基于区块链的多关键字属性基可搜索加密方案

牛淑芬[1], 韩　松[1], 谢亚亚[1], 王彩芬[2]

[1]西北师范大学 计算机科学与工程学院 兰州 中国 730070
[2]深圳技术大学 大数据与互联网学院 深圳 中国 518118

摘要　云存储服务的出现可将文件上传至云服务器, 节约了本地的信息存储空间以及管理开销。文件以明文的形式存储显然无法满足隐私保护和安全需求, 但若将加密后的文件上传至云服务器, 将失去搜索原文件的能力。因此, 可搜索加密技术的出现解决了用户如何在文件不解密的情况下搜索加密数据。目前现有的单关键字可搜索加密方案会产生许多与检索内容不符合的信息, 没有考虑数据用户细粒度搜索权限和搜索效率, 以及因云存储的集中化带来的数据安全和隐私保护等问题。针对以上问题, 该文提出了基于区块链的多关键字属性基可搜索加密方案。该方案使用多关键字可搜索加密技术实现了加密数据的有效搜索; 利用基于属性的加密技术实现加密数据的细粒度访问控制; 结合区块链的智能合约技术, 经过多笔交易获得搜索结果。并且利用区块链的不可篡改性, 满足了方案中相关性质的公平性, 保证了在方案中三方的公平性和安全性并进行了相关分析。在随机预言机模型下, 基于困难问题假设证明了方案的关键字安全及陷门安全, 即所提方案满足在选择关键字攻击下的关键字密文不可区分性安全和陷门不可区分性安全。最后通过数值分析表明该方案在关键字密文生成阶段和关键字搜索阶段具有较高的效率。并展望了在未来的工作中考虑将其应用于电子病历数据共享等场景中, 以获得更实用的价值。

关键词　区块链; 云存储; 可搜索加密; 属性基加密
中图法分类号　TP309.7　DOI 号 10.19363/J.cnki.cn10-1380/tn.2023.01.10

Attribute-based Searchable Encryption Scheme Supporting Multiple Keywords Based on Blockchain

NIU Shufen[1], HAN Song[1], XIE Yaya[1], WANG Caifen[2]

[1]College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
[2]College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China

Abstract　The emergence of cloud storage services, now files can be uploaded to cloud servers, it saves local storage space and management overhead. The storage of files in plaintext obviously cannot meet the privacy and security requirements. However, if the encrypted files are uploaded to the server by traditional encryption, the server will lose the ability to search them by keywords. Therefore, the emergence of searchable encryption technology can effectively solve how to search encrypted data without decryption. At the moment, the existing traditional searchable encryption scheme supporting single keyword will produce many information that is not consistent with the retrieval content. And it does not consider the problem of fine-grained search permission and search efficiency of data users, as well as the problem of data security and privacy preservation caused by the centralization of cloud storage in the existing searchable encryption scheme. According to the above problems, an attribute-based searchable encryption scheme supporting multiple keywords based on blockchain was proposed. In this scheme, multi-keyword searchable encryption technology is used to achieve effective search of encrypted data, attributed-based encryption technology is used to realize fine-grained access control of data. Through combining the smart contract technology of blockchain, the search results are obtained through multiple transactions to guarantee the fairness and security of the scheme. The scheme should also satisfy the fairness of the relevant property, so the immutability property of blockchain is used to ensure the fairness and security of data users, the data owner and the cloud server in the scheme. In addition, we conducted a relevant analysis. Under the random oracle model, based on the decisional bilinear Diffie-Hellman assumption and decisional Diffie-Hellman assumption of difficult problems, it is proved that the scheme can guarantee the security of keyword and trapdoor. The numerical experimental results show that the proposed scheme is more efficient in the ciphertext generation phase and keyword search phase. In the future work, it is considered to be applied to electronic medical record data sharing and other scenarios in order to obtain more practical value.

Key words　blockchain; cloud storage; searchable encryption; attribute-based encryption

通讯作者：韩松, 硕士, Email:565904313@qq.com。
国家自然科学基金资助项目(No. 61662069, No. 61662071, No. 61772022)。
收稿日期: 2021-04-14; 修改日期: 2021-12-25; 定稿日期: 2022-11-03

1 引言

随着基于云的外包服务模式日益普及, 越来越多的如医疗数据或者公司报表等信息被外包存储到云服务器。为了保护用户的隐私, 数据在上传到云服务器前是经过加密的。而加密通常会隐藏原始数据的特征, 因此云服务器很难使用传统的数据搜索机制对加密数据进行搜索, 故传统的加密方式在云环境下并不适用, 可搜索加密[1]的概念被及时地提出。与传统的加密方式相比, 可搜索加密机制支持对加密数据的有效搜索。

Song 等人[1]首次在对称密钥的基础下解决了对加密数据的检索问题, 并对所提出的方案给出了安全性证明, 其不足之处在于搜索效率较低。Boneh 等人[2]针对加密邮件系统, 讨论了使用公钥加密数据的搜索问题, 提出公钥关键字可搜索加密的概念。并给出了几种公钥可搜索加密的构造方案, 但效率较低。虽然关键字可搜索加密技术可以解决在不解密的情况下搜索加密数据的问题, 但单关键字查询会浪费网络带宽和计算资源, 不适用于现实的场景。Fan 等人[3]提出了一种支持搜索结果验证的多关键字可搜索加密方案, 但其在半可信的云服务器下并不能保证返回的搜索结果一定是正确的, 且其没有考虑用户搜索授权的问题。

文献[1-3]中搜索用户均被赋予了无限的搜索能力, 可以采用任意关键字从云服务器获取包含所要搜索关键字的加密数据, 这导致数据属主无法对外包加密数据实施有效的细粒度访问控制。作为实现细粒度数据共享的实用技术, 基于属性的加密技术得到了广泛的关注。Waters 和 Sahai[4]首次提出了基于属性加密的密码原语, 该方法被认为是一种高效的加密媒介, 在云存储中具有细粒度的访问控制。基于属性的加密可分为密文策略的属性基加密[5]和密钥策略的属性基加密[6]。在密文策略的属性基加密中, 访问策略被嵌入到密文中, 用户的密钥与其属性相关联。文献[5-11]提出了几种带关键字搜索授权的可搜索加密方案, 实现了对加密数据的有效搜索和数据访问控制。在文献[5]中, 属性用于描述用户的凭证, 数据属主决定了谁可以解密的策略, 在安全性方面可以抵抗合谋攻击。Li 等人[7]提出了一种基于属性、支持多关键字搜索的可搜索加密方案。当且仅当由一系列属性定义的用户满足访问结构时, 该方案允许多个用户正确查询关键字密文。文献[8]设计了云环境中可验证的基于属性的关键字搜索方案, 该方案支持可伸缩的、细粒度的所有者强制加密数据搜索, 可以同时实现系统的可扩展性和细粒度性。文献[9]针对恶意用户和恶意云服务提供商对加密的数据文件进行非法搜索的问题, 提出了基于云存储的可信属性基可搜索加密方案。

文献[10-12]指明区块链是一个去中心化的分布式存储系统, 能够提供平台支持, 生成永久、不可逆向修改的记录。文献[13]提出了区块链中高效的、保护隐私的、可追踪的属性基可搜索加密方案, 利用区块链技术保证了数据的完整性和不可篡改性。Li 等人[14]基于区块链技术, 给出了一个可行的解决方案来解决对称可搜索加密中出现的公平性问题。在该方案中, 用户无需验证即可自动获得搜索结果, 云服务器将导致其丢失押金且无法获得服务费。文献[15]介绍了区块链技术智能合约的发展与前景, 文献[16-20]分别介绍了基于云存储的属性基密码算法。其中胡等人[18] 提出了一个符合实际应用需求的支持代理重加密的隐藏访问结构的、基于属性的密文检索方案, 实现了当授权用户不在线时将密文搜索和解密权限委托给其他用户, 从而实现数据和密文的进一步安全有效的共享, 但其为单关键字可搜索加密, 不能验证所得到的搜索结果是否正确, 且未考虑云服务器的半可信问题。Zheng 等人[19]提出了可验证的基于属性的关键字可搜索加密方案, 其基于云存储且为单关键字, 效率较低。而区块链具有去信任、去中心化、开放自治、匿名可溯源、信息不可篡改等特性, 基于区块链的分布式架构、共识算法等, 智能合约允许相互不信任的用户在不需要任何第三方可信中介或权威的情况下完成交易, 同时, 数字形式的智能合约可灵活嵌入各种有形或无形的资产、交易和数据中, 实现主动或被动的资产、信息管理与控制, 逐步构建可编程的智能资产、系统及社会[15], 能够有效地解决云存储不可信等问题。

本文利用区块链构造了一个公平的多关键字属性基可搜索加密方案。数据文件被加密并存储在云服务器上。本文主要有如下三部分的工作和贡献:

(1) 本文利用可搜索加密和基于属性的加密技术, 设计了一个密文策略的属性基多关键字可搜索加密方案。可搜索加密用以实现对加密数据的有效搜索, 且多关键字保证了其搜索效率。基于属性的加密可以实施加密数据的细粒度访问控制以提高数据使用的灵活性。

(2) 针对云服务器因集中化对数据安全和隐私保护带来的威胁, 并基于区块链去中心化、匿名性、不可篡改性和可验证性等特点, 将区块链技术应用于上述方案。其中, 区块链中的智能合约技术保证了

数据用户和云服务器之间的公平性, 若数据用户不诚实, 则不能从云服务器获得正确的结果, 若云服务器是恶意的, 则不能得到服务费。在整个过程中, 云服务器不能得到任何关于关键字和数据明文的有效信息。且在整个过程中, 需要多笔交易才能获得正确的搜索结果。

(3) 本文在 Linux Ubuntu-10.10 操作系统下利用 PBC, 用 C 语言进行编程, 对本文方案、文献[18]和文献[19]的方案进行了数值实验。实验结果表明本文方案的效率较高于文献[18]和[19]。

2 相关知识

本节给出双线性对定义、困难问题假设、基于属性加密方案以及区块链的相关定义。

2.1 双线性对和困难问题假设

定义1(双线性对)令 G_1 和 G_2 为两个阶为素数 p 的乘法循环群, 定义一个双线性映射 $e:G_1\times G_1\to G_2$ 满足如下性质:

1) 双线性: 对任意的 $u,v\in G_1$, 存在 $a,b\in Z_p^*$, 使得 $e(u^a,v^b)=e(u,v)^{ab}$。

2) 非退化性: 存在 $u,v\in G_1$, 使得 $e(u,v)\neq 1$。

3) 可计算性: 对任意的 $u,v\in G_1$, 存在有效算法计算 $e(u,v)$。

定义 2 判定性双线性(Diffie-Hellman Decisional bilinear Diffie-Hellman, DBDH)假设。G_1 和 G_2 是阶为素数 p 的循环群 $e:G_1\times G_1\to G_2$ 为一个双线性映射, g 为 G_1 的生成元, 给定两个元组 $(g,g^a,g^b,g^c,e(g,g)^{abc})$ 和 $(g,g^a,g^b,g^c,e(g,g)^z)$, 对随机的 $a,b,c,z\in Z_q^*$, 不存在概率多项式时间的攻击者以不可忽略的优势区分 $\left(g^a,g^b,g^z\right)$ 和 $\left(g^a,g^b,g^{ab}\right)$。

定义 3 判定性 Diffie-Hellman(Decisional Diffie-Hellman,DDH)假设。g 为 G_1 的生成元, 给定两个三元组 $\left(g^a,g^b,g^z\right)$ 和 $\left(g^a,g^b,g^{ab}\right)$, 对随机的 $a,b,z\in Z_q^*$, 不存在概率多项式时间的攻击者以不可忽略的优势区分 $\left(g^a,g^b,g^z\right)$ 和 (g^a,g^b,g^{ab})。

2.2 访问结构和访问树

定义 4(访问结构)令 $A=\{A_1,A_2,...,A_n\}$ 为一个属性集合。令集合 $\Gamma\in 2^{\{A_1,A_2,...,A_n\}}$, $\forall B,C$:若 $B\in\Gamma$ 且 $B\subseteq C$, 那么 $C\in\Gamma$。若上述关系成立, 则称 Γ 是单调的。因此一个访问结构 Γ 就是由 $A=\{A_1,A_2,\cdots,A_n\}$ 的非空属性子集构成的集合。在 Γ 中的集合被称为授权访问集合, 不在 Γ 中的集合称为非授权访问集合。

定义 5(访问树) T 表示一个访问树。T 中的每个非叶子节点 x 都可以表示成如 (n_x,k_x) 的门限结构, 其中 n_x 表示的孩子节点个数, k_x 表示门限值, 其中 $0\leqslant k_x\leqslant n_x$。$k_x=1$时表示“OR”门, $k_x=n_x$ 则表示“AND”门。叶子节点 x 用来描述属性, 我们规定其门限值 $k_x=1$。

2.3 智能合约和交易单

定义 6(以太坊智能合约)智能合约一般具有值和属性两个状态, 智能合约经多方共同协定, 各自签署后随用户发起的交易提交, 经 P2P 网络传播、矿工验证后存储在区块链特定区块中, 用户得到返回的合约地址及合约接口信息后即可通过发起交易来调用合约。矿工受系统预设的激励机制激励, 将贡献自身算力来验证交易, 矿工收到合约创建或调用交易后在以太坊虚拟机中创建合约或执行合约代码, 合约代码根据可信外部数据源和世界状态的检查信息自动判断当前所处场景是否满足合约触发条件以严格执行响应规则并更新世界状态。交易验证有效后被打包进新的数据区块, 新区块经共识机制认证后链接到区块链主链, 所有更新生效[15]。

定义 7(交易单)比特币系统由地址和它们之间的交易组成。地址通常是用户公钥的散列值。当用户希望构建一个交易单时, 每个用户都可以拥有一对密钥, 即一个私钥和一个公钥。公钥用于验证交易的签名是否有效, 而私钥用于对该事务签名。为了简洁起见, 这对密钥被表示为 $\left(A.pk,A.sk\right)$。令 $\sigma=\mathrm{sig}_A(T)$ 表示 A 的私钥对交易 T 的签名, $\mathrm{Ver}_A(T,\sigma)$ 表示 A 的公钥对交易 T 的验证。在比特币系统中, 每笔交易可以有多个输入和两个输出。交易的输入脚本与前一个交易的输出脚本相关联。每个交易存在一个时间戳 t, 表示交易只在时间 t 之后生效。当输出脚本被验证为有效, 所涉及的交易不会被赎回, 且至少经过 6 个区块确认后, 一个交易才能被证明有效。

3 形式化定义与安全模型

3.1 系统模型

基于区块链的多关键字属性基可搜索加密方案的系统模型如图 1 所示, 适用于一对多的搜索场景。

主要包括数据属主、数据用户、云服务器、区块链和可信的属性授权中心 5 个实体。

(1) 数据属主: 数据属主将加密的数据文件 $C=\{C_1,C_2,...,C_n\}$ 和关键字索引 I 上传至云服务器, 当用户的属性满足访问结构时, 用户对关键字具有搜索权限。并构建交易向满足属性的用户提供正确的对称密钥 K, 否则将失去交易的费用。

(2) 数据用户: 数据用户要搜索包含关键字集 $w'=\{w_1',w_2',...,w_m'\}$ 的文件时, 产生陷门信息 T 并将其发送给云服务器。并构建交易保证与云服务器诚实履行协议, 否则将失去保证金。

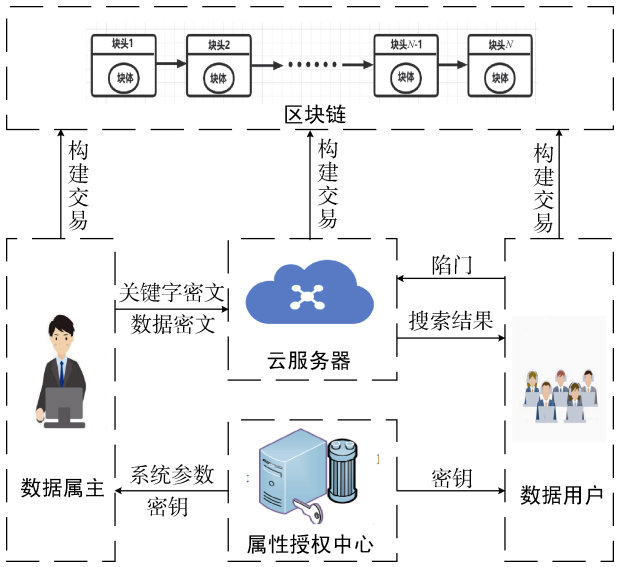

图 1 系统模型

Figure 1 System model

(3) 云服务器: 云服务器需要向提供正确陷门的数据用户提供数据搜索结果, 并通过用户生成的交易获得搜索功能相应的服务器费用, 否则云服务器无法得到费用。

(4) 区块链: 基于区块链中的智能合约技术, 将构建的交易广播在区块链中, 拥有自动执行以及内容不可篡改等性质, 用来保证每一方诚实的履行交易协议。

(5) 属性授权中心: 为系统产生参数, 并为数据属主以及数据用户生成相应的密钥。

本文方案中在区块链中的交易过程可以分为 4 个阶段, 且一次完整的数据搜索可以执行 6 次交易。本文方案通过区块链上的交易使数据用户和数据属主以及云服务器之间需要诚实的执行协议, 保证了方案的安全性和公平性, 模型如图 2 所示。

(1) 第一阶段: 数据用户首先构建一个交易 A 并设置违约金, 其中交易接收方是用户自身或云服务器。

(2) 第二阶段: 数据用户构建交易 Ask 来向用户属主询问对称 密钥 K 并设置保证金。如果数据属主提供正确, 则可以使用交易 Pay 来获得交易 Ask 中的保证金, 否则数据用户构建交易 Withdraw 来赎回交易 Ask 中的保证金。

(3) 第三阶段: 数据用户构建交易 G 用来获得正确的搜索结果并设置服务费, 若云服务器提供正确的结果, 则使用交易 P 来获得交易 G 中的服务费, 否则数据用户构建交易 F 来追回交易 G 中的服务费。

(4) 第四阶段: 当最后用户获得正确的搜索数据后, 数据用户将验证结果构建交易 R 赎回违约金,否则云服务器构建交易 C 来得到交易 A 中的违约金。

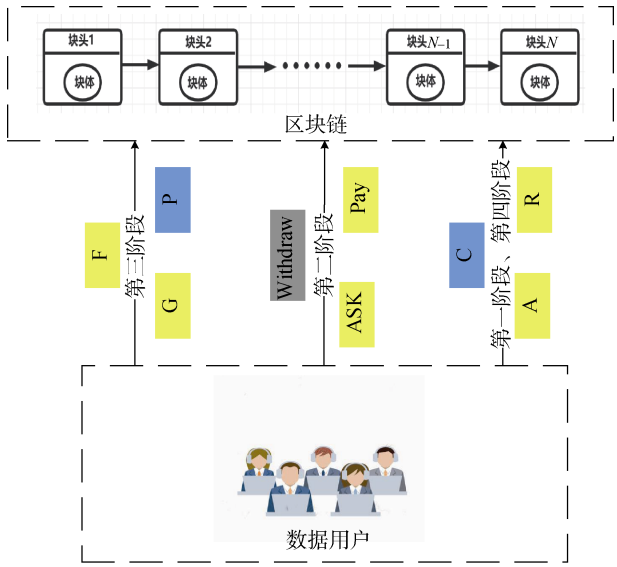

图 2 交易过程模型

Figure 2 Transactions model

3.2 算法形式化定义

基于区块链的多关键字属性基可搜索加密方案包括以下 7 个概率多项式时间算法。

$\text{SetUp}(1^{\lambda})\rightarrow(\text{Param},SK)$: 输入安全参数 λ 和公开的双线性映射 (G_1,G_2,e,p,g), 属性授权中心 (Attribute Authority, AA) 运行该算法输出系统公开参数 Param 和密钥 SK。

$\text{KeyGen}(\text{Param},S)\rightarrow SK_u$: 输入系统公开参数 Param 和用户的属性集 S, AA 运行该算法输出用户的密钥 SK_u。

$\text{Encrypt}(\text{Param},SK,w,T,F_i,K_i)\rightarrow(I_w,C_i,h(F_i))$: 输入系统公开参数 Param、数据属主的密钥 SK、关

键字集 w 和访问树结构 T, 数据属主运行该算法输出关键字密文 I_w、加密的数据文件 C_i 和数据文件 F_i 对应的哈希值 $h(F_i)$。

Trapdoor(Param, SK_u, w', T_u) $\rightarrow (T_w, \mathrm{A}, \mathrm{C})$: 输入系统公开参数 Param、用户的密钥 SK_u 和要查询的关键字集 w' 和一个未被赎回的交易 T_u, 用户运行该算法输出陷门信息 T_w、交易 A 和交易 C。

Search(Param, I_w, T_w) $\rightarrow 1/0$: 输入系统公开参数 Param、关键字密文 I_w 和用户陷门 T_w, 云服务器运行该算法。若用户的属性集 S 满足访问树结构, 关键字密文 I_w 中包含的关键字和用户陷门 T_w 中包含的关键字相同, 则输出 1 并将对应的加密数据文件 C_i 和 $h(F_i)$ 发送给用户, 否则输出 0 并终止。

Decrypt(Param, C_i, K, T_{u1}) $\rightarrow (F_i, \mathrm{Ask}, \mathrm{Pay/Withdraw})$: 输入系统公开参数 Param、数据文件密文 C_i、对称密钥 K_i 和一个未被赎回的交易 T_{u1}, 用户运行该算法输出数据文件明文 F_i、交易 Ask 和交易 Pay (或交易 Withdraw)。

Verify(T_{u2}, T_{u3}, $h(f)$, z_1, z_2, DB(w)) $\rightarrow (\mathrm{R}, \mathrm{G}, \mathrm{P/F})$: 输入未被赎回的交易 T_{u2}, T_{u3} 和计算的文件哈希值 $h(f)$ 以及 z_1, z_2, DB(w), 用户输出交易 R、交易 G、交易 P (或交易 F), 计算解密得到的数据明文的哈希值 $h'(f)$ 并验证 $h(f)$=$h'(f)$ 是否成立。

3.3 安全模型

本文通过概率多项式时间攻击者 $\mathcal{A}$ 和挑战者 $\mathcal{B}$ 之间的游戏来定义方案在选择关键字攻击下的关键字密文不可区分性安全和陷门不可区分性安全。

游戏 1. 关键字密文不可区分性.

初始阶段:挑战者 $\mathcal{B}$ 运行系统建立算法返回系统公开参数, $\mathcal{A}$ 定义挑战访问树 T^*.

阶段1: 在这个阶段 $\mathcal{A}$ 适应性地进行多项式有界次以下询问.

密钥提取询问: $\mathcal{A}$ 适应性地向挑战者 $\mathcal{B}$ 询问与属性集 $S_1, S_2, \cdots, S_n$ 相关联的密钥.

关键字密文询问: $\mathcal{A}$ 适应性地向 $\mathcal{B}$ 询问索引的关键字 w_i, $1 \leqslant i \leqslant m$ 的密文.在这个过程中询问到的密钥都不满足访问树 T^*.

挑战: $\mathcal{A}$ 向 $\mathcal{B}$ 提交两个挑战关键字 w_0 和 w_1. $\mathcal{B}$ 随机地选择一个比特 $\mu \in \{0,1\}$, 然后 w_μ 得到关键字密文 I_{w_μ}, 并将其返回给攻击者 $\mathcal{A}$.

阶段2: $\mathcal{A}$ 像阶段1一样继续发起一系列对应于属性集为 $S_{q+1}, S_{q+2}, \cdots$ 的询问, 要求询问到的密钥均不满足访问树 T^*.

猜测: $\mathcal{A}$ 输出 μ 的猜测 $\mu' \in \{0,1\}$, 若 $\mu' = \mu$ 则 $\mathcal{A}$ 赢得游戏 1.

攻击者 $\mathcal{A}$ 成功地赢得游戏 1 的优势可被表示为: $\mathrm{Adv}_{\mathcal{A}}^{C}(\lambda) = |\Pr[\mu' = \mu] - \frac{1}{2}|$

对于概率多项式时间的攻击者 $\mathcal{A}$, 若 $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{C}}(\lambda)$ 是可忽略的, 则称方案满足关键字密文不可区分性.

游戏 2. 陷门不可区分性.

假设 $\mathcal{A}$ 是一个试图攻破陷门不可区分性的多项式时间攻击者.挑战者 $\mathcal{B}$ 通过建立算法解决 DDH 问题, 挑战者 $\mathcal{B}$ 获得实例 $F = (G_1, G_2, e, p, g, a, b, g^{ab})$.

初始阶段: $\mathcal{B}$ 运行系统建立算法返回系统公开参数.

阶段 1:在这个阶段 $\mathcal{A}$ 适应性地进行多项式有界次以下询问.

密钥提取询问: $\mathcal{B}$ 运行密钥生成算法计算 SK_u, 并将密钥 SK_u 返回给 $\mathcal{A}$.

陷门询问: $\mathcal{A}$ 适应性地向 $\mathcal{B}$ 询问想要搜索的关键字 w_i, $1 \leqslant i \leqslant m$ 的陷门时, $\mathcal{B}$ 计算产生相应的陷门 T_w, 并将其返回给 $\mathcal{A}$.

挑战: $\mathcal{A}$ 向 $\mathcal{B}$ 提交两个挑战关键字 w_0 和 w_1. $\mathcal{B}$ 随机选取 $\mu \in \{0,1\}$, 并且利用 w_μ 得到陷门 T_{w_μ}, 并将其返回给 $\mathcal{A}$.

阶段 2: $\mathcal{A}$ 像阶段 1 一样继续发起一系列询问, 但不能询问与挑战关键字有关的信息.

猜测: $\mathcal{A}$ 输出 $\mu' \in \{0,1\}$, 若 $\mu' = \mu$ 则 $\mathcal{A}$ 赢得游戏 2.

$\mathcal{A}$ 成功地赢得游戏 2 的优势可被表示为: $\mathrm{Adv}_{\mathcal{A}}^{T}(\lambda) = |\Pr[\mu' = \mu] - \frac{1}{2}|$.

对于概率多项式时间的攻击者 $\mathcal{A}$, 若 $\mathrm{Adv}_{\mathcal{A}}^{T}(\lambda)$ 可忽略, 则称方案满足陷门不可区分性。

3.4 设计目标

在该方案的设计中, 除了需要保证密文不可区分性安全和陷门不可区分性安全, 还应满足以下相关性质的公平性:

如果三方诚实公平地执行协议, 用户可以得到正确的搜索结果, 数据属主和云服务器也可以获得相应的服务费。

(1) 如果用户不诚实或者首先终止了搜索协议, 除了丢失押金外, 也无法得到正确的搜索结果。

(2) 如果数据属主不诚实或者提供了错误信息, 将会失去相应的保证金。

(3) 如果服务器不诚实或恶意提供错误数据, 除了无法获取相关明文信息, 同时也无法获得服务费。

4 方案构造

基于区块链的多关键字属性基可搜索加密方案可以分为 4 个阶段: 系统建立、数据加密、数据搜索、数据验证。

阶段 1 本阶段主要包括系统初始化和密钥生成两个步骤:

系统初始化 (SetUp) :属性授权中心 AA 执行该算法, 输入安全参数 λ。

(1) 产生一个双线性映射 $e: G_1 \times G_1 \to G_2$, 其中 G_1 和 G_2 是阶为素数 p 的循环乘群, g 是 G_1 的生成元。

(2) 定义两个抗碰撞的哈希函数 $H: \{0,1\}^* \to Z_q^*$, $H_1: \{0,1\}^* \to G_1$。

(3) 定义 Lagrange 系数: $\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$, S 表示属性集合, $i, j \in Z_q^*$。

(4) 随机选择 $\alpha, \beta \in Z_q^*$, 计算 $g^\alpha, g^\beta, e(g,g)^\alpha$。返回系统公开参数 $\text{Param} = \{G_1, G_2, e, g, H, H_1\}$ 和数据属主的密钥 $SK = \{e(g,g)^\alpha, g^\beta\}$。

密钥生成 (KeyGen) :属性授权中心 AA 执行该算法, 为数据用户产生与其属性集 S 相关联的密钥。

(1) 随机选择 $r \in Z_q^*$, 计算 $SK_{u1} = g^{\frac{\alpha+r}{\beta}}$, $SK_{u2} = g^{\frac{1}{\beta}}$, $SK_{u3} = g^r$。

(2) 对 $\forall \text{att} \in S$, 随机选择 $r_a \in Z_q^*$, 并计算 $SK_{ua} = SK_{u3} \times H_1(att)^{r_a} = g^r \times H_1(att)^{r_a}, SK_{ua}{}' = g^{r_a}$。最后得到 $SK_u = \{SK_{u1}, SK_{u2}, SK_{u3}, \{SK_{ua}, SK_{ua}{}'\}_{\text{att} \in S}\}$, 并将其返回给数据用户。

阶段 2 本阶段主要包括数据明文的加密和关键字的加密两个步骤:

明文加密 (Encrypt) : 数据属主选定文件集 $F = \{F_1, F_2, ..., F_n\}$, 并随机选择 $K_j \in Z_q^*$ 作为对称密钥, 其中 $j \in [1,n]$。对于 $f \in F$ 计算对应的哈希值 $h(f)$, 并计算得到加密后的数据文件 $C = \varepsilon.\text{Enc}_K f$。其中, $\varepsilon.\text{Enc}$ 表示安全的对称加密算法。

关键字加密 (Encrypt) :由数据属主执行该算法, 对于 $f \in F$, 提取关键字集 $w = \{w_1, w_2, ..., w_m\}$。选择一个大小为 n 的空数组 $\text{DB}(w_i)$:若第 j 个文档包含关键字 w_i, 则 $\text{DB}(w_i)[j] = 1$, 否则 $\text{DB}(w_i)[j] = 0$。

(1) 随机选择 $s \in Z_q^*$ 并作为秘密值, 计算 $C_{w_i} = e(g^{H(w_i)s}, g) e(g,g)^{\alpha s}$, $i \in [1,m]$ 和 $C_w{}' = g^{\beta s}$。

(2) 首先执行秘密共享算法, 对于每个在访问树 T 中的节点 x (包含叶子节点 t 在内), 选择一个多项式 q_x。这些多项式从根节点 t 开始选择, 具体步骤如下:

①对于访问树 T 中的每个节点, 使得多项式 q_x 的次数 $d_x = k_x - 1$, 其中 k_x 为该节点的门限值。

②从根节点 t 开始, 定义 $q_t(0) = s$, 然后随机选择多项式 q_t 的 d_t 个点, 完成对 q_t 的定义。对于其他节点 x, 定义 $q_x(0) = q_{\text{parent}(x)}(\text{index}(x))$, 并随机选择 d_x 个点完成对 q_x 的定义。

③令 X 为访问树 T 中的叶子节点集合, 对 $\forall x \in X$,计算 $C_x = g^{q_x(0)}, C_x{}' = H_1(\text{attr}(x))^{q_x(0)}$。

最后将关键字加密并生成索引为 $I_w = \{C_{w_i}, C_w{}', \{C_x, C_x{}'\}_{x \in X}\}$, 加密后的数据文件 C 以及数据明文的哈希值 $h(f)$ 返回给云服务器。

阶段 3 本阶段主要包括陷门生成和关键字搜索两个步骤:

陷门生成 (Trapdoor) :给定搜索关键字集 $w' = \{w_1', w_2', ..., w_m'\}$, 由数据用户执行该算法。

(1) 数据用户随机选择 $r_1 \in Z_q^*$, 并计算 $T_{1,i} = SK_{u1} \times SK_{u2}^{\sum_{i=1}^{n} H(w_i')} \times SK_{u2}^{r_1} = g^{\frac{\alpha + r + \sum_{i=1}^{n} H(w_i') + r_1}{\beta}}, i \in [1,m]$。

(2) 对于 $\forall \text{att} \in S$, 数据用户计算 $T_a{}' = SK_{ua}{}'$ 和 $T_a = SK_{ua} \times g^{r_1} = g^{r+r_1} \times H(att)^{r_a}$。

最后得到搜索陷门 $T_w = \{T_{1,i}, \{T_a, T_a{}'\}_{\text{att} \in S}\}$, 并将其返回给云服务器。与此同时, 数据用户按照以下方式构造如图 3 所示的交易 A :

①构建一笔未花费的价值为 d\$ 的交易 T_u, 接收者为数据用户自己或云服务器。

②将云服务器构建的智能合约 $\phi(\cdot)$ 嵌入到交易 A 的输出脚本中, 在这个智能合约中主要执行哈希操作, 形如 $\phi(x,y)$ 并判断 $h(x) = y$ 是否成立。

③用户和云服务器将交易 A 作为输入来计算交

易 C 的主体。然后用户将自己对交易 C 的签名发送给云服务器, 云服务器对其进行签名。交易 C 的时间戳 $t_{\max 1}$。表示在时间 $t_{\max 1}$ 之后, 云服务器广播交易 C。

④用户对交易 A 签名, 并将其广播至区块链。

⑤若交易 A 直到时间 $t_{\max 1}-\max_0$ 仍未出现在区块链上, 用户使用其私钥赎回交易 T_u 并退出协议, 其中 $\max_0$ 表示交易 A 加入到区块链上可能的最大时延。其中, 图 3 中 z_1 是云服务器产生的会话密钥, $z_2=h(\mathrm{DB}(w)\|z_1)$。

搜索 (Search) :给定加密的关键字索引 I_w 和搜索陷门 T_w, 由云服务器执行该算法。x 表示访问树 T 中的节点,算法运行如下:

(1) 若节点 x 是叶子节点, 令 $\mathrm{att}=\mathrm{attr}(x)$, 即 att 表示与叶子节点 x 相关联的属性。定义如下:

①若 $\mathrm{att}\in S$, 进行以下计算:

$$F_x=\frac{e(T_a,C_x)}{e(T_a',C_x')}=\frac{e(g^{r+r_1}\times H_1(\mathrm{att})^{r_a},g^{q_x(0)})}{e(g^{r_a},H_1(\mathrm{attr}(x))^{q_x(0)})}=$$

$$\frac{e(g^{r+r_1},g^{q_x(0)})e(H_1(\mathrm{att})^{r_a},g^{q_x(0)})}{e(g^{r_a},H_1(\mathrm{attr}(x))^{q_x(0)})}=$$

$$e(g,g)^{(r+r_1)q_x(0)}$$

②若 $\mathrm{att}\notin S$, 定义 $F_x=\perp$。

(2) 若该节点 x 是非叶子节点, 对于节点 x 的所有孩子节点 z, 执行算法后的结果记为 F_z, 集合 U_x 中保留 $F_z\neq\perp$ 的所有值:

①若 $|U_x|<k_x$, 表明 x 节点的孩子节点属性集合不满足该节点的门限值, 则终止并输出 $\perp$。

②若 $|U_x|\geq k_x$, 则表明 x 节点的孩子节点属性集合满足该节点的门限值, 则从集合 U_x 中随机挑选 k_x 个 F_z 的值, 结合 Lagrange 系数计算 F_x 的值:

$$F_x=\prod_{z\in U_x}F_z^{\Delta_{i,S_x}(0)}=\prod_{z\in U_x}(e(g,g)^{(r+r_1)q_z(0)})^{\Delta_{i,S_x}(0)}=$$

$$\prod_{z\in U_x}(e(g,g)^{(r+r_1)q_{\mathrm{parent}(z)}(\mathrm{index}(z))})^{\Delta_{i,S_x}(0)}=$$

$$\prod_{z\in U_x}e(g,g)^{(r+r_1)q_x(i)\Delta_{i,S_x}(0)}=e(g,g)^{(r+r_1)q_x(0)}$$

其中 $i=\mathrm{index}(z)$, $S_x=\{\forall z\in U_x:\mathrm{index}(z)\}$, Δ_{i,S_x} 为 Lagrange 系数。

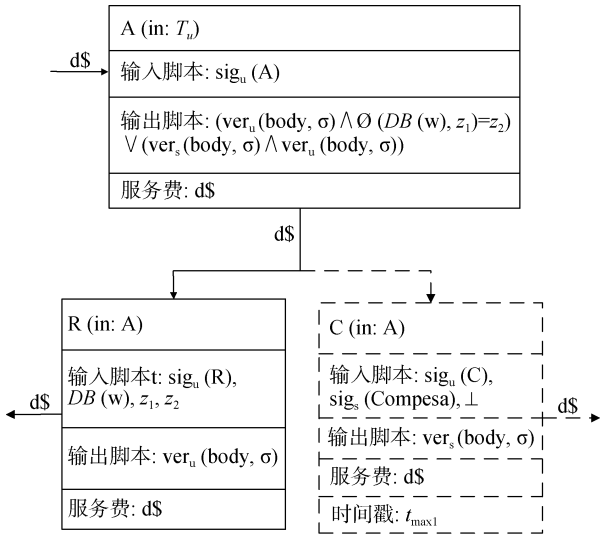

图 3 第一和第四阶段交易

Figure 3 Phase 1 and 4 transactions

(3) 若用户的属性集满足访问树, 递归计算得到最终执行结果为 $F_t=e(g,g)^{(r+r_1)q_t(0)}=e(g,g)^{(r+r_1)s}$。

正确性证明:

云服务器计算 $A=\dfrac{e(C_w',T_{1,i})}{F_t}$, 并验证 $A=\sum_{i=1}^{n}C_{w_i}$ 是否成立, 若等式成立, 则表明搜索成功, 说明用户的属性集 S 满足嵌在 I_w 中的访问树且 w 和 w' 一致。若等式不成立, 则表明搜索失败。搜索失败的情况可分为以下两种: 用户的属性集 S 不满足嵌在 I_w 中的访问树, 算法终止, 也就是说用户对 w 不具有搜索权限, 或者是用户对 w 有搜索权限, 但搜索时发现 w 和 w' 并不相同。

$$\sum_{i=1}^{n}C_{w_i}=\sum_{i=1}^{n}e(g^{H(w_i)s},g)e(g,g)^{\alpha s}=e(g^{s\sum_{i=1}^{n}H(w_i)},g)e(g,g)^{\alpha s}$$

$$A=\frac{e(C_w',T_{1,i})}{F_t}=\frac{e(g^{\beta s},g^{\frac{\alpha+r+\sum_{i=1}^{n}H(w_i)+r_1}{\beta}})}{e(g,g)^{(r+r_1)s}}=$$

$$\frac{e(g^s,g^{r+r_1})e(g^s,g^{\alpha+\sum_{i=1}^{n}H(w_i)})}{e(g,g)^{(r+r_1)s}}=e(g^s,g^{\alpha+\sum_{i=1}^{n}H(w_i)})=$$

$$e(g,g)^{\alpha s}e(g^{s\sum_{i=1}^{n}H(w_i)},g)$$

综上可得, $A=\sum_{i=1}^{n}C_{w_i}$。若搜索成功, 则云服务器将加密的数据文件 C 和 $h(f)$ 返回给用户, 由用户

进行后续的解密和验证工作。

阶段 4 本阶段主要包括数据解密和验证两个步骤:

解密 (Decrypt) :给定数据文件密文 C 和对称密钥 K，用户执行解密算法得到数据文件明文 $f=\varepsilon.\mathrm{Dec}_K\,C$，且 $\varepsilon.\mathrm{Dec}$ 表示安全的对称解密算法。通过以下方式获得数据明文，$\varphi(x,y)$ 表示数据用户创建的智能合约。在云服务器执行搜索并返回搜索结果后，用户构造交易 Ask 以得到数据明文，如图 4 所示:

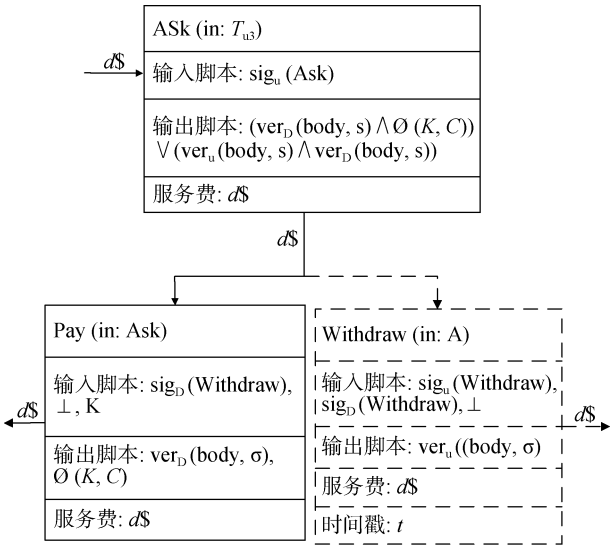

图 4 第二阶段交易

Figure 4 Phase 2 transactions

①构建一笔未花费的价值为 d\$ 的交易 T_{u1}，接收者为用户自己。

②将 $\varphi(K,C)$ 嵌入到交易 Ask 的输出脚本中。

③数据属主和用户将交易 Ask 作为输入来计算交易 Withdraw 的主体。然后数据属主将其对交易 Withdraw 的签名发送给用户，用户对其进行签名。交易 Withdraw 有时间戳 t，表示在时间 t 之后，用户可以广播交易 Withdraw 。

④数据属主对交易 Ask 签名，并将其广播至区块链。

⑤若交易 Ask 直到时间 $t-\max_1$ 仍未出现在区块链上，用户可以使用其私钥赎回交易 T_{u1} 并退出协议，其中 $\max_1$ 表示交易 Ask 加入到区块链上可能的最大时延。

数据属主利用交易 Ask 计算交易 Pay 的主体，并将 K 嵌入交易 Pay 的输入脚本。对其签名之后，广播交易 Pay 。用户在 P2P 网络上收交易 Pay 得到对称密钥 K，用户进行解密得到: $f=\varepsilon.\mathrm{Dec}_K\,C$ 。若能解密，交易被接受，否则被拒绝。若交易 Pay 直到时间 t 仍未出现在区块链上，用户广播交易 Withdraw 并追回他的钱。

验证 (Verify) :用户构造如图 5 所示的交易 G 来获得正确的搜索结果，如下所示:

①构建两笔未花费的价值分别为 d\$ 的交易 T_{u2} 和价值为 d_1\$ 的交易 T_{u3}，接收者为用户。

②将 $\psi(\mathrm{DB}(w),h(f),z_1,z_2)$ 嵌入到交易 G 的输出脚本中，其中 z_1 是云服务器产生的会话密钥，$z_2=h(\mathrm{DB}(w)\|z_1)$ 。

③将交易 T_{u2} 和 T_{u3} 作为输入来计算交易 G 的主体。用户和云服务器以交易 G 作为输入来计算交易 F 的主体，然后云服务器将自己对交易 F 的签名发送给用户，用户对其进行签名。交易 F 的时间戳 t_1，表示在时间 t_1 之后，用户可以广播交易 F 。

④用户对交易 G 签名，并将其广播至区块链。若交易 G 直到时间 $t_1-\max_3$ 仍未出现在区块链上，用户可以立即赎回交易 T_{u2} 和 T_{u3} 并退出协议，其中 $\max_3$ 表示交易 G 加入到区块链上可能的最大时延。

⑤用户使用交易 G 计算交易 P 的主体，并将数据属主计算的数据文件 f 的哈希值 $h(f)$ 和由用户自己计算的 $h'(f)$ 放入交易 P 的输入脚本中。然后，向云服务器发送对交易 P 的签名。

⑥云服务器将交易 G 作为输入来计算交易 P 的主体，然后在交易 P 的输入脚本中添加 $\mathrm{DB}(w),C_w,z_1,z_2$ 。对其签名之后，云服务器广播交易 P 。

⑦用户在 P2P 网络上收集交易 P，并验证: $h(f)=h'(f)$ 是否成立。若成立，则交易 P 被接受，否则被拒绝。

最后，用户以交易 A 为输入，将 $z_1,z_2,\mathrm{DB}(w)$ 放入如图 3 所示的交易 R 的输入脚本。对其签名之后，将其广播至区块链。用户在 P2P 网络上验证 $z_2=H(\mathrm{DB}(w)\|z_1)$ 是否成立，若成立则输出 1，表示交易 R 被接受。若交易 R 直到时间 $t_{\max}$ 仍未出现在区块链上，云服务器广播如图 3 所示的交易 C 去赎回交易 A 。

5 安全性证明和分析

5.1 关键字密文不可区分性

证明 $\mathcal{A}$ 是一个试图攻破关键字密文安全性的

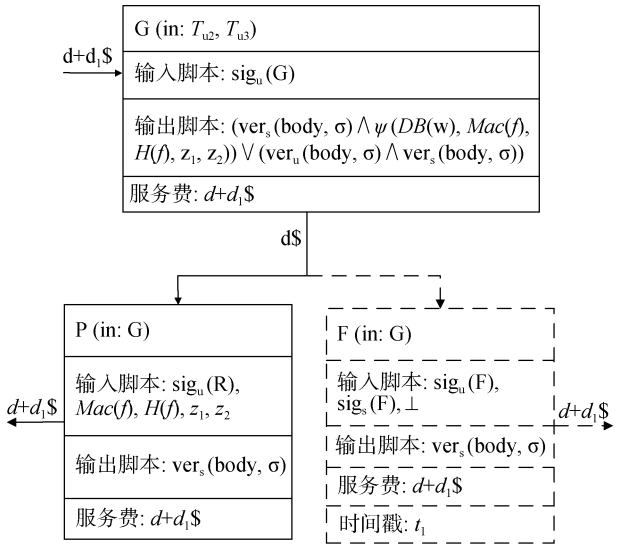

图 5　第三阶段交易

Figure 5　Phase 3 transactions

概率多项式时间攻击者, 挑战者 $\mathcal{B}$ 通过建立算法解决 DBDH 问题。

初始阶段: $\mathcal{B}$ 随机选择一个比特 $v\in\{0,1\}$, $t_0=(g,A=g^a,B=g^b,C=g^c,Z=e(g,g)^{abc})$, $t_1=(g,A=g^a,B=g^b,C=g^c,Z=e(g,g)^z)$ 且 a,b,c,z 都是从 Z_q^* 中随机选取的。

阶段 1:在此阶段 $\mathcal{A}$ 适应性地向由挑战者 $\mathcal{B}$ 模拟的预言机进行一系列询问。挑战者 $\mathcal{B}$ 计算公开参数 $Y=e(B,C)=e(g,g)^{bc}$, 并将其返回给攻击者 $\mathcal{A}$, 同时 $\mathcal{A}$ 定义挑战访问树 T^*。

密钥提取询问 O_{E}: $\mathcal{A}$ 自适应性地询问属性集 $S_1,S_2,...,S_n$ 对应的私钥 $SK_{\mathcal{A}}^{S_1},SK_{\mathcal{A}}^{S_2},...,SK_{\mathcal{A}}^{S_n}$。要求嵌入到相应私钥中的属性集 $S_1,S_2,...,S_n$ 都不满足挑战访问树, 且所有的私钥都能产生合法的搜索陷门。

关键字密文询问 O_{I}: $\mathcal{A}$ 自适应性地询问关键字 $k_1,k_2,...,k_m$ 对应的密文 $I_{k_1},I_{k_2},...,I_{k_m}$。

以上询问到的密钥和密文满足以下条件:给定一个私钥 $SK_{\mathcal{A}}^{S_i},\{i\in[1,n]\}$ 和一个关键字 $w_j,\{j\in[1,m]\}$, 计算得到搜索陷门为 $SK_{\mathcal{A}}^{S_i}(w_j)$。存在关键字密文 $I_{k_j\{j\in[1,m]\}}$, 使得搜索算法 $\mathrm{Search}(\mathrm{Param},I_{k_j},\mathrm{SK}_{\mathcal{A}}^{S_i}(w_j))=1$。

挑战: $\mathcal{A}$ 向 $\mathcal{B}$ 提交两个挑战关键字 w_0 和 w_1, 并将挑战访问树 T^* 返回给 $\mathcal{B}$。$\mathcal{B}$ 从 Z_q^* 中随机地选择一个比特 $\mu\in\{0,1\}$, 将关键字密文 $I_{w_\mu}^*(T^*,C_{w_\mu}^*=e(A^{H(w_\mu)},g)Z,(C_w')^*=A^\beta,\{C_x^*=g^{q_x(0)},(C_x')^*=H_1(\mathrm{attr}(x))^{q_x(0)}\}_{x\in X^*})$ 返回给 $\mathcal{A}$, 其中 X^* 表示挑战访问树 T^* 的叶子节点集合。

阶段 2:与阶段 1 类似, $\mathcal{A}$ 自适应性地询问属性集 $S_{n+1},S_{n+2},...$ 对应的私钥 $\mathrm{SK}_{\mathcal{A}}^{S_{n+1}},\mathrm{SK}_{\mathcal{A}}^{S_{n+2}},...$ 和关键字 $k_{m+1},k_{m+2},...$ 对应的密文 $I_{k_{m+1}},I_{k_{m+2}},...$。要求嵌入到相应密钥中的属性集 $S_{n+1},S_{n+2},...$ 都不满足挑战访问树 T^*。

猜测: $\mathcal{A}$ 输出 μ 的猜测比特 μ'。因为 $\mathcal{A}$ 询问到的属性集均不满足访问树 T^*, 故 $\mathcal{A}$ 无法通过搜索算法 $\mathrm{Search}(\mathrm{Param},I_{w_\mu}^*,\mathrm{SK}_{\mathcal{A}}^{S_i}(w_v))=1$ 来判断 μ=0 或 μ=1。因此 $\mathcal{A}$ 必须从 $I_{w_\mu}^*$ 中恢复关键字信息 $H(w_\mu)$ 以判定 μ=0 或 μ=1。

若 $v=0,Z=e(g,g)^{abc},I_{w_\mu}^*=\{T^*,C_{w_\mu}^*=e(A^{H(w_\mu)},g)e(g,g)^{abc},(C_w')^*=g^{a\beta},\{C_x^*=g^{q_x(0)},(C_x')^*=H_1(\mathrm{attr}(x))^{q_x(0)}\}_{\forall x\in X^*}\}$。在加密关键字时 α,s 都是随机选择的, 令 $a=s$, $bc=\alpha$ 则关键字密文可以被表示为 $I_{w_\mu}^*=\{T^*,C_{w_\mu}^*=e(g^{sH(w_\mu)},g)e(g,g)^{\alpha s},(C_w')^*=g^{s\beta},\{C_x^*=g^{q_x(0)},(C_x')^*=H_1(\mathrm{attr}(x))^{q_x(0)}\}_{x\in X^*}\}$。

若 $v=1$, $Z=e(g,g)^z$, 则关键字密文可以表示为 $I_{w_\mu}^*=\{T^*,C_{w_\mu}^*=e(g^{sH(w_\mu)},g)e(g,g)^z,(C_w')^*=g^{s\beta},\{C_x^*=g^{q_x(0)},(C_x')^*=H_1(\mathrm{attr}(x))^{q_x(0)}\}_{x\in X^*}\}$。因为 z 是一个随机元素, 故对攻击者 $\mathcal{A}$ 来说 $I_{w_\mu}^*$ 也是一个随机元素。

$\mathcal{A}$ 输出 μ 的猜测 μ'。若 μ=μ', $\mathcal{B}$ 输出 v 的猜测 v'。$\mathcal{A}$ 从 $I_{w_\mu}^*$ 恢复关键字信息 $H(w_\mu)$ 的优势为 $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{C}}(\lambda)$, 则 $\mathcal{A}$ 输出 μ=μ' 的概率是 $\frac{1}{2}+\mathrm{Adv}_{\mathcal{A}}^{\mathrm{C}}(\lambda)$。若 μ=μ', $\mathcal{B}$ 输出 μ 的猜测 μ'=1。$\mathcal{A}$ 输出 μ=μ' 的概率是 $\frac{1}{2}$。

故 $\mathcal{B}$ 在以上游戏中解决 DBDH 问题的优势为

$$\mathrm{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(\lambda)=|\frac{1}{2}\times\Pr[v=v'|v=0]+\frac{1}{2}\times\Pr[v=v'|v=1]-\frac{1}{2}|=$$

$$|[\frac{1}{2}\times(\frac{1}{2}+\mathrm{Adv}_{\mathcal{A}}^{\mathrm{C}}(\lambda))+\frac{1}{2}\times\frac{1}{2}]-\frac{1}{2}|=\frac{1}{2}\mathrm{Adv}_{\mathcal{A}}^{\mathrm{C}}(\lambda),$$

其中如果 $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{C}}(\lambda)$ 不可忽略, 则 $\mathrm{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(\lambda)$ 也不可忽略, 这与 DBDH 问题的困难性假设矛盾。

5.2 陷门不可区分性

证明 假设 $\mathcal{A}$ 是一个试图攻破陷门安全的概率多项式时间攻击者。挑战者 $\mathcal{B}$ 通过建立算法解决 DDH 问题, $\mathcal{B}$ 获得实例 $F=(G_1,G_2,e,p,g,a,b,g^{ab})$。

初始阶段: $\mathcal{B}$ 随机地选择 $a,r,r_1\in Z_q^*$, $\text{Param}=\{G_1,G_2,e,q,H,H_1,g,e(g,g)^{\alpha}=e(g,g)^{a-r-r_1-H_1(w)}, g^{\beta}=g^{1/b}\}$, 将 Param 返回给 $\mathcal{A}$。随机地选择 $v\in\{0,1\}$, 若 $v=1$, 令 $g^{\beta}=g^{1/b}$。若 $v=0$, 则随机选择 $y\in Z_q^*$, 得到 $g^{\beta}=g^{y}$。$\mathcal{B}$ 维护一个初始为空的列表 $L_H=\{<\cdot,\cdot,\cdot,\cdot>\}$。

阶段 1:在此阶段 $\mathcal{A}$ 适应性地进行如下询问, 并且假设 $\mathcal{A}$ 不会执行重复的询问。

哈希询问 O_H: 挑战者 $\mathcal{B}$ 从 L_H 中查找 $\{<w,\alpha,R_{(1)},v>\}$, 若 L_H 中 $R_{(1)}$ 不为空, 挑战者 $\mathcal{B}$ 提取私钥发送给攻击者 $\mathcal{A}$。否则, 若 $v=1$, 计算 $R_{(1)}=a-\alpha-r_1$ 并写入 L_H, 若 $v=0$, 计算 $R_{(1)}=g^{y}$ 并写入 L_H。

哈希询问 O_{H_1}:随机选择 $k\in G_1$, 并返回它作为 $H_1(\text{att})$ 的输出, 即 $R_{(2)}=k$。

密钥提取询问 O_{E}:若 $v=1$, 则中止。否则, 查看列表 L_H, $\mathcal{B}$ 执行密钥生成算法计算 SK_u, 并将密钥 SK_u 返回给 $\mathcal{A}$。其中 $\text{SK}_u=\{\text{SK}_{u1}=g^{\frac{\alpha+r}{y}},\text{SK}_{u2}=g^{\frac{1}{y}}, \text{SK}_{u3}=g^{r},\{r_a\in Z_q^*,\text{SK}_{ua}=g^{r}\times R_{(2)}^{r_a},\text{SK}_{ua}{}'=g^{r_a}\}_{\text{att}\in S}\}$

陷门询问 O_{T}:若 $v=1$, 查看列表 L_H 计算 T_w 并将其返回给 $\mathcal{A}$, 其中 $T_w=\{T_1=g^{\frac{\alpha+r+\sum_{i=1}^{m}H(w_i)+r_1}{\beta}}=g^{ab}, \{r_a\in Z_q^*,T_a=g^{r+r_1}\times H_1(\text{att})^{r_a}=g^{a-R_{(1)}-\alpha}\times R_{(2)}^{r_a},T_a'=g^{r_a}\}_{\text{att}\in S}\}$。否则 $v=0$, 则中止。

挑战: $\mathcal{A}$ 向 $\mathcal{B}$ 提交两个挑战关键字 w_0 和 w_1。$\mathcal{B}$ 随机地选择 $\mu\in\{0,1\}$, 计算陷门信息 $T_{w_\mu}^*=\{T_1^*=g^{\frac{\alpha+r+H(w_\mu)+r_1}{\beta}}=g^{ab},\{r_a\in Z_q^*,T_a^*=g^{r+r_1}\times H_1(\text{att})^{r_a} =g^{a-R_{(1)}-\alpha}\times R_{(2)}^{r_a},(T_a')^*=g^{r_a}\}_{\text{att}\in S}\}$ 并返回给 $\mathcal{A}$。

阶段 2:与阶段 1 一样。但不能询问与挑战关键字有关的信息。

猜测: $\mathcal{A}$ 输出 μ 的猜测 μ'。若 $\mu'=\mu$, 则表示挑战者 $\mathcal{B}$ 挑战成功, 输出 1, 否则输出 0。

$\mathcal{A}$ 输出 μ 的猜测 μ'。若 $\mu'=\mu$, $\mathcal{B}$ 输出 v 的猜测 v'。$\mathcal{A}$ 成功地赢得这个游戏的优势为 $\text{Adv}_{\mathcal{A}}^{\text{T}}(\lambda)$, 则 $\mathcal{A}$ 输出 $\mu'=\mu$ 的概率是 $\frac{1}{2}+\text{Adv}_{\mathcal{A}}^{\text{T}}(\lambda)$。若 $\mu'=\mu$, $\mathcal{B}$ 输出 μ 的猜测 $\mu'=1$。$\mathcal{A}$ 输出 $\mu'=\mu$ 的概率是 $\frac{1}{2}$。

故 $\mathcal{B}$ 在以上游戏中解决 DDH 问题的优势为 $\text{Adv}_{\mathcal{B}}^{\text{DDH}}(\lambda)=|\frac{1}{2}\times\Pr[v=v'|v=0]+\frac{1}{2}\times\Pr[v=v'|v=1]-\frac{1}{2}| =|[\frac{1}{2}\times(\frac{1}{2}+\text{Adv}_{\mathcal{A}}^{\text{T}}(\lambda))+\frac{1}{2}\times\frac{1}{2}]-\frac{1}{2}|=\frac{1}{2}\text{Adv}_{\mathcal{A}}^{\text{T}}(\lambda)$, 其中如果 $\text{Adv}_{\mathcal{A}}^{\text{T}}(\lambda)$ 是不可忽略的, 则 $\text{Adv}_{\mathcal{B}}^{\text{DDH}}(\lambda)$ 也是不可忽略的, 这与 DDH 问题的困难性假设矛盾。

5.3 区块链的不可篡改性

利用区块链的不可篡改性, 保证我们的方案可以满足第 3.4 节中提出的以下相关公平性:

如果用户不诚实, 则意味着区块链将拒绝生成交易 Ask。因此, 用户无法得到相关的数据文件密文 C 和对称密钥 K。在此过程中, 用户将受到处罚。

如果数据属主不诚实, 则意味着它嵌入在交易 Pay 中的文件密文 C 和对称密钥 K 是错误的, 区块链将拒绝生成交易 Pay 且数据属主将受到处罚。

如果服务器不诚实, 则意味着它嵌入在交易 G 中的值 DB(w) 是错误的, 要么它没有提供正确的结果。对于前者, 服务器无法获取任何明文信息;对于后者, 它不能得到服务费。

如果三方都诚实公平地执行协议, 用户可以从交易 P 中得到正确的搜索结果, 数据属主和服务器也可以得到相应的服务费用。

6 性能及效率分析

6.1 功能特性比较

本文与近几年的属性基加密文献[16-19]中的方案进行功能性对比, 其中访问控制策略主要包括访问树 (Access Tree) 和线性秘密共享方案 (Linear Secret Sharing Scheme, LSSS)两种。对比结果如表 1 所示, 表 1 表明, 本文方案在功能特性上有一定优势。

6.2 理论分析与比较

在表 2 中, T_{p} 表示配对运算的时间, T_{e} 表示指数运算的时间, T_{m} 表示乘法运算的时间, T_{h} 表示哈希运算的时间, T_{inv} 表示乘法逆元运算的时间, 且 $T_{\text{p}}>T_{\text{e}}>T_{\text{m}}>T_{\text{h}}>T_{\text{inv}}$。

6.2.1 计算量比较

在表 2 和表 3 中, 分别用$|S|$, $|X|$, $|N|$表示一个用户的属性集, 一个访问树的叶子节点集合和满足访问控制策略的最小属性集。其中$T_{\text{Ref[16]}}=(2|N|+3)T_{\text{p}}+|N|T_{\text{e}}+(|N|+2)T_{\text{m}}+2T_{\text{inv}}$, $T_{\text{Ours}}=(2|N|+3)T_{\text{p}}+|N|T_{\text{e}}+(|N|+3)T_{\text{m}}+|N|T_{\text{inv}}$。

6.2.2 通信量比较

在表 3 中, 分别用$|G_1|,|G_2|,|G|,|G_T|,|Z_q^*|$表示$G_1,G_2,G,G_T,Z_q^*$中元素的长度。

6.3 数值实验与比较

在 Linux 操作系统下利用双线性对包 PBC (pairing-based cryptography library)[21], 使用的椭圆曲线基域为 512bit, 双线性对包参数类型为 Type-A。基于 C 语言进行编程, 在 2.9 GHz CPU, 4 GB RAM PC 机上运行。实验结果如图 6 和图 7 所示:

由图 6(a)~(c)可得出结论, 本文方案在 KeyGen 阶段和 Encrypt 阶段的效率与文献[19]方案的效率相当。在 Search 阶段, 本文方案的效率高于文献[18]和文献[19]方案的效率。

图 7(a)~(b)分别表示, 本文方案在关键字个数和属性个数取不同的值时, Encrypt 和 Search 算法的运行时间。如 Search 阶段, 当关键字个数为 100, 属性个数为 20 时, 本文方案的运行时间为 0.5349s。

7 结论

本文提出了一种基于区块链的多关键字属性基可搜索加密方案。本文方案利用属性基加密技术使数据属主能为多个数据用户执行细粒度的搜索授权, 利用多关键字可搜索加密技术完成对加密数据的有效搜索, 利用区块链技术保证数据用户和云服务器之间的公平性, 若数据用户不诚实, 则不能从云服务器获得正确的结果, 若云服务器是恶意的, 则不

表 1 功能特性比较

Table 1 Function comparison

References	Access Control	Keyword Search	Verifiable	Multi-Keyword
Ref[16]	LSSS	√	×	×
Ref[17]	Access Tree	√	×	×
Ref[18]	LSSS	√	×	×
Ref[19]	Access Tree	√	√	×
Ours	Access Tree	√	√	√

表 2 计算量比较

Table 2 Computation comparison

	Ref[18]	Ref[19]	Ours
SetUp	$T_{\text{p}}+4T_{\text{e}}+T_{\text{m}}$	$3T_{\text{e}}$	$T_{\text{p}}+3T_{\text{e}}$
KeyGen	$(2+\|S\|)T_{\text{e}}+(2+\|S\|)T_{\text{m}}$	$(3\|S\|+1)T_{\text{e}}+(\|S\|+2)T_{\text{m}}+\|S\|T_{\text{h}}+T_{\text{inv}}$	$(2\|S\|+1)T_{\text{e}}+(\|S\|+1)T_{\text{m}}+\|S\|T_{\text{h}}+T_{\text{inv}}$
Ecrypt	$T_{\text{P}}+(4\|X\|+6)T_{\text{e}}+(4\|X\|+8)T_{\text{m}}+T_{\text{h}}$	$(2\|X\|+4)T_{\text{e}}+2T_{\text{m}}+(\|X\|+1)T_{\text{h}}$	$T_{\text{p}}+(2\|X\|+3)T_{\text{e}}+3T_{\text{m}}+(\|X\|+1)T_{\text{h}}$
Trapdoor	$(\|S\|+5)T_{\text{e}}+3T_{\text{m}}+T_{\text{h}}$	$(2\|S\|+4)T_{\text{e}}+T_{\text{m}}+T_{\text{h}}$	$(\|S\|+1)T_{\text{e}}+(\|S\|+1)T_{\text{m}}+T_{\text{h}}$
Search	$(2\|N\|+3)T_{\text{p}}+\|N\|T_{\text{e}}+(2\|N\|+1)T_{\text{m}}$	$(2\|N\|+3)T_{\text{p}}+\|N\|T_{\text{e}}+(\|N\|+2)T_{\text{m}}+2T_{\text{inv}}$	$(2\|N\|+3)T_{\text{p}}+\|N\|T_{\text{e}}+(\|N\|+3)T_{\text{m}}+\|N\|T_{\text{inv}}$

表 3 存储量比较

Table 3 Storage comparison

Algorithms	Ref[18]	Ref[19]	Ours
SetUp	$4\|G\|+\|G_T\|+3\|Z_q^*\|$	$4\|G_1\|+3\|Z_q^*\|$	$3\|G_1\|+\|G_2\|+\|Z_q^*\|$
KeyGen	$(2+\|S\|)\|G\|+\|S\|\|Z_q^*\|$	$(2\|S\|+1)\|G_1\|$	$(2\|S\|+2)\|G_1\|$
Encrypt	$(2\|X\|+4)\|G\|+\|G_T\|$	$(2\|X\|+3)\|G_1\|$	$(2\|X\|+1)\|G_1\|+\|G_2\|$
Trapdoor	$(\|S\|+4)\|G\|+\|S\|\|Z_q^*\|$	$(2\|S\|+3)\|G_1\|$	$(2\|S\|+1)\|G_1\|$
Search	\	\	\

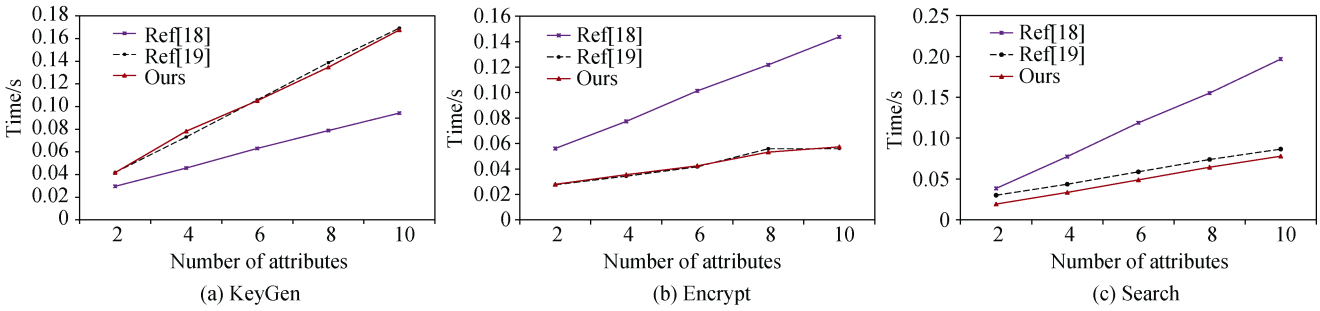

图 6　算法效率比较

Figure 6　Algorithm efficiency comparison

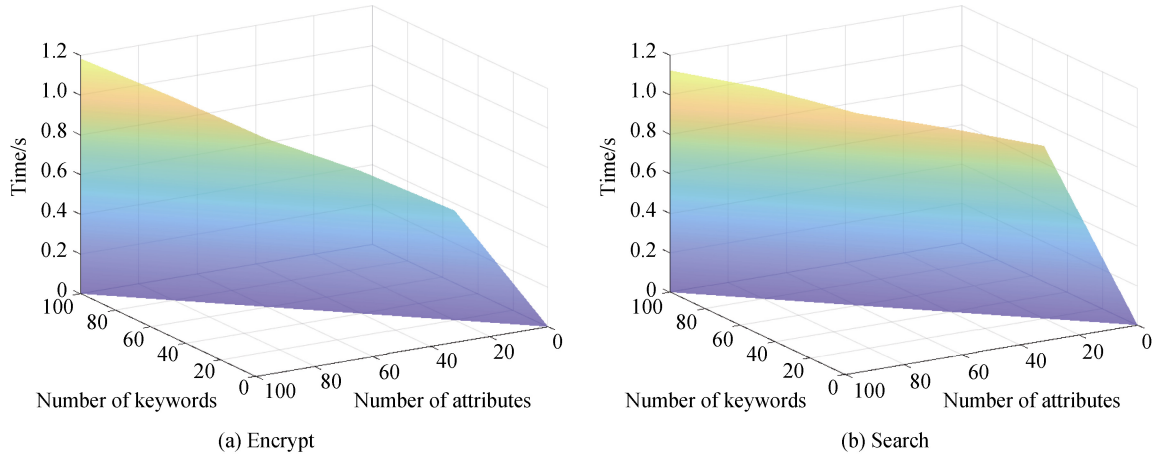

图 7　算法计算开销

Figure 7　Computation cost

能得到服务费并失去其保证金。在整个过程中, 不会向云服务器泄露任何关于关键字和数据明文的重要信息。最后给出了详细的正确性证明、性能分析和安全性证明。数值实验结果表明, 本文方案具有较高的效率。在未来的工作中考虑将其应用于电子病历数据共享等场景中, 以获得更实用的价值。

参考文献

[1] Song D X, Wagner D, Perrig A, et al. Practical techniques for searches on encrypted data[C]. *Proceeding 2000 IEEE Symposium on Security and Privacy*. *S&P*, 2002: 44-55.

[2] Boneh D, di Crescenzo G, Ostrovsky R, et al. Public Key Encryption with Keyword Search[M]. Advances in Cryptology - EUROCRYPT 2004. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004: 506-522.

[3] Fan Y Q, Liu Z H, Processing C A. Verifiable attribute-based multi-keyword search over encrypted cloud data in multi-owner setting[C]. *2017 IEEE Second International Conference on Data Science in Cyberspace*, 2017: 441-449.

[4] Sahai A,Waters B.Fuzzy identity-based encryption[C]. *International Conference on Theory & Applications of Cryptographic Techniques*, 2005:457-473.

[5] Bethencourt J, Sahai A, Waters B, et al. Ciphertext-policy attribute-based encryption[C]. *2007 IEEE Symposium on Security and Privacy*, 2007: 321-334.

[6] Goyal V, Pandey O, Sahai A, et al. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data[C]. *The 13th ACM conference on Computer and communications security*, 2006: 89-98.

[7] Li R H, Zheng D, Zhang Y H, et al. Attribute-based encryption with multi-keyword search[C]. *2017 IEEE Second International Conference on Data Science in Cyberspace*, 2017: 172-177.

[8] Sun W H, Yu S C, Lou W J, et al. Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud[C]. *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014: 226-234.

[9] Zhang Y H, Deng R H, Shu J G, et al. TKSE: Trustworthy Keyword Search over Encrypted Data with Two-Side Verifiability via Blockchain[J]. *IEEE Access*, 6: 31077-31087.

[10] Han X, Yuan Y, Wang F Y. Security Problems on Blockchain: The State of the Art and Future Trends[J]. *Acta Automatica Sinica*, 2019, 45(1): 206-225.
(韩璇, 袁勇, 王飞跃. 区块链安全问题: 研究现状与展望[J]. *自动化学报*, 2019, 45(1): 206-225.)

[11] Zeng S, Yuan Y, Ni X C, et al. Scaling Blockchain towards Bitcoin: Key Technologies, Constraints and Related Issues[J]. *Acta Automatica Sinica*, 2019, 45(6): 1015-1030.
(曾帅, 袁勇, 倪晓春, 等. 面向比特币的区块链扩容: 关键技

术，制约因素与衍生问题[J]. *自动化学报*，2019，45(6): 1015-1030.)

[12] Yuan Y, Ni X C, Zeng S, et al. Blockchain Consensus Algorithms: The State of the Art and Future Trends[J]. *Acta Automatica Sinica*, 2018, 44(11): 2011-2022.
(袁勇，倪晓春，曾帅，等. 区块链共识算法的发展现状与展望[J]. *自动化学报*, 2018, 44(11): 2011-2022.)

[13] Wu A X, Zhang Y H, Zheng X K, et al. Efficient and Privacy-Preserving Traceable Attribute-Based Encryption in Blockchain[J]. *Annals of Telecommunications*, 2019, 74(7): 401-411.

[14] Li H G, Tian H B, Zhang F G, et al. Blockchain-Based Searchable Symmetric Encryption Scheme[J]. *Computers & Electrical Engineering*, 2019, 73: 32-45.

[15] Ouyang L W, Wang S, Yuan Y, et al. Smart Contracts: Architecture and Research Progresses[J]. *Acta Automatica Sinica*, 2019, 45(3): 445-457.
(欧阳丽炜，王帅，袁勇，等. 智能合约：架构及进展[J]. *自动化学报*, 2019, 45(3): 445-457.)

[16] Wang S P, Zhang D, Zhang Y L, et al. Efficiently Revocable and Searchable Attribute-Based Encryption Scheme for Mobile Cloud Storage[J]. *IEEE Access*, 6: 30444-30457.

[17] Zhu H J, Wang L C, Ahmad H, et al. Key-Policy Attribute-Based Encryption with Equality Test in Cloud Computing[J]. *IEEE Access*, 5: 20428-20439.

[18] Hu Y Y, Chen Y L, Zhu M H. Privacy Protection Attribute-Based Ciphertext Search Scheme[J]. *Application Research of Computers*, 2019, 36(4): 1158-1164.
(胡媛媛，陈燕俐，朱敏惠. 可实现隐私保护的基于属性密文可搜索方案[J]. *计算机应用研究*, 2019, 36(4): 1158-1164.)

[19] Zheng Q J, Xu S H, Ateniese G, et al. VABKS: Verifiable attribute-based keyword search over outsourced encrypted data[C]. *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014: 522-530.

[20] Nie X Y, Bao Y Y, Sun J F, et al. A Multi-Authority Attribute-Based Signcryption Scheme[J]. *Journal of Cyber Security*, 2018, 3(5): 15-24.
(聂旭云，鲍阳阳，孙剑飞，等. 一个多授权中心的属性基签密方案[J]. *信息安全学报*, 2018, 3(5): 15-24.)

[21] The pairing-based cryptography library.http://crypto.stanford.edu/pbc/,2015.

牛淑芬 于 2013 年在西北师范大学基础数学专业获得博士学位。现任西北师范大学副教授。研究领域为大数据网络隐私保护。Email: sfniu76@nwnu.edu.cn

韩松 于 2019 年在郑州大学软件工程专业获得学士学位。现在西北师范大学计算机技术专业攻读硕士学位。研究领域为大数据网络隐私保护。Email：565904313@qq.com

谢亚亚 于 2018 年在天水师范学院计算机科学与技术专业获得学士学位。现在西北师范大学计算机技术专业攻读硕士学位。研究领域为大数据网络隐私保护。Email: 2418606113@qq.com

王彩芬 于 2003 年在西安电子科技大学密码学专业获得博士学位。现任深圳技术大学计算机系教授。研究领域为网络安全、密码协议。Email: wangcf@nwnu.edu.cn