基于访问代理的数据加密及搜索技术研究

王国峰[1]，刘川意[2]，韩培义[1]，潘鹤中[1]，方滨兴[2]
（1. 北京邮电大学网络空间安全学院，北京 100876；
2. 哈尔滨工业大学（深圳）计算机科学与技术学院，广东 深圳 518055）

摘 要：针对云应用程序数据机密性问题，提出一种访问代理执行的密文搜索方案。此方案不需要修改云应用程序且不改变用户使用习惯，具有很强的可适用性。首先从功能性、效率性和安全性等方面分析了基于访问代理的密文搜索方案，并指出其所面临的关键问题，包括代理间索引和密文的安全分享，并设计解决方案。实验结果表明，此方案可有效保护云服务用户数据，实现多种搜索功能，且具有很高的效率性和安全性。

关键词：云安全；数据保护；密文搜索；密文分享
中图分类号：TP302
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2018114

Research on technology of data encryption and search based on access broker

WANG Guofeng[1], LIU Chuanyi[2], HAN Peiyi[1], PAN Hezhong[1], FANG Binxing[2]
1. College of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China

Abstract: Broker executed searchable encryption (BESE) scheme was proposed for the confidentiality issues of cloud application data. The scheme did not need to modify the cloud application or user habits, thus had strong applicability. Firstly, systematic and quantitative analysis on BESE scheme was conducted in terms of query expressiveness, performance and security. Then, the main challenges of BESE scheme including securely sharing index and encrypted data between brokers were pointed out, and corresponding schemes were proposed to address the above challenges. The experimental results show that the BESE scheme can effectively protect the user data in the cloud, achieve a variety of search functions, and has high efficiency and security.

Key words: cloud security, data protection, searchable encryption, encrypted data sharing

1 引言

根据云安全联盟（CSA）的报告[1]，数据泄露是云计算面临的重要威胁之一。2016 年上半年，共有 974 个公开披露的数据泄露事件，导致 5.44 亿条数据记录被窃取[2]。在数据外包到云服务之前将敏感数据加密是解决数据泄露的一个有效措施[3]。

然而，在客户端加密数据势必与云应用程序的数据计算功能产生冲突，其中搜索是最常见的计算

收稿日期：2017−09−25；**修回日期**：2018−04−28
通信作者：刘川意，cy-liu04@mails.tsinghua.edu.cn
基金项目：国家高技术研究发展计划（“863”计划）基金资助项目（No.2015AA016001）；国家重点研发计划基金资助项目（No.2017YFB0801801）；国家科技重大专项基金资助项目（No.BB29100002）；国家科研发展咨询基金资助项目（No.BA25500031, No.BB25500019）
Foundation Items: The National High Technology Research and Development Program of China (863 Program) (No. 2015AA016001), The National Key Research and Development Program of China (No.2017YFB0801801), The National Science and Technology Major Project of China (No.BB29100002), The National Research Development Consulting Project (No.BA25500031, No. BB25500019)

功能之一。目前，密文搜索（SE, searchable encryption）方案需要在查询功能、安全性和效率之间做出不同的权衡和取舍。Oblivious RAM（ORAM）[4]可以很好地满足安全性和查询功能的需求，但对于大规模的实际应用来说效率极其低下。确定性加密可最大限度地提高效率，但不具有很高的安全性。许多研究人员专注于基于加密索引的密文搜索技术，其包括以下步骤：用户生成加密文档和可搜索的加密索引；加密的文件和加密的索引上传到云服务器；为了搜索某关键字，用户生成一个对应的搜索陷门（trapdoor）；使用陷门，云服务器可以搜索加密索引并返回对应的加密文档。本文将这种方法称为云端执行的密文搜索（CESE, cloud executed SE）方案。CESE 方案可以很好地满足效率性和安全性。然而，CESE 方案往往会失去一些查询表达能力[5]，并且需要修改当前的云应用程序。

近年来，产业界开始提倡云访问安全代理（CASB, cloud access security broker）技术[6]，其中访问代理透明地位于云应用程序和用户之间。在敏感数据传入云端之前，CASB 对其拦截并加密，实现数据保护。然而，目前大多数 CASB 解决方案都来自工业界，技术细节不是公开的。本文针对云应用程序提出了访问代理执行的密文搜索（BESE, broker executed SE）方案，并从查询功能、性能和安全性方面对 BESE 方案做了系统的定量分析。为了不失一般性，选择 3 个常见的搜索功能作为代表，即多关键字排序搜索、模糊搜索和动态更新[7]。从定量比较和安全分析方面可以看出，BESE 方案是保护云应用程序数据的有效解决方案。

然而，并不是把密文搜索的体系结构一改，所有的问题就都自然而然地解决了。在实用性方面，本文指出了 BESE 方案面临的一些重要问题，这些问题在其他基于访问代理执行的密文搜索方案中均未涉及，总结如下。

1) 不同访问代理间的索引分享。例如，位于代理 B_1 下的用户 u_1，迁移到代理 B_2 下并想使用原来位于代理 B_1 下的索引进行搜索。

2) 不同访问代理间的密文分享。例如，位于代理 B_1 下的用户 u_1，想要和位于代理 B_2 下的用户 u_2 分享部分数据。然而，由于数据在 B_1 下加密，故需要在代理 B_1 与 B_2 之间安全地分享密钥。

为了解决上述挑战，本文提出一种索引分享方案以及一种结合身份加密（IBE）和公钥加密（PKE）的双层加密方案，以在代理之间安全地分享索引和加密密钥。最后本文在实际云应用程序上对 BESE 方案进行了评估，并对实验结果进行了评价。

2 相关工作

2.1 加密系统

在用户和客户端之间，ShadowCrypt[8]以浏览器插件模式运行，以执行加密和解密功能。但 ShadowCrypt 仅支持文本输入数据，不支持移动平台。Lau 等[9]提出了基于移动平台的用户和应用层之间的 7.5 层的 M-Aegis。但 M-Aegis 也是只支持文本数据，而 BESE 方案支持非文本数据和移动平台数据。

在客户端和服务器端之间，Mylar[10]基于 Meteor JavaScript 框架保护数据免受恶意服务器管理员的窃取，影响了后向兼容性。Virtru 作为另一个浏览器插件执行电子邮件加密，使网络邮件提供商无法访问用户的数据。Virtru 仅支持 Gmail 等电子邮件，无法适应其他云应用程序。BESE 方案也处于客户端和服务器端之间，但可以透明地与多个云应用程序集成。

在服务器端和数据库之间，CryptDB[11]将数据加密后上传到数据库，并可对加密数据执行查询请求，从而有效防护有恶意的数据库管理员。CryptDB 无法防护服务器破坏程序。同时查询请求可能需要多次加密和解密操作。相比之下，BESE 方案可以有效地保护数据隐私，防止云服务器破坏程序。

2.2 密文搜索

Song 等[12]提出了第一个可实用的密文搜索方案，其搜索操作简单，但云端需要全文扫描，计算量与数据大小成正比。另外，云端可以使用查询统计信息来获得额外的信息。Goh[13]使用 Bloom Filter 设计了一种安全的索引方案，实现密文搜索功能。但是，该方案由于使用 Bloom Filter，可能导致搜索结果不正确。另外，它不是一个亚线性搜索方案。Curtmola 等[14]构建基于反向索引的密文搜索机制，大大提高了搜索效率和密文搜索的安全性。但它只能支持精确的关键字搜索，文档无法动态更新。根据文献[14]，搜索模式意味着给定 2 个具有相同结

果的搜索陷门，是否可以确定 2 个陷门对应相同的关键字。访问模式是指查询结果中可能泄露的信息。如果密文没有泄露明文信息，则密文搜索方案为“选择明文攻击”（CPA, chosen plaintext attack）安全。如果用户一次生成所有查询，搜索过程中密文和索引不会泄露除访问模式和搜索模式之外的任何有关文本和查询的信息，则密文搜索方案是“非自适应选择关键字攻击”（CKA1, non-adaptive chosen keyword attack）安全。在 CKA1 的基础上，“自适应选择关键字攻击”（CKA2, adaptive chosen keyword attack）安全允许用户根据已查询的陷门和返回的搜索结果进行查询。

近年来，支持高级搜索功能的密文搜索方案得到进一步发展。Xia 等[15]提出了一种基于树的多关键字排序搜索方案，满足亚线性搜索时间。它是一种 CKA1 安全方案，会泄露访问模式和一定量的相关性分数。方案中关键字字典的大小是固定的，不能动态地改变。如果将新关键字添加到字典中，则必须更新整个索引。文档更新操作要求用户存储完整的索引以生成更新信息，并将更新信息发送给云端。Li 等[16]通过预先设定各关键字基于通配符的模糊集合来构建模糊搜索方案。此方案是一种 CKA1 安全方案，允许加密索引泄露一定的编辑距离信息。它构建基于通配符的模糊集合，索引需要耗费一定的空间，且查询需要耗费一定的时间。Kamara 等[17]设计动态更新机制，但需要维护复杂的数据结构。该方案在特定的条件下是 CKA2 安全的，但它泄露搜索模式和访问模式，并且在更新过程中泄露某些关键字出现在特定文档中的信息。Boneh 等[18]通过使用非对称加密算法实现多用户密文搜索（PEKS），这种方案会带来很高的时间开销。Liu 等[19]提出了基于 PEKS 的密文搜索（SPKS）方案，允许云服务提供商参与部分解密，而不需要知道明文的具体内容。

上述所有密文搜索方案都要修改当前的云端编程接口。对于每个搜索功能，CESE 方案需要生成一个特定结构的索引，然后用特定算法加密索引并上传到云端。为了搜索，用户需要生成特定的陷门，然后云端可以使用特定的算法对加密的索引执行特定的查询。它们给云端应用程序带来了额外的负担，同时在一个加密索引上实现多个功能是不切实际的。BESE 方案可以适用于多个云应用程序，而不需要修改云服务和用户程序。

2.3　密钥分享

如果数据用不同的密钥加密，密文分享就需要相应的密钥分享。常用的密钥分享机制是 PKE 和 IBE。2 个知名的 PKE 证书验证方案是证书吊销列表（CRL）和在线证书状态协议（OCSP）。但是，在撤销大量证书的情况下，2 种方案的效率都很低[20]。“Novomodo”系统[21]提升了效率，但它带来第三方证书状态查询的问题。为了消除第三方证书状态查询，IBE 方案是一种有效的方式。Shamir[22]提出了基于身份的加密，并引入了基于身份的签名方案。但是，这并不是可完全实用的方案。Boneh 等[23]提出了基于 Weil 对的可实用方案。然而，IBE 本身具有私钥托管问题：私钥授权方可以解密用户的密文数据。为了解决这个问题，基于证书的加密[20]和无证书公钥加密[24]结合 IBE 和 PKE 来实现双重加密，Lewko 等[25]设计了一个支持多个授权方的方案。为了安全地跨代理分享密钥，并可方便地实现密钥更新，本文设计了实用的 TLES 方案，结合 IBE 和 PKE 各自的优势安全地在访问代理间分享密钥，并对其性能做了测试。

3　BESE 方案

3.1　系统模型

如图 1 所示，BESE 方案由用户、云服务和访问代理 3 个关键角色组成。CESE 方案中密文搜索功能由云端执行，而 BESE 方案中搜索功能在访问代理的索引服务器中执行。云服务的用户可对文件进行上传、搜索、下载、分享等操作。访问代理对用户和云服务之间的应用层（HTTP 或 HTTPS）连接进行处理，以保护用户的敏感数据。访问代理对用户的明文数据和文件标识符进行索引实现搜索功能，其中文件标识符指向云服务中的加密数据。

3.2　威胁模型

在本文的威胁模型中，云服务器被认为是诚实但好奇的，即遵循服务协议，但是有窃取用户敏感数据的动机。在 CESE 密文搜索方案中，尽管用户的数据已经被加密，但云服务器也可通过分析加密数据和索引，统计查询请求和结果获得其他的敏感信息。BESE 方案确保整个搜索过程以安全的方式进行，数据在被检索的同时，向云端和攻击者泄露尽可能少的信息。而在 BESE 方案中，访问代理位于客户端，是可被充分信任的，同时确保访问代理内的通道和索引服务器是安全的，不会受到外部访

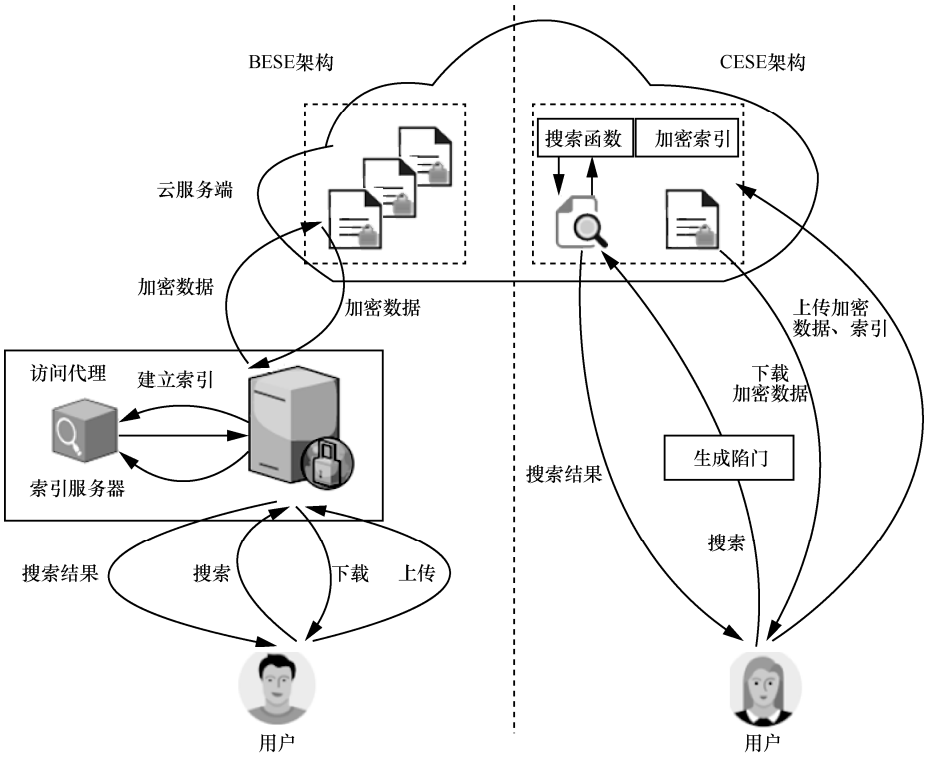

图 1　BESE 和 CESE 系统模型

问和攻击。访问代理以外的链路（包括云服务器和其他访问代理）被认为是不受信任的，需要受到防护。敏感数据在传递给云端之前由访问代理加密，从而有效防止云服务器窃取用户私有信息。另外，即使云服务器和其他访问代理联系，它们也无法恢复明文，因为它们无法获得位于用户访问代理中的相应密钥。在访问代理之外，即使用户账户被攻击者窃取，攻击者也只能得到加密数据。

3.3　BESE 系统架构

在 BESE 系统架构中，访问代理是其主要组成部分，它通过分析网络协议识别和加密用户的敏感数据。如图 2 所示，访问代理主要由以下 4 个部分组成。

1) 过滤器：主要包括客户端过滤器和服务端过滤器，用于识别特定的应用层协议。根据特征字段（URL、SNI 等），调用相应的解析器逻辑。

2) 解析器：主要用于协议识别和语义分析。它通过分析请求内容格式（如键值对、multi-part 等）来识别敏感字段，并缓存要搜索的文件内容。当从响应内容中得到文件标识符（指向云端存储的加密

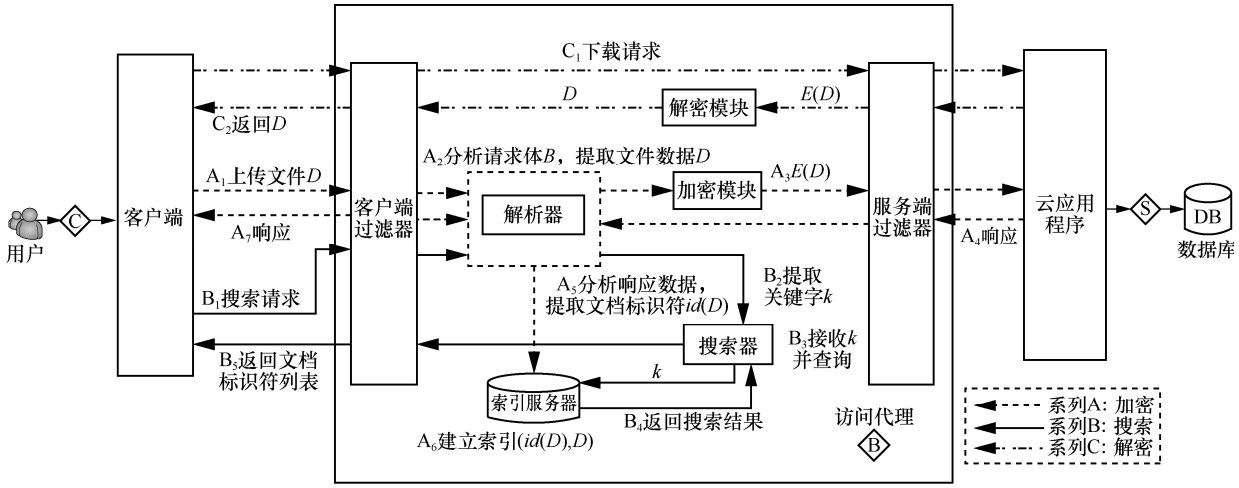

图 2　BESE 架构

的文件数据）后，解析器将其与缓存的文件数据相关联并建立可搜索的索引。

3) 加密/解密：加密模块对敏感数据进行对称加密，属于同一文档的密文数据具有相同的元数据，包括密钥 *ID*、代理 *ID*、特征数据、头长度等。特征数据是用于标识加密数据的符号串，以使解密模块可以快速识别密文数据。解密模块首先根据元数据中的特征数据找到密文，然后利用密钥 *ID* 找到和密文对应的解密密钥，解密得到明文数据。

4) 索引服务器/搜索器：索引服务器使用指向云端加密文件数据的文件标识符和缓存的文件数据建立搜索索引，并维护和管理索引数据。搜索器执行搜索操作，并返回相应的文件标识符（可用于向云服务器请求加密的文件数据）。

加密、解密和搜索的执行过程如下。

加密。① 应用层文档数据首先到达客户端过滤器，过滤后到达解析器模块。解析器分析请求数据并提取由加密模块加密的敏感数据。② 加密模块首先生成一个对称密钥 *k*，将密钥 *k* 和密钥 *ID* 存储在数据库中，然后用对称密钥 *k* 加密数据并将密文有关的元数据附加到密文头，最后它将加密的数据发送到云端。③ 访问代理将加密文档上传到云端后，云端会返回文档标识符。④ 使用缓存的文档数据和文档标识符，索引服务器索引数据并将其与文档标识符相关联。

解密。① 云服务应用层数据首先到达服务端过滤器。② 被过滤后，数据由解密模块进行分析。解密模块通过扫描特征数据来定位密文。③ 解密模块根据密文元数据中的密钥 *ID* 查询数据库获取对应的对称密钥 *k* 来解密数据，得到明文数据内容，返回给用户。

搜索。① 用户输入关键字发起查询请求。② 访问代理拦截并分析查询请求获取搜索关键字。访问代理继而使用搜索关键字向索引服务器发起搜索请求。索引服务器返回对应的文档标识符，访问代理将文档标识符作为列表转发给用户。③ 此后，用户可以根据指向云端相关加密文件的标识符选择要下载的相关文档。④ 当云端返回相关的加密文档后，访问代理解密并将明文数据转发给用户。

如上所述，访问代理依次加密并发送上传的文档数据块，而不是等待文档的所有内容被上传后才进行加密并发送。在加密之前，同一文档的数据被缓存在某一存储结构中。当文档的整个内容上传之后，云端返回指向加密文档的文档标识符。访问代理使用缓存数据和文档标识符建立本地索引，以支持加密文档的搜索功能。算法 1 描述数据加密及索引建立的过程，其中，*E* 表示对称加密方案，$L(w)$ 表示关键字 w 的倒排表，表中每个节点包含关键字 w 的文档 D_i 的文档标识符 $id(D_i)$以及关键字在文档 D_i 中的频率（TF, term frequency）。逆向文档频率（IDF, inverse document frequency）指某关键字在所有文档中出现的频率，$f(w)$表示关键字 w 与其倒排表 $L(w)$之间的映射关系。

算法 1　数据加密及索引建立

输入　上传数据文件 D_i，对称密钥 k

输出　文件 D_i 对应的搜索索引

for 文件 D_i 中每个上传的数据块 M do
　缓存文件 D_i 中的数据内容 M;
　利用对称密钥 k 加密 M 得到 $E_k(M)$并上传到云服务端。
文件的所有数据块上传完成后会从云端得到文档标识符 $id(D_i)$;
为缓存文件 D_i 的所有数据内容和文档标识符 $id(D_i)$之间建立映射关系。
for 文件 D_i 中的每个关键字 w_i do
　if w_i 在字典 W 中 then
　　根据 $f(w_j)$获取倒排表 $L(w_j)$，并将 $id(D_i)$、$TF(w_j, id(D_i))$添加到倒排表 $L(w_j)$中；更新字典 W 中的 $IDF(w_j)$值。
　else
　　添加 w_j、$IDF(w_j)$到字典 W 中；
　　根据 $f(w_j)$初始化倒排表 $L(w_j)$，并将 $id(D_i)$、$TF(w_j, id(D_i))$添加到 $L(w_j)$中。
return 搜索索引 $I = (W, L)$。

3.4　查询表达能力

为了分析 BESE 的查询表达能力，选择 3 个具有代表性的搜索功能进行比较，即多关键字排序搜索、模糊搜索和动态更新。

3.4.1　多关键字排序搜索

在多关键字排序搜索中，*D* 代表文档集合，$\boldsymbol{Q}$ 代表包含多个关键字的查询向量，正整数 n 代表在查询后返回具有和查询向量最相关的 n 个文档。

在 BESE 中，索引在访问代理中构建和搜索，而不是云服务器，故可以很方便地构建多关键字排

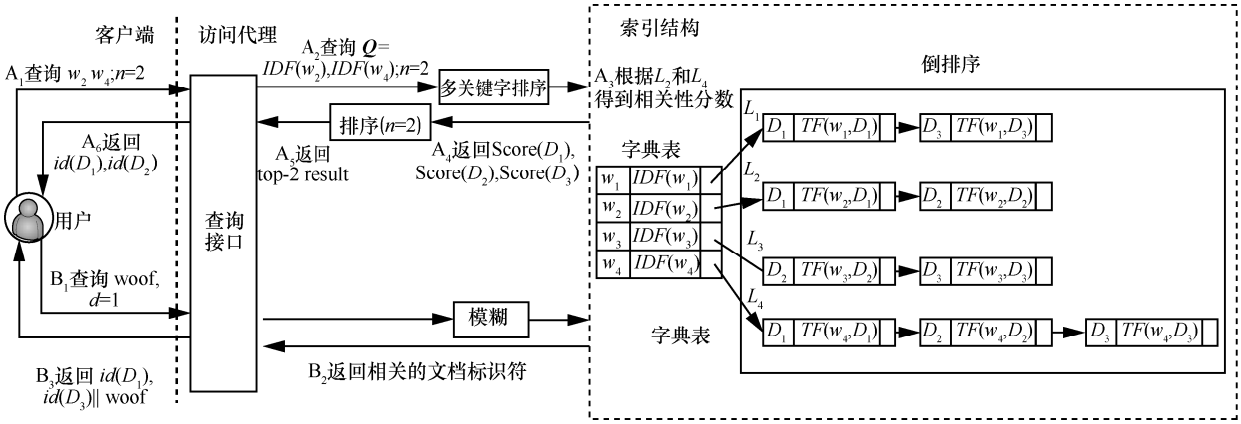

图 3　BESE 系统模型中多关键字排序和模糊搜索过程

序搜索方案，并可以有效地计算查询向量与文档的相关性分数。如图 3 所示，访问代理首先扫描文件集 D，构建包含所有关键字的字典 W，并计算各关键字出现在某文档中的频率 TF 和出现在所有文档中的逆向文档频率 IDF 值。为了对每个关键字进行快速搜索，访问代理为每个关键字 $w \in W$ 构建了具有长度$|D(w)|$的倒排表 $L(w)$。算法 2 用于在文档集 D 上计算各文档与查询向量的相关性分数。数组 $RScores$ 保存了包含任意一个查询词的 N 个文档的分数，$D_{i,j}$ 表示$|D(w_j)|$中的第 i 个文档。当得到最终的 N 个分数后，选择具有最高分数的 n 个文档作为搜索结果。

算法 2　BESE 多关键字排序搜索

输入　关键字查询向量 $\boldsymbol{Q}$，返回文档数量 n

float $Rscores[N] = 0$;

for each $w_j \in \boldsymbol{Q}$

获取 w_j 对应的 $IDF(w_j)$ 和倒排表 $L(w_j)$;

for $1 \leqslant i \leqslant |D(w_j)|$:

从倒排表 $L(w_j)$ 中获取 $id(D_{i,j})$ 和 $TF(w_j,id(D_{i,j}))$;

$Rscores[id(D_{i,j})]+=TF(w_j,id(D_{i,j}))\cdot IDF(w_j)$。

从 $Rscores[N]$中得到与查询最相关的 n 个文档。

3.4.2　模糊搜索

与确切的关键字搜索不同，当关键字拼错或格式有误时，模糊搜索仍能找到近似的关键字并对其进行搜索。例如，原本需要检索包含关键字“wood”的文档，当将关键字拼写为“woof”时仍能返回同样正确的结果。

BESE 构建的模糊搜索方案不必预先计算字典中每个关键字的模糊集合，可利用 Levenshtein 距离计算 2 个关键字的相似度。根据 Levenshtein 自动机[26]，对于长度为 L_1 的关键字 W_1 和长度为 L_2 的关键字 W_2，则可以在时间复杂度 $O(\max(L_1, L_2))$中判定 2 个关键字是否在编辑距离 d 内。使用该方案，给定一个编辑距离 d，如果在字典中有 m 个不同的关键字，最长关键字的长度为 l，则一次模糊搜索的最大时间复杂度为 $O(ml)$。进一步地，更快速的方法是为字典中的关键字构建一个关键字查找树，然后可以在搜索某关键字时及早去除不相关的关键字。图 4 为包含关键字“wood”“can”和“cash”的关键字查找树。当查找与查询词“woof”Levenshtein 距离不大于 1 的关键字时，按序测试这个树中的字母，以确定是否需要继续搜索。比较“ca”后，左侧分支可以被去除，因为所有以“ca”开头的关键字与“woof”的 Levenshtein 距离肯定大于 1。最后，得到并搜索与“woof”的 Levenshtein 距离不大于 1 的关键字“wood”。

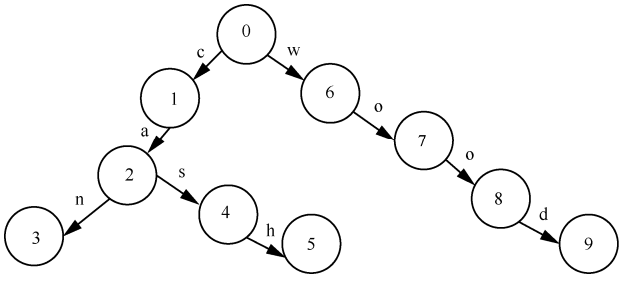

图 4　关键字“wood”“can”和“cash”的单词查找树

3.4.3　动态更新

实用的 SE 方案应支持密文数据的动态更新，即可以添加和删除可搜索的加密文件。为了满足动态更新功能，SE 方案需支持向字典添加新的关键字

的操作，并且可以对现有关键字的倒排表中的文档标识符进行修改。

对于 BESE 建立的反向索引，在主索引上就地更新非常耗时[7]。如图 5 所示，要快速添加新的文档，BESE 方案维护了 2 个索引：一个主索引和一个辅助索引。主索引存储在磁盘中，辅助索引用于索引新文档并保存在内存中。当辅助索引消耗空间大于一定阈值时，它会被合并到主索引中。

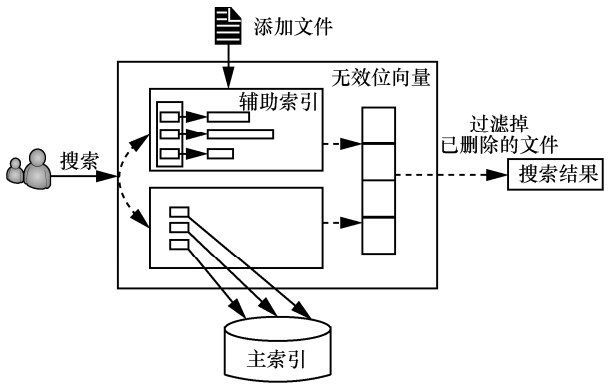

图 5　BESE 动态更新模型

当在 BESE 动态更新方案中搜索时，需要在主索引和辅助索引上分别处理搜索请求，然后合并结果。添加新文档时，访问代理程序中的索引服务器将为文档构建辅助索引。删除文档时，将更新标识文档被删除的无效位向量，以表明文档已被删除。然后，索引服务器将在返回搜索结果之前根据无效位向量过滤掉已删除的文档。如果文档内容被更新，它将被删除并重新插入文档集中。

BESE 方案需要索引服务器来管理索引并执行查询功能，并可以在反向索引结构中很好地实现多种搜索功能。它支持访问代理中的多用户交互。此外，它还可以适应多个云应用程序，而不需要修改云应用程序和用户程序。

3.5　性能分析

表 1 对比了典型的具有特殊搜索功能的 CESE 方案和 BESE 方案之间的性能，其中，m 为关键字个数，n 为文档个数，q 为一次查询关键字个数，p 为平均每个文档含有不同关键字的个数，l 为关键字的平均长度，t 为关键字对应的模糊集的大小，e 为编辑距离，$N=\sum_{w}|D(w)|$，$|W_{id}|$表示在一次更新中变化的关键字个数，θ 表示至少含有一个查询中关键字的文档个数。MRSE 表示多关键字排序搜索方案，FSE 表示模糊搜索方案，DSE 表示动态更新方案。

Xia 等[15]提出的多关键字排序搜索方案可实现亚线性搜索，但其索引建立、搜索和陷门操作均涉及矩阵运算，处理时间随矩阵维数增加而增加。另外，用户还需要额外的空间来存储矩阵和索引。如果字典中关键字数量 m 很大，那么计算时间会大大增加。此外，为了动态更新索引，用户需要存储整个明文索引。如果用户没有存储索引，则需要从云中下载加密索引并解密，从而将花费更多的时间和吞吐量。而 BESE 使用反向索引进行搜索，可大大节省时间和空间消耗，实现最优查询效率。

Li 等[16]提出的模糊搜索方案需要为文档中的每个关键字创建一个模糊集，所以对于大型数据集，存储开销将显著增加。

BESE 方案不需要预先计算字典中每个关键字的模糊集，搜索时间的复杂度与字典中 m 的基数和关键字长度 l 有关。

Kamara 等[17]提出的动态更新方案需要维护关键字—文档数组和文档—关键字数组，所以建立的索引将会消耗相当大的空间，以实现较高搜索效率和动态更新功能。在更新操作期间，需要对多个相应的指针进行同态加密操作。

BESE 动态更新方案在添加或删除文档时不需要对主索引进行就地更新，而是建立一个小的辅助索引，从而大大减少了执行时间。但是，添加文档对应的临时辅助索引会消耗一定的内存。另外，主索引和辅助索引之间的合并过程需要消耗一定的时间。

表 1　　**性能对比**

方案	效率			存储		
	建立索引时间	搜索时间	更新时间	索引大小	客户端存储	更新成本
文献[15]方案	$O(nm^2)$	$O(\theta m\log n)$	$O(m^2\log n)$	$O(nm)$	$O(m^2)$	$O(m\log n)$
文献[16]方案	$O(nwl^e)$	$O(nwl^e)$	不支持	$O(nwl^e)$	$O(tl^e)$	不支持
文献[17]方案	$O(N)$	$O(q)$	$O(\|W_{id}\|)$	$O(m+n)$	$O(1)$	$O(\|W_{id}\|)$
BESE	$O(N)$	$O(q)_{\text{MRSE}}, O(ml)_{\text{FSE}}$	$O(\|W_{id}\|)$	$O(m+n)$	$O(1)$	$O(\|W_{id}\|)$

通过以上对比可以看出，BESE 方案在本地构建索引，并在访问代理中执行搜索操作，从而可支持多种搜索功能，且具有很高的性能。但是，BESE 方案不可避免地在访问代理中维护索引，从而为访问代理带来一定的空间开销。

3.6 安全分析

常见的密文搜索方案往往允许泄露某些信息以获得更好的性能。信息泄露情况使用泄露函数 L 进行量化。泄露函数以一系列查询 Q 作为输入，并把攻击者参与执行搜索过程可学到的信息内容作为输出。一个密文搜索方案泄露的信息量取决于索引是如何构建和保存的以及如何根据陷门执行查询过程。

依据文献[14,17]，对于某密文搜索方案，如果存在一个模拟器 S，将 $L(P)$作为输入，其中，P 表示搜索历史。输出的视图 $S(L(P))$和攻击者以 P 为输入参与执行真实的搜索过程看到的视图不可区分，则称此密文搜索方案是 L-安全的。给定函数 f，如果对于任意多项式 $p(\cdot)$和任意足够大的 s，$f(s)<\frac{1}{p(s)}$，则称 $f(s)$是可忽略的。如果对于任意概率多项式时间算法 F，分布 X 和 Y 满足式(1)，则称 X 和 Y 是计算不可区分的。

$$|\Pr[F(X)=1]-\Pr[F(Y)=1]|<\frac{1}{p(s)} \tag{1}$$

密文搜索的安全定义如下。给定一个密文搜索方案，P 表示一个有状态的概率多项式时间（PPT, probabilistic polynomial-time）攻击者，S 是一个有状态的 PPT 模拟器，L_1 和 L_2 是在 *Ideal* 安全游戏中有状态的泄露函数，s 为安全参数。定义 $Real_P(s)$ 和 $Ideal_{P,S}(s)$游戏如下。

$Real_P(s)$：挑战者根据安全参数 s 产生密钥 k。P 给定数据文件 D，挑战者利用数据 D 和密钥 k 产生加密索引 I 和加密数据 C，并将 I 和 C 发送给攻击者。然后攻击者进行多项式数量的自适应查询 Q，其中，对于每个查询 q 对应的关键字 w，攻击者都会从挑战者处接收到关键字对应的搜索陷门 TD，最后 P 返回一个比特 b 作为游戏的输出。

$Ideal_{P,S}(s)$：P 输出数据文件 D。给定 $L_1(D)$，S 产生并发送(I^*,C^*)到 P。然后攻击者进行多项式数量的自适应查询 Q，对于每个查询 q 对应的关键字 w，模拟器接收 $L_2(D, w)$并返回对应的陷门 TD^*。最后，P 返回一个比特 b 作为游戏的输出。

如果对于任意 PPT 攻击者 P，任意多项式 p 和足够大的 s，存在一个 PPT 模拟器 S，满足以下条件。

$$|\Pr[Real_P(s)=1]-\Pr[Ideal_{P,S}(s)=1]|<\frac{1}{p(s)} \tag{2}$$

则密文搜索方案对于自适应攻击 CKA2 是(L_1, L_2)安全的。用同样的方法可定义密文搜索方案对于非自适应攻击 CKA1 是(L_1, L_2)安全的，但 P 必须在游戏开始时就已经选好了所有的查询 Q，即用户的查询独立于搜索索引和以前的查询结果。

1) L_1 安全性：在 BESE 中，由于数据通过安全强度高的对称加密方式加密，故攻击者很难从密文中获取额外的信息。由于索引是在访问代理处产生和保存的，故攻击者无法获得索引，从而保证了索引的安全性，使攻击者无法获得更多的信息。

2) L_2 安全性：在搜索过程中，查询关键字交给访问代理，并在访问代理中执行查询，从而向云端隐藏了搜索模式。当执行查询获得搜索结果（文档标识符列表）后，用户向云服务器发送其他请求以检索特定文档。这些请求可以嵌入其他的检索中，从而混淆搜索请求和密文检索请求之间的相关性，使云服务器不能得出一次查询请求对应于哪些文件，从而在一定程度上向云端隐藏了访问模式。

参考文献[27-28]，如果攻击者拥有加密文件的先验知识，泄露搜索模式或访问模式可能会泄露更多关于查询或文档的信息。而在 BESE 方案中，查询过程在一定程度上隐藏了搜索模式和访问模式，所以即使攻击者拥有加密文件对应的所有明文，也不能推理得到查询对应的关键字。

基于上述分析和比较可以看出，对于云应用程序，BESE 方案比 CESE 方案更有优势。然而，BESE 方案中仍然存在着诸多挑战，且这些挑战在大多数基于 CASB 的解决方案中没有被提及和解决。

4 不同代理间索引的分享

在 BESE 方案中，与用户相关的搜索索引位于其访问代理内，用户如何跨代理搜索是需要解决的第一个问题。考虑 2 种情况：1) 如果用户仅在另一个代理中暂时进行搜索，则查询请求可转发给其原始代理，由原始代理处理该请求并返回文件标识符，查询关键字和搜索结果使用代理之间共享的密钥进行加密，以确保安全性；2) 如果用户永久切换

到另一个代理，则必须设计一个有效的方案将用户相关的索引从原始代理迁移到现在所在的代理。以下部分将重点介绍此场景。

大多数以CASB为基础的方案均未涉及不同访问代理间索引迁移的问题，因此迫切需要一种将用户相关索引从一个代理传输到另一个代理的有效方式。在BESE方案中，属于不同代理的索引服务器没有连接。此外，索引组织与物理架构密切相关，因此，难以在不同索引服务器之间直接迁移索引。BESE 采用的方法是在索引服务器中保留索引的源数据，然后对其进行过滤，并在另一个索引服务器中重新建立索引。此方案引入控制节点来协调传输过程，这里，源数据是一种 JSON 格式的结构化数据，根据原始文件内容和用户属性生成，适用于建立索引。不同代理间索引的分享如图 6 所示。如果原来在代理A的用户迁移到代理B下工作，那么需要将用户的索引从代理 A 传送到代理 B，过程如下所示。

1) 用户使用用户 *ID* 向代理 B 执行身份验证。

2)~4) 代理B检查它是否包含与用户*ID*相关的索引；如果没有，则从控制节点请求用户的索引。

5) 控制节点查询哪个代理与用户 *ID* 相关联，并查询到代理 A。然后，控制节点使用用户 *ID* 向代理 A 请求索引。控制节点最后更新用户和代理之间的对应关系。

6) 代理 A 从控制节点接收与用户 *ID* 相关索引的请求。

7)~9) 代理 A 首先向索引服务器发送请求，以过滤与用户 *ID* 相关的索引源数据。

10)~11) 代理 A 使用代理之间共享的对称密钥加密源数据并将其转发给代理 B。之后，代理 A 删除用户的索引和相关信息。

12)~15) 代理 B 接收加密数据，提取用户的源数据，然后重新索引源数据。之后，用户可以在代理 B 下进行搜索。

由于代理之间使用共享的对称密钥通过对称加密方式加密索引，故可保证索引分享的安全性，使没有密钥的攻击者难以破解密文得到明文索引。不同代理间的密钥分享机制将在第 5 节介绍。当索引源数据的量很大时，此方案将导致一定的存储和传输开销。优化方法按时间过滤并传输索引源数据，如最近 2 个月的索引数据。

5　不同代理间密钥的分享

数据在访问代理处加密，即秘密密钥在访问代理中，所以只有访问代理可以解密加密的数据。如果在代理 A 中的用户 u_A 想要与另一个代理 B 下的用户 u_B 分享一个文件，当用户 u_B 从云端接收加密的文件后，无法解密文件，因为它是被代理 A 加密的。因此如何在不同代理之间分享用于解密数据的密钥是密文分享需要解决的问题。

秘密密钥必须在代理间安全地共享。为此，PKE 方案[21]可有效保证密钥的安全传输。但为了证明证书的合法性，PKE 方案为证书设置了有效期。当证书过期或撤销大量证书时，PKE 方案效率低下，且会导致很多查询证书状态的请求。IBE 方案[23]可有效解决证书验证问题，但其具有私钥托管问题，即私钥生成器（PKG）可以解密用户通过 IBE 加密的数据。BESE

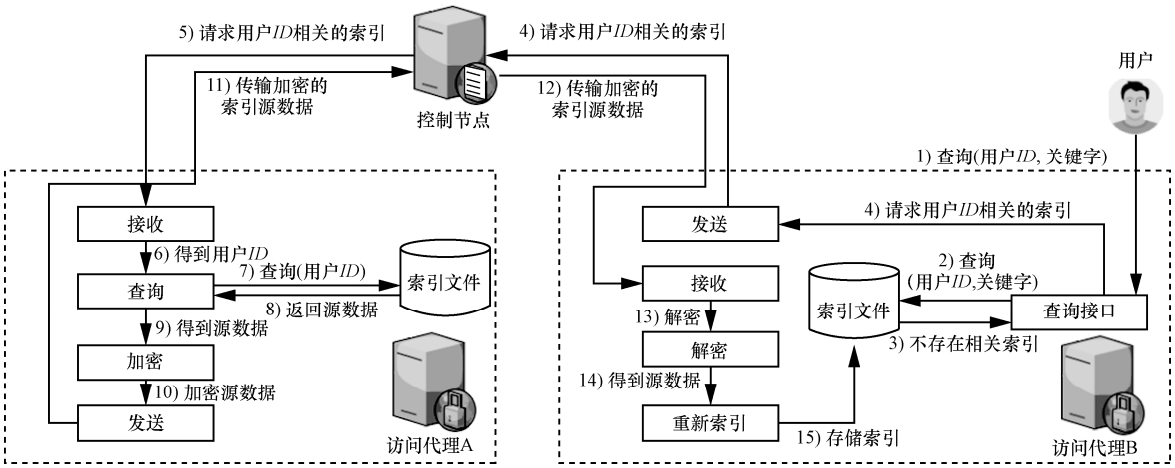

图 6　不同代理间索引的分享

结合 IBE 和 PKE 实现双层加密方案（TLES, two-layer encryption scheme），采用其各自的优势弥补各自的不足，保证密钥在代理之间安全传输。最后，本文设计了一个可实用的 TLES 原型系统并测试了其性能。

为了实现 TLES，BESE 方案使用控制节点充当 IBE 方案的 PKG。当访问代理被初始化后，它使用自己的 PKE 公钥—私钥对及其身份 *ID* 向控制节点进行身份验证。认证后，控制节点发出与身份 *ID* 相对应的 IBE 私钥。控制节点还负责初始化和更新访问代理基本信息，如访问代理 *ID*、访问代理公钥、密钥 *ID* 等。

令 PK_{I} 表示 IBE 公钥，SK_{I} 表示 IBE 私钥，PK_{P} 表示 PKE 公钥，SK_{P} 表示 PKE 私钥。如图 7 所示，不同访问代理之间的密钥分享过程如下。

1)~3) 代理 B 从云端接收代理 A 加密的数据。

4) 代理 B 将密文元数据上传到控制节点以请求解密密钥。

5)~7) 控制节点获取密文元数据，根据元数据中的代理 *ID* 查找到代理 A，然后从代理 A 请求对应的密钥。请求参数包括密钥 *ID*、代理 B 的时间参数 t_{B}、代理 B 的身份 ID_{B}、代理 B 的 PKE 公钥 $PK_{\mathrm{P\text{-}B}}$。

8)~10) 代理 A 根据参数 t_{B}、ID_{B} 得到 $PK_{\mathrm{I\text{-}B}}$，并结合 $PK_{\mathrm{P\text{-}B}}$ 双重加密相应密钥，将加密密钥转发给代理 B。

11) 代理 B 使用 IBE 私钥 $SK_{\mathrm{I\text{-}B}}$ 和 PKE 私钥 $SK_{\mathrm{P\text{-}B}}$ 来解密接收到的消息并获得相应的密钥。

12) 代理 B 用解密密钥解密密文以获得明文。

为了描述密钥产生分配的具体过程，给出以下数学化定义。$\{0,1\}^*$表示所有有限的字符串的集合。$\{0,1\}^s$ 表示所有 s 位字符串的集合。Z_q 表示模为 q 的加法群$\{0,1,\cdots,q-1\}$。$Z_q^*=Z_q\backslash\{O\}$表示 Z_q 中除去单位元 O 的集合。TLES 方案使用双线性对来构建。双线性对的定义如下。

假设 G_1 和 G_2 是 2 个阶为大素数 p 的循环群。G_1 是在有限域 F_q 上某一椭圆曲线上的点群，G_2 是有限域 F_c^*的子群，其中 $c=q^2$。如果对于任意 $P, Q \in G_1$ 和任意 $a, b \in Z$，映射 $e(aP, bQ) = e(P, Q)^{ab}$，则映射 $e{:}G_1\times G_1\rightarrow G_2$ 被认为是双线性的。

根据文献[21,23]中的安全性分析，TLES 方案各模块描述如下。

初始化：控制节点生成系统参数 $params=<q, n, P, P_{\mathrm{pub}}, G, H>$和主密钥 sk_{IBE}，*params* 是公开的，可由各代理获取。详细过程如下。

1) 在有限域为 F_q 上的椭圆曲线 $y^2=x^3+x$ 上构建双线性对，其中，素数 q=3mod 4。q 由 $q+1=th$ 生成，其中，t 为素数，h 为 12 的倍数。例如，对于整数 a 和 b，$0<b<a$，$t=2^a-2^b-1$。为了系统的安全性，设置 t 为 160 位，q 为 512 位。E 表示在域 Fq 上由 $y^2=x^3+x$ 定义的椭圆曲线，选择任意 P 使 $P\in\dfrac{E}{F_q}$。

2) 随机选择 sk_{IBE} 使 $sk_{\mathrm{IBE}}\in Z_q^*$，使 $P_{\mathrm{pub}}=sk_{\mathrm{IBE}}P$。

3) 选择散列函数 $H{:}F_c\rightarrow\{0,1\}^n$。选择散列函数 $G{:}\{0,1\}^*\rightarrow F_q$。其中，$H$ 和 G 被认为是随机的。信息空间为$\{0,1\}^n$，密文空间为 $\dfrac{E}{F_q}\times\{0,1\}^n$。

私钥签发：对于表征代理身份的字符串 *ID* 和时间参数 t，使用如下步骤生成私钥 d。

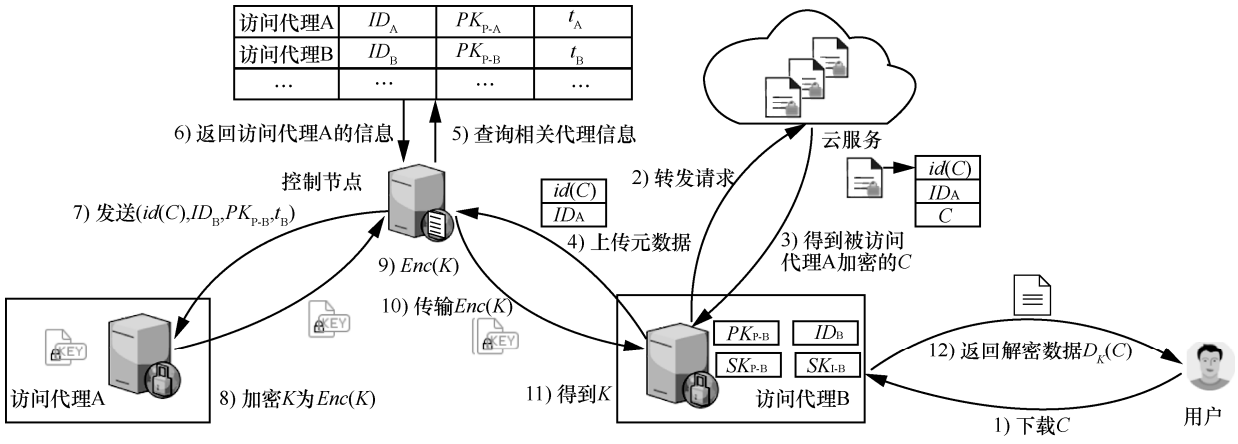

图 7 不同代理间密文数据的分享

1) 将 ID 和 t 连接并映射到点 $Q_{ID}\in\frac{E}{F_q}$。

2) 使 $d=sk_{\mathrm{IBE}}Q_{ID}$，其中 sk_{IBE} 为主密钥。控制节点使用代理的 PKE 公钥加密私钥 d，并将其转发给代理。代理解密得到私钥 d。

加密：使用 ID、t 和 PK_{B} 加密 $M\in\{0,1\}^n$。

1) 将 ID 和 t 连接并映射到点 Q_{ID}。

2) 从 Z_q 中随机选择 r。

3) 使 $C_1=<rP, M\oplus H(g^r_{ID})>$，其中，$g_{ID}=e(Q_{ID}, P_{\mathrm{pub}})\in F_q$。

4) 使用代理 PKE 公钥加密 rP 得到密文 U，最后生成密文 $C=<U, M\oplus H(g^r_{ID})>$。

解密。代理使用 PKE 私钥和 IBE 私钥解密密文获得明文 M。

1) 给定密文 $C=<U, M\oplus H(g^r_{ID})>$，代理首先使用 PKE 私钥解密 U 获得 rP。

2) 使 $C_1=<rP, M\oplus H(g^r_{ID})>$，使用私钥 d 解密 C_1 得到明文：$M\oplus H(g^r_{ID})\oplus H(e(d, rP))=M$。

在 TLES 方案密钥传输过程中，密钥是被双层加密传输的。即使控制节点具有代理的 IBE 私钥，也不能解密密文获得密钥，因为它不具有代理的 PKE 私钥。如果一个代理被攻击者替换了 PKE 公钥，攻击者具有了相应的 PKE 私钥，但是没有 IBE 对应的私钥，因此攻击者无法解密密文获得密钥。以上密钥传输过程中，系统参数选择、密钥生成和加密过程等均遵循安全的 IBE 加密方案[23]和 PKE 加密方案[21]，可保证密钥仅被发送方代理和接收方代理获得，而控制节点和其他代理等攻击者均无法破解双重加密的密文以获取密钥。如上所述，TLES 方案有效地确保了密钥的安全传输，只有与 ID 匹配的代理可以获得密钥。利用时间参数 t，可以有效地更新代理的 IBE 私钥以提高安全性。

6　实验分析

实验分析的主要内容和目的包括各种云应用程序中访问代理引入的额外开销、不同代理之间的索引迁移以及密钥分享的效率。

6.1　实验设置

搭建一个原型系统来测试所提出的 BESE 方案的有效性和性能。所用的虚拟机的配置包括英特尔 2.5 GHz 双核，内存为 4 GB，上行速度约为 1 000 KB/s，下行速度约为 7 500 KB/s。访问代理仲裁用户和云服务器之间的应用层连接，以保护用户的私有数据。

对于索引构建、搜索和分享方案，集成常用的开源搜索引擎如 Elastic Search 来构建索引服务器。当需要将索引从一个代理迁移到另一个代理时，源代理首先从索引服务器迁移索引源数据，并将加密的源数据传输到另一个代理，另一个代理使用 Nginx 反向代理来接收数据并对其进行解密得到源数据，然后将其转发到索引服务器以重新建立索引。TLES 借助 OpenSSL 库和 PBC 库实现。

6.2　案例研究

针对处理文本数据、文件数据或其他格式的数据，选择 3 类真实云应用程序来测试 BESE 方案，包括电子邮件（QQ、163 等）、网络硬盘（百度网盘等）和企业资源规划（ERP, enterprise resource planning）系统。实验结果表明，BESE 方案不影响电子邮件的正常功能。在网络硬盘服务中，用户无法对上传的加密文件（如图像和视频）使用预览功能。BESE 还支持 ERP 服务，缺点是服务器无法计算加密的数字数据，这将在未来的工作中得到解决。

6.3　性能测试

首先，选择一个典型的网络电子邮件服务——腾讯 QQ 邮件作为测试程序。为了测试访问代理引入的额外开销，主要通过对比打开代理服务和没有打开代理服务 2 种情况下发送和接收不同大小的电子邮件的时间开销。如图 8 所示，broker、解析、加密和元数据表示打开代理服务的数据传输和处理的时间，而 normal 表示没有打开代理服务的数据传输和处理的时间。

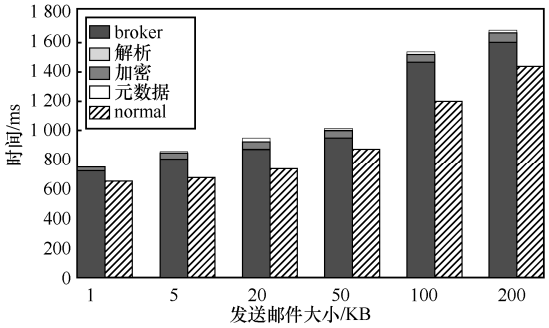

图 8　发送 QQ 邮件

从图 8 可以得出结论，代理在发送低于 200 KB 的电子邮件正文的情况下，平均增加 15%的额外开销。额外的开销包括解析、加密、元数据和数据传输及处理所耗费的时间。在额外的开销中，加密操作（包括密钥生成、加密以及形成包含加密数据的

新请求体）比解析或元数据操作花费更多的时间。另外，根据图 9 的测试结果，在代理中处理接收到的加密邮件仅仅带来平均 10%的额外开销，其中，解密操作比解析操作消耗更多的时间。

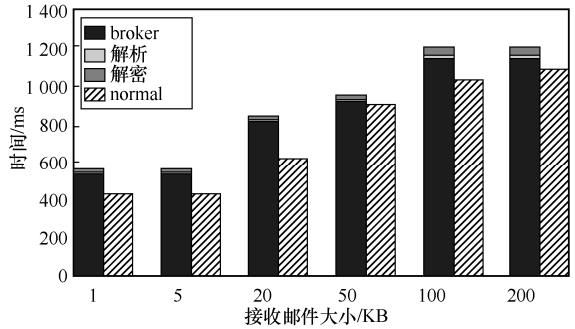

图 9　接收 QQ 邮件

为了测试文件加密，在百度网络磁盘中发送和接收 5 种类型的文件：256 B 文本、360 KB 图像、4.6 MB 和 10.3 MB 文档以及 32.1 MB 可执行程序。测试结果如图 10 所示。broker、缓存、加密和元数据表示打开代理服务的数据传输和处理的时间，而 normal 表示没有打开代理服务的数据传输和处理的时间。缓存操作存储原始数据以构建本地索引。从图 10 可以看出，代理在发送文件时仅带来 2.5%~4.5%的额外成本，因为代理可以分别加密和发送大文件的各数据块，而不是等到文件上传完所有内容后发送。图 11 显示接收加密文件时，代理带来少于 1%的额外开销。综上所述，代理加密或解密文件带来的额外开销较小。

使用基准框架 Rally 来评估 BESE 方案中索引的搜索效率，如表 2 所示。随着文件集合大小和数量的增加，索引存储空间增长和搜索性能下降都很缓慢。即使是大文件集合（如 2.2 GB），索引仅占用 1.7 GB 的空间，查询性能可达到每秒 11 次。

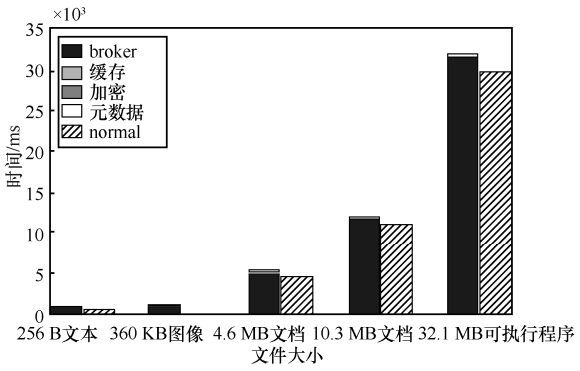

图 10　在百度网盘中上传文件

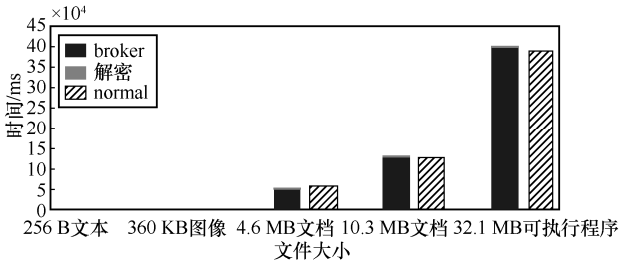

图 11　在百度网盘中下载文件

表 2　　搜索性能评估

文档数量/个	文档大小/GB	索引大小/GB	查询/s
9 697 882	1.212 332	0.947 194	19.567 5
10 716 760	1.339 540	1.050 98	16.188 5
11 961 342	1.487 816	1.179 34	14.677 4
13 053 463	1.624 238	1.255 26	13.830 9
17 647 279	2.201 861	1.706 34	10.957 6

如果用户在另一个代理中注册，则相应地，需要将原始代理中用户的相关索引传送到另一个代理中。如图 12 所示，对于不同大小的索引，测试了 2 个代理之间索引迁移的性能。实线表示将索引数据从一个代理迁移到另一个代理的总时间，包括导出所有索引源数据的时间、索引数据加密和传输时间以及数据解密和重新索引时间。虚线仅表示导出所有索引源数据的时间。实验结果表明，索引迁移所消耗的总时间与索引大小成正比。随着索引大小的增加，导出索引源数据的时间没有太大变化，源数据重新索引占用更多的时间。

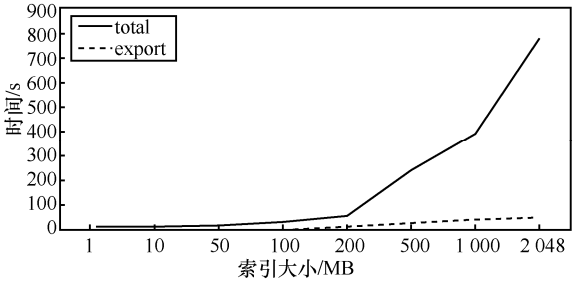

图 12　不同代理索引迁移开销

在密钥分享场景下，代理和控制节点之间的通信主要由套接字传输完成。使用 OpenSSL 库生成 4 096 位的 PKE 密钥对，调用 PBC 库生成 IBE 方案的双线性对。TLES 各个过程（包括系统参数生成、私钥签发、数据加密和数据解密）的性能测试如表 3 所示。从测试结果看出，解密时间要比加密时间长。总的来说，TLES 方案引入的开销是毫秒级

的，对于用户体验来说是微不足道的。

表 3　　TLES 各过程性能测试

过程	时间/ms
系统参数生成	50
私钥签发	20
加密	12
解密	38

7　结束语

本文首先提出了访问代理执行的密文搜索方案 BESE，并分析其效率性和安全性。针对实际云应用程序的实验进一步表明，BESE 方案确实引入很低的开销。为了实现密文搜索，BESE 方案使用加密文件的标识符在访问代理中建立搜索索引。为了在代理之间进行搜索，它实现了代理之间的索引分享方案。但是，当用户从一个代理迁移到另一个代理时，用户的原始代理仍然可以保留用户的密钥和索引。如果原始代理获得与密钥相关联的加密数据，则仍然可以对其进行解密。这个问题将在未来的研究中利用数据重加密技术解决。为了在不同代理之间共享加密数据，提出了 2 层加密方案 TLES 来传输密钥。当然，BESE 仍然面临诸多技术挑战，需要进一步的研究和改进。例如，由于应用程序种类繁多，BESE 需要自动匹配更多协议并构建数据索引以供搜索。另外，为了提升大数据量索引迁移的性能，需要进一步优化索引分享方案。

参考文献：

[1] Cloud Security Alliance, Top Threats Working Group. CSA's cloud computing top threats in 2016[R]. 2016.

[2] It's all about identity theft - first half findings from the 2016 breach level index[R]. 2016.

[3] 王国峰, 刘川意, 潘鹤中, 等. 云计算模式内部威胁综述[J]. 计算机学报,
WANG G F, LIU C Y, PAN H Z, et al. Survey on insider threats to cloud computing[J]. Chinese Journal of Computers, 2017, 40(2): 296-316.

[4] GOLDREICH O, OSTROVSKY R. Software protection and simulation on oblivious RAMs[J]. Journal of the ACM, 1996, 43(3): 431-473.

[5] BÖSCH C, HARTEL P, JONKER W, et al. A survey of provably secure searchable encryption[J]. ACM Computing Surveys (CSUR), 2015, 47(2): 18.

[6] Gartner report: how to evaluate and operate a cloud access security broker[R]. 2015.

[7] SCHÜTZE H. Introduction to information retrieval[C]//International Communication of Association for Computing Machinery Conference. 2008.

[8] HE W, AKHAWE D, JAIN S, et al. Shadowcrypt: encrypted Web applications for everyone[C]//The 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014: 1028-1039.

[9] LAU B, CHUNG S, SONG C, et al. Mimesis aegis: a mimicry privacy shield–a system's approach to data privacy on public cloud[C]//23rd USENIX Security Symposium (USENIX Security 14). 2014: 33-48.

[10] POPA R A, STARK E, VALDEZ S, et al. Building Web applications on top of encrypted data using Mylar[C]//11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14). 2014: 157-172.

[11] POPA R A, REDFIELD C, ZELDOVICH N, et al. CryptDB: protecting confidentiality with encrypted query processing[C]//The Twenty-Third ACM Symposium on Operating Systems Principles. 2011: 85-100.

[12] SONG D X, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[C]// 2000 IEEE Symposium on Security and Privacy. 2000: 44-55.

[13] GOH E J. Secure indexes[J]. International Association for Cryptologic Research Cryptology ePrint Archive, 2003: 216.

[14] 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.

[15] XIA Z, WANG X, SUN X, et al. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data[J]. IEEE Transactions on Parallel and Distributed Systems. 2016, 27(2): 340-352.

[16] LI J, WANG Q, WANG C, et al. Fuzzy keyword search over encrypted data in cloud computing[C]//INFOCOM. 2010: 1-5.

[17] KAMARA S, PAPAMANTHOU C, ROEDER T. Dynamic searchable symmetric encryption[C]// The 2012 ACM Conference on Computer and Communications Security. 2012: 965-976.

[18] 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. 2004:506-522.

[19] LIU Q, WANG G, WU J. Secure and privacy preserving keyword searching for cloud storage services[J]. Journal of Network and Computer Applications, 2012, 35(3):927-933.

[20] GENTRY C. Certificate-based encryption and the certificate revocation problem[C]//International Conference on the Theory and Applications of Cryptographic Techniques. 2003: 272-293.

[21] MICALI S. Scalable certificate validation and simplified pki management[C]//1st Annual PKI Research Workshop. 2002:15.

[22] SHAMIR A. Identity-based cryptosystems and signature schemes[C]// Workshop on the Theory and Application of Cryptographic Techniques. 1984: 47-53.

[23] BONEH D, FRANKLIN M. Identity-based encryption from the Weil pairing[C]//Annual International Cryptology Conference. 2001: 213-229.

[24] AL-RIYAMI S S, PATERSON K G. Certificateless public key cryptography[C]//International Conference on the Theory and Application of Cryptology and Information Security. 2003: 452-473.

[25] LEWKO A, WATERS B. Decentralizing attribute-based encryption[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2011: 568-588.

[26] SCHULZ K U, MIHOV S. Fast string correction with Levenshtein automata[J]. International Journal on Document Analysis and Recog-

nition, 2002, 5(1): 67-85.
[27] ISLAM M S, KUZU M, KANTARCIOGLU M. Access pattern disclosure on searchable encryption: ramification, attack and mitigation[C]//NDSS. 2012: 12.
[28] CASH D, GRUBBS P, PERRY J, et al. Leakage-abuse attacks against searchable encryption[C]// The 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015: 668-679.

[作者简介]

王国峰（1988−），男，山东济宁人，北京邮电大学博士生，主要研究方向为数据安全、云计算与云安全。

刘川意（1982−），男，四川乐山人，哈尔滨工业大学（深圳）副教授，主要研究方向为云计算与云安全、大规模存储系统、数据保护与数据安全。

韩培义（1992−），男，山西吕梁人，北京邮电大学博士生，主要研究方向为数据安全、云安全。

潘鹤中（1991−），男，辽宁本溪人，北京邮电大学博士生，主要研究方向为数据安全、云安全。

方滨兴（1960−），男，江西上饶人，中国工程院院士，哈尔滨工业大学（深圳）教授，主要研究方向为网络与信息安全、内容安全。