高效的可验证无证书可搜索加密方案

崔新华[1,2,3]，田有亮[1,2,4]，张起嘉[1,2]

（1. 公共大数据国家重点实验室，贵州 贵阳 550025；2. 贵州大学计算机科学与技术学院，贵州 贵阳 550025；
3. 贵州师范大学经济与管理学院，贵州 贵阳 550025；4. 贵州大学密码学与数据安全研究所，贵州 贵阳 550025）

摘 要：在云计算环境中，可搜索加密方案是一种实现数据隐私保护和关键词搜索的有效方法。目前，现有方案不仅难以实现高效验证与动态更新，同时也存在证书管理和密钥分配问题。为了解决上述问题，近期有相关学者提出了一种基于改进 Merkle-Tree 认证方法的可验证多关键词搜索方案，然而经过安全性分析，该方案不能满足密文的不可区分性。通过改进，提出了一种新的高效的可验证无证书可搜索加密方案。分析表明，所提方案不仅能够满足无证书环境下的密文不可区分性与签名的不可伪造性，还实现了更高的计算效率与更小的通信开销，更能适用于资源有限的终端设备。

关键词：无证书加密；可搜索加密；可验证性；动态更新

中图分类号：TP309

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2023156

Efficient certificateless searchable encryption scheme with verifiability

CUI Xinhua[1,2,3], TIAN Youliang[1,2,4], ZHANG Qijia[1,2]

1. State Key Laboratory of Public Big Data, Guiyang 550025, China
2. College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
3. College of Economics and Management, Guizhou Normal University, Guiyang 550025, China
4. Institute of Cryptography & Data Security, Guizhou University, Guiyang 550025, China

Abstract: Searchable encryption offers an effective way to achieve data privacy protection and keyword search in cloud computing environments. Currently, the existing schemes not only lack dynamic update and efficient verification mechanism, but also suffer from the certificate management burden and key escrow issue. To address these issues, a verifiable multi-keyword searchable encryption scheme based on improved Merkle-Tree had been proposed recently. However, through cryptoanalysis, that scheme could not achieve the indistinguishability. With improvement, an efficient able certificateless searchable encryption scheme with verifiability was proposed. Rigorous analysis show that the proposed scheme not only supports the indistinguishability and the unforgeability, but also enjoys higher computing efficiency and lower communication cost, which is more suitable for terminal devices with limited resources.

Keywords: certificateless encryption, searchable encryption, verifiability, dynamic updating

---

收稿日期：2023−03−17；**修回日期：**2023−06−06

通信作者：田有亮，youliangtian@163.com

基金项目：国家重点研发计划基金资助项目（No.2021YFB3101100）；国家自然科学基金资助项目（No.U1836205, No.62272123）；贵州省高层次创新型人才基金资助项目（黔科合平台人才[2020]6008）；贵阳市科技计划基金资助项目（筑科合[2021]1-5, 筑科合[2022]2-4）；贵州省科技计划基金资助项目（黔科合平台人才[2020]5017, 黔科合支撑[2022]一般 065）

Foundation Items: The National Key Research and Development Program of China (No.2021YFB3101100), The National Natural Science Foundation of China (No.U1836205, No.62272123), Project of High-level Innovative Talents of Guizhou Province (No.[2020]6008), Science and Technology Program of Guiyang (No.[2021]1-5, No.[2022]2-4), Science and Technology Program of Guizhou Province (No.[2020]5017, No.[2022]065)

0 引言

随着云计算技术的快速发展，为了满足用户日益增长的需求，互联网设备产生的数据量飞速增长。越来越多的企业与个人选择将数据存放在云服务器中，以缓解本地存储带来的压力。然而，当数据被上传至云服务器后，数据拥有者便失去了对数据的控制。特别是当数据中包含敏感信息时，数据拥有者难以对敏感信息提供安全保护。为了避免云服务器侵犯敏感数据的隐私性，需要将文件加密后再上传。然而，加密会导致数据之间的关联性被破坏，给文件检索带来了困难。为了解决云服务器对密文实现高效检索的问题，可搜索加密[1]成为最受研究者关注的方法之一。

现有的可搜索加密方案中，公钥可搜索加密（PEKS）[2-3]相比于对称可搜索加密（SSE, symmetric searchable encryption）[4-5]更适用于云存储环境中的文件共享等场景，因此成为一个重要的研究热点。PEKS 引入了非对称的搜索陷门，以关键词密文与搜索陷门能否匹配作为搜索成功的判断依据。具体来说，数据拥有者首先需要依次对准备上传的文件提取关键词，然后利用文件接收者的公钥计算生成关键词密文。接下来，数据拥有者将数据文件加密，再将文件密文与关键词密文相连接，形成密文索引并一同上传至云服务器中进行存储。数据用户在对目标文件发起搜索前，首先需要确定目标文件的关键词，并使用私钥生成搜索陷门，然后作为搜索请求发送给云服务器。云服务器接收到搜索请求后，对密文索引进行匹配测试，最后将匹配到的密文作为搜索结果返回给数据用户。在此期间，不可信的云服务器虽然执行存储与匹配任务，却无法得知任何关于数据文件的信息。

然而，在现实环境中云服务器可能会因为经济利益或黑客攻击等因素，恶意地发动选择性转发攻击[6]，即只返回部分搜索结果给用户。而在目前大多数的 PEKS 方案中，用户缺乏搜索结果的验证机制，无法检测返回的搜索结果是否完整或是否已经被恶意篡改。为了抵抗恶意云服务的攻击，可验证性[7-9]为用户提供了搜索结果的审查机制，成为研究目标之一。Sun 等[10]提出了一种可验证的多关键词可搜索加密方案。该方案采用基于树的数据结构作为索引，以实现搜索结果的完整性验证。Wang 等[11]结合布隆过滤器（BF, Bloom filter）、Merkle 哈希树（MHT, Merkle hash tree）等数据结构，提出了一种外包数据库的审计协议。然而，该协议只能实现静态审计，无法适用于云计算中的动态更新场景。随后，大量支持动态更新的可验证的可搜索加密方案被提出[6,12-13]。然而，这些方案在更新过程中带来的计算开销与通信开销较大，有待进一步降低。因此，对搜索结果实现高效验证与密文的高效更新仍是亟待解决的难题。

大多数 PEKS 方案都是基于证书或基于身份的公钥密码，存在证书管理以及密钥托管问题。为了解决该问题，Peng 等[14]提出了第一个基于无证书的公钥可搜索加密（CLPEKS, certificateless public key encryption with keyword search）方案，该方案将无证书的公钥密码引入 PEKS 中，为传统的 PEKS 方案提升了抗恶意用户与半诚实的密钥生成中心的安全性。此后，大量的研究人员对基于无证书的 PEKS 方案进行了研究[15-18]。Zheng 等[19]提出了一个不需要用户执行对运算的无证书可搜索加密方案，该方案显著降低了用户的计算量，提升了方案的实用性。Miao 等[20]实现了一种基于无证书的可验证的多关键词可搜索加密方案，该方案不仅基于 Zheng 等[19]方案实现了无证书加密，还将云审计与可搜索加密相结合，采用可信的第三方审计服务器协助用户对搜索结果进行验证。田有亮等[21]在 Miao 等[20]方案的基础上提出了一种基于改进 Merkle 树认证方法的可验证多关键词可搜索加密方案，与之前的方案相比，该方案将验证过程中的计算开销由经典 MHT 的 $O(n)$降低到 $O(\log n)$，实现了高效验证与更新。然而，经过本文分析，该方案存在明显的安全性缺陷，无法达到文中声称的安全要求。张键红等[6]提出了一种高效、可验证的多关键词搜索加密方案，该方案不仅能够支持多关键词搜索，还实现了搜索模式的隐私性和文件的前向安全。

为了实现安全高效可验证的密文检索，同时满足无证书公钥加密环境中的安全性，本文提出了一种高效的可验证无证书可搜索加密方案。本文的主要贡献如下。

1) 首先，对文献[21]方案进行了安全性分析，指出了其既不能满足密文的选择明文攻击的不可区分（IND-CPA）安全，也不能满足适用于可搜索加密的选择关键词攻击的不可区分（IND-CKA）安全。然后，给出了 2 种具体的攻击，即外部攻

击者发动的密文猜测攻击，以及恶意用户发动的在线关键词猜测攻击。

2) 提出了一种高效的基于无证书的可搜索加密方案。在随机预言机模型下，基于判定性线性（DLIN, decisional linear）假设与计算性 DH（co-CDH, co-computational Diffie-Hellman）假设，证明了所提方案满足选择关键词攻击下的不可区分性与签名的不可伪造性。

3) 通过结合改进的 MHT 与无证书签名，实现了高效可验证性与高效动态更新，通过实验与其他现有主流方案进行对比，分析表明了所提方案在计算效率方面与通信效率方面均优于其他相关方案，具有更高的实用价值。

1　基础知识

本节首先介绍了双线性对的概念和本文用到困难假设，然后给出了无证书可搜索加密方案的形式化定义和相关的安全模型。

1.1　双线性对

设 λ 为安全参数， p 为与 λ 相关的大素数，G_1、G_2 和 G_T 为 3 个阶为 p 的乘法循环群。

定义 1　双线性映射。若映射 $e: G_1 \times G_2 \to G_T$ 满足以下 3 个条件，则该映射为一个双线性映射。

1) 双线性：对于任意元素 $g_1 \in G_1, g_2 \in G_2$ 和 $a,b \in Z_p$，均有 $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$。

2) 非退化性：至少存在元素 $g_1 \in G_1$ 和 $g_2 \in G_2$，满足 $e(g_1, g_2) \neq 1$。

3) 可计算性：对于任意元素 $g_1 \in G_1$ 和 $g_2 \in G_2$，均有多项式时间算法计算 $e(g_1, g_2)$。

根据群 G_1 与 G_2 的不同关系，双线性映射可分为 2 类。若 $G_1 = G_2$，则该双线性映射为 Type-1 类型；若 $G_1 \neq G_2$，且 G_1 与 G_2 之间存在一个同构映射 ϕ，使 $\phi(G_2) \to G_1$，则该双线性映射为 Type-2 类型。

1.2　困难假设

定义 2　co-CDH 假设。给定四元组 $(g_1, g_2, g_1^a, g_2^b) \in G_1^2 \times G_2^2$，其中 $a,b \in Z_p$ 未知，对于任意的多项式时间算法 A，计算得出 g_1^{ab} 的优势是可忽略的，数学形式表示为

$$\mathrm{Adv}_A^{\text{co-CDH}}(\lambda) = \Pr[A(g_1, g_2, g_1^a, g_2^b) \to g_1^{ab}]$$

定义 3　DLIN 假设。给定六元组 (u, v, w, v^c, w^d, Z)，其中 $c,d \in Z_p$ 未知，且 $(u, v, v^c, Z) \in G_1$，$(w, w^d) \in G_2$。对于任意的多项式时间算法 A，判定 $Z = u^{c+d}$ 是否成立的优势是可忽略的，数学形式表示为

$$\mathrm{Adv}_A^{\text{DLIN}}(\lambda) = \Pr[A(u, v, w, v^c, w^d, u^{c+d}) = 1] - \Pr[A(u, v, w, v^c, w^d, Z) = 1]$$

1.3　Merkle 哈希树

经典 Merkle 哈希树是一种二叉树数据结构，将数据分成若干个块，然后对每个块计算哈希值，作为叶子节点。输入相邻的 2 个叶子节点的哈希值，再计算生成一个新的哈希值，作为这 2 个叶子节点的父节点。递归进行该操作，直到得到一个根节点，它可以确保所有节点的完整性，因此可以作为整个数据的哈希值。Garg 等[22]提出了一种基于相对索引和时间戳的 Merkle 哈希树，通过修改搜索算法实现了高效动态更新。

改进 Merkle 树的每个节点包括 2 个信息：一个是数据块的哈希值，另一个是相对索引。与节点 T 关联的相对索引是指属于 T 的子树的叶子节点的数量，叶子节点的索引值为 1。例如，如果 T 的左右节点分别为 h_{Left} 和 h_{Right} 且相对索引为 i_{Left} 和 i_{Right}，则 T 的哈希值为 $H(h_{\text{Left}}, h_{\text{Right}})$，索引值为左右节点的相对索引之和 $(i_{\text{Left}} + i_{\text{Right}})$。根节点 h_{Root} 的生成方式为 $h_{\text{Root}} = H(h_{\text{Left}}, h_{\text{Right}}, d_t)$，其中，时间戳 d_t 表示 Merkle 树创建的时间，根节点只对树中任意数据块做出修改而更新，且只有在有效时间内才能通过验证，因此能够保证数据的有效性。

1.4　形式化定义

如图 1 所示，无证书可搜索加密系统模型中包含 4 个不同的实体：密钥生成中心（KGC, key generation center）、数据拥有者（DO, data owner）、数据用户（DU, data user）以及云服务器（CSP, cloud service provider）。

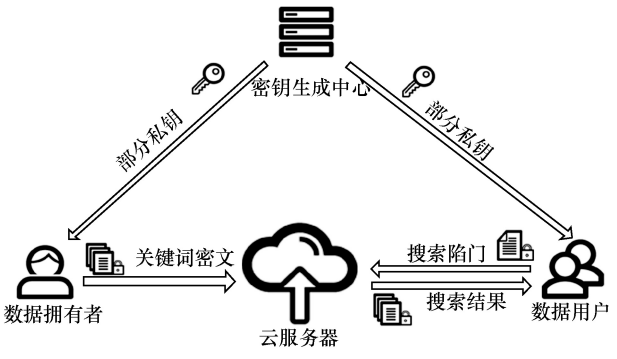


图 1　系统模型

1) 密钥生成中心。KGC 为一个半诚实的实体，负责根据安全参数生成系统主密钥与系统公开参数，以及为每个数据拥有者与用户生成部分私钥。KGC 会对密文中包含的信息产生好奇，能够利用已知的系统主密钥等关键信息发动猜测攻击，以及勾结云服务器伪造验证信息。

2) 数据拥有者。数据拥有者为一个诚实的实体，负责生成自己的公私钥，并使用私钥生成验证信息；还负责使用用户的公钥生成关键词密文，创建密文索引，最后将密文文件、密文索引与验证信息一同上传至云服务器中。除此之外，数据拥有者还能够对密文进行修改、添加、删除等更新操作。

3) 数据用户。数据用户为一个可能存在恶意的实体，但不与其他实体相互勾结。数据用户负责向云服务器提出搜索请求，以及收到结果后对结果进行验证。恶意用户可能伪装成合法用户伪造搜索陷门，也可能对验证信息发动伪造攻击。

4) 云服务器。云服务器为一个半诚实的实体，负责存储数据拥有者上传的信息、帮助数据用户执行搜索、协助数据拥有者完成密文更新操作。在操作过程中，云服务器可能会对密文中包含的信息产生好奇。

定义 4 可验证的无证书可搜索加密方案。可验证的无证书可搜索加密方案通常包括以下 7 个多项式时间算法。

1) Setup(λ)。该算法由 KGC 执行。输入安全参数 λ，生成系统主密钥 msk，发布系统公开参数 params。

2) ParKeyGen(params, msk, ID)。该算法由 KGC 执行。输入系统主密钥 msk、系统公开参数 params 以及用户 ID，生成用户的部分私钥 psk。

3) KeyGen(params, psk, ID)。该算法由用户执行。输入部分私钥 psk、系统公开参数 params 以及用户 ID，生成用户的私钥 $\mathrm{sk_U}$ 与公钥 $\mathrm{pk_U}$。

4) Encrypt(params, w, Files, $\mathrm{pk_U}$, $\mathrm{sk_O}$, ID)。该算法由数据拥有者执行。输入系统公开参数 params、数据拥有者私钥 $\mathrm{sk_O}$ 以及数据文件 Files，生成验证信息 v。然后输入用户公钥 $\mathrm{pk_U}$ 与关键词 w，生成关键词密文 CT。

5) Trapdoor(params, w, $\mathrm{sk_U}$)。该算法由数据用户执行。输入系统公开参数 params、私钥 $\mathrm{sk_U}$、关键词 w，生成搜索陷门 T。

6) Search(T, CT)。该算法由 CSP 执行。输入搜索陷门 T 和关键词密文 CT，由 CSP 计算是否匹配，若匹配成功，则将文件密文以及相应的验证信息返回给用户。

7) Verify(params, v, $\mathrm{pk_O}$, Files)。该算法由数据用户执行。输入系统公开参数 params、数据拥有者公钥 $\mathrm{pk_O}$、验证信息 v 以及解密文件 Files，输出验证结果 1（成功）或 0（失败）。

1.5 安全模型

无证书可搜索加密考虑 2 种类型的敌手：Type-1 类型的敌手 A_{I} 代表恶意用户，无法得知系统主密钥 msk，但可以发动公钥替换攻击，伪装成一个合法用户伪造搜索陷门或签名；Type-2 类型的敌手 A_{II} 代表诚实且好奇的 KGC，能够掌握系统主密钥 msk，以及掌握每个用户的部分私钥 psk，但无法得知用户私钥，也无法伪装成其他用户。

本文通过以下 4 个挑战者 C 与敌手 A_{I}、A_{II} 之间的安全游戏来定义方案的安全模型，其中，Game1 与 Game2 为针对密文的 IND-CKA 游戏，Game3 与 Game4 为针对签名的 EUF-CMA 游戏。

Game1 此处的敌手为 A_{I}。

1) 初始化。挑战者 C 运行 Setup 算法，设置公共参数。然后设置系统主密钥，计算系统公钥。最后将生成的系统公共参数与公钥发送给敌手 A_I。

2) 哈希询问。敌手 A_{I} 在该阶段对 2 个随机预言机分别进行多项式有限次的询问，挑战者 C 维护 2 个初始为空的列表，用来记录每次询问。

3) 阶段 1。在该阶段，敌手 A_{I} 向挑战者 C 进行多项式有界次适应性询问，内容包含以下 4 项。

① ParKeyGen 询问。敌手 A_{I} 向挑战者 C 发送一个身份 ID。当 ID 与挑战者身份相同时，中止游戏；否则，挑战者 C 在收到 ID 后，查询哈希列表并计算部分私钥 psk，返回给敌手。

② KeyGen 询问。敌手 A_{I} 向挑战者 C 发送一个身份 ID。当 ID 与挑战者身份相同时，中止游戏；否则，挑战者 C 执行 ParKeyGen 算法，然后计算公私钥对(sk, pk)，并返回给敌手。

③ Replace-KeyGen 询问。挑战者 C 首先建立一个初始为空的列表。敌手 A_{I} 随机生成一个公钥 pk'。随后向挑战者 C 发送想要替换的身份 ID 与替换公钥 pk'，挑战者 C 将身份 ID 对应的公钥 pk 使用 pk' 替换，并将其存储在列表中。

④ Trapdoor 询问。敌手 A_{I} 向挑战者 C 发送一个

身份 ID 与一个关键词 w。挑战者 C 首先对于 ID 执行 ParKeyGen 询问与 KeyGen 询问，然后计算 Trapdoor 并返回给敌手 A_{I}。

4) 挑战。敌手 A_{I} 向挑战者 C 发送一个身份 ID 与 2 个长度相同的关键词 w_0, w_1。ID 与 w_0, w_1 曾经被询问过，则中止游戏；否则，挑战者 C 随机抛出一枚硬币 $b=\{0,1\}$，计算关于 ID 与 w_b 的密文，返回给敌手。

5) 阶段 2。与阶段 1 相同，唯一的限制是敌手 A_{I} 不能询问关于挑战密文的关键词，也不能询问挑战者的私钥与部分私钥。

6) 猜测。敌手 A_{I} 输出一个比特 $b'=\{0,1\}$，若 $b'=b$，则敌手获胜。

定义 5　对于多项式时间内的敌手 A_{I}，赢得 IND-CKA 游戏的优势为

$$\mathrm{Adv}_{A_{\mathrm{I}}}^{\text{IND-CKA}}(\lambda)=\left|\Pr[b'=b]-\frac{1}{2}\right|$$

Game2　此处的敌手为 A_{II}。Game2 与 Game1 相似，区别是不需要执行 ParKeyGen 询问与 Replace-KeyGen 询问，且 Setup 阶段需要发送系统密钥给敌手 A_{II}。

Game3　此处的敌手为 A_{I}。

1) 初始化。挑战者 C 运行 Setup 算法，设置公共参数。然后设置系统主密钥，计算系统公钥。最后将生成的系统公共参数与公钥发送给敌手 A_{I}。

2) 哈希询问。敌手 A_{I} 在该阶段对 2 个随机预言机分别进行多项式有限次的询问，挑战者 C 维护 2 个初始为空的列表，用来记录的每次询问。

3) 询问阶段。在该阶段，敌手 A_{I} 向挑战者 C 进行多项式有界次适应性询问，内容包含以下 4 项。

① ParKeyGen 询问。敌手 A_{I} 向挑战者 C 发送一个身份 ID。当 ID 与挑战者身份相同时，中止游戏；否则，挑战者 C 在收到 ID 后，查询哈希列表并计算部分私钥 psk，返回给敌手。

② KeyGen 询问。敌手 A_{I} 向挑战者 C 发送一个身份 ID。当 ID 与挑战者身份相同时，中止游戏；否则，挑战者 C 执行 ParKeyGen 算法，然后计算公私钥对 $(\mathrm{sk},\mathrm{pk})$，并返回给敌手。

③ Replace-KeyGen 询问。挑战者 C 首先建立一个初始为空的列表。敌手 A_{I} 随机生成一个公钥 pk'。随后向挑战者 C 发送想要替换的身份 ID 与替换公钥 pk'，挑战者 C 将身份 ID 对应的公钥 pk 使用 pk' 替换，并将其存储在列表中。

④ Signature 询问。敌手 A_{I} 向挑战者 C 发送一个身份 ID 与一个消息 m。挑战者 C 首先对 ID 执行 ParKeyGen 询问与 KeyGen 询问，然后计算签名并返回给敌手 A_{I}。

4) 伪造。敌手 A_{I} 向挑战者 C 发送一个身份 ID、一个关于 ID 的公钥、一个消息 m*以及对应的签名 σ*。若 ID 与消息 m*的签名曾经被询问过，则中止游戏；否则，敌手 A_{I} 赢得该游戏。

定义 6　对于多项式时间内的敌手 A_{I}，赢得 EUF-CMA 游戏的优势为

$$\mathrm{Adv}_{A_{\mathrm{I}}}^{\text{EUF-CMA}}(\lambda)=\Pr[\mathrm{Ver}(m^*,\sigma^*,\mathrm{ID},\mathrm{pk}')=1]$$

Game4　此处的敌手为 A_{II}。Game4 与 Game3 相似，区别是不需要执行 ParKeyGen 询问与 Replace-KeyGen 询问，且 Setup 阶段需要发送系统密钥给敌手 A_{II}。

2　对文献[21]方案的安全性分析

本节首先简要回顾文献[21]中的方案，然后指出其不能满足密文不可区分性，并给出 2 种具体的攻击过程。

2.1　对文献[21]中方案的描述

由于篇幅限制，本节只简要描述其 Setup、ParKeyGen、KeyGen、Encryption、Trapdoor Generation 以及 Search 算法。方案的详细设计请参考文献[21]。

1) Setup。输入安全参数 λ，选定一个双线性映射 $e: G\times G\to G_T$，其中，G 是阶为素数 p 的一个乘法循环群，g 是群 G 的生成元。KGC 首先选择 2 个随机元素 $x,y\in Z_p^*$ 作为系统主密钥，计算 g^x, g^y 并作为系统公钥。选择 n+1 个随机元素 $u,u_1,\cdots,u_n\in G$，然后定义一个哈希函数 $H_1(\mathrm{ID})=u\prod_{i=1}^{n}u_i^{\mathrm{ID}_i}$，其中 $\mathrm{ID}=\{\mathrm{ID}_1,\cdots,\mathrm{ID}_n\}$。选择一个哈希函数 H_0:$\{0,1\}^*\to Z_p^*$。最后，保留系统主密钥 $\mathrm{msk}=\{x,y\}$，公开系统公共参数

$$\mathrm{param}=\{g,H_0,H_1,u,u_1,\cdots,u_n,g^x,g^y\}$$

2) ParKeyGen。给定用户身份 ID，KGC 随机选择 $t\in Z_p^*$，为用户生成部分私钥

$$\mathrm{psk}=\{g^t, g^xH_1(\mathrm{ID})^t\}$$

3) KeyGen。用户随机选择 $x',y'\in Z_p^*$，计算用户公钥 $\mathrm{pk}=\{g^{x'},g^{y'}\}$，并保留用户私钥

$\mathrm{sk}=\{\mathrm{psk},x',y'\}$。

4) Encryption。给定文件对应的关键词集$W=\{w_1,\cdots,w_n\}$，以及搜索用户的身份 ID，数据拥有者随机选择$r_1,r_2\in Z_p^*$，并计算关键词密文 CT

$$C_1=g^{r_1}$$
$$C_2=g^{(x+x')(r_1+r_2)}g^{yH_0(W)r_1}$$
$$C_3=g^{r_2}$$
$$C_4=H_1(\mathrm{ID})^{r_2}$$

5) Trapdoor Generation。给定搜索关键词集$W'=\{w_1',\cdots,w_n'\}$，搜索用户随机选择$s\in Z_p^*$，并计算搜索陷门T_w

$$T_1=g^{ts}$$
$$T_2=g^{(x+x')s}H_1(\mathrm{ID})^{ts}$$
$$T_3=g^{s}$$
$$T_4=\left(g^{(x+x')}g^{yH_0(W')}\right)^{s}$$

6) Search。服务器收到搜索用户发来的T_w后，计算并验证式(1)是否成立。

$$e(C_2,T_3)=e(C_1,T_4)\frac{e(C_3,T_2)}{e(C_4,T_1)} \tag{1}$$

若式(1)成立，则返回相应的密文给搜索用户；否则，匹配失败，终止搜索。

2.2 针对 IND-CKA 的攻击

本节通过分析证明文献[21]方案无法达到其声称的 IND-CKA 安全。

在 IND-CKA 安全游戏中，敌手在挑战阶段向挑战者发送 2 个长度相等的关键词集W_0^*和W_1^*，挑战者随机选择$b\in\{0,1\}$，生成相应的关键词密文CT_b^*，并发送给敌手。而敌手在收到关键词密文后，不需要知道用户私钥便可以区分出其中的关键词信息，进而得出正确的b。敌手通过伪造生成合法的搜索陷门，与挑战密文进行匹配，实现对挑战关键词的区分。具体操作如下。

1) 随机选择$s'\in Z_p^*$，利用公开的系统参数与用户公钥，生成W_1^*相关的伪造陷门$T'_{w_1^*}$

$$T_1'=g^{s'}$$
$$T_2'=g^{(x+x')s'}H_1(\mathrm{ID})^{s'}$$
$$T_3'=T_1'$$
$$T_4'=\left(g^{(x+x')}g^{yH_0(W_1^*)}\right)^{s'}$$

2) 根据挑战阶段中接收到的挑战密文，验证式(2)是否成立。

$$e(C_2^*,T_3')=e(C_1^*,T_4')\frac{e(C_3^*,T_2')}{e(C_4^*,T_1')} \tag{2}$$

显然，若式(2)成立，则密文中包含的关键词集为W_1^*；若式(2)不成立，则密文中包含的关键词集为W_0^*。

因此，敌手攻击成功，同时证明该方案不满足可搜索加密的 IND-CKA 安全。

2.3 针对密文机密性的关键词猜测攻击

本节通过分析证明文献[21]方案无法实现密文的机密性。具体来说，诚实且好奇的云服务器能够对存储的关键词密文发动关键词猜测攻击。详细过程如下。

对于一个包含关键词集$W=\{w_1,\cdots,w_n\}$的密文 CT，有

$$C_1=g^{r_1}$$
$$C_2=g^{(x+x')(r_1+r_2)}g^{yH_0(W)r_1}$$
$$C_3=g^{r_2}$$
$$C_4=H_1(\mathrm{ID})^{r_2}$$

诚实且好奇的云服务器首先计算

$$\sigma_1=e(C_2,g)=e(g^{(x+x')(r_1+r_2)}g^{yH_0(W)r_1},g)$$

然后，利用给定的系统公钥g^x,g^y以及用户公钥$g^{x'}$，计算

$$\sigma_2=e(g^xg^{x'},C_1C_3)$$
$$\sigma_3=e(g^y,C_1)$$

假设猜测的关键词集为W^*，则云服务器计算

$$\sigma_{W^*}=\sigma_2\sigma_3^{H_0(W^*)} \tag{3}$$

最后，验证式(4)是否成立。

$$\sigma_1=\sigma_{W^*} \tag{4}$$

当$W^*=W$时，则有

$$\begin{aligned}&\sigma_{W^*}=\sigma_2\sigma_3^{H_0(W)}=\\&e(g^xg^{x'},C_1C_3)e(g^y,C_1)^{H_0(W)}=\\&e(g^{(x+x')},g^{(r_1+r_2)})e(g^y,g^{r_1})^{H_0(W)}=\\&e(g^{(x+x')(r_1+r_2)},g)e(g^{yH_0(W)r_1},g)=\\&e(g^{(x+x')(r_1+r_2)}g^{yH_0(W)r_1},g)=\sigma_1\end{aligned} \tag{5}$$

由于现实世界中的关键词的明文空间有限，攻击者可以通过对所有可能的关键词集 W^* 进行有限次穷举，对式(3)进行大量计算，得到的 σ_{W^*} 分别使用式(4)进行对比，便可在多项式时间内成功猜测出密文 CT 中所包含的关键词明文信息。

因此，该方案不满足关键词密文的机密性。

3　所提方案

下面给出本文提出的高效的可验证无证书可搜索加密方案。

3.1　系统初始化

给定安全参数 λ，KGC 首先生成一个 Type-2 类型的双线性映射 $e: G_1 \times G_2 \to G_T$，其中 G_1, G_2 均是阶为素数 p 的乘法循环群，g_1 为 G_1 群的生成元，g_2 为 G_2 群的生成元。KGC 选择 2 个随机元素 $x, y \in Z_p^*$ 作为系统主密钥，并计算 g_1^x, g_1^y, g_2^y 作为系统公钥。然后，KGC 选择如下 5 个哈希函数。

$$H_1: \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^{l_{\text{date}}} \to \{0,1\}^n,$$
$$H_2: \{0,1\}^{l_{\text{ID}}} \to G_2, H_3: \{0,1\}^{l_{\text{ID}}} \to G_1,$$
$$H_4: \{0,1\}^* \to G_1, H_5: \{0,1\}^{l_{\text{kw}}} \to Z_p$$

最后，KGC 保留系统主密钥 $\text{msk} = \{x, y\}$，公开系统公共参数

$$\text{params} = \{g_1, g_2, H_1, H_2, H_3, H_4, H_5, g_1^x, g_1^y, g_2^y\}$$

3.2　部分私钥生成

给定身份 ID，通过执行该算法，KGC 分别生成数据用户与数据拥有者的部分私钥，具体步骤如下。

1) 对于数据用户，输入 $(\text{msk}, \text{ID}_\text{U}, \text{params})$，其中，$\text{ID}_\text{U}$ 为长度为 l_{ID} 的比特串，表示用户的身份。KGC 随机选择 $t \in Z_p$，并计算

$$\text{psk}_{\text{U},1} = g_1^t, \text{psk}_{\text{U},2} = g_2^x H_2(\text{ID}_\text{U})^t$$

2) 对于数据拥有者，输入 $(\text{msk}, \text{ID}_\text{O}, \text{params})$，其中，$\text{ID}_\text{O}$ 为数据拥有者的身份。KGC 随机选择 $t' \in Z_p$，并计算

$$\text{psk}_{\text{O},1} = g_2^{t'}, \text{psk}_{\text{O},2} = g_1^x H_3(\text{ID}_\text{O})^{t'}$$

然后将 2 个部分私钥 $\text{psk}_\text{U} = (\text{psk}_{\text{U},1}, \text{psk}_{\text{U},2})$ 与 $\text{psk}_\text{O} = (\text{psk}_{\text{O},1}, \text{psk}_{\text{O},2})$ 通过安全信道分别发送给数据用户与数据拥有者。

3.3　公私钥生成

在收到部分私钥后，数据用户随机选择一个 $x' \in Z_p^*$，然后计算 $g_1^{x'}$。则数据用户的公私钥为 $\text{pk}_\text{U} = g_1^{x'}$，$\text{sk}_\text{U} = \{\text{psk}_\text{U}, x'\}$。

同时，数据拥有者随机选择一个 $y' \in Z_p^*$，计算生成数据拥有者的公私钥 $\text{pk}_\text{O} = g_2^{y'}$，$\text{sk}_\text{O} = \{\text{psk}_\text{O}, y'\}$。

3.4　密文生成

首先，数据拥有者对数据文件进行提取关键词处理。对于包含相同关键词集 $W = \{w_1, \cdots, w_n\}$ 的文件 f_i，数据拥有者使用对称加密算法（如 SM4）对其进行加密处理，生成数据文件密文 $c = \{c_1, \cdots, c_q\}$。

其次，计算每个密文 c_i 的哈希值 h_i，再将每个文件的哈希值作为叶子节点，每个叶子节点作为哈希函数 H_3 的输入，迭代生成 MHT。其中，根节点 $h_{\text{Root}} = H_1(h_{\text{Left}} \| h_{\text{Right}} \| d_t)$ 为一个 n 位长的序列。由左右叶子节点与系统时间戳 d_t 共同计算生成。系统时间戳 d_t 表示一个有效时间段（如一天、一周或一个月），若时间段有效，则系统生成的时间戳相同。数据拥有者输入自己的身份 ID_O、私钥 sk_O、公钥 pk_O，生成根节点的签名

$$\sigma_1 = \text{psk}_{\text{O},1}$$
$$\sigma_2 = \text{psk}_{\text{O},2} H_4(\text{ID}_\text{O}, g_2^{y'}, h_{\text{Root}}, d_t)^{y'}$$

并将 $v = (h_{\text{Root}}, \sigma_1, \sigma_2)$ 作为搜索结果的验证信息。

然后，给定系统公钥 g_1^x, g_1^y 以及数据用户的 ID_U 与公钥 $g_1^{x'}$，数据拥有者随机选择 $r \in Z_p^*$，对关键词 $w_i \in W$ 计算生成关键词密文 C_w

$$C_1 = g_1^r$$
$$C_2 = (g_1^{(x+x'y')} g_1^{yH_5(w)})^r$$
$$C_3 = H_2(\text{ID}_\text{U})^r$$

最后，数据拥有者将 $\{C_W, v, c\}$ 一同上传至云服务器中进行存储，其中，$C_W = (C_1, C_2, C_3)$。

3.5　陷门生成

数据用户想要在云服务器中搜索感兴趣的文件，首先需要输入特定的关键词 $w' \in W$，然后随机选择一个元素 $s \in Z_p^*$，计算生成搜索陷门 $T_{w'}$

$$T_1 = g_1^{ts}$$
$$T_2 = (g_2^{(x+x'y')} H_2(\text{ID}_\text{U})^t g_2^{yH_5(w')})^s$$
$$T_3 = g_2^s$$

最后，将搜索陷门 $T_{w'}$ 提交给云服务器。

3.6 密文搜索

云服务器收到用户发送的搜索陷门 $T_{w'}$ 后，将其与存储的关键词密文 C_w 进行匹配，计算并判断式(6)是否成立。

$$e(T_1,C_3)e(C_2,T_3)=e(C_1,T_2) \tag{6}$$

若式(6)成立，则说明关键词密文与陷门匹配一致，将数据文件密文 $c=\{c_1,\cdots,c_q\}$ 以及相应的验证信息 $v=(h_{\text{Root}},\sigma_1,\sigma_2)$ 返回给用户。

3.7 结果验证

数据用户收到返回的搜索结果后，出于对云服务器的不信任，需要对搜索结果进行验证

首先，将数据文件密文 $c=\{c_1,\cdots,c_q\}$ 作为输入，依次计算哈希值 $\{h'_1,\cdots,h'_q\}$ 。

然后，将得到的哈希值 $\{h'_1,\cdots,h'_q\}$ 作为叶子节点重建 MHT，同时输入系统生成的时间戳 d_t '，得到根节点 $h'_{\text{Root}}=H_1(h'_{\text{Left}},h'_{\text{Right}},d'_t)$ 。

最后，输入 h'_{Root} 、数据拥有者的公钥与系统公钥，验证签名

$$e(\sigma_2,g_2)\overset{?}{=}e(g_1^x,g_2)e(H_3(\text{ID}_\text{O}),\sigma_1)e(\theta,\text{pk}_\text{O}) \tag{7}$$

其中， $\theta=H_4(\text{ID}_\text{O},\text{pk}_\text{O},h'_{\text{Root}},d'_t)$ 。若式(7)成立，则通过验证，同时证明搜索结果满足正确性与完备性。

3.8 动态更新

当数据拥有者希望对存储在云服务器上的文件进行修改或添加时，首先向服务器认证自己的身份。当身份验证通过后，通过以下方式对文件进行更新。

1) 数据修改。数据拥有者向云服务器发送修改令牌 $\pi=(X,i,c'_i,d'_t)$ ，其中， X 表示修改操作， i 表示待修改的文件所处的位置信息， c'_i 表示更新后的文件， d'_t 表示系统时间戳。云服务器收到修改令牌后，执行以下操作，原始密文 MHT 如图 2 所示。首先，使用 c'_i 替换原来的文件。其次，计算 c'_i 的哈希值，并根据文件的位置信息对原有的 MHT 中对应的路径上的节点进行更新。最后，输入更新后的 MHT 以及 d'_t ，计算新的根节点 $h'_{\text{Root}}=H_1(h'_{\text{Left}}\,\|\,h'_{\text{Right}}\,\|\,d'_t)$ ，并将 h'_{Root} 与 MHT 发送给数据拥有者。数据拥有者根据收到的 MHT 对云服务器进行挑战，挑战内容包括验证文件的哈希值，以及文件是否被正确更新。通过挑战后，数据拥有者计算 h'_{Root} 的签名，生成新的验证信息 $v'=(h'_{\text{Root}},\text{Sig}')$ ，并发送给云服务器。云服务器使用收到的 $v'=(h'_{\text{Root}},\text{Sig}')$ 替换原有的验证信息。

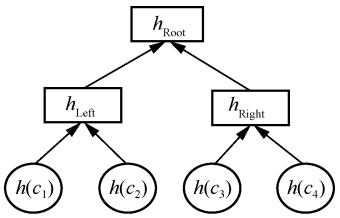


图 2 原始密文 MHT

2) 数据添加。数据拥有者向云服务器发送文件令牌 $\pi=(I,i,c'_i,d'_t)$ ，其中， I 表示插入操作， i 表示想要将文件插入的位置信息， c'_i 表示想要插入的文件， d'_t 表示系统时间戳。云服务器收到插入令牌后，执行以下操作（如图 3 所示）。首先，在 MHT 中找到位置 i 的对应的叶子节点，复制该节点的哈希值。将该节点作为父节点，使用该哈希值生成右叶子节点。然后，计算待插入文件 c'_i 的哈希值，并作为原位置 i 节点的左叶子节点。接下来，根据新生成的两个叶子节点计算生成哈希值，并替换原有的父节点，依次类推，计算并替换该路径上除根节点外的全部节点的值，得到更新后的 MHT。最后，输入 d'_t 与得到的 MHT，计算 $h'_{\text{Root}}=H_1(h'_{\text{Left}}\,\|\,h'_{\text{Right}}\,\|\,d'_t)$ ，并将 h'_{Root} 与 MHT 发送给数据拥有者。数据拥有者根据收到的 MHT 对云服务器进行挑战。通过挑战后，数据拥有者计算生成新的验证信息 $v'=(h'_{\text{Root}},\text{Sig}')$ ，并发送给云服务器。云服务器使用收到的 $v'=(h'_{\text{Root}},\text{Sig}')$ 替换原有的验证信息，然后将文件 c'_i 插入指定位置。

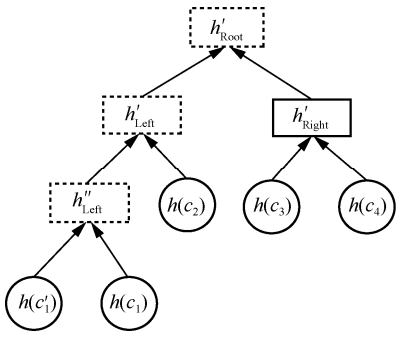


图 3 数据添加过程

3) 数据删除。数据拥有者向云服务器发送文件令牌 $\pi=(D,i,d'_t)$ ，其中， D 表示插入操作， i 表示待删除的文件的位置信息， d'_t 表示系统时间戳。云服务器收到删除指令后，执行以下操作（如图 4 所示）。首先，删除位置 i 对应的文件。其次，在 MHT 中找到位置 i 对应叶子节点，以及相同父节点下的另一个叶子节点 i' 。使用节点 i' 的值替换父节点的

值，并将叶子节点 i 删除。计算并替换该路径上除根节点外的全部节点的值，得到更新后的 MHT。最后，输入 d_t' 与得到的 MHT，计算新的根节点 h_{Root}'，并将 h_{Root}' 与 MHT 发送给数据拥有者。数据拥有者根据收到的 MHT 对云服务器进行挑战。通过挑战后，数据拥有者计算生成新的验证信息 $v'=(h_{\text{Root}}',\text{Sig}')$，并发送给云服务器。云服务器使用收到的 $v'=(h_{\text{Root}}',\text{Sig}')$ 替换原有的验证信息。

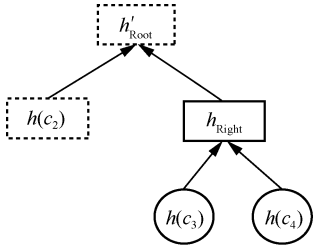


图 4　数据删除过程

4　方案分析

本节针对所提方案的正确性与安全性给出了具体的证明。

4.1　正确性分析

在搜索阶段，为了保证方案中的云服务器能够严格地对关键词密文与搜索陷门进行匹配，对于式(6)，有

$$\begin{aligned}&\text{Left}=e(T_1,C_3)e(C_2,T_3)=\\&e(g_1^{ts},H_2(\text{ID})^r)e(g_1^{(x+x'y')r}g_1^{yH_5(w)r},g_2^s)=\\&e(g_1,H_2(\text{ID}))^{tsr}e(g_1,g_2)^{sr(x+x'y'+yH_5(w))}\end{aligned}$$

$$\begin{aligned}&\text{Right}=e(C_1,T_2)=\\&e(g_1^r,(g_2^{(x+x'y')}H_2(\text{ID})^t g_2^{yH_5(w')})^s)=\\&e(g_1,H_2(\text{ID}))^{tsr}e(g_1,g_2)^{sr(x+x'y'+yH_5(w'))}\end{aligned}$$

当 $T_{w'}$ 与 C_w 中包含的关键词相同时，即 $w'=w$，则有 $e(T_1,C_3)e(C_2,T_3)=e(C_1,T_2)$，因此式(6)成立。

在结果验证阶段，为了保证搜索结果的正确性与完备性，对于式(7)，有

$$\begin{aligned}&e(\sigma_2,g_2)=e(\text{psk}_{\text{O},2}\theta^{y'},g_2)=\\&e(g_1^x H_3(\text{ID}_\text{O})^{t'}\theta^{y'},g_2)=\\&e(g_1^x,g_2)e(H_3(\text{ID}_\text{O})^{t'},g_2)e(\theta^{y'},g_2)=\\&e(g_1^x,g_2)e(H_3(\text{ID}_\text{O}),\sigma_1)e(\theta,\text{pk}_\text{O})\end{aligned}$$

其中，$\theta=H_4(\text{ID}_\text{O},\text{pk}_\text{O},h_{\text{Root}}',d_t')$。当根节点与验证信息一致时，即 $h_{\text{Root}}'=h_{\text{Root}}$，则式(7)成立，验证通过。

方案的正确性证毕。

4.2　安全性分析

本节通过定理 1～定理 3，证明了所提方案在 CDH 假设与 DLIN 假设下能够抵抗无证书环境中 Type-1 与 Type-2 这 2 种类型的敌手的攻击，实现选择关键词攻击下的 IND-CKA 安全，并且满足签名的 EUF-CMA 安全。

定理 1　若 DLIN 问题是困难的，则所提方案在随机预言机模型下能够满足 IND-CKA 安全。

引理 1　在随机预言机模型下，若存在一个 Type-1 类型的敌手 A_I，能够以不可忽略的优势 ε 攻破所提方案的 IND-CKA 安全性，则能够构造一个算法 B，利用敌手 A 的能力以不可忽略的优势 $\text{Adv}_B^{\text{DLIN}}(\lambda)$ 求解 DLIN 问题。

$$\begin{aligned}&\text{Adv}_B^{\text{DLIN}}(\lambda)\geqslant\\&\left(1-\frac{1}{q_{H_2}}\right)^{q_{\text{psk}}}\left(1-\frac{1}{q_{H_2}}\right)^{q_{\text{sk}}}\frac{2}{q_{H_2}q_{H_5}}\left(\frac{1}{2}+\varepsilon\right)-\frac{1}{2}\end{aligned}$$

证明　给定 B 一个 DLIN 问题的实例 (u,v,w,v^c,w^d,Z)，其中 $c,d\in Z_p$ 未知，B 通过与敌手 A_I 进行下面的 IND-CKA 游戏，最后输出一个比特 $b^*=\{0,1\}$，$b^*=1$ 表示 $Z=u^{c+d}$，令 $b^*=0$ 表示 Z 为一个随机元素。

初始化。B 运行 Setup 算法，设置公共参数 $g_1=v,g_2=w$。再选择一个随机数 $R\in Z_p$，隐式地设置系统主密钥为 $x=cR,\ y=d$，令 $g_1^x=(v^c)^R=v^{cR},\ g_2^y=w^d$，并利用同态映射 $\phi:G_2\to G_1$ 计算 $g_1^y=\phi(g_2^y)$。最后，随机选取 $y'\in Z_p^*$ 并计算 $g_2^{y'}$ 作为数据拥有者的公钥，将与生成的系统公共参数一并发送给 A_I。

$$\begin{aligned}&\text{params}=\{G_1,G_2,G_T,g_1,g_2\}\\&\text{pk}=\{g_1^x,g_1^y,g_2^y,g_2^{y'}\}\end{aligned}$$

哈希询问。敌手 A_I 在该阶段对 H_5,H_2 这 2 个随机预言机分别进行至多 q_{H_5},q_{H_2} 次的询问，B 维护 2 个初始为空的列表 L_1,L_2，用来记录 A_I 的每次询问。

H_5 询问。令 W_i 为敌手向 H_1 预言机进行的第 i 次询问。B 首先检查 L_1 中是否包含 W_i 的记录，若存在，则按照 L_1 中对应的内容答复敌手；否则，B 抛掷一枚硬币 $\sigma_i=\{0,1\}$，$\sigma_i=0$ 则随机选择 $u_i\in Z_p$，设置 $H_5(W_i)=u_i$，$\sigma_i=1$ 则随机选择设置 $H_5(W_i)=q_i+R$。

然后将 $H_5(W_i)$ 作为答复返回给敌手，并将三元组 $(W_i, H_5(W_i), \sigma_i)$ 作为新元素添加到列表 L_1 中。令概率 $\Pr[\sigma_i = 1] = \frac{1}{q_{H_5}}$，则 $\Pr[\sigma_i = 0] = 1 - \frac{1}{q_{H_5}}$。

H_2 询问。B 首先随机指定一个 $j^* \in (0, q_{H_2})$。当敌手向 H_2 预言机询问 ID_j 时，若 L_2 中没有记录，则 B 随机选择 $v_j \in Z_p$。令 $H_2(\mathrm{ID}_j) = g_2^{v_j}$，并返回给敌手作为对敌手询问的应答。同时，将三元组 $(\mathrm{ID}_j, g_2^{v_j}, v_j)$ 添加至 L_2 中。若 L_2 中已有关于 ID_j 的记录，则挑战者按照 L_2 中的记录应答敌手的询问。

阶段 1。在该阶段，敌手 A_I 向 B 进行多项式有界次适应性询问，内容包含以下 4 项。

1) ParKeyGen 询问。敌手 A_I 向 B 发送一个身份 ID_j。当 j=j*时，中止游戏；否则，B 在收到 ID_j 后，查询 L_2 列表获取对应的 $H_2(\mathrm{ID}_j)$ 与 v_j，再随机选择 $t_j \in Z_p^*$，计算

$$\mathrm{psk}_j = ((g_1^x)^{-v_j^{-1}} g_1^{t_j}, g_2^{v_j t_j})$$

最后将 psk_j 返回给敌手。该 psk_j 于敌手视角中为合法部分私钥。由于一个合法 psk 需要满足以下关系

$$e(g_1, \mathrm{psk}_2) = e(g_1^x, g_2) e(\mathrm{psk}_1, H_2(\mathrm{ID}))$$

敌手在收到 psk_j 后，可以利用已知信息对 psk_j 进行如下验证

$$\begin{aligned} &e(g_1^x, g_2) e(\mathrm{psk}_{j_1}, H_2(\mathrm{ID}_j)) = \\ &e(g_1^x, g_2) e((g_1^x)^{-v_j^{-1}} g_1^{t_j}, g_2^{v_j}) = \\ &e(g_1^x, g_2) e(g_1, g_2)^{-x + v_j t_j} = \\ &e(g_1, g_2)^{v_j t_j} \end{aligned}$$

由于 $e(g_1, \mathrm{psk}_{j_2}) = e(g_1, g_2^{v_j t_j})$，因此验证通过，同时证明该 psk_j 是合法的，敌手无法通过该参数区分自身是否处于模拟游戏中。

2) KeyGen 询问。敌手 A_I 向 B 发送一个身份 ID_j，当 j=j*时，中止游戏；否则，B 执行 ParKeyGen 算法，然后随机选择 $x'_j \in Z_p$，计算公私钥对

$$\mathrm{pk}_j = g_1^{x'_j}, \mathrm{sk}_j = (\mathrm{psk}_j, x'_j)$$

并返回给敌手。

3) Replace-KeyGen 询问。B 首先建立一个初始为空的列表 L_3。敌手 A_I 随机选择 $\mathrm{sk}'_j \in Z_p$，计算 $\mathrm{pk}'_j = g_1^{\mathrm{sk}'_j}$。随后向 B 发送想要替换的身份 ID_j 与替换公钥 pk'_j，B 将身份 ID_j 对应的公钥 pk_j 使用 pk'_j 替换，并将 $(\mathrm{ID}_j, \mathrm{pk}'_j)$ 存储在 L_3 中。

4) Trapdoor 询问。敌手 A_I 向 B 发送一个身份 ID_j 与一个关键词 W_i。当 j=j*且 $\sigma_i = 1$时，游戏中止；否则，B 首先对于 ID_j 执行 ParKeyGen 询问与 KeyGen 询问。若 L_3 中包含关于 ID_j 的记录，则需要 A_I 发送私钥 sk'_j。B 收到 sk'_j 后随机选择 $s \in Z_p$，计算 Trapdoor

$$\begin{aligned} T_1 &= ((g_1^x)^{-v_j^{-1}} g_1^{t_j})^s \\ T_2 &= (g_2^{v_j t_j + \mathrm{sk}'_j y'} g_2^{y q_i})^s \\ T_3 &= g_2^s \end{aligned}$$

最后将 Trapdoor 发送给敌手。在敌手 A_I 视角中该陷门是合法的，因为 $T_1 = ((g_1^x)^{-v_j^{-1}} g_1^{t_j})^s = \mathrm{psk}_{j_1}^s$，$T_2 = (g_2^{v_j t_j + \mathrm{sk}'_j y'} g_2^{y q_i})^s = (\mathrm{psk}'_{j_2} g_2^{\mathrm{sk}'_j y'} g_2^{y H_5(W)})^s$。

挑战。敌手 A_I 向 B 发送一个身份 ID_j 与 2 个长度相同的关键词 W_0, W_1。其中，若 j≠j*，或 $\sigma_0 \cap \sigma_1 \neq 1$，则中止游戏。当 j=j*且 $\sigma_0 = \sigma_1 = 1$时，若敌手在 Trapdoor 询问中询问过关于 W_0 或 W_1 的 Trapdoor，则游戏中止；否则，B 随机抛出一枚硬币 $b = \{0,1\}$，从 L_1 中获取 $H_5(W_b) = R + q_b$，分为以下 2 种情况计算关于 W_b 的密文。

1) 当 L_3 中没有关于身份 ID_{j^*} 的记录时，敌手 A_I 没有针对身份 ID_{j^*} 发动过公钥替换询问。则令 ID_{j^*} 的公钥为 $\mathrm{pk}_{j^*} = (g_1^c)^{\frac{q_b}{y'}}$，再令 $u = \phi(C_3)^{v^{-1}} = g_1^r$。计算

$$\begin{aligned} C_1 &= u \\ C_2 &= Z^{(R+q_b)} \\ C_3 &= H_2(W_b)^r \end{aligned}$$

该密文在敌手的视角中是合法密文，因为当 DLIN 实例中的 $Z = u^{c+d}$ 时，有 $C_2 = Z^{(R+q_b)} = u^{(c+d)(R+q_b)} = g_1^{r(c+d)(R+q_b)} = (g_1^{c(R+q_b)} g_1^{d(R+q_b)})^r = \left(v^{cR + \left(\frac{cq_b}{y'}\right) y'} \phi(w^d)^{(R+q_b)}\right)^r = (g_1^{x + x' y'} g_1^{y H_5(W_b)})^r$。

2) 当身份 ID_{j^*} 存在于 L_3 中时，对应的公钥被 Type-1 类型的敌手替换为 pk'_{j^*}，此时根据文献[19]中的方法，设置参数 $g_1^x = \frac{(v^c)^{(R+q_b)}}{(\mathrm{pk}'_{j^*})^{y'}}$，再令 $u = \phi(C_3)^{v^{-1}} = g_1^r$。计算

$$C_1 = u$$
$$C_2 = Z^{(R+q_b)}$$
$$C_3 = H_2(W_b)^r$$

该密文在敌手的视角中同样为合法密文，理由同上。

因此，当 $Z = u^{c+d}$ 时，这是一个合法的关于身份 ID_{j^*} 的密文；否则，该密文是随机的。最后，将密文 $C_b = \{C_1, C_2, C_3\}$ 作为挑战密文，发送给敌手。

阶段 2。在该阶段，敌手 A_{I} 向 B 进行与阶段 1 中相同的 ParKeyGen、KeyGen、Replace-KeyGen、Trapdoor 询问。唯一的限制是 A_{I} 不能进行关于身份 ID_{j^*} 的 ParKeyGen、KeyGen 询问，也不能询问关于 (ID_{j^*}, W_0) 与 (ID_{j^*}, W_1) 的 Trapdoor。另外，此时也不允许 A_{I} 再次替换目标身份 ID_{j^*} 的公钥。

猜测。敌手 A_{I} 输出一个比特 $b' = \{0,1\}$。若 $b' = b$，则敌手在该游戏获胜，同时 B 输出 b^*。

当敌手 A_{I} 在游戏中获胜时，若 B 在以上模拟游戏中不退出，则需要以下 3 个条件同时满足。

1) 敌手 A_{I} 没有在 ParKeyGen 询问与 PrivateKey 询问阶段向 B 发送过身份 ID_{i^*}。

2) 敌手 A_{I} 没有在陷门询问阶段向 B 同时发送过身份 ID_{i^*} 与 $\sigma_i = 1$ 的消息 M_{j^*}。

3) 敌手 A_{I} 在挑战阶段必须发送的是身份 ID_{i^*} 与 $\sigma_i = 1$ 的消息 M_{j^*}。

则 B 在以上条件下利用 A_{I} 的攻击成功解决 DLIN 问题的优势为

$$\mathrm{Adv}_B^{\mathrm{DLIN}}(\lambda) \geqslant$$
$$\Pr[\mathrm{Event}_1]\Pr[\mathrm{Event}_2]\Pr[\mathrm{Event}_3]\left(\frac{1}{2}+\varepsilon\right)-\frac{1}{2} \geqslant$$
$$\left(1-\frac{1}{q_{H_2}}\right)^{q_{\mathrm{psk}}}\left(1-\frac{1}{q_{H_2}}\right)^{q_{\mathrm{sk}}}\frac{2}{q_{H_2}q_{H_5}}\left(\frac{1}{2}+\varepsilon\right)-\frac{1}{2}$$

引理 1 证毕。

引理 2　在随机预言机模型下，若存在一个 Type-2 类型的敌手 A_{II}，能够以不可忽略的优势 ε 攻破所提方案的 IND-CKA 安全性，则能够构造一个算法 B，利用敌手 A 的能力以不可忽略的优势 $\mathrm{Adv}_B^{\mathrm{DLIH}}(\lambda)$ 求解 DLIN 问题。

证明　引理 2 的证明与引理 1 采用的思路相似，区别是 Type-2 类型的敌手 A_{II} 能够得知系统主密钥 msk，因此需要将 DLIN 问题的元素隐藏在用户公钥 pk_j 中，而非系统公钥 g_1^x 中。为节省篇幅，此处不再展开详细证明。

需要注意的是，Type-2 类型的敌手 A_{II} 不能进行 replace-public-key 询问。由于 Type-2 类型的敌手拥有 msk，因此也不需要进行 ParKeyGen 询问。在挑战阶段，限制 A_{II} 不能得知目标身份的私钥 sk_j^*。

定理 2　若 co-CDH 问题是困难的，则所提方案在随机预言机模型下能够满足 EUF-CMA 安全。

引理 3　在随机预言机模型下，若存在一个 Type-1 类型的敌手 A_{I}，能够以不可忽略的优势 ε 攻破所提方案的 EUF-CMA 安全性，则能够构造一个算法 B，利用敌手 A_{I} 的能力以不可忽略的优势 $\mathrm{Adv}_B^{\text{co-CDH}}(\lambda)$ 求解 co-CDH 问题。

$$\mathrm{Adv}_B^{\text{co-CDH}}(\lambda) \geqslant \left(1-\frac{1}{q_{H_3}}\right)^{q_{\mathrm{psk}}}\left(1-\frac{1}{q_{H_3}}\right)^{q_{\mathrm{sk}}}\frac{1}{q_{H_4}}\varepsilon$$

证明　给定 B 一个 co-CDH 问题的实例 (g_1, g_2, g_1^a, g_2^b)，其中 $a, b \in Z_p$ 未知，B 通过与敌手 A_{I} 进行下面的 EUF-CMA 游戏，最后输出 CDH 问题的解 g_1^{ab}。

初始化。B 运行 Setup 算法，随机选择 $x \in Z_p$ 作为系统主密钥。设置群 G_1, G_2, G_T，其中令 G_1, G_2 中的生成元 g_1, g_2 为 co-CDH 问题实例中的元素。最后，将生成的系统公共参数与公钥发送给 A_{I}。

$$\mathrm{params} = \{G_1, G_2, G_T, g_1, g_2\}$$
$$\mathrm{pk} = g_1^x$$

哈希询问。敌手 A_{I} 在该阶段对 H_3, H_4 这 2 个随机预言机分别进行至多 q_{H_3}, q_{H_4} 次的询问，B 维护 2 个初始为空的列表 L_1, L_2，用来记录 A_{I} 的每次询问。此外，随机指定 2 个 $i^* \in (0, q_{H_3})$，$j^* \in (0, q_{H_4})$。

H_3 询问。令 ID_i 为 A_I 向 H_3 预言机进行的第 i 次询问。B 首先检查 L_1 中是否包含 ID_i 的记录，若存在，则按照 L_1 中对应的内容答复敌手；否则，B 考虑以下 2 种情况。

① 当 $i \neq i^*$ 时，随机选择一个 $a_i \in Z_p$，设置 $H_3(\mathrm{ID}_i) = g_1^{a_i}$。然后将 $g_1^{a_i}$ 作为随机预言机 H_3 的答复返回给敌手，并将三元组 $(\mathrm{ID}_i, H_3(\mathrm{ID}_i), a_i)$ 作为新元素添加到列表 L_1 中。

② 当 $i = i^*$ 时，将 co-CDH 问题实例的元素 g_1^a 作为随机预言机 H_3 的答复，即令 $H_3(\mathrm{ID}_{i^*}) = g_1^a$，并返回给敌手 A_{I}。再将三元组 $(\mathrm{ID}_{i^*}, g_1^a, \perp)$ 添加到列表 L_1 中。

H_4 询问。当 A_I 向 H_4 预言机询问 $(\mathrm{ID}_i,\mathrm{pk}_i,M_j,d_t)$ 时，若 L_2 中存在记录，则 B 按照 L_2 中的内容答复 A_{I}；若 L_2 中不包含关于 $(\mathrm{ID}_i,\mathrm{pk}_i,M_j,d_t)$ 的记录，则 B 考虑以下 2 种情况。

① 当 $i=i^*$ 且 $j\neq j^*$ 时，B 随机选择 $c_j\in Z_p$。将 co-CDH 问题实例的元素 g_1^a 隐藏到 $H_4(\mathrm{ID}_{i^*},\mathrm{pk}_{i^*},M_j,d_t)=g_1^{c_j}(g_1^a)^{-1}=g_1^{-a+c_j}$，并返回给 A_{I} 作为对 A_{I} 询问随机预言机 H_4 的应答。同时，将六元组 $(\mathrm{ID}_{i^*},\mathrm{pk}_{i^*},M_j,d_t,g_1^{-a+c_j},c_j)$ 添加至 L_2 中。

② 当 $i\neq i^*$，或 $i=i^*$ 且 $j=j^*$，B 随机选择 $c_j\in Z_p$。令 $H_4(\mathrm{ID}_i,\mathrm{pk}_i,M_j,d_t)=g_1^{c_j}$，并返回给 A_I 作为对 A_I 询问随机预言机 H_4 的应答。同时，将六元组 $(\mathrm{ID}_i,\mathrm{pk}_i,M_j,d_t,g_1^{c_j},c_j)$ 添加至 L_2 中。

阶段 1。在该阶段，敌手 A_{I} 向 B 进行多项式有界次适应性询问，内容包含以下 5 项。

1) ParKeyGen 询问。敌手 A_{I} 向 B 发送一个身份 ID_i。当 $i=i^*$ 时，中止游戏；否则，B 在收到 ID_i 后，查询 L_1 列表获取到对应的 $H_3(\mathrm{ID}_i)$ 与 a_i，再随机选择 $t_i\in Z_p$，计算 $\mathrm{psk}_i=(g_2^{t_i},g_1^x g_1^{a_i t_i})$，然后将 psk_i 返回给敌手。

2) PrivateKey 询问。敌手 A_{I} 向 B 发送一个身份 ID_j，当 $i=i^*$ 时，中止游戏。否则，B 首先执行 ParKeyGen 算法，然后随机选择 $y_i'\in Z_p$，计算私钥 $\mathrm{sk}_i=(\mathrm{psk}_i,y_i')$ 并返回给敌手。

3) PublicKey 询问。敌手 A_{I} 向 B 发送一个身份 ID_i，当 $i=i^*$ 时，B 隐式地令 ID_{i^*} 的私钥为 $b+r$，其中 $r\in Z_p$，然后令 ID_{i^*} 的公钥为 $\mathrm{pk}_{i^*}=g_2^b g_2^r$；否则，$B$ 执行 KeyGen 算法，得到私钥 y_i'，然后计算公钥 $\mathrm{pk}_i=g_2^{y_i'}$，并将公钥询问的结果返回给敌手。

4) Replace-KeyGen 询问。B 首先建立一个初始为空的列表 L_3。敌手 A_I 随机选择 $\mathrm{sk}_i'\in Z_p$，计算 $\mathrm{pk}_i'=g_1^{\mathrm{sk}_j'}$。随后向 B 发送想要替换的身份 ID_i 与替换公钥 pk_i'，B 将身份 ID_i 对应的公钥 pk_i 使用 pk_i' 替换，并将 $(\mathrm{ID}_i,\mathrm{pk}_i')$ 存储在 L_3 中。

5) Signature 询问。A_{I} 向 B 请求关于身份 ID_i 与消息 M_j 的签名，B 首先判断 $i=i^*$ 且 $j=j^*$ 是否成立，若成立则游戏中止；否则，B 首先发起关于身份 ID_i 的公钥询问，然后考虑以下 2 种情况。

① 当 $i\neq i^*$ 时，无论关于身份 ID_i 的公钥是否被替换，B 将公钥询问中得到的 y_i' 作为私钥，并检查列表 L_1,L_2，然后随机选择 $t_i'\in Z_p$，计算签名

$$\sigma_1=g_1^{t_i'}$$
$$\sigma_2=g_1^x g_1^{a_i t_i'} g_1^{c_i y_i'}$$

然后将签名 $\sigma=(\sigma_1,\sigma_2)$ 发送给 A_I。容易验证 B 模拟的签名是一个 ID_i 关于消息 M_j 的合法签名。

② 当 $i=i^*$ 且 $j\neq j^*$ 时，B 首先令 ID_{i^*} 的部分私钥 $\mathrm{psk}_{i^*1}=g_2^b$，其中隐式设置 $t^*=b$。B 无法使用公钥询问中隐式设置的 y_i' 作为私钥，也无法使用 t^* 计算签名，但 B 可以根据签名的结构，利用 co-CDH 问题实例的元素构造出 ID_{i^*} 的签名

$$\sigma_1=g_2^b$$
$$\sigma_2=g_1^x(g_1^a)^{-r}g_1^{bc_i}g_1^{c_i r}$$

该模拟签名在敌手 A_{I} 的视角中与真实签名不可区分，因为有

$$\sigma_1=g_2^b=\mathrm{psk}_{i^*1}$$
$$\sigma_2=g_1^x(g_1^a)^{-r}g_1^{bc_i}g_1^{c_i r}=$$
$$g_1^x(g_1^a)^{-r}g_1^{ab}g_1^{-ab}g_1^{bc_i}g_1^{c_i r}=$$
$$g_1^x g_1^{ab}g_1^{(-a+c_i)(b+r)}=$$
$$g_1^x H_3(\mathrm{ID}_{i^*})^{t^*}H_4(\mathrm{ID}_{i^*},\mathrm{pk}_{i^*},M_j,d_t)^{y'}$$

且该签名能够通过验证，敌手 A_{I} 验证

$$e(g_1^x,g_2)e(H_3(\mathrm{ID}_{i^*}),\sigma_1)e(H_4(\theta),\mathrm{pk}_{i^*})=$$
$$e(g_1^x,g_2)e(g_1^a,g_2^b)e(g_1^{-a+c_j},g_2^{b+r})=$$
$$e(g_1^x g_1^{ab}g_1^{(-a+c_j)(b+r)},g_2)=$$
$$e(\sigma_2,g_2)$$

其中，$H_4(\theta)=H_4(\mathrm{ID}_{i^*},pk_{i^*},M_j,d_t)$。

伪造。敌手 A_{I} 向 B 发送一个身份 ID_i 关于消息 M_j 的伪造签名 $(\mathrm{ID}_i,\mathrm{pk}_i,M_j,\sigma^*)$。其中，若 $i\neq i^*$ 或 $j\neq j^*$，则中止游戏；若 $i=i^*$ 且 $j=j^*$，B 检查列表 L_1,L_2，然后使用 $H_3(\mathrm{ID}_{i^*})$ 与 $H_4(\mathrm{ID}_{i^*},\mathrm{pk}_{i^*},M_{j^*},d_t)$ 计算 co-CDH 问题实例的元素 g_1^{ab}

$$\frac{\sigma_2^*}{g_1^x(\mathrm{pk}_{i^*}')^{c_j}}=$$
$$\frac{g_1^x H_3(\mathrm{ID}_{i^*})^{t^*}H_4(\mathrm{ID}_{i^*},\mathrm{pk}_{i^*}',M_{j^*},dt)^{\mathrm{sk}'}}{g_1^x(\mathrm{pk}_{i^*}')^{c_j}}=$$
$$\frac{g_1^{ab}g_1^{c_j\mathrm{sk}'}}{(\mathrm{pk}_{i^*}')^{c_j}}=\frac{g_1^{ab}g_1^{c_j\mathrm{sk}'}}{(g_1^{\mathrm{sk}'})^{c_j}}=g_1^{ab}$$

若 B 在以上模拟游戏中不退出，则需要以下 3 个条件同时满足。

1) 敌手 A_{I} 没有在 ParKeyGen 询问与 PrivateKey 询问阶段向 B 发送过身份 ID_{i^*}。

2) 敌手 A_{I} 没有在签名询问阶段向 B 同时发送过身份 ID_{i^*} 与消息 M_{j^*}。

3) 敌手 A_{I} 在伪造阶段必须发送的是关于身份 ID_{i^*} 与消息 M_{j^*} 的签名。

则 B 在以上条件下利用 A_{I} 伪造的签名成功解决 co-CDH 问题的概率为

$$\mathrm{Adv}_B^{\mathrm{co-CDH}}(\lambda)\geqslant$$
$$\Pr[\mathrm{Event}_1]\Pr[\mathrm{Event}_2]\Pr[\mathrm{Event}_3]\varepsilon\geqslant$$
$$\left(1-\frac{1}{q_{H_3}}\right)^{q_{\mathrm{psk}}}\left(1-\frac{1}{q_{H_3}}\right)^{q_{\mathrm{sk}}}\frac{1}{q_{H_4}}\varepsilon$$

引理 3 证毕。

引理 4　在随机预言机模型下，若存在一个 Type-2 类型的敌手 A_{II}，能够以不可忽略的优势 ε 攻破所提方案的 EUF-CMA 安全性，则能够构造一个算法 B，利用敌手 A_{II} 的能力以不可忽略的优势 $\mathrm{Adv}_B^{\mathrm{co-CDH}}(\lambda)$ 求解 co-CDH 问题。

$$\mathrm{Adv}_B^{\mathrm{co-CDH}}(\lambda)\geqslant\left(1-\frac{1}{q_{H_3}}\right)^{q_{\mathrm{sk}}}\frac{1}{q_{H_4}}\varepsilon$$

证明　给定 B 一个 co-CDH 问题的实例 (g_1,g_2,g_1^a,g_2^b)，其中 $a,b\in Z_p$ 未知，B 通过与敌手 A_{II} 进行下面的 EUF-CMA 游戏，最后输出 CDH 问题的解 g_1^{ab}。区别于引理 3，敌手 A_{II} 在该 EUF-CMA 游戏中的询问阶段无法向 B 进行 ParKeyGen 询问与 Replace-PublicKey 询问。

初始化。B 运行 Setup 算法，设置群 G_1,G_2,G_T，其中令 G_1,G_2 中的生成元 g_1,g_2 作为 co-CDH 问题实例中的元素。然后，随机选择 $x\in Z_p$ 作为系统主密钥 $\mathrm{msk}=x$。最后，将生成的系统主密钥、系统公钥、系统公共参数一同发送给 A_{II}。

$$\mathrm{msk}=x$$
$$\mathrm{pk}=g_1^x$$
$$\mathrm{params}=\{G_1,G_2,G_T,g_1,g_2\}$$

哈希询问。敌手 A_{II} 在该阶段对 H_3,H_4 这 2 个随机预言机分别进行至多 q_{H_3},q_{H_4} 次的询问，B 维护 2 个初始为空的列表 L_1,L_2，用来记录 A_{II} 的每次询问。此外，随机指定 2 个 $i^*\in(0,q_{H_3})$，$j^*\in(0,q_{H_4})$。

H_3 询问。令 ID_i 为 A_{II} 向 H_3 预言机进行的第 i 次询问。B 首先检查 L_1 中是否包含 ID_i 的记录，若存在，则按照 L_1 中对应的内容答复敌手；否则，B 随机选择一个 $a_i\in Z_p$，设置 $H_3(\mathrm{ID}_i)=g_1^{a_i}$。然后将 $g_1^{a_i}$ 作为随机预言机 H_3 的答复返回给敌手，并将三元组 $(\mathrm{ID}_i,H_3(\mathrm{ID}_i),a_i)$ 作为新元素添加到列表 L_1 中。

H_4 询问。当 A_{II} 向 H_4 预言机询问 $(\mathrm{ID}_i,\mathrm{pk}_i,M_j,d_t)$ 时，若 L_2 中存在记录，则 B 按照 L_2 中的内容答复 A_{II}；若 L_2 中不包含关于 $(\mathrm{ID}_i,\mathrm{pk}_i,M_j,d_t)$ 的记录，则 B 考虑以下 2 种情况。

① 当 $i=i^*$ 且 $j=j^*$ 时，B 将 co-CDH 问题实例的元素 g_1^a 隐藏到 $H_4(\mathrm{ID}_{i^*},\mathrm{pk}_{i^*},M_j,d_t)=g_1^a$，并返回给 A_{II} 作为对 A_{II} 询问随机预言机 H_4 的应答。同时，将六元组 $(\mathrm{ID}_{i^*},\mathrm{pk}_{i^*},M_j,d_t,g_1^a,\perp)$ 添加至 L_2 中。

② 当 $i\neq i^*$ 或 $j\neq j^*$ 时，B 随机选择 $c_j\in Z_p$。令 $H_4(\mathrm{ID}_i,\mathrm{pk}_i,M_j,d_t)=g_1^{c_j}$，并返回给 A_{II} 作为对 A_{II} 询问随机预言机 H_4 的应答。同时，将六元组 $(\mathrm{ID}_i,\mathrm{pk}_i,M_j,d_t,g_1^{c_j},c_j)$ 添加至 L_2 中。

询问。在该阶段，敌手 A_{II} 向 B 进行多项式有界次适应性询问，内容包含以下 3 项。

1) PrivateKey 询问。敌手 A_{II} 向 B 发送一个身份 ID_i，当 $i=i^*$ 时，中止游戏；否则，B 首先执行 ParKeyGen 算法，随机选择 $t_i\in Z_p$，查询 L_1，计算 $\mathrm{psk}_i=(g_2^{t_i},g_1^x g_1^{a_i t_i})$，然后随机选择 $y_i'\in Z_p$，计算私钥 $\mathrm{sk}_i=(\mathrm{psk}_i,y_i')$ 并返回给 A_{II}。

2) PublicKey 询问。敌手 A_{II} 向 B 发送一个身份 ID_i，当 $i=i^*$ 时，B 隐式地令 ID_{i^*} 的私钥为 $b+r$，其中 $r\in Z_p$，然后令 ID_{i^*} 的公钥为 $\mathrm{pk}_{i^*}=g_2^b g_2^r$；否则，$B$ 执行 KeyGen 算法，得到私钥 y_i'，然后计算公钥 $\mathrm{pk}_i=g_2^{y_i'}$，然后将公钥询问的结果返回给敌手。

3) Signature 询问。A_{II} 向 B 请求关于身份 ID_i 与消息 M_j 的签名，若 $i=i^*$ 且 $j=j^*$，则游戏中止；否则，B 首先发起关于身份 ID_i 的 PublicKey 询问，然后考虑以下 2 种情况。

① 当 $i\neq i^*$ 时，B 检查列表 L_1,L_2，然后进行 ID_i 的公钥询问，并使用 y_i' 作为私钥，再随机选择一个 $t_i'\in Z_p$，计算签名

$$\sigma_1=g_1^{t_i'}$$
$$\sigma_2=g_1^x g_1^{a_i t_i'} g_1^{c_i y_i'}$$

② 当 $i=i^*$ 且 $j\neq j^*$ 时，B 随机选择一个 $t_i'\in Z_p$，构造签名

$$\sigma_1=g_1^{t_i'}$$

$$\sigma_2=g_1^{x}g_1^{a_it_i'}g_1^{bc_i}g_1^{rc_i}$$

然后将签名 $\sigma=(\sigma_1,\sigma_2)$ 发送给 A_{II}。容易验证 B 模拟的签名是一个 ID_i 关于消息 M_j 的合法签名。

伪造。敌手 A_{II} 向 B 发送一个身份 ID_i 关于消息 M_j 的伪造签名 $(\mathrm{ID}_i,\mathrm{pk}_i,M_j,\sigma^*)$。其中，若 $i\neq i^*$ 或 $j\neq j^*$，则中止游戏；若 $i=i^*$ 且 $j=j^*$，B 检查列表 L_1,L_2，然后使用 $H_3(\mathrm{ID}_{i^*})$ 与 $H_4(\mathrm{ID}_{i^*},\mathrm{pk}_{i^*},M_{j^*},d_t)$ 计算 co-CDH 问题实例的元素 g_1^{ab}

$$\frac{\sigma_2^*}{g_1^{x}H_3(\mathrm{ID}_{i^*})^{t^*}H_4(\mathrm{ID}_{i^*},\mathrm{pk}'_{i^*},M_{j^*},dt)^{r}}=$$

$$\frac{g_1^{x}H_3(\mathrm{ID}_{i^*})^{t^*}H_4(\mathrm{ID}_{i^*},\mathrm{pk}'_{i^*},M_{j^*},dt)^{b+r}}{g_1^{x}H_3(\mathrm{ID}_{i^*})^{t^*}H_4(\mathrm{ID}_{i^*},\mathrm{pk}'_{i^*},M_{j^*},dt)^{r}}=$$

$$H_4(\mathrm{ID}_{i^*},\mathrm{pk}'_{i^*},M_{j^*},dt)^{b}=g_1^{ab}$$

若 B 在以上模拟游戏中不退出，则需要以下 3 个条件同时满足。

1) 敌手 A_{II} 没有在 PrivateKey 询问阶段向 B 发送过身份 ID_{i^*}。

2) 敌手 A_{II} 没有在签名询问阶段向 B 同时发送过身份 ID_{i^*} 与消息 M_{j^*}。

3) 敌手 A_{II} 在伪造阶段必须发送的是关于身份 ID_{i^*} 与消息 M_{j^*} 的签名。

则 B 在以上条件下利用 A_{II} 伪造的签名成功解决 co-CDH 问题的概率为

$$\mathrm{Adv}_B^{\mathrm{co-CDH}}(\lambda)\geqslant$$

$$\Pr[\mathrm{Event}_1]\Pr[\mathrm{Event}_2]\Pr[\mathrm{Event}_3]\varepsilon\geqslant$$

$$\left(1-\frac{1}{q_{H_3}}\right)^{q_{\mathrm{sk}}}\frac{1}{q_{H_4}}\varepsilon$$

引理 4 证毕。

5 性能分析

为了展示所提方案在效率方面的优势，本节对其与 4 种对比方案进行了理论性能分析，4 种对比方案分别为高被引的文献[19]、文献[20]、文献[21]以及最新的文献[6]。最后，将所提方案与 4 种对比方案进行了实现，通过实验对比，验证了理论分析的结果。参数含义如表 1 所示。

表 1 参数含义

参数	含义
E	群 G 中的指数运算(Type-1 pairing)
E_1	群 G_1 中的指数运算
E_2	群 G_2 中的指数运算
P	对运算
H_0	群 G 中的 map to point 运算(Type-1 pairing)
H_1	群 G_1 中的 map to point 运算
H_2	群 G_2 中的 map to point 运算

5.1 理论分析

在通信开销方面，文献[19]在密文生成阶段的通信开销为 4 096 bit，且不提供可验证性。文献[6]在密文阶段的通信开销为 7 168 bit，其中验证信息为 2 048 bit。文献[20]在密文生成阶段的通信开销为 5 120 bit，其中包含的关键词密文为 4 096 bit，验证信息为 1 024 bit。由于基于相同的思想，文献[21]在密文生成阶段的通信开销同样为 5 120 bit。而所提方案在该阶段仅需生成密文为 2 226 bit 长度、验证信息为 1 272 bit 长度的无证书签名。由于所提方案采用的群满足 Type-2 类型的双线性对，因此在满足相同的安全等级的同时，群中元素的长度要短于另外 3 种方案中的群元素。通过比较可知，所提方案在密文生成阶段的通信开销明显低于其他对比方案。

在用户执行的陷门生成阶段，文献[19]、文献[21]、文献[20]、文献[6]的通信开销分别为 4 096 bit、4 096 bit、5 120 bit、2 048 bit，而所提方案仅需 1 590 bit，明显优于这 4 种方案。表 2 展示了几种方案的通信开销。

表 2 通信开销

方案	密文生成阶段	陷门生成阶段
文献[19]	4|G|	4|G|
文献[20]	5|G|	4|G|
文献[21]	5|G|	5|G|
文献[6]	5|G|+2|G_{T}|	2|G|
所提方案	2|G_1|+|G_2|	|G_1|+2|G_2|

表 3 展示了几种方案的计算开销。为便于展示，省略了耗时较短的乘法运算与整数群哈希运算，仅展示对计算开销影响较大的 7 种运算（如表 1 所示）。由表 3 可知，所提方案在涉及计算资源受限的终端用户的密钥生成、密文生成阶段的计算效率显著高于其他方案，在陷门生成阶段的计算效率虽与文献[6]持平，但明显高于其他方案。在涉及数据拥有者与

表 3　　计算开销

方案	密钥生成（用户侧）	密文生成	陷门生成	密文搜索	结果验证
文献[19]	$2E$	$9E$	$10E$	$4P$	—
文献[20]	$2E$	$11E+H_0$	$10E$	$4P$	$4P$
文献[21]	E	$11E$	$10E$	$4P$	$E+4P$
文献[6]	$2E$	$8E+H_0+2P$	$3E$	$2P+2E$	$2E+H_0+2P$
所提方案	E_1	$4E_1+H_1$	E_1+3E_2	$3P$	$4P$

云服务器的密文搜索阶段与结果阶段的计算开销也低于其他 4 种方案。

5.2　实验分析

实验设备为树莓派 4B，处理器为四核 ARM Cortex-A72，内存为 2 GB，操作系统为 Raspberry Pi OS。使用 C 语言作为编程语言，调用 PBC 密码库实现群元素运算。本文实验采用 PBC 库推荐的 Type-A 曲线实现满足对称双线性映射的群，群中所有元素均采样自 Type-A 曲线上的点，用于构建文献[19]、文献[20]、文献[21]、文献[6]方案中的实验。同时，为了达到与 Type-A 曲线近似的安全位数，实验采用了 D159 曲线实现满足所提方案的非对称双线性映射的群 G_1、G_2。同理，G_1、G_2 群中的元素均由 D159 曲线上的点构成。实验设置在 G、G_1、G_2 群中元素的安全性一致的条件下（即攻击者破解离散对数问题的代价相同），从构建的群中选取参数，实现方案中的算法与功能，衡量不同方案的时间开销。Type-A 曲线与 D159 曲线的详细实验参数与树莓派运行 benchmark 的时间如表 4 所示。

表 4　　实验参数

曲线	表达式	基域的阶/bit	安全位数/bit	群 G_1 中指数运算耗时/ms	群 G_2 中指数运算耗时/ms	对运算耗时/ms	map to point 运算耗时/ms
Type-A	$y^2=x^3+x \bmod q$	512	1 024	6.726	—	9.151	14.280
D159	$y^2=x^3+ax+b \bmod q$	159	954	12.322	1.262	10.532	0.115(G_2)

实验分别模拟实现了 KGC 的初始化与部分私钥生成算法、数据拥有者执行的密钥生成算法与加密算法、云服务器执行的搜索算法以及用户执行的密钥生成算法与陷门算法。实验中各算法执行 100 次，然后记录各方案的单次运行平均耗时。

在关键词密文生成阶段，4 种对比方案为了实现关键词不可区分性，分别采用了 9 次、11 次、11 次、8 次群 G 中的指数运算，且文献[20]中包含了 1 次 map to point 运算、文献[6]中包含了 1 次 map to point 运算和 2 次对运算。而所提方案实现相同的安全特性仅需 4 次 G_1 群中的指数运算和 1 次 map to point 运算。根据图 5 中的实验结果可知，所提方案的计算开销为 49.403 ms，分别低于文献[19]的 60.534 ms、文献[20]的 88.266 ms、文献[21]的 73.986 ms 和文献[6]的 86.390 ms。对比实验结果表明，所提方案在密文生成算法上的计算效率优于其他 4 种方案。

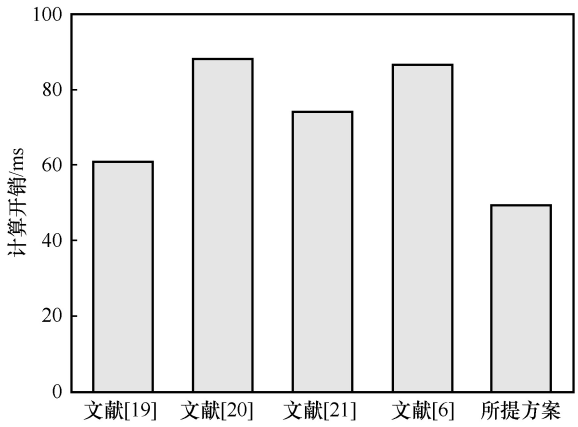


图 5　关键词密文生成阶段的计算开销

终端用户负责执行陷门生成算法，其计算开销如图 6 所示。4 种对比方案在该阶段为了生成与密文相匹配的陷门，同时保障陷门内关键词的机密性，文献[19]、文献[20]和文献[21]均需要用户执行 10 次群 G 中的指数运算，计算开销较大。由于文献[6]优化了陷门生成算法，仅需执行 3 次群 G 中的

指数运算，大幅降低了计算量。在所提方案中，用户需要执行 1 次 G_1 群中的指数运算和 3 次 G_2 群中的指数运算。然而，由表 4 可知，所提方案采用了带有 Type-2 类型 pairing 特性的曲线，其 G_2 群中的指数运算开销要显著低于其他方案的指数运算开销，因此计算效率依然优于文献[6]。由图 6 可知，所提方案的计算效率不仅优于文献[6]，更较文献[19]、文献[20]、文献[21]提升了 77.68%，具有明显优势。

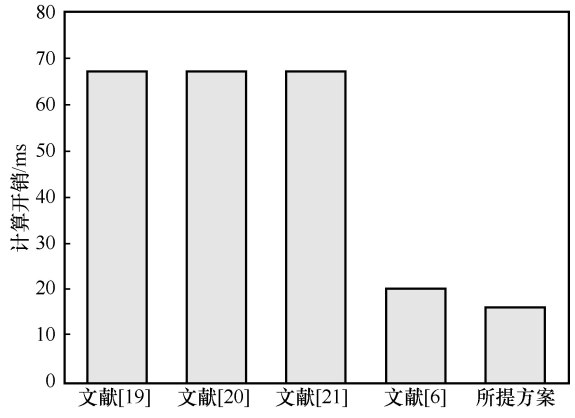


图 6 陷门生成阶段的计算开销

用户总负载记录了在方案实现的全部过程中用户端花费的全部计算开销。文献[19]由于不支持搜索结果的公开验证性，因此用户总计算开销低于文献[20]、文献[21]、文献[6]。而文献[20]、文献[21]的算法设计导致 2 种方案中用户的计算开销（如表 3 所示）相近。文献[6]虽然优化了陷门生成算法和验证算法，其密文生成算法涉及了多种耗时的运算，但计算效率仍低于所提方案。由图 7 可知，所提方案不仅实现了搜索结果的公开验证性，还采用了更少的指数运算次数，以及计算开销更低的曲线，因此效率显著高于其他方案，更能适用于如智能手机等资源有限的终端设备，具有较高的实用价值。

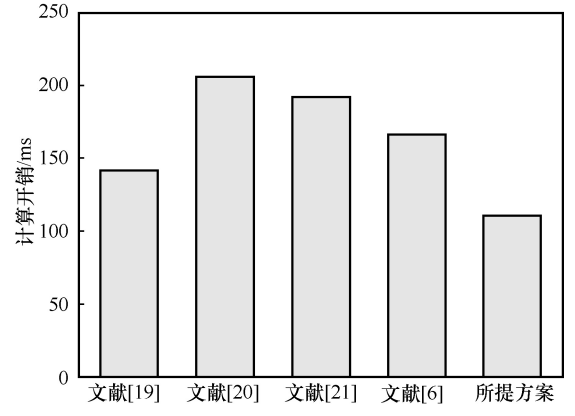


图 7 用户总负载

6 结束语

本文首先对可验证的无证书可搜索加密进行了研究，指出了文献[21]方案存在安全缺陷，无法抵抗恶意用户的猜测攻击。然后，结合无证书加密、无证书签名与改进的 Merkle Tree 方法，提出了一种高效的可验证无证书可搜索加密方案。与现有的无证书可搜索加密方案相比，所提方案既实现了搜索结果的可验证性，又能满足无证书场景中的安全性，同时，所提方案具有更高的计算效率与更小的通信开销，在结合云计算的实际应用场景中具备更强的实用性。

参考文献：

[1] SONG D X, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[C]//Proceedings of IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 2002: 44-55.

[2] BONEH D, CRESCENZO G D, OSTROVSKY R, et al. Public key encryption with keyword search[C]//International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2004: 506-522.

[3] GOLLE P, STADDON J, WATERS B. Secure conjunctive keyword search over encrypted data[C]//International Conference on Applied Cryptography and Network Security. Berlin: Springer, 2004: 31-45.

[4] KAMARA S, PAPAMANTHOU C, ROEDER T. Dynamic searchable symmetric encryption[C]//Proceedings of the 2012 ACM Conference on Computer and Communications Security. New York: ACM Press, 2012: 965-976.

[5] CURTMOLA R, GARAY J, KAMARA S, et al. Searchable symmetric encryption: improved definitions and efficient constructions[J]. Journal of Computer Security, 2011, 19(5): 895-934.

[6] 张键红, 武梦龙, 王晶, 等. 云环境下安全的可验证多关键词搜索加密方案[J]. 通信学报, 2021, 42(4): 139-149.
ZHANG J H, WU M L, WANG J, et al. Secure and verifiable multi-keyword searchable encryption scheme in cloud[J]. Journal on Communications, 2021, 42(4): 139-149.

[7] SHACHAM H, WATERS B. Compact proofs of retrievability[J]. Journal of Cryptology, 2013, 26(3): 442-483.

[8] GUO L F, LU B, LI X Y, et al. A verifiable proxy re-encryption with keyword search without random oracle[C]//Proceedings of Ninth International Conference on Computational Intelligence and Security. Piscataway: IEEE Press, 2014: 474-478.

[9] YANG Y, MA M D. Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for E-health clouds[J]. IEEE Transactions on Information Forensics and Security, 2016, 11(4): 746-759.

[10] SUN W H, WANG B, CAO N, et al. Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking[J]. IEEE Transactions on Parallel and Distributed Systems,

2014, 25(11): 3025-3035.

[11] WANG J F, CHEN X F, HUANG X Y, et al. Verifiable auditing for outsourced database in cloud computing[J]. IEEE Transactions on Computers, 2015, 64(11): 3293-3303.

[12] MIAO Y B, MA J F, LIU X M, et al. VMKDO: verifiable multi-keyword search over encrypted cloud data for dynamic data-owner[J]. Peer-to-Peer Networking and Applications, 2018, 11(2): 287-297.

[13] GUO Y, ZHANG C, JIA X H. Verifiable and forward-secure encrypted search using blockchain techniques[C]//Proceedings of 2020 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2020: 1-7.

[14] PENG Y G, CUI J T, PENG C G, et al. Certificateless public key encryption with keyword search[J]. China Communications, 2014, 11(11): 100-113.

[15] 郑东, 朱天泽, 郭瑞. 基于区块链的多用户环境中公钥可搜索加密方案[J]. 通信学报, 2021, 42(10): 140-152.
ZHENG D, ZHU T Z, GUO R. Public key searchable encryption scheme in blockchain-enabled multi-user environment[J]. Journal on Communications, 2021, 42(10): 140-152.

[16] MA M M, HE D B, KUMAR N, et al. Certificateless searchable public key encryption scheme for industrial Internet of things[J]. IEEE Transactions on Industrial Informatics, 2018, 14(2): 759-767.

[17] 张玉磊, 文龙, 王浩浩, 等. 多用户环境下无证书认证可搜索加密方案[J]. 电子与信息学报, 2020, 42(5): 1094-1101.
ZHANG Y L, WEN L, WANG H H, et al. Certificateless authentication searchable encryption scheme for multi-user[J]. Journal of Electronics & Information Technology, 2020, 42(5): 1094-1101.

[18] WU L B, ZHANG Y B, MA M M, et al. Certificateless searchable public key authenticated encryption with designated tester for cloud-assisted medical Internet of things[J]. Annals of Telecommunications, 2019, 74(7): 423-434.

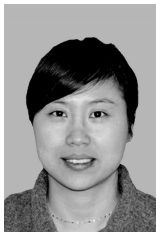
[19] ZHENG Q J, LI X X, AZGIN A. CLKS: certificateless keyword search on encrypted data[C]//International Conference on Network and System Security. Berlin: Springer, 2015: 239-253.

[20] MIAO Y, WENG J, LIU X, et al. Enabling verifiable multiple keywords search over encrypted cloud data[J]. Information Sciences, 2018, 465(10): 21-37.

[21] 田有亮, 骆琴. 基于改进 Merkle-Tree 认证方法的可验证多关键词搜索方案[J]. 通信学报, 2020, 41(9): 118-129.
TIAN Y L, LUO Q. Verifiable multi-keyword search scheme based on improved Merkle-Tree authentication method[J]. Journal on Communications, 2020, 41(9): 118-129.

[22] GARG N, BAWA S. RITS-MHT: relative indexed and time stamped Merkle hash tree based data auditing protocol for cloud computing[J]. Journal of Network and Computer Applications, 2017, 84: 1-13.

[作者简介]



崔新华（1982− ），女，侗族，贵州凯里人，贵州大学博士生，主要研究方向为密码学与信息安全、公钥密码。

田有亮（1982− ），男，贵州盘县人，博士，贵州大学教授、博士生导师，主要研究方向为算法博弈论、密码学与安全协议、大数据安全与隐私保护等。

张起嘉（1995− ），男，河北衡水人，贵州大学博士生，主要研究方向为公钥密码学。