(κ,U) 算法生成公共参数 Para 和主密钥 MK。随后，$\mathcal{C}$ 将公共参数 Para 发送给敌手 $\mathcal{A}$。

查询阶段 1　敌手 $\mathcal{A}$ 进行以下查询，挑战者 $\mathcal{C}$ 将作出如下回应。

$\mathcal{O}_{\mathrm{sk}}(\varphi)$：敌手 $\mathcal{A}$ 提供属性 φ。挑战者 $\mathcal{C}$ 运行算法 SKGen(MK,φ) 获得 sk 并将其发送给敌手 $\mathcal{A}$。

$\mathcal{O}_{\mathrm{DecK}_\varphi}(\varphi)$：敌手 $\mathcal{A}$ 提供属性 φ。挑战者 $\mathcal{C}$ 通过 $\mathcal{O}_{\mathrm{sk}}(\varphi)$ 查询获得 sk，然后运行算法 PreKeyGen(sk) 获得 PreK 和 DecK，最后将 DecK 发送给 $\mathcal{A}$。

$\mathcal{O}_{\mathrm{TD}}(\varphi,\omega)$：敌手 $\mathcal{A}$ 提供一个关键词 ω 和属性 φ。挑战者 $\mathcal{C}$ 分别通过查询 $\mathcal{O}_{\mathrm{sk}}(\varphi)$ 和 $\mathcal{O}_{\mathrm{DecK}}(r)$ 获得 sk 和 DecK，然后运行算法 TDGen(sk,DecK,ω) 生成陷门 TD 并将其发送给敌手 $\mathcal{A}$。

挑战阶段　敌手 $\mathcal{A}$ 首先选择两个大小相等的关键词 ω_0 和 ω_1，然后将它们发送给挑战者 $\mathcal{C}$。挑战者 $\mathcal{C}$ 随机选择 $b\in\{0,1\}$ 并通过运行算法 IndexGen(ω_b) 生成索引 I，然后将其发送给敌手 $\mathcal{A}$。

查询阶段 2　敌手 $\mathcal{A}$ 可以继续像查询阶段 1 那样进行查询。唯一的限制是，$\mathcal{A}$ 不能执行 $\mathcal{O}_{\mathrm{TDGen}}(\omega_0)$ 和 $\mathcal{O}_{\mathrm{TDGen}}(\omega_1)$ 查询。

猜测阶段　敌手 $\mathcal{A}$ 输出一个猜测值 b'，如果 $b'=b$，则 $\mathcal{A}$ 赢得游戏。

敌手 $\mathcal{A}$ 在此游戏中获胜的优势定义为

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{GameC_I}} = |\Pr[b'=b]-\frac{1}{2}| \tag{3}$$

3　方案构造

3.1　系统初始化与密文上传阶段

系统初始化与密文上传流程如图 2 所示。

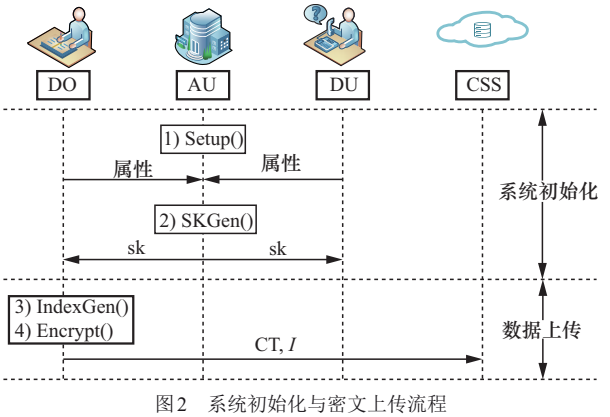

图 2　系统初始化与密文上传流程

1) Setup(κ,U)：AU 选择安全参数 $\mathbb{R}^*$ 和属性空间 U，然后执行以下步骤。

① 生成 $(p,e,\mathbb{G},\mathbb{G}_T)$。

② 定义哈希函数 $H:\{0,1\}^* \to \mathbb{G}_T$，$H_0:\{0,1\}^\kappa \to \mathbb{G}$，$H_1:\mathbb{G}_T \to \{0,1\}^\kappa$，$H_2:\mathbb{G}_T \to \{0,1\}^*$，$H_3:\{0,1\}^* \to \mathbb{Z}_p^*$，$H_4:\{0,1\}^* \to \{0,1\}^\kappa$，$H_5:\mathbb{G}_T \to \mathbb{Z}_p^*$。

③ 选择 $\alpha,\beta,a,b,c \in_R \mathbb{Z}_p^*$，$g,Q,h_1,\cdots,h_U \in_R \mathbb{G}$。

④ 主密钥 $\mathrm{MK}=(\alpha,\beta,a,b,c)$，系统公共参数 $\mathrm{Para}=(p,g,Q,\mathbb{G},\mathbb{G}_T,e,H,H_0,H_1,H_2,H_3,H_4,H_5,g^a,g^b,g^{abc},g^\beta,e(g,g)^\alpha,h_1,\cdots,h_U)$。

2) SKGen(MK,φ)：用户提交属性 φ 和真实身份 RID，AU 运行 SKGen(MK,φ) 生成属性密钥 sk 并发送给用户。算法选择 $u \in_R \mathbb{Z}_p^*$，计算 $d_1=ac$，$d_2=bc$，$\mathrm{AK}_1=g^\alpha g^{\beta u}$，$\mathrm{AK}_2=g^u$，$\mathrm{AK}_x=(H(x))^u$，随后计算

$$\mathrm{TK}_1=g^{\frac{\alpha}{a}}g^{\frac{\beta u}{a}},\ \mathrm{TK}_2=g^{\frac{\alpha}{b}}g^{\frac{\beta u}{b}}$$

将属性密钥 $\mathrm{sk}=(d_1,d_2,\mathrm{AK}_1,\mathrm{AK}_2,\{\mathrm{AK}_x\}_{x\in\varphi},\mathrm{TK}_1,\mathrm{TK}_2)$ 发送给用户。

3) IndexGen(ω_n)：工厂服务器提供关键词 ω_n ($n>3$)，选择 $D \in_R \{0,1\}^\kappa, v_1,v_2 \in_R \mathbb{Z}_p^*$，令 $v_t=v_1+v_2$，生成索引

$$I=(L_1=(g^b)^{v_1},L_2=(g^a)^{v_2},\{I_{1,i}=D\oplus H_1(C_{\omega_i}),$$

$$I_{2,i}=H_2(C_{\omega_i}),L_{3,i}=g^{v_t}H_0(\omega_i)^{v_t}\}_{i=[1,n]}),D,v_t)$$

4) Encrypt($m,\overline{\mathbb{R}},v_t,D$)：生成索引后，工厂服务器将明文 m、访问策略 $\overline{\mathbb{R}}=(\bar{M},\bar{\pi})$、随机值 v_t 和 D 作为输入并运行此算法生成密文 CT。算法选择 $k_{\mathrm{DO}} \in_R \{0,1\}^\kappa$，计算 $\bar{r}=H_3(k_{\mathrm{DO}}\|m)$。随后，继续选择 $\boldsymbol{\mu}=(\bar{r},y_2,\cdots,y_n) \in_R \mathbb{Z}_p^{*n}$，计算 $\lambda_j=\boldsymbol{\mu}\cdot\bar{M}_j$，其中 $j=[1,\bar{l}]$：选择 $\bar{r}_j \in_R \mathbb{Z}_p^*$，计算 $C_{4,j}=g^{\beta\lambda_j}\cdot H(\bar{\pi}(j))^{-\bar{r}_j}$，$C_{5,j}=g^{\bar{r}_j}$，其中 $j=[1,\bar{l}]$。最后计算

$$C_0=m\oplus H_4(k_{\mathrm{DO}}),C=k_{\mathrm{DO}}\cdot e(g,g)^{\alpha\bar{r}},$$

$$C_1=g^{v_t+\bar{r}}\cdot H_0(D),C_2=Q^{\bar{r}},C_3=g^{\bar{r}}$$

得到密文 $\mathrm{CT}=(C_0,C,C_1,C_2,C_3,\{C_{4,j},C_{5,j}\}_{j=[1,\bar{l}]},\overline{\mathbb{R}}=(\bar{M},\bar{\pi}))$。最终，工厂服务器将 I 和 CT 上传到 CSS。

3.2 密文检索阶段

密文检索阶段流程如图 3 所示，其中，List1 和 List2 根据 DU 的需求组织文件，List1 包含与 DU 需求匹配且 DU 可解密的文件，List2 包含与 DU 需求一致但 DU 不可解密的文件。

5) PreKenGen(sk)：数据使用者将其属性密钥 sk 作为输入并运行此算法生成预解密密钥 PreK，解密密钥 $\mathrm{DecK}=z$，其中 $z \in_R \mathbb{Z}_p^*$。计算过程如下

$$\overline{\mathrm{AK}}_1=(\mathrm{AK}_1)^z,\overline{\mathrm{AK}}_2=(\mathrm{AK}_2)^z,\{\overline{\mathrm{AK}}_x=(\mathrm{AK}_x)^z\}_{x\in\varphi}$$

预解密密钥 $\mathrm{PreK}=(\overline{\mathrm{AK}}_1,\overline{\mathrm{AK}}_2,\{\overline{\mathrm{AK}}_x\}_{x\in\varphi})$。

6) TDGen(sk,DecK,ω_{n_1})：数据使用者使用 sk、DecK 和关键词 ω_{n_1} 作为输入并运行此算法生成陷门 TD。

算法选择 $t_1,t_2,t_3,t_4 \in_R \mathbb{Z}_p^*, t_5 \in_R [1,n_1]$，计算

$$T_{1,i}=g^{d_1t_1}\cdot(H_0(\omega_i))^{d_1+d_1t_1}$$

$$\bar{T}_{1,i}=(\mathrm{TK}_2)^z\cdot(gH_0(\omega_i))^{d_1t_2}$$

$$T_{2,i}=g^{d_2t_1}(H_0(\omega_i))^{d_2+d_2t_1}$$

$$\bar{T}_{2,i}=(\mathrm{TK}_1)^z\cdot(gH_0(\omega_i))^{d_2t_2}$$

其中，$i=[1,n_1]$，同时计算

$$T_{1,n_1+j}=g^{d_1t_3}\cdot(H_0(\omega_j))^{d_1+d_1t_3}$$

$$\bar{T}_{1,n_1+j}=(\mathrm{TK}_2)^z\cdot(gH_0(\omega_j))^{d_1t_4}$$

$$T_{2,n_1+j}=g^{d_2t_3}(H_0(\omega_j))^{d_2+d_2t_3}$$

$$\bar{T}_{2,n_1+j}=(\mathrm{TK}_1)^z\cdot(gH_0(\omega_j))^{d_2t_4}$$

其中，$j\leqslant 5, n_2=n_1+t_5$。

此外，需要计算 $T_3=(g^{abc})^{t_1},\bar{T}_3=(g^{abc})^{t_2}$。陷门 $\mathrm{TD}=(\{T_{1,i},\bar{T}_{1,i},T_{2,i},\bar{T}_{2,i}\}_{i=[1,n_2]},T_3,\bar{T}_3)$。

数据使用者使用哈希函数 SHA-256 将其账户地址 Addr、随机值 R_1 作为输入获取查询 ID-QID。

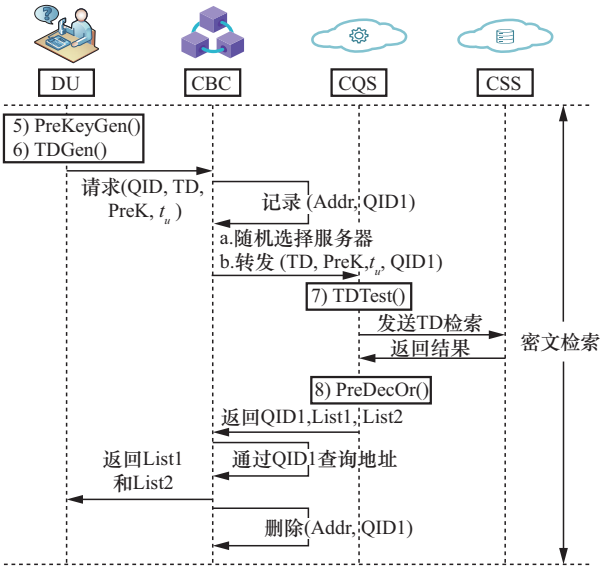

图3 密文检索流程

选择匹配精准度 $t_u \in [1-\frac{1}{(n_3+1)},\frac{1}{2}]$ 通过智能合约 A 发起请求，包括 (QID,TD,PreK,t_u)。

接收到请求后，智能合约 A 自动检索用户地址 Addr，将 (QID,TD,PreK,t_u,Addr) 通过中间合约转发给智能合约 B。智能合约 B 将产生随机数 R_2，计算 $\mathrm{QID}_1 = \mathrm{QID} \oplus \mathrm{Addr} \oplus R_2$，同时将 ($\mathrm{QID}_1$,Addr) 记录在表 $\mathrm{Tab}_{\mathrm{record}}$ 中。最后从表 $\mathrm{Tab}_{\mathrm{CQS}}$ 中随机选择 CQS 并转发 QID_1,TD,PreK,t_u。

7) TDTest (TD,I,t_u)：CQS 接收到请求后，将 TD 和 t_u 输入此算法。算法定义 count = 0,i_c = 0,j_c = 0，进行如下计算。

```
for i=1; i≤n; i++ do
 for j=1; j≤n_2; j++ do
  (φ̂, sk_φ̂);
  if (H_2(C_{ω_ij})! = I_{2,i} && i==n && j==n_1)
   return ⊥;
  else if (H_2(C_{ω_ij}) == I_{2,i})
   i_c = i, j_c = j, count ++;
 end for
end for
```

若 count ⩾ $t_u n_2$，则计算

$$D = I_{1,i_c} \oplus H_1(C_{\omega_{i_j}})$$

$$E = \frac{e(\bar{T}_{1,j_c},L_1)\cdot e(\bar{T}_{2,j_c},L_2)}{e(\bar{T}_3,L_{3,i_c})}$$

若匹配成功，算法输出(D, E)，CQS 在 List1 中记录与数据用户需求相匹配的数据条目；否则，算法输出 ⊥。

8) $\mathrm{PreDec}_{\mathrm{Or}}$ (PreK,CT,D,E)：CQS 输入 PreK、(D,E) 和 CT。如果预解密密钥中属性 φ 不满足密文中的访问策略 ($\bar{M}$,$\bar{\pi}$)，算法输出 ⊥，CQS 从 List1 中删除相应数据条目，并将其添加到 List2 中；否则，算法令 $J = \{ j:\bar{\pi}(j) \in \varphi \}$，选择 $\{\theta_j\} \in {}_R\mathbb{Z}_p^*$，$\sum_{j\in J}\theta_j \cdot \bar{M}_j = (1,0,\cdots,0)$，计算

$$\hat{C}_1 = \frac{C_1}{H_0(D)}$$

$$\hat{C} = \frac{e(\hat{C}_1,\overline{\mathrm{AK}}_1)}{E\cdot\prod_{j\in J}(e(C_{4,j},\overline{\mathrm{AK}}_2)\cdot e(C_{5,j},\overline{\mathrm{AK}}_{\bar{\pi}(j)}))^{\theta_j}}$$

原始密文的预解密密文 $\mathrm{CT}_{\mathrm{OrPre}} = \{C_0, C, \hat{C}\}$。

CQS 将 QID_1、List1 和 List2 转发给智能合约 C，智能合约 C 自动调用智能合约 D，将 QID_1、List1 和 List2 转发给智能合约 D。智能合约 D 使用 QID_1 在表 $\mathrm{Tab}_{\mathrm{record}}$ 中查找对应的账户地址 Addr。然后，通过地址 Addr 向数据用户发送 List1 和 List2，同时发送到表中与 Addr 相邻的两个记录的地址。最后，从 $\mathrm{Tab}_{\mathrm{record}}$ 中删除与 Addr 相关的数据。

3.3　请求密文授权与解密阶段

请求密文授权与解密阶段流程如图 4 所示。

数据用户在接收到智能合约 D 发送的 List1 和 List2 后，向 List2 中列出的数据所有者发送请求。

9) RKGen ($\mathrm{sk}_{\mathrm{DO}}$,$\mathbb{R}'$)：数据拥有者接收到请求并验证数据用户身份，验证通过则将属性密钥 $\mathrm{sk}_{\mathrm{DO}}$ 及与数据使用者相关的访问策略 $\mathbb{R}' = (M',\pi')$ 作为此算法输入生成重加密密钥 rk。算法选择 $\gamma \in {}_R\mathbb{G}_T, \delta \in {}_R\mathbb{Z}_p^*$，计算 $\mathrm{rk}_1 = \mathrm{AK}_1^{H_5(\gamma)}\cdot Q^{\delta}$, $\mathrm{rk}_2 = g^{\delta}$, $\mathrm{rk}_3 = (\mathrm{AK}_2)^{H_5(\gamma)}$, $\mathrm{rk}_{4,x} = (H(x))^{u\cdot H_5(\gamma)}$，其中 $x \in \varphi$。选择 $\boldsymbol{u}' = (r',y_1,\cdots,y_{n'}) \in {}_R\mathbb{Z}_p^{*n}$ 并计算 $\lambda_{j'} = \boldsymbol{u}'\cdot M_{j'}$，选择 $r_j \in {}_R\mathbb{Z}_p^*$ 并计算 $\mathrm{rk}_{7,j} = g^{\beta\lambda_{j'}}(H(x))^{-r_{j'}}$, $\mathrm{rk}_{8,j} = g^{r_{j'}}$，其中 $j = [1,l']$。最后计算 $\mathrm{rk}_5 = \gamma\cdot e(g,g)^{\alpha r'}$, $\mathrm{rk}_6 = g^{r'}$。

重加密密钥 rk = { rk_1, rk_2, rk_3, $\mathrm{rk}_{4,x}$, rk_5, rk_6, { rk_7,rk_8 }$_{j=[1,l']}$,$\mathbb{R}'$}。

数据用户在接收到数据所有者发送的 rk 后，使用 SHA-256 哈希函数，以其账户地址和随机数 R_3 作为输入来生成一个新的查询 ID-QID_2。然后，数据用户向智能合约 A 发送请求，该请求包括 QID_2、PreK、rk 以及来自 List2 中与 rk 相对应的数据标识符 Da。

智能合约 A 在接收到数据用户的请求后，自动获取数据用户的账户地址 Addr 并将 (QID_2,rk,PreK,Addr) 转发给智能合约 B。合约 B 选择随机数 R_4，然后计算新的查询 ID-$\mathrm{QID}_3 = \mathrm{QID}_2 \oplus \mathrm{Addr} \oplus R_4$，并在表 $\mathrm{Tab}_{\mathrm{record}}$ 中记录 (QID_3,Addr)。随后，合约 B 从云服务器节点列表中随机选择 CQS 并转发请求，该请求包括 (QID_3,Da,rk,PreK)。

10) ReEnc (rk,CT)：CQS 运行此算法获取重加密密文 CT′。输入 rk 和 CT，此算法首先设置 $J = \{ j:\pi(j) \in \varphi \} \subset \{1,\cdots,\bar{l}\}$，选择 $\{\theta_j\} \in {}_R\mathbb{Z}_p^*$ 使 $\sum_{j\in J}\theta_j \cdot \bar{M}_j = (1,0,\cdots,0)$，计算

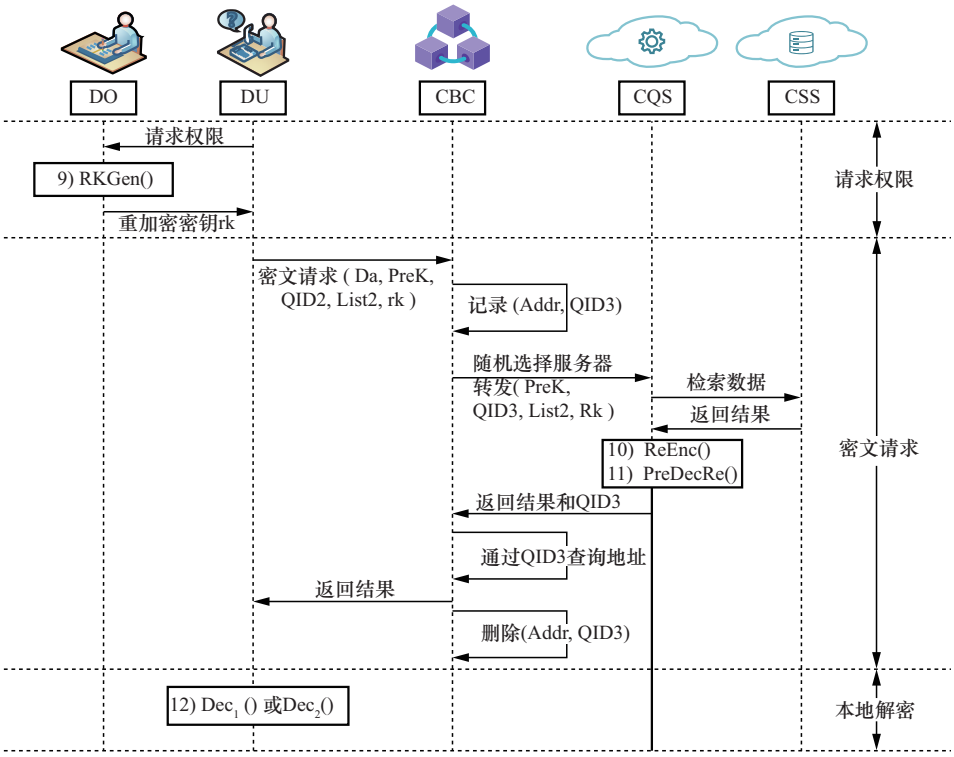

图4　请求密文授权与解密流程

$$C_R=\frac{e(\mathrm{rk}_1,C_3)e(\mathrm{rk}_2,C_2)^{-1}}{\prod_{j\in J}(e(\mathrm{rk}_3,C_{4,j})e(\mathrm{rk}_{4,x},C_{5,j}))^{\theta_j}}$$

重加密密文 $\mathrm{CT}'=(C_0,C,\{\mathrm{rk}_{7,j},\mathrm{rk}_{8,j}\}_{j=[1,l']},C_R,\mathrm{rk}_5,\mathrm{rk}_6,\mathbb{R}')$。

11) $\mathrm{PreDec}_{\mathrm{Re}}(\mathrm{PreK},\mathrm{CT}',\mathrm{rk})$：如果解密密钥中的属性 φ 不满足重加密密文的访问策略 (M',π')，算法输出 $\perp$；否则，设 $J=\{j:\pi(j)'\in\varphi\}$，选择 $\{\theta_j\}\in{}_R\mathbb{Z}_p^*$ 使

$$\sum_{j\in J}\theta_j\cdot M'=(1,0,\cdots,0)$$

计算

$$W=\frac{e(\overline{\mathrm{AK}}_1,\mathrm{rk}_6)}{\prod_{j\in J}(e(\overline{\mathrm{AK}}_2,\mathrm{rk}_{7,j})\cdot e(\overline{\mathrm{AK}}_x,\mathrm{rk}_{8,j}))^{\theta_j}}$$

重加密密文的预解密密文 $\mathrm{CT}_{\mathrm{RePre}}=\{C_0,C,C_R,W,C_{\mathrm{Re}}=\mathrm{rk}_5\}$。

CQS 将 QID_3 和预解密结果 $\mathrm{CT}_{\mathrm{RPreDec}}$ 发送给智能合约 C。随后，智能合约 C 自动调用智能合约 D，并将结果转发给智能合约 D。智能合约 D 通过 QID_3 在表 $\mathrm{Tab}_{\mathrm{record}}$ 中查找对应的账户地址 Addr。然后通过地址 Addr 向数据用户发送结果 $\mathrm{CT}_{\mathrm{RPreDec}}$，同时也发送给表中与 Addr 相邻的两个记录的地址。这样做确保了云服务器无法识别真实用户的身份。最后，从 $\mathrm{Tab}_{\mathrm{record}}$ 中删除与 Addr 相关的数据。

数据使用者运行算法 $\mathrm{Dec}_1(\mathrm{DecK},\mathrm{CT}_{\mathrm{OrPre}})$ 和 $\mathrm{Dec}_2(\mathrm{CT}_{\mathrm{RePre}},\mathrm{DecK})$ 解密密文。若正确输出明文，则表明密文未被篡改。

12) $\mathrm{Dec}_1(\mathrm{DecK},\mathrm{CT}_{\mathrm{OrPre}})$：此算法依次计算 $K_{\mathrm{DO}}=\frac{C}{\hat{C}^{1/\mathrm{DecK}}},m=C_0\oplus H_4(\mathrm{K}_{\mathrm{DO}}),\bar{r}'=H_3(K_{\mathrm{DO}},m)$。若 $C=\!=K_{\mathrm{DO}}\cdot e(g,g)^{\alpha\bar{r}'}$，则输出 m；否则，输出 $\perp$。

13) $\mathrm{Dec}_2(\mathrm{CT}_{\mathrm{RePre}},\mathrm{DecK})$：算法计算

$$\gamma=\frac{C_{\mathrm{Re}}}{W^{\frac{1}{\mathrm{DecK}}}},K_{\mathrm{DO}}=\frac{C}{(C_R)^{\frac{1}{H_5(\gamma)}}}$$

$$m=C_0\oplus H_4(K_{\mathrm{DO}}),\bar{r}'=H_3(K_{\mathrm{DO}},m)$$

若 $C=\!=K_{\mathrm{DO}}\cdot e(g,g)^{\alpha\bar{r}'}$，输出 m；否则，输出 $\perp$。

4　正确性证明

方案中相关公式正确性证明如下。

在算法 TDTest() 中，E 的计算如下所示。如果陷门中的关键词 ω_i 与索引中的关键词 ω_j 相匹配，即 $\omega_i=\omega_j$，那么可以通过继续计算得到 E 的值，即 $E=e(g,g)^{(\alpha+\beta u)zv_t}$。

$$E=\frac{e(\bar{T}_1,L_1)e(\bar{T}_2,L_2)}{e(\bar{T}_3,L_{3,i})}=$$
$$\frac{e(g^{\frac{\alpha z}{b}}\cdot g^{\frac{\beta uz}{b}}\cdot g^{d_1t_2}\cdot(H_0(\omega_j))^{d_1t_2},g^{bv_1})\cdot e(g^{\frac{\alpha z}{a}}\cdot g^{\frac{\beta uz}{a}}\cdot g^{d_2t_2}\cdot(H_0(\omega_j))^{d_2t_2},g^{av_1})}{e(g^{abct_2},g^{v_t}\cdot(H_0(\omega_i))^{v_t})}=$$
$$\frac{e(g,g)^{\alpha zv_t}\cdot e(g,g)^{\beta uzv_t}\cdot e(g,H_0(\omega_j))^{abct_2v_t}}{e(g,H_0(\omega_i))^{abct_2v_t}}$$

在算法TDTest()中，$C_{\omega_{i,j}}$的计算如下所示。如果陷门中的关键词ω_i与索引中的关键词ω_j匹配，那么可以得到$D=I_1\oplus H_1(C_{\omega_j})=I_1\oplus H_1(C_{\omega_i})$。

$$C_{\omega_{i,j}}=\frac{e(T_1,L_1)\cdot e(T_2,L_2)}{e(T_3,L_{3,i})}=$$
$$\frac{e(g^{act_1}\cdot(H_0(\omega_j))^{d_1+d_1t_1},g^{bv_1})e(g^{d_2t_1}\cdot(H_0(\omega_j))^{d_2+d_2t_1},g^{av_2})}{e(g^{abct_1},g^{v_t}\cdot(H_0(\omega_i))^{v_t})}=$$
$$\frac{e(g,g)^{abct_1v_1}\cdot e(H_0(\omega_j),g)^{bv_1(d_1+d_1t_1)}\cdot e(g,g)^{abct_1v_2}\cdot e(H_0(\omega_j),g)^{av_2(d_2+d_2t_1)}}{e(g,g)^{abct_1v_t}\cdot e(H_0(\omega_i),g)^{abct_1v_t}}=$$
$$\frac{e(H_0(\omega_j),g)^{abcv_1}\cdot e(H_0(\omega_j),g)^{abcv_1t_1}\cdot e(H_0(\omega_j),g)^{abcv_2}\cdot e(H_0(\omega_j),g)^{abct_1v_2}}{e(H_0(\omega_i),g)^{abct_1v_t}}$$

在算法$\mathrm{PreDec}_{\mathrm{Or}}()$中，$\hat{C}$的计算如下所示。

$$\hat{C}=\frac{e(\hat{C}_1,\overline{\mathrm{AK}}_1)}{E\cdot\prod_{j\in J}(e(C_{4,j},\overline{\mathrm{AK}}_2)e(C_{5,j},\overline{\mathrm{AK}}_{\bar{\pi}(j)}))^{\theta_j}}=$$
$$\frac{e(g^{v_t}g^{\bar{r}},g^{z\alpha}g^{zu\beta})}{E\cdot\prod_{j\in J}\cdot(e(g^{\beta\lambda_j}(H(\bar{\pi}(j)))^{-\bar{r}_j},g^{zu})e(g^{\bar{r}_j},(H(\bar{\pi}(j)))^{zu}))^{\theta_j}}=$$
$$\frac{e(g,g)^{z\alpha v_t}e(g,g)^{z\alpha\bar{r}}e(g,g)^{zu\beta\bar{r}}e(g,g)^{zu\beta v_t}}{e(g,g)^{z\alpha v_t}e(g,g)^{zu\beta v_t}e(g,g)^{zu\beta\bar{r}}}=e(g,g)^{z\bar{r}\alpha}$$

在算法ReEnc()中，C_R的计算如下所示。

$$C_R=\frac{e(\mathrm{rk}_1,C_3),e(\mathrm{rk}_2,C_2)^{-1}}{\prod_{j\in J}(e(\mathrm{rk}_3,C_{4,j})e(\mathrm{rk}_{4,x},C_{5,j}))^{\theta_j}}=$$
$$\frac{e(g,g)^{\alpha\bar{r}H_5(\gamma)}e(g,g)^{\beta u\bar{r}H_5(\gamma)}e(Q,g)^{\delta\bar{r}}e(Q,g)^{-\delta\bar{r}}}{e(g,g)^{\beta u\bar{r}H_5(\gamma)}}=$$
$$e(g,g)^{\alpha\bar{r}H_5(\gamma)}$$

在算法$\mathrm{PreDec}_{\mathrm{Re}}()$中，$W$的计算如下所示。

$$W=\frac{e(\overline{\mathrm{AK}}_1,\mathrm{rk}_6)}{\prod_{j\in J}(e(\overline{\mathrm{AK}}_2,\mathrm{rk}_{7,j})\cdot e(\overline{\mathrm{AK}}_x,rk_{8,j}))^{\theta_j}}=$$
$$\frac{e(g,g)^{\alpha zr'}\cdot e(g,g)^{\beta uzr'}}{\prod_{j\in J}(e(g,g)^{\beta uz\lambda_j}\cdot e(g,H(x))^{-uzr_{j'}}\cdot e(g,H(x))^{uzr_{j'}})^{\theta_j}}=$$
$$e(g,g)^{\alpha zr'}$$

5　方案对比分析与性能分析

表1展示了几种方案的功能比较，其中，DPR为动态权限请求，UDS为用户自定义搜索精度，SS为选择性共享，MM为多关键词匹配，BC为区块链，OKGA为离线关键词猜测。从表1可以看出，只有本文方案支持动态权限申请和用户定义的搜索精度。不同于DSA[3]，本文方案具备抵抗离线关键词猜测攻击的能力。虽然G-ABEET[4]也能够抵抗离线关键词猜测攻击，但其实现机制与本文方案不同。相较其他方案，本文方案在完整性、动态权限请求和用户友好性方面具有显著优势，这增强了其在工业物联网环境中云辅助数据共享的适用性。

为评估本文方案性能，本节采用如表2所示的配置进行实验。

表 1　　几种方案的功能比较

方案	DPR	UDS	SS	MM	BC	OKGA
DSA[3]	×	×	√	×	×	×
G-ABEET [4]	×	×	√	×	×	√
MK-SVS[22]	×	×	√	√	×	—
HTAC [27]	×	×	√	—	×	—
BE-TRDSS[28]	×	×	√	—	√	—
本文方案	√	√	√	√	√	√

表 2　　实验配置

环境	配置项	参数/版本
硬件环境	处理器	11th Gen Intel Core i5-11500 (2.70 GHz)
	主机内存	16 GB RAM
	虚拟机内存	4 GB
	虚拟机磁盘空间	60 GB
软件环境	开发工具	IntelliJ IDEA 2023.2.5
	编程语言	Java 11.0.20.1
	加密库	Jpbc（A1 型椭圆曲线，160 位群阶）
	操作系统	Ubuntu 22.04.3 (VM Host OS)
	区块链平台	FISCO BCOS 2.9.1
	区块链中间件	WeBASE 1.5.5
	容器工具	Docker 24.0.7，Docker-Compose 1.29.2

算法 SKGen 和 Encrypt 的时间消耗如图 5 和图 6 所示。由图 5 和图 6 可知，本文方案在属性密钥生成算法 SKGen 的效率方面优于 DSA[3]、HTAC[27]和 BE-TRDSS[28]。虽然在加密算法 Encrypt 效率上低于 BE-TRDSS[28]，但仍高于 DSA[3]和 HTAC[27]。

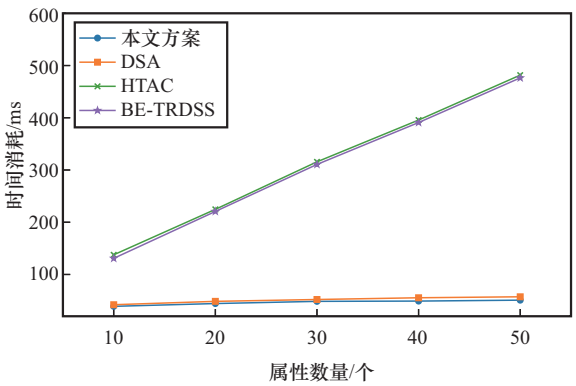

图 5　算法 SKGen 的时间消耗

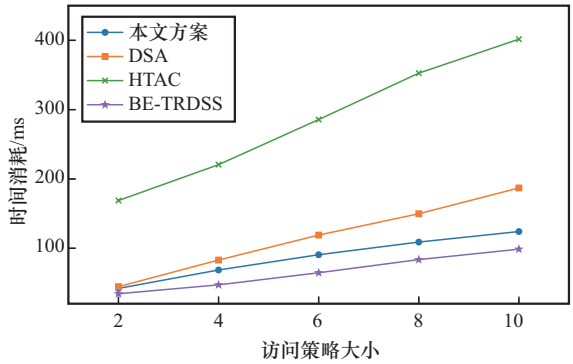

图 6　算法 Encrypt 的时间消耗

图 7 对比了索引生成算法 IndexGen 和陷门生成算法 TDGen 的时间消耗。由图 7 可知，本文方案的效率与 DSA[3]基本一致，尽管 TDGen 的效率略有下降，但其陷门生成过程中引入的冗余设计增强了安全性。图 8 表明，得益于云辅助预解密机制，本文方案的解密算法 Dec 效率显著高于 BE-TRDSS[28]和 HTAC[27]，与 DSA[3]基本一致。

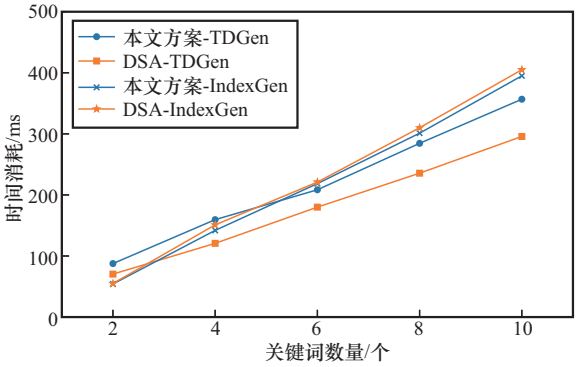

图 7　算法 IndexGen 与 TDGen 的时间消耗

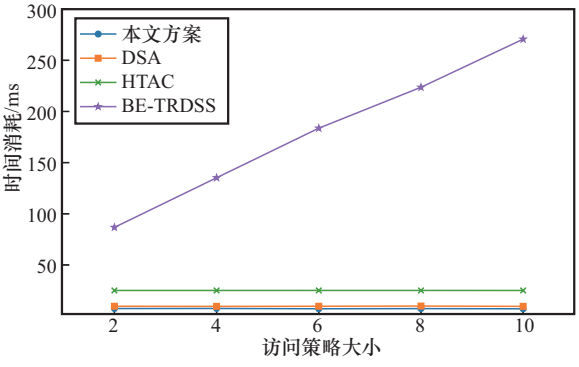

图 8　算法 Dec 的时间消耗

预解密密钥生成算法 PreKeyGen 与预解密算法 PreDec 的时间消耗如图 9 所示。由图 9 可知，本文方案在效率上相较 DSA[3]有所提升。本文方案其他

算法的时间消耗如图 10 所示。由图 10 可知，本文方案其他算法的时间消耗在正常范围内。

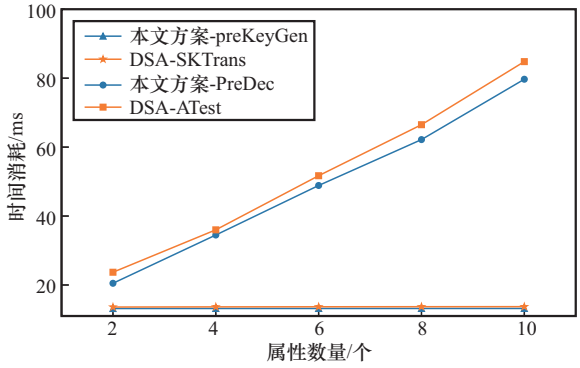

图 9　算法 PreKeyGen 与 PreDec 的时间消耗

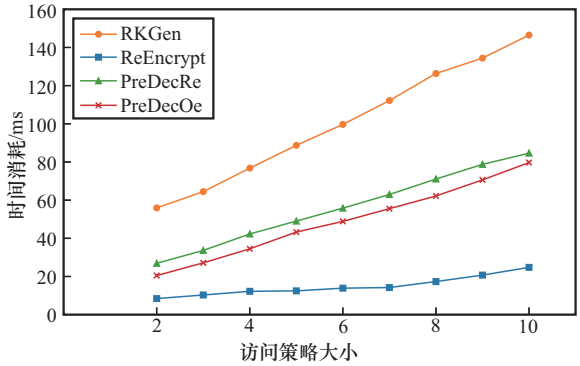

图 10　本文方案其他算法的时间消耗

在 FISCO BCOS[29]平台的测试结果表明，当请求大小从 5 KB 增加到 50 KB 时，各操作的时间消耗保持高度稳定，结果如表 3 所示。由表 3 可知，发送检索请求（SRR）操作时间从 538.510 ms 增至 549.651 ms，转发检索结果（FRR）从 537.600 ms 增至 537.766 ms，发送转换请求（STR）从 532.186 ms 增至 533.922 ms，转发转换结果（FTR）从 534.908 ms 增至 535.612 ms，波动幅度均小于 0.7%，绝对时间增量不超过 3.2 ms，这表明处理时间与请求大小基本无关（$|r|<0.01$）。在 Gas 成本方面，虽然请求大小增加了 9 倍，但 Gas 成本仅增加了 6.90~7.59 倍。执行智能合约时区块链吞吐量的对比如图 11 所示。由图 11 可知，在系统区块链吞吐量方面，随着交易发送速率的增加，本文方案较对比方案 HTAC[27] 和 BE-TRDSS[28]更晚达到性能瓶颈且始终优于这两种方案。这些结果证实了本文方案在处理不同规模请求时具有稳定的时间效率和可控的资源消耗特性，且能适应高并发场景。

表 3　不同请求大小的时间消耗对比

操作	时间消耗/ms		Gas 成本	
	请求大小为 5 KB	请求大小为 50 KB	请求大小为 5 KB	请求大小为 50 KB
SRR	538.510	549.651	471 449	3 748 088
FRR	537.600	537.766	366 827	3 114 745
STR	532.186	533.922	474 543	3 751 237
FTR	534.908	535.612	361 980	3 109 842

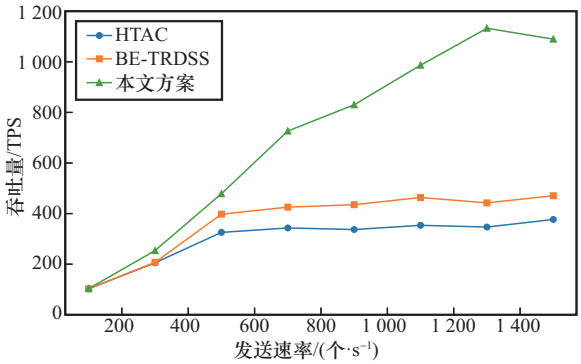

图 11　区块链智能合约吞吐量

6　结束语

本文提出了一种基于区块链的、用户友好的云辅助工业物联网数据检索和共享方案。该方案正式定义为允许数据请求者请求两种类型的数据：被授权解密的数据和感兴趣但未被授权解密的数据。它支持多关键词搜索，并利用云服务器对密文进行预解密，减轻用户的计算负担。引入智能合约机制生成唯一的请求 ID，并将查询结果发送给另外两个用户，以隐藏用户身份，防止攻击者将关键词猜测结果与特定用户关联，从而增强安全性和抵抗离线关键词猜测攻击的能力。本文详细描述了该方案的构建，提供了确保保密性的形式化安全证明，并包含了确认其可行性和实用性的综合分析。未来的研究将集中在提高多关键词匹配过程的效率上。

参考文献：

[1] GUPTA I, SINGH A K, LEE C N, et al. Secure data storage and sharing techniques for data protection in cloud environments: a systematic review, analysis, and future directions[J]. IEEE Access, 2022, 10: 71247-71277.

[2] BOUCHELAGHEM S, OMAR M. Secure and efficient pseudonymization for privacy-preserving vehicular communications in smart cities[J]. Computers & Electrical Engineering, 2020, 82: 106557.

[3] YANG K, SHU J G, XIE R T. Efficient and provably secure data selective sharing and acquisition in cloud-based systems[J]. IEEE Transac-

tions on Information Forensics and Security, 2022, 18: 71-84.
[4] XIONG H, WANG H X, MENG W Z, et al. Attribute-based data sharing scheme with flexible search functionality for cloud-assisted autonomous transportation system[J]. IEEE Transactions on Industrial Informatics, 2023, 19(11): 10977-10986.
[5] SAHAI A, WATERS B. Fuzzy identity-based encryption[C]//Advances in Cryptology-EUROCRYPT 2005. Berlin: Springer, 2005: 457-473.
[6] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]//Proceedings of the 13th ACM Conference on Computer and Communications Security. New York: ACM Press, 2006: 89-98.
[7] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[C]//Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07). Piscataway: IEEE Press, 2007: 321-334.
[8] ZHANG Y H, CHEN X F, LI J, et al. Anonymous attribute-based encryption supporting efficient decryption test[C]//Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. New York: ACM Press, 2013: 511-516.
[9] ZHAO C B, XU L, LI J G, et al. Toward secure and privacy-preserving cloud data sharing: online/offline multiauthority CP-ABE with hidden policy[J]. IEEE Systems Journal, 2022, 16(3): 4804-4815.
[10] CHEN J, WEE H. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula[C]//Security and Cryptography for Networks. Berlin: Springer, 2014: 277-297.
[11] KOPPULA V, WATERS B. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption[C]//Advances in Cryptology-CRYPTO 2019. Berlin: Springer, 2019: 671-700.
[12] LIANG X H, CAO Z F, LIN H, et al. Attribute based proxy re-encryption with delegating capabilities[C]//Proceedings of the 4th International Symposium on Information, Computer, and Communications Security. New York: ACM Press, 2009: 276-286.
[13] SUN J F, XU G W, ZHANG T W, et al. Verifiable, fair and privacy-preserving broadcast authorization for flexible data sharing in clouds[J]. IEEE Transactions on Information Forensics and Security, 2022, 18: 683-698.
[14] WANG L L, LIN Y, YAO T, et al. FABRIC: fast and secure unbounded cross-system encrypted data sharing in cloud computing[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(6): 5130-5142.
[15] GE C P, SUSILO W, FANG L M, et al. A CCA-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system[J]. Designs, Codes and Cryptography, 2018, 86(11): 2587-2603.
[16] LIANG K T, AU M H, LIU J K, et al. A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing[J]. Future Generation Computer Systems, 2015, 52: 95-108.
[17] ZHENG Q J, XU S H, ATENIESE G. VABKS: verifiable attribute-based keyword search over outsourced encrypted data[C]//Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2014: 522-530.
[18] CHENG L X, MENG F. Server-aided public key authenticated searchable encryption with constant ciphertext and constant trapdoor[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 1388-1400.
[19] LUO F C, WANG H Y, LIN C L, et al. ABAEKS: attribute-based authenticated encryption with keyword search over outsourced encrypted data[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 4970-4983.
[20] JIANG J W, WANG D. QPASE: quantum-resistant password-authenticated searchable encryption for cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 4231-4246.
[21] LIU X Y, YANG X T, LUO Y K, et al. Verifiable multikeyword search encryption scheme with anonymous key generation for medical Internet of things[J]. IEEE Internet of Things Journal, 2022, 9(22): 22315-22326.
[22] ZHANG Y H, ZHU T, GUO R, et al. Multi-keyword searchable and verifiable attribute-based encryption over cloud data[J]. IEEE Transactions on Cloud Computing, 2023, 11(1): 971-983.
[23] JIANG S H, WU J. A blockchain-powered data market for multi-user cooperative search[J]. IEEE Transactions on Network and Service Management, 2022, 19(1): 203-215.
[24] ZHOU Y J, CAO Z F, DONG X L, et al. BLDSS: a blockchain-based lightweight searchable data sharing scheme in vehicular social networks[J]. IEEE Internet of Things Journal, 2022, 10(9): 7974-7992.
[25] AGYEKUM K O O, XIA Q, SIFAH E B, et al. A proxy re-encryption approach to secure data sharing in the Internet of Things based on blockchain[J]. IEEE Systems Journal, 2022, 16(1): 1685-1696.
[26] ZHANG D, WANG S P, ZHANG Q, et al. Attribute based conjunctive keywords search with verifiability and fair payment using blockchain[J]. IEEE Transactions on Services Computing, 2023, 16(6): 4168-4182.
[27] LI Q, ZHANG Y H, ZHANG T, et al. HTAC: fine-grained policy-hiding and traceable access control in mHealth[J]. IEEE Access, 2020, 8: 123430-123439.
[28] MA R N, ZHANG L Y, WU Q, et al. BE-TRDSS: blockchain-enabled secure and efficient traceable-revocable data-sharing scheme in industrial Internet of things[J]. IEEE Transactions on Industrial Informatics, 2023, 19(11): 10821-10830.
[29] FISCO-BCOS[EB]. 2025.

[作者简介]

张波（1981– ），男，山东德州人，博士，济南大学副教授、硕士生导师，主要研究方向为网络与信息安全、区块链等。

李哲成（2001– ），男，山东枣庄人，济南大学硕士生，主要研究方向为区块链、数据共享等。

徐兴帅（1999– ），男，山东菏泽人，济南大学硕士生，主要研究方向为区块链、数据共享等。