计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201000026

多云环境中基于属性加密的高效多关键词检索方案

何 亨 蒋俊君 冯 可 李 鹏 徐芳芳
武汉科技大学计算机科学与技术学院 武汉 430065
武汉科技大学智能信息处理与实时工业系统湖北省重点实验室 武汉 430065

摘 要 随着云计算技术的快速发展和广泛应用,云环境中的数据安全问题成为用户关注的焦点。为了保障数据隐私,用户将隐私数据加密后上传至云服务器。然而,如何从多个云服务器中的海量加密数据里检索到包含特定信息的密文是富有挑战性的。传统的可搜索加密方案无法直接应用于多云环境的密文数据检索中。基于属性的加密技术为密文关键词检索提供了一种新的解决思路,但是,现有的相关方案存在仅支持单个或连接关键词检索、访问控制策略不灵活、检索效率低、计算和存储开销大以及无法有效适用于多云环境等问题。因此,文中提出了一种多云环境中基于属性加密的高效多关键词检索方案(MRAM)。MRAM基于高性能的密文策略的属性加密算法,实现了任意密文多关键词检索,细粒度的访问控制,并且通过引入检索服务器有效支持多云环境中高效准确的密文检索。安全分析表明,MRAM能够实现安全索引机密性、检索陷门机密性、抗共谋攻击等重要安全特性,性能评估验证了MRAM相较于已有的方案,在安全索引生成、检索陷门生成和检索阶段具有更低的计算开销,且安全索引和检索陷门的存储开销也更小。

关键词:多云环境;属性加密;多关键词检索;密文检索;访问控制
中图法分类号 TP309

Efficient Multi-keyword Retrieval Scheme Based on Attribute Encryption in Multi-cloud Environment

HE Heng,JIANG Jun-jun,FENG Ke,LI Peng and XU Fang-fang
School of Computer Science and Technology,Wuhan University of Science and Technology,Wuhan 430065,China
Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System,Wuhan 430065,China

Abstract With the rapid development and wide application of cloud computing technology,data security issues in the cloud environment have become the focus of users' attention. To ensure data privacy,users encrypt the private data and upload it to the cloud server. Nevertheless,it is challenging to retrieve ciphertext containing specific information from massive encrypted data of multiple cloud servers. Traditional searchable encryption schemes cannot be directly applied to ciphertext data retrieval in the multi-cloud environment. The attribute-based encryption provides a new solution for ciphertext keyword retrieval. However,the existing related schemes have some problems,such as only supporting single or conjunctive keyword retrieval,inflexible access control policy,low retrieval efficiency,large calculation and storage overhead,and not applying to the multi-cloud environment effectively. Therefore,this paper proposed an efficient Multi-keyword Retrieval scheme based on Attribute encryption in the Multi-cloud environment (MRAM). MRAM is based on the high-performance ciphertext-policy attribute-based encryption algorithm,and realizes multi-keyword ciphertext retrieval and fine-grained access control. By introducing a retrieval server,MRAM effectively supports efficient and accurate ciphertext retrieval in multi-cloud environment. Security analysis shows that MRAM can achieve important security features such as security index confidentiality,trapdoor confidentiality,and resistance to collusion attacks. The performance evaluation verifies that MRAM has lower computational overhead in the secure index generation,trapdoor generation,and retrieval stages compared with existing solutions,and the storage overhead of the secure index and trapdoor is also smaller.

Keywords Multi-cloud environment,Attribute-based encryption,Multi-keyword retrieval,Ciphertext retrieval,Access control policy

1 引言

随着云计算技术的飞速发展[1],因其可按需部署、可拓展性强、性价比高等优点,越来越多的企业用户开始关注到云计算,并逐步将他们的数据库和应用部署到云上,以享受云计算带来的各种高性能、便捷和廉价的服务。各大互联网服务供

基金项目:国家自然科学基金项目(61602351,61802286);湖北省自然科学基金(2018CFB424);湖北省教育厅科学研究计划(B2019009)
This work was supported by the National Natural Science Foundation of China(61602351,61802286),Natural Science Foundation of Hubei Province,China(2018CFB424) and Scientific Research Project of Education Department of Hubei Province,China(B2019009).
通信作者:何亨(heheng@wust.edu.cn)

应商纷纷推出自己的云计算服务平台,例如 GCP、AWS、阿里云和腾讯云等。

云计算环境中的数据安全问题一直是用户关注的焦点。一方面,云服务供应商通常是半可信的,即“诚实但好奇”的[2]。云服务供应商会诚实地执行用户命令和系统任务,但是也会出于某种商业目的尝试获取用户的数据。另一方面,云服务供应商的云服务器处于公网环境中,面对复杂的网络形势,云服务供应商需要提供一套完整的安全方案,以保护自身存储的海量用户数据。然而近年来,存储在云服务器上的数据被泄露的新闻屡见不鲜[3],一旦云服务器受到攻击,导致数据泄露,则会对用户安全造成巨大的威胁。因此,为了保障用户自身的数据安全和隐私,用户通常会把自己的隐私数据加密后,再将密文数据上传给云服务供应商,这种做法成为了目前用户保护自己隐私数据的普遍做法。

然而,在云环境中,用户规模和数据规模都是巨大的。如何在云环境中从海量的加密数据里,安全、高效、准确地检索到包含特定信息的数据是富有挑战性的。传统的可搜索加密方案主要分为基于对称加密[4]和基于公钥加密[5]的可搜索加密方案。基于对称加密的可搜索加密方案[6]通常具有计算开销小的优点,但是在用户和数据的数量都很大的场景中,存在密钥管理复杂和难以实现细粒度的访问控制策略等缺点。基于公钥加密的可搜索加密方案[7]中密钥管理方便,可以有效应用于一对一和多对一用户的加密文件共享和检索的场景中,此时一个或多个数据拥有者对文件利用待共享用户的公钥进行加密后提供给用户检索即可。但是,对于一对多用户的加密文件的共享和检索,数据拥有者需要对同一文件分别使用多个用户的公钥进行加密,再将多个密文提供给不同用户来检索。当待共享的用户数量很大时,数据拥有者的计算和存储资源的消耗都非常大。与此同时,由于各个云服务供应商提供的资费、带宽、容量各自不同,同一用户通常会根据自己的实际情况选择一种或多种云服务供应商存储数据,不同用户使用的云服务供应商也可能完全不同。因此,用户需要使用不同的密钥分别在多个云服务供应商的云存储平台上进行密文数据的多次检索,并自己对检索结果进行整理归纳,导致整体的检索效率较低。由此可见,传统的可搜索加密方案存在密钥管理复杂、难以实现细粒度的访问控制策略、检索效率低、计算和存储开销大及应用场景有限等问题。因此,这些方案都无法直接应用于实际的多云环境的密文数据检索中。

属性加密[8](Attribute-Based Encryption,ABE)的概念最早由 Sahai 等于 2005 年提出,是对基于身份的加密[9]的拓展。密文策略的属性加密[10](Ciphertext-Policy ABE,CP-ABE)是 ABE 技术的进一步发展,最早由 Bethencourt 等提出。他们将访问结构引入到密文中,实现对每个加密数据部署不同的访问控制策略。在 CP-ABE 中,可以将每一个用户用一组属性而不是唯一的标识信息进行标识,并在每一个密文文件中嵌入基于属性的访问控制策略。当且仅当用户密钥中的属性集合满足密文文件中的访问控制策略时,用户才能将密文正确地解密成有效的明文。与传统的公钥加密方案相比,CP-ABE 打破了一对一的加密通信模式的限制,它可以看作是向特定组的用户进行的加密广播,使得数据拥有者在面向多个用户进行加密文件共享时,不需要对相同的文件进行多次加密,只需要执行一次 CP-ABE 操作即可,从而可以实现高效的细粒度数据访问控制和方便的密钥管理。与 CP-ABE 相反,密钥策略的属性加密[11](Key-PolicyABE,KP-ABE)是将访问控制策略嵌入在用户密钥中,只有当密文中的属性满足密钥中的访问控制策略,用户才能正确地将密文解密。从二者的实现原理来看,CP-ABE 更加适合于解决云环境中用户隐私数据的安全共享和检索的问题,因为 CP-ABE 能够由数据拥有者决定谁能够解密密文,而 KP-ABE 需要依赖于密钥分发的可信第三方来做决策。

ABE 的出现为实现支持细粒度访问授权的密文检索方案提供了新的思路。因此,国内外研究人员对基于 ABE 的密文数据的安全共享和检索方案展开了深入的研究[12-17]。文献[12-14]使用 KP-ABE 设计可搜索加密文献,实现单关键词的密文检索,其访问控制策略依赖于密钥授权中心;文献[15]基于 CP-ABE 实现密文连接关键词检索,但检索时有较大的计算开销;文献[16]基于 CP-ABE 实现云环境中的密文连接关键词检索,但存在访问控制结构不灵活的问题;文献[17]基于 CP-ABE 实现密文多关键词检索,但使用的算法效率不高,导致了系统整体效率较低。此外,上述方案均不能有效应用于多云环境。

针对多云环境的特点,以及现有基于 ABE 的密文关键词检索方案中存在的问题,文中提出了一种多云环境中基于属性加密的高效多关键词检索方案(efficient Multi-keyword Retrieval scheme based on Attribute encryption in theMulti-cloud environment,MRAM)。MRAM 以高性能的 CP-ABE 算法[19]为基础,实现了安全、高效、准确的密文多关键词检索,并支持细粒度的访问控制,能够有效应用于多云环境中大量数据的安全共享与检索的场景。MRAM 具有如下特点:

(1)多云环境中高效准确的密文检索。文献[18]提出的 CP-ABE 算法是基于文献[19]设计的,计算效率高。MRAM 文献[18]的基础上,实现多关键词的密文检索,相比已有相关方案[14-17],有更高的检索效率。此外,MRAM 通过引入专门的检索服务器连接多个云服务供应商,实现多云环境中的密文检索,使用户无需关心要检索的内容可能在哪个云服务供应商,仅需要通过检索服务器发送一次检索请求,便能够快速准确地检索到多个云服务供应商上所需要的密文文件,同时还能够有效减少对其他云服务供应商的访问次数,从而降低不必要的访问开销。

(2)密文多关键词检索。通过在安全索引和检索陷门中引入密文多关键词集合,减少用户一次逻辑检索的实际检索次数,以提高整体的检索效率和准确度。例如,需要完成密文状态下的一次逻辑检索“猪猫狗”,即完整检索出以“猪猫狗”中任意一个或多个密文关键词建立索引的所有文件。密文单关键词检索方案可能会检索不全目标文件,即无法检索出不以密文单关键词建立索引的文件[12-14,16];密文连接关键词检索方案则需要进行 7 次检索,即要分别以“猪”“猫”“狗”“猪猫”“猪狗”“猫狗”“猪猫狗”的对应密文关键词进行检索;而密文多关键词检索方案仅需要进行一次密文检索即可检索出所有需要的文件。

(3)细粒度的访问控制。采用树型结构来实现数据的访问控制,树型结构能够灵活支持属性间的多种门限操作,从而使数据拥有者可以方便地定义细粒度的访问控制策略。当且仅当嵌入在用户密钥中的属性集能够满足密文的访问控制策略时,该用户才能检索相应的密文关键词索引。

2 相关工作

ABE技术被提出以来，国内外研究人员对基于ABE的可搜索加密技术展开了深入的研究，以实现密文数据的安全共享和检索方案。

Li等[12]利用KP-ABE对公钥可搜索加密方案[11]进行改进，提出了一种基于属性的可搜索加密方案，利用ABE适应群组的特点，解决了密文关键词索引只能由特定用户检索的问题，但是该方案仅支持密文单关键词的检索，且数据拥有者在上传密文关键词索引之前需要进行大量的计算，方案效率较低。Zheng等[13]提出了基于KP-ABE的可验证外包数据的密文关键词检索方案，将复杂的计算任务外包给云服务供应商，提高了方案的效率，但是该方案只能按密文单关键词进行检索。Liang等[14]提出了一种基于KP-ABE的可搜索加密方案，应用于安全云存储的数据共享中，该方案支持关键词更新，具有较高的检索效率，但是仍然只能按密文单关键词进行检索。此外，文献[12-14]都是基于KP-ABE的方案，它们的访问控制策略不能由数据拥有者制定，而需要依赖于授权中心。

Song等[15]通过构造多项式方程，实现基于CP-ABE的可搜索加密方案，支持关键词任意连接检索，但检索效率不高，一次逻辑检索的效率仍然低于多关键词的可搜索加密方案。Sun等[16]提出了一种基于CP-ABE的可搜索加密方案，该方案引入代理重加密技术，将系统更新和用户撤销时大量的计算任务代理到云服务供应商上，减少系统的计算开销，但其算法只能实现“and”门限的访问控制逻辑，使得系统的访问控制策略所能够描述的属性关系有限。Wu等[17]提出了一种基于CP-ABE的密文多关键词检索方案，通过支持多关键词检索，减少了一次逻辑检索的实际检索次数，但是该方案的算法不够高效，不能有效适用于多云环境。

3 预备知识

3.1 Shamir秘密共享方案

Shamir秘密共享方案基于多项式插值技术，其基本思想是将一个秘密 s 分成 q 份子秘密，分发给 q 个人，只有当任意 $t(1\leqslant t\leqslant q)$ 个及以上的人一起共享他们所拥有的子秘密时，才能正确的得到秘密 s，任意 $t-1$ 个或更少的人共享子秘密都无法得到关于秘密 s 的任何信息，其中 (t,q) 被称为门限。

Shamir (t,q) 秘密共享方案的具体流程如下：

(1)分发秘密

1)随机生成一个大素数 $p\in Z_p$，且 $p>\max(s,q)$。

2)随机选取正整数 $a_0,a_1,\cdots,a_{t-1}\in N^*$。

3)定义多项式(1)，令 $s=a_0$：

$$p(x)=a_0+a_1x^1+a_2x^2+\cdots+a_{t-1}x^{t-1} \tag{1}$$

4)为 q 个待共享的用户分别随机选取 q 个非零元素 $x_u\in Z_p$，计算 x_u 对应的子秘密 $s_u=p(x_u) \bmod p(1\leqslant u\leqslant q)$，然后将每一份子秘密 s_u 分发给对应的用户。

(2)重构秘密

任意 t 个或以上的用户将各自的子秘密 s_u 共享时，可以通过式(2)计算出对应的拉格朗日基本多项式 $l_u(x)$，再通过式(3)重构出多项式 $p(x)$。

$$l_u(x)=\prod_{v=0,v\neq u}^{t}(x-x_v)/(x_u-x_v) \tag{2}$$

$$p(x)=\sum_{u=0}^{t-1}s_ul_u(x) \tag{3}$$

最后令 $x=0$，即可恢复秘密 s，即 $p(0)=a_0=s$。

3.2 属性和访问结构树

每一个用户都有其或大众或独特的属性，即用户可以用一组属性来描述，例如对于一个计算机学院的学生来说，他拥有以下属性“年级：2018级”“专业：网络工程”“职业：学生”“性别：男”和“年龄：19”等，这些属性描述了该用户的身份信息。反过来讲，通过指定特定的用户属性，就能够描述一组特定的用户，例如用“年级：2018级”“专业：网络工程”“职业：学生”来指定2018级的全体网络工程专业的学生。在ABE中，存在一个全局属性集 $\Omega=\{a_1,a_2,\cdots,a_n\}$，它包含所有用户的所有合法属性，若用户的属性不在 Ω 中时，说明该属性为非授权属性，则不能用于对应密钥的生成。用户属性集合 A 表示用户的合法属性集合，显然应该满足关系：$A\subseteq\Omega$。

在CP-ABE中，数据拥有者可以指定加密数据的访问控制策略，允许哪一用户群组可以正确解密该数据。MRAM采用灵活、拓展性强的树型结构设计访问控制策略，实现对数据细粒度的访问控制。

令访问结构树 T 表示访问控制策略，r 表示 T 的根节点。T 的每一个叶子节点表示用户的属性信息，中间节点表示逻辑关系，如“and”“or”和“of”门限。令 n 表示非叶子节点 x 的子节点数量，k 为 x 的门限值，显然 $1\leqslant k\leqslant n$。当 $k=n$ 时，x 表示为门限“and”；当 $1<k<n$ 时，x 表示为门限“of”；当 $k=1$ 时，x 表示为门限“or”。同时，每一个叶子节点的 k 为1。

图1给出了CP-ABE中访问控制策略的示例。只有当用户密钥中的属性能够满足密文的访问结构树时，用户才能够使用自己的密钥正确的将密文解密，得到明文。图1中User1和User2能够成功解密密文，而User3无法解密。

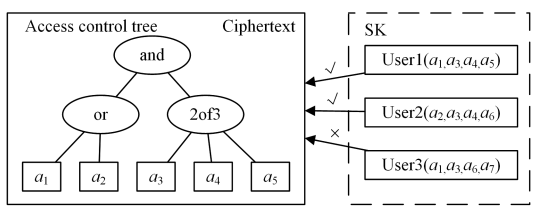

图1 CP-ABE的访问控制策略示例图

Fig. 1 Example of access control policy in CP-ABE

3.3 双线性映射

定义 G_0 和 G_T 是阶为素数 p 的两个乘法循环群，g 是 G_0 的一个生成元，$\hat{e}$ 表示双线性映射：$G_0\times G_0\rightarrow G_T$，并具有以下属性。

(1)双线性。对于 $\forall u,v\in G_0$，选取随机元素 $a,b\in Z_p$，满足条件：

$$\hat{e}(u^a,v^b)=\hat{e}(u,v)^{ab}$$

(2)非退化性。$\hat{e}(g^a,g^b)\neq 1$。

(3)可计算性。对于 $\forall u,v\in G_0$，存在一种高效的算法可以计算出 $\hat{e}(u,v)$。

4 方案设计

4.1 方案概述

MRAM的系统框架如图2所示。首先，数据拥有者根据共享文件的关键词集合和访问结构树生成安全索引，并上传至检索服务器；然后，检索服务器会复制并保留该安全索引中

的访问结构树部分,并将安全索引上传至数据拥有者指定的云服务供应商的云存储平台上。在用户检索文件时,根据自己的密钥和检索关键词集合生成检索陷门,并发送给检索服务器;然后,检索服务器将保留的访问结构树与检索陷门做匹配,根据匹配结果,将检索陷门转发至检索用户有权限检索的一个或多个云服务供应商的云存储平台;接着,由云存储平台根据检索陷门检索安全索引,将与检索陷门相匹配的安全索引所对应的密文文件存储信息返回给检索服务器,检索服务器对所有云服务供应商返回的密文文件信息进行整合排序,再将结果返回给检索用户;最后,检索用户根据得到的密文文件信息下载需要的密文文件。

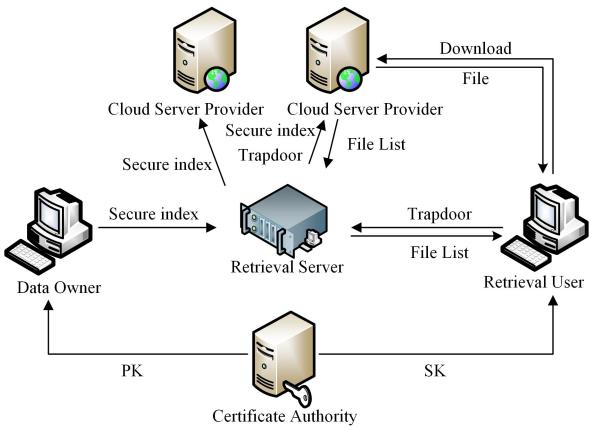

图 2　MRAM 系统框架

Fig. 2　System framework of MRAM

4.2　系统模型和安全假设

如图 2 所示,MRAM 主要存在 5 个实体。

(1)授权中心(Certificate Authority,CA):执行系统安全操作的可信第三方机构。CA 选取安全参数,产生全局属性集,生成系统公共密钥和主密钥,以及用户密钥。

(2)数据拥有者(Data Owner,DO):密文文件和安全索引的创建者。DO 对待共享的文件进行加密,生成密文文件,并根据文件的关键词集合和自定义的文件访问控制策略,生成对应关键词安全索引。

(3)检索用户(RetrievalUser,RU):向检索服务器发起密文检索请求的用户。RU 根据待检索的关键词集合和密钥,生成对应的检索陷门。

(4)检索服务器(RetrievalServer,RS):面向用户提供多云密文检索服务的专用服务器。RS 负责将安全索引上传给云服务供应商,验证用户的访问权限并向多个云服务器转发用户检索陷门。

(5)云服务供应商(Cloud ServerProvider,CSP):为用户提供廉价的高性能云计算和存储资源。CSP 负责向用户提供密文文件和安全索引的存储、检索和下载等服务。

同相关文献[15-17],MRAM 假设 CSP 是半可信的,即“诚实但好奇”的。CSP 能够正确执行用户的命令,完成用户的要求,包括存储密文文件、安全索引以及根据检索陷门进行密文检索操作等,然而,CSP 会出于某些商业目的,尝试获取密文文件、安全索引和检索陷门里存在的明文信息。RS 安全假设与 CSP 相同,并且不会与 CSP 进行串谋。与此同时,CA 完全可信,并且可以将用户密钥通过安全通道发送给用户。此外,存在任意数量未被授权的用户 RU 可能通过共谋以扩大他们的检索能力。

如文献[20]所述,MRAM 基于两种威胁模型:1)已知密文模型,即 CSP 只能获取到安全索引、检索陷门和检索结果;2)已知背景模型,在已知密文模型的基础上,CSP 可以进一步收集 RU 发送的不同的检索陷门,并通过分析这些检索陷门来推测文件的关键词。

4.3　核心算法

MRAM 包含 5 个核心算法,分别是系统初始化、安全索引生成、用户授权、检索陷门生成和检索。算法中定义了一个哈希函数 $H_0:G_0\rightarrow G_0$,G_0 和 G_T 是阶为素数 p 的两个乘法循环群,H_0 用于将关键词映射到 G_0 群中的一个元素,方案的核心算法如算法 1 所示。

4.3.1　系统初始化

算法 1 用于整个系统的初始化操作,将安全参数 k 和全局属性集合 Ω 作为输入,生成系统公共密钥 PK 和主密钥 MK。

算法 1　系统初始化

输入:安全参数 k 和全局属性集合 Ω

输出:公共密钥 PK,主密钥 MK

1. 生成阶为素数 p 的双线性群 G_0,g 是 G_0 的一个生成元,可以得出双线性映射关系:$\hat{e}:G_0\times G_0\rightarrow G_T$;
2. 生成随机元素 $\alpha\in Z_p$,对于全局属性集合 $\Omega=\{a_1,a_2,\cdots,a_n\}$ 中的每一个元素,生成与之对应的随机元素 $t_1,t_2,\cdots,t_n\in Z_p$;
3. 计算 $y=\hat{e}(g,g)^{\alpha}$,$T_j=g^{t_j}(1\leqslant j\leqslant n)$;
4. 返回公共密钥:$PK=(\hat{e},g,y,T_j(1\leqslant j\leqslant n))$,主密钥:$MK=(\alpha,t_j(1\leqslant j\leqslant n))$。

4.3.2　安全索引生成

算法 2 用于多关键词的安全索引生成,将关键词集合 $RW=\{rw_1,rw_2,\cdots,rw_x\}$ 中每一个关键词映射到 G_0 中的一个元素得 $W=\{w_1,w_2,\cdots,w_x\}$、访问结构树 T 和公共密钥 PK 作为输入,生成安全索引 C_T。

算法 2　安全索引生成

输入:文件关键词集合 W,访问结构树 T,公共密钥 PK

输出:安全索引 C_T

1. 生成随机元素 $s\in Z_p$,计算 $c_0=g^s$;
2. 计算 $y^s=\hat{e}(g,g)^{\alpha s}$;
3. 对于 $\forall w_i\in W$,计算 $k_i=H_0(w_i)$,$c_i=k_i\cdot y^s=H_0(w_i)\cdot\hat{e}(g,g)^{\alpha s}$;
4. 为访问结构树 T 的根节点 r 分配秘密值 s,将 r 设置为已标记,并将所有其他节点设置为未标记。从 r 开始按从上至下的方式,对每个未标记的非叶子节点递归地执行以下操作:
   4.1. 如果门限为“of”,并且它的子节点为未标记,则根据 Shamir(t,q)秘密共享方案共享该节点的秘密值给子节点,此时 $1<t<q$,其中 q 表示所有子节点的总数,t 表示恢复秘密值必须的子节点数,并设置所有子节点为已标记;
   4.2. 如果门限为“and”,并且它的子节点为未标记,则根据 Shamir(t,q)秘密共享方案共享该节点的秘密值给子节点,此时 t=q,并设置所有子节点为已标记;
   4.3　如果门限为“or”,并且它的子节点为未标记,则根据 Shamir(t,q)秘密共享方案共享该节点的秘密值给子节点,此时 t=1,并设置所有子节点为已标记。
5. 对于 T 中每一个叶节点 f,其属性为 a_j,计算 $c_{f,j}=T_j^{s_f}$,其中 s_f 为 f 保存的秘密值;
6. 返回安全索引:

$C_T=(T,c_0,\forall w_i\in W:c_i,\forall a_j\in T:c_{f,j})$。

4.3.3 用户授权

算法 3 用于对检索用户 RU 授权，将用户属性集合 A 和主密钥 MK 作为输入，生成对应的用户密钥 SK。

算法 3 用户授权

输入：用户属性集合 $A=\{a_1, a_2, \cdots, a_m\}$，主密钥 MK

输出：用户密钥 SK

1. 生成随机元素 $\tau, r \in Z_p$，计算 $d_0 = g^{\alpha-\tau r}$；
2. 对于 $\forall a_j \in A$，计算 $d_j = g^{r/t_j}$；
3. 返回用户密钥：$SK = (\tau, d_0, \forall a_j \in A: d_j)$。

4.3.4 检索陷门生成

算法 4 用于检索陷门生成，将检索关键词集合 $RW' = \{rw_1', rw_2', \cdots, rw_y'\}$ 中每一个关键词映射到 G_0 中的一个元素得 $W' = \{w_1', w_2', \cdots, w_y'\}$ 和用户密钥 SK 作为输入，生成该用户该次检索对应的检索陷门 $T_{W'}$。

算法 4 检索陷门生成

输入：检索关键词集合 W′，用户密钥 SK

输出：检索陷门 $T_{W'}$

1. 对于 $\forall w_j' \in W'$，计算 $D_j = H_0(w_j')$；
2. 对于每一个 d_j，计算 $d_j' = d_j^{\tau} = g^{\tau r/t_j}$；
3. 返回检索陷门：$T_{W'} = (d_0, \forall a_j \in A: d_j', \forall w_j' \in W': D_j)$。

4.3.5 检索

算法 5 用于根据检索陷门对安全索引进行检索，将检索陷门 $T_{W'}$ 和安全索引 C_T 作为输入，若检索到对应的文件，则生成与 $T_{W'}$ 对应的文件下载信息列表 FL；若检索不成功则产生 NULL。当安全索引数量很大时，可以通过对安全索引进行分区等方式，实现并行检索，以提高检索效率。

算法 5 检索

输入：检索陷门 $T_{W'}$，安全索引 C_T

输出：文件下载列表 FL

1. 选择满足访问结构树 T 的最小集合 $A' \subseteq A$，计算：

$$\prod_{a_j \in A'} \hat{e}(c_{f,j}, d_j')^{l_f(0)} = \prod_{a_j \in A'} \hat{e}(g^{t_j s_f}, g^{\tau r/t_j})^{l_f(0)} = \prod_{a_j \in A'} \hat{e}(g,g)^{\tau r s_f l_f(0)} = \hat{e}(g,g)^{\tau r s}$$

其中，$l_f(0)$ 是叶节点 f 对应的拉格朗日系数，并且能被满足 T 的用户根据 Shamir 秘密共享方案计算出来；

2. 计算：

$$\hat{e}(c_0, d_0) \cdot \hat{e}(g,g)^{\tau r s} = \hat{e}(g^s, g^{\alpha-\tau r}) \cdot \hat{e}(g,g)^{\tau r s} = \hat{e}(g^s, g^{\alpha})$$

3. 对于每一个 c_i，计算：

$$K_i = \frac{c_i}{\hat{e}(g^s, g^{\alpha})} = \frac{H_0(w_i) \cdot \hat{e}(g,g)^{\alpha s}}{\hat{e}(g^s, g^{\alpha})} = H_0(w_i)$$

4. 令 K 为 $K_i (1 < i < x)$ 的集合，D 为 $D_j (1 < j < y)$ 的集合，x 为 K 中元素的个数，y 为 D 中元素的个数，记 z 为 $K \cap D$ 的元素个数；
5. 令 $z=0$，将 K 中的每一个元素 K_i 映射到哈希表 H 中，再依次将 D 中的元素 D_j 映射到 H 中，当 D_j 与 K_i 发生哈希碰撞时，若两元素相同，则 $z=z+1$，若不相同，则继续计算 D 中的下一个元素的哈希值，重复上述操作，直到 D 中所有元素处理完毕；
6. 计算检索权值 $v = z^2/xy$，显然 $v \leqslant 1$，且文件的 C_T 与 $T_{W'}$ 匹配的程度越高，其 v 越接近于 1；
7. 若 $v \neq 0$，则根据 v 将该 C_T 对应文件的下载信息降序添加到文件下载列表 FL 中，返回 FL；否则返回 NULL。

4.5 系统流程

MRAM 的系统流程如下。

(1)授权中心 CA 选取安全参数，产生全局属性集，运行算法 1，生成系统的主密钥 MK 和公共密钥 PK，CA 负责安全的保存 MK，并将 PK 公开。

(2)数据拥有者 DO 提取待共享文件的关键词集合 W，并构造对应的访问结构树 T，运行算法 2，生成安全索引 C_T，再将 C_T 上传给检索服务器 RS。需要说明的是，关于文件自身的加密处理并不属于 MRAM 的主要研究内容，可以对文件基于文献[19]提出的 CP-ABE 算法进行加密，然后将密文文件上传至某一云服务供应商，再将密文文件存储信息写入 C_T。云环境中基于 CP-ABE 的文件加密和共享的有关方案在文献[21-22]等中进行了深入的研究。

(3)RS 在收到 DO 上传的 C_T 后，首先会复制并保存 C_T 中的 T，再将 C_T 上传至用户指定的云服务供应商 CSP 中，若用户未指定，则上传至默认 CSP。

(4)检索用户 RU 首次使用系统时，CA 会根据 RU 的属性集运行算法 3，生成 RU 对应的用户密钥 SK，用于 RU 检索密文文件。

(5)RU 在获得 SK 后，根据自己要检索的关键词集合 W' 运行算法 4，生成检索陷门 $T_{W'}$，向 RS 发起检索请求。这一步能够让用户在一次逻辑检索过程中，不用分别向多个 CSP 发起检索请求，只需要向 RS 发送一次请求即可，减少了用户一次逻辑检索中实际的检索次数。

(6)RS 在收到 RU 的检索请求后，会将 $T_{W'}$ 中的属性与保存的访问结构树 T' 进行匹配，判断出该 $T_{W'}$ 有权限检索的文件存储在哪些 CSP 上。若匹配成功，说明至少一个 CSP 中有该 RU 有权限检索的密文文件，则将该 $T_{W'}$ 转发至所有匹配的 CSP 进行下一步检索；若匹配失败，说明该 RU 的权限无法检索任何 CSP 中的任何一个密文文件，检索失败。这一步能够显著减少 CSP 的被访问和检索次数，提升系统整体的检索效率。

(7)CSP 在收到 RS 转发的 $T_{W'}$ 后，遍历所有的 C_T，依次将 $T_{W'}$ 中的属性与 C_T 中的 T 进行匹配，若 $T_{W'}$ 中的属性不满足 T，跳过该索引；若满足 T，则运行算法 5，进一步判断 $T_{W'}$ 中密文关键词集合与 C_T 中密文关键词集合的匹配程度。CSP 根据匹配成功的 C_T，将其对应的密文文件信息按照检索权值降序排列后添加到文件列表 FL_i 中，并返回给 RS。

(8)RS 在收齐所有检索的 CSP 返回的 FL_i 后，对所有 FL_i 进行整合，将其中密文文件信息根据检索权值进行降序排列后最终得到文件列表 FL，并发送给 RU。因为 FL 是检索多个 CSP 得到的综合结果，所以 FL 比单一 CSP 的检索结果更加全面准确，即更符合用户的检索期望。

(9)RU 收到 RS 转发的 FL 后，根据自己的需要从 CSP 上下载对应的密文文件。

5 安全分析与性能评估

5.1 安全分析

下面从安全索引机密性、检索陷门机密性、抗共谋攻击和细粒度的访问控制 4 个方面讨论 MRAM 的安全性。所有的安全性分析均在第 4.2 节中的系统模型和安全假设下进行。

5.1.1 安全索引机密性

MRAM 在 Luan 等[19]的 CP-ABE 算法的基础上实现了多云环境中多关键词的密文检索，而原方案只用于加解密原始数据，所以二者之间存在明显的区别。在检索过程中，首先

要确保存放在 CSP 上安全索引 C_T 的机密性,因为 C_T 包含了检索相关的一些关键信息。MRAM 在生成 C_T 时,使用伪随机函数 $H_0(\cdot)$对关键词集合进行处理,再根据 Shamir 秘密共享方案,将秘密值 s 分享到访问结构树 T 的每一个子节点,不满足 T 的用户不能计算出 s,从而无法正确解密出有效的明文数据。因此尽管 CSP 拿到了 C_T,也无法从密文状态下的 C_T 中获得有效的明文关键词信息。在文献[18-19]中,Yan 等的 CP-ABE 算法都被证明在 Decisional Bilinear Diffie-Hellman(DBDH)假设下是安全的,因此,MRAM 在 DBDH 假设下同样是安全的。

5.1.2　检索陷门机密性

在检索过程中,CSP 可能会对检索陷门 $T_{W'}$ 进行分析,尝试获取某些私密信息。但是,MRAM 对关键词集合做了预处理,将每一个真实的关键词 rw_i 映射到 G_0 中的一个元素 w_i,如图 3 所示。文献[23]中为了隐藏用户属性的描述域也采用了相似的操作,再考虑到 $H_0(\cdot)$是伪随机函数,因此 CSP 无法从 $T_{W'}$ 中猜测出有意义的关键词信息。而且,由于在算法 3 中,授权中心 CA 为每个用户选取了一个随机元素 r 和 τ,并将 r 和 τ 嵌入在 SK 的 d_0 和 d_j' 中,故 CSP 是无法从多个不同的 $T_{W'}$ 中分析出 SK 中用户属性等相关信息。因此,MRAM 中的检索陷门是安全的。

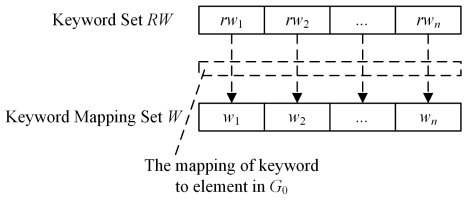

图 3　关键词到 G_0 群中的映射

Fig. 3　Mapping of keyword to element in G_0

5.1.3　抗共谋攻击

任意数量的非授权用户都可能为了扩大自己的检索能力而进行共谋攻击。MRAM 在检索阶段,根据算法 5,首先计算 $\hat{e}(g,g)^{\tau rs}$,再计算 $\hat{e}(c_0,d_0)$ 即 $\hat{e}(g^s,g^{\alpha-\tau r})$,当且仅当 $\hat{e}(g,g)^{\tau rs}$ 与 $\hat{e}(g^s,g^{\alpha-\tau r})$ 中的 τ 和 r 保持一致,才能计算出 $\hat{e}(g^s,g^{\alpha-\tau r})\cdot\hat{e}(g,g)^{\tau rs}=\hat{e}(g^s,g^{\alpha})$,进而能够检索出与陷门相匹配的密文文件信息。如果有多个用户通过共享、组合密钥的方式来伪造检索陷门,扩大检索范围,是无法检索到正确的密文文件信息的,因为共谋用户组合的密钥中的随机元素 τ 和 r 是不相同的。因此,MRAM 可以防御任意未授权用户的共谋攻击。

5.1.4　细粒度的访问控制

在 MRAM 中,数据拥有者可以对不同安全需求的隐私数据的安全索引采用各自不同的访问控制策略,即每个 C_T 都包含自身的访问结构树 T。T 通过实现“and”“or”和“of”门限,来表示复杂灵活的访问控制策略。只有用户属性满足 T,才有权限通过检索陷门 $T_{W'}$ 检索 C_T 以获得与 $T_{W'}$ 相匹配的密文文件信息。此外,在 MRAM 中,可以对 C_T 和 $T_{W'}$ 中的属性使用对称加密或哈希运算的方式进行预处理,以隐藏访问结构树和用户属性的含义。因此,MRAM 实现了细粒度的密文数据访问控制。

5.2　性能评估

对于 MRAM 的性能评估从计算开销和存储开销两个方面展开,并与 MKSE-UR[17],ABKS-UR[16],ABE-CKS[15] 和 SABM-DS[14] 进行比较。

下文所涉及到的主要计算操作为:P 表示双线性对运算,E_0 和 E_T 分别表示群 G_0 和 G_T 中的指数运算。另外,乘法运算和哈希运算的计算开销相对于上述操作可以忽略不计。同时,记 n 为全局属性集中属性数量,m 为用户属性集中属性数量,a 为用户属性满足访问结构树的属性数量,x 为文件关键词集合中关键词数量,y 为检索关键词集合中关键词数量,l_0 和 l_T 分别表示群 G_0 和 G_T 中元素的长度。上述变量通常具有以下关系:E_T 的运算开销远小于 E_0 和 P 的运算开销,且 $n\gg m\geqslant a$。

5.2.1　计算开销

表 1 对 MRAM,MKSE-UR,ABKS-UR,ABE-CKS 和 SABM-DS 中 3 个主要阶段的理论计算开销进行比较,包括安全索引生成、陷门生成和检索阶段。从表 1 可以看出,在关键词安全索引生成阶段,MRAM 和 MKSE-UR 的计算开销主要都是关于 a 和 x 的线性函数,但 MKSE-UR 的自变量 a 的系数是 MRAM 的 2 倍,ABKS-UR 和 SABM-DS 的计算开销是关于 n 的线性函数,而 ABE-CKS 的计算开销是关于 m 的线性函数,且系数为 2,考虑到系数大小和 $n\gg m\geqslant a$,即 m 和 a 远小于 n,所以 MRAM 在安全索引生成阶段的计算开销理论上略低于 MKSE-UR,且远低于其他方案。在陷门生成阶段,MRAM 和 MKSE-UR 的计算开销均是关于 m 和 y 的二元一次函数,但是 MKSE-UR 的关于 m 的函数系数是 MRAM 的 2 倍,ABE-CKS 的计算开销是关于 x 和 y 的二元一次函数,ABKS-UR 的计算开销是关于 n 的一次函数,而 SABM-DS 的计算开销是关于 m 的二次函数,考虑到 y 是检索关键词集中关键词数量,通常比较小,且小于 n,m 和 x,所以 MRAM 在陷门生成阶段的计算开销理论上优于其他 4 个方案。在检索阶段,从主要影响计算开销的 E_0 和 P 相关的自变量来看,MRAM 和 ABE-CKS 的计算开销都是关于 a 的线性函数,但 ABE-CKS 的函数系数是 MRAM 的 2 倍,ABKS-UR 和 SABM-DS 是关于 n 的线性函数,而 MKSE-UR 的计算开销是关于 m 的线性函数,又存在 $n\gg m\geqslant a$ 的关系,所以 MRAM 在检索阶段的计算开销理论上优于其他方案。此外,MRAM 和 MKSE-UR 支持多关键词检索,ABKS-UR 和 ABE-CKS 支持连接关键词检索,而 SABM-DS 仅支持单关键词检索。

表 1　理论计算开销比较

Table 1　Comparison of theoretical computational consumptions

Scheme	Secure index generation	Trapdoor generation	Retrieval	Multi-keyword
MRAM	$(a+1)E_0+E_T+xP$	$(m+y+1)E_0$	$aP+aE_T$	√
MKSE-UR	$(2a+x+6)E_0$	$(2m+y+3)E_0$	$2mE_0+5P+E_T$	√
ABKS-UR	$(n+1)E_0+E_T$	$(2n+1)E_0$	$(n+1)E_0+E_T$	×
ABE-CKS	$(2m+x+3)E_0$	$(2x+y+2)E_0$	$(2a+2x+2)P$	×
SABM-DS	$(n+1)E_0+E_T$	m^2E_0	nE_0+P	×

为了更直观地了解 MRAM 的性能，下面通过两组仿真实验测试 MRAM 分别随用户属性数量和关键词数量变化时的计算开销，并与 MKSE-UR 和 ABKS-UR 进行比较。三者均基于 JAVA 语言，使用 IntelliJIDEA 2018 工具和 jPBC2.0 开源加密库实现。实验中，硬件配置为 Intel Core i5 8500 @ 3.0GHz，16G RAM，操作系统为 Windows 10 专业版，使用阶为 160 bit 的 A 类型超奇异椭圆曲线 $y^2=x^3+x$ 来初始化加密环境，结合实际云环境中密文检索应用的情况和为了便于进行比较，具体参数设置为：第一组中 $n=60$，$m\in[10,60]$，$a=m$，$x=2y$，$y=5$；第二组中 $n=60$，$m=10$，$a=m$，$x=2y$，$y\in[5,30]$。当参数取不同的合理值时，也能够得到类似的结果，此外所有的实验结果均为 10 次实验的平均值。

图 4—图 6 分别给出了第一组实验中，随着用户属性数量的增加，MRAM，MKSE-UR 和 ABKS-UR 在安全索引生成、陷门生成和检索阶段的时间消耗情况。仿真实验结果表明，在安全索引生成阶段，ABKS-UR 的时间消耗则维持在 530 ms 左右，因为其时间复杂度不依赖于用户属性数量，但当用户属性较少时，其时间消耗仍然保持在较高水平，而 MRAM 和 MKSE-UR 的时间消耗呈线性增长趋势，且 MKSE-UR 的斜率将近 MRAM 的 2 倍，当 $m\geqslant 30$ 时，超过 ABKS-UR 的时间消耗。在陷门生成阶段，ABKS-UR 的时间消耗较大，并保持稳定，而 MRAM 和 MKSE-UR 的时间消耗呈线性增长趋势，但 MRAM 的增长速度比 MKSE-UR 的增长速度慢，当 $m=n=60$ 时，MKSE-UR 略微超过 ABKS-UR 的时间消耗，MRAM 则在 ABKS-UR 的 50% 左右。在检索阶段，ABKS-UR 的时间消耗始终维持在 450 ms 左右，MRAM 和 MKSE-UR 的时间消耗随 m 的增大呈线性增长趋势，当 $m\geqslant 30$ 时，MKSE-UR 的时间消耗超过 ABKS-UR，当 $m=60$ 时，MRAM 的时间消耗略微超过 ABKS-UR，考虑到 MRAM 和 MKSE-UR 支持多关键词检索，且 m 和 a 始终小于或等于 n，因此在实际的检索过程中，MRAM 和 MKSE-UR 的效率是优于 ABKS-UR 的。

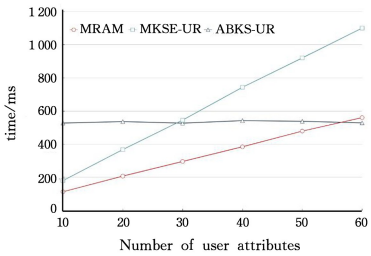

图 4 安全索引生成阶段用户属性与时间消耗的关系

Fig. 4 Relationship between user attributes and time consumption during security index generation phase

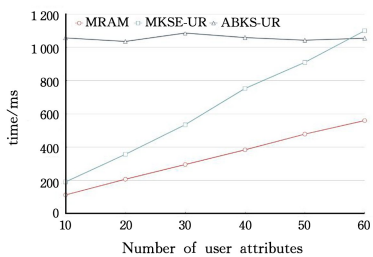

图 5 陷门生成阶段用户属性与时间消耗的关系

Fig. 5 Relationship between user attributes and time consumption during trapdoor generationphase

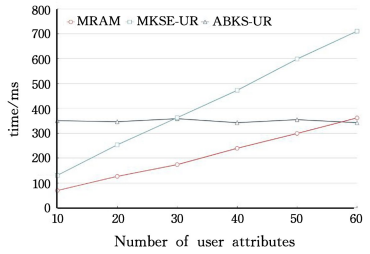

图 6 检索单个安全索引时用户属性与时间消耗的关系

Fig. 6 Relationship between user attributes and time consumption during retrieving a single security index

综上所述，随着用户属性数量 $m(a)$ 的增加，MRAM 在陷门生成和检索阶段的时间消耗均优于 MKSE-UR 和 ABKS-UR，在安全索引生成阶段，只有当 m 接近 n 时，其时间消耗超过 ABKS-UR。

图 7—图 9 分别给出了分别展示了第二组实验中，随着关键词数量的增加，MRAM，MKSE-UR 和 ABKS-UR 在安全索引生成、陷门生成和检索阶段的时间消耗情况。

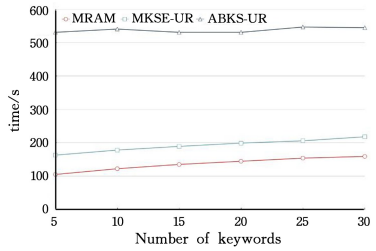

图 7 安全索引生成阶段关键词与时间消耗的关系

Fig. 7 Relationship between keywords and time consumption during security index generation phase

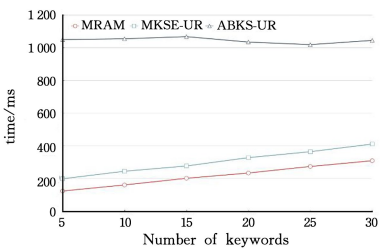

图 8 陷门生成阶段关键词与时间消耗的关系

Fig. 8 Relationship between keywords and time consumption during trapdoor generation phase

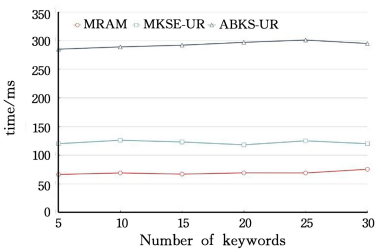

图 9 检索单个安全索引时关键词与时间消耗的关系

Fig. 9 Relationship between keywords and time consumption during retrieving a single security index

实验结果表明，在安全索引生成阶段，ABKS-UR 的时间消耗保持在 530 ms 的水平，MRAM 和 MKSE-UR 时间消耗的增长速率接近，但 MRAM 比 MKSE-UR 低 60 ms 左右。陷门生成阶段的结果与安全索引生成阶段的结果类似，ABKS-UR 的时间消耗保持在较高水平，超过 1000 ms，而 MRAM 和

MKSE-UR 的时间消耗的增长速率较低，MKSE-UR 比 MRAM 高出 80 ms 左右。在检索阶段，三者时间消耗的增长趋势均呈现水平状态，其中，MRAM 的时间消耗在 65 ms 左右。MKSE-UR 接近 MRAM 的 2 倍，ABKS-UR 接近 MRAM 的 4 倍。因此，随着关键词数量 $y(x)$ 的增加，MRAM 在安全索引生成、陷门生成和检索阶段的时间消耗是优于 MKSE-UR 和 ABKS-UR 的。

检索服务器 RS 在 MRAM 中的主要工作是将检索陷门中的属性与 RS 保存的所有访问结构树 T' 进行比较，判断该检索陷门符合哪些 T' 的检索权限，再将符合检索权限的检索陷门转发给 T' 对应的一个或多个云服务供应商。图 10 展示了随着 T' 数量增长，RS 判断检索陷门权限的时间消耗情况。可知，RS 完成单次权限匹配的时间消耗是微秒级的，当 T' 的数量上千时，其时间开销也不超过 10ms，因此 RS 的计算消耗是非常低的。但是，考虑到当用户和数据规模非常大时，检索服务器承担的计算开销也会增加，因此可以进一步对检索服务器进行集群化设计，使多个检索服务器能够协同并行工作，从而更有效地支持大数据环境中的密文检索操作。这部分工作将是 MRAM 未来的研究方向。

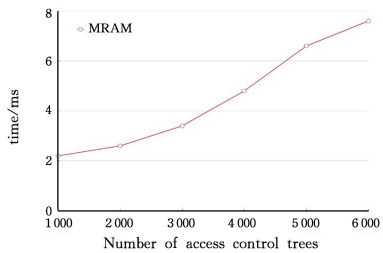

图 10　判断检索陷门权限时访问结构树数量与时间消耗的关系

Fig. 10　Relationship between access control treesand time consumption during judging privilege of trapdoor

5.2.2　存储开销

表 2 对 MRAM，MKSE-UR，ABKS-UR，ABE-CKS 和 SABM-DS 的理论存储开销进行了比较。可以得出，在安全索引生成阶段，MRAM，MKSE-UR 和 ABE-CKS 的存储开销均是关于 a 和 x 的二元一次函数，但 MRAM 的系数小于 MKSE-UR 和 ABE-CKS，故 MRAM 的存储开销理论上低于 MKSE-UR 和 ABE-CKS。ABKS-UR 和 SABM-DS 的存储开销是关于 n 的一次函数，考虑到全局属性集中元素的数量 n 非常大，所以 MRAM 的存储开销理论上也低于 ABKS-UR 和 SABM-DS。因此 MRAM 的安全索引存储开销理论上低于其他方案。从检索陷门的存储开销来说，MRAM，MKSE-UR 和 ABE-CKS 主要的存储开销是关于 m 的线性函数，且 MKSE-UR 和 ABE-CKS 的系数都为 MRAM 的 2 倍，SABM-DS 的存储开销则是关于 m 的二次函数，而 ABKS-UR 的存储开销则是关于 n 的一次函数。因此随着 n 和 m 的增长，MRAM 检索陷门存储开销的增长速度理论上会低于其他方案。

表 2　理论存储开销比较

Table 2　Comparison of theoretical storage overheads

Scheme	Secure index	Trapdoor
MRAM	$(a+2)l_0+xl_T$	$(m+y+1)l_0$
MKSE-UR	$(2a+x+6)l_0$	$(2m+3)l_0$
ABKS-UR	$(n+2)l_0$	$(2n+2)l_0$
ABE-CKS	$(2a+x+2)l_0$	$(2m+y+3)l_0$
SABM-DS	$(n+4)l_0+l_T$	m^2l_0

下面通过仿真实验测量并比较了 MRAM，MKSE-UR 和 ABKS-UR 中安全索引与检索陷门的存储长度。实验测量了 3 个方案在 $n=60$，$m=10$，$a=m$，$x=10$，$y=5$ 时的存储开销，结果如图 11 所示，MRAM 的安全索引和检索陷门都接近 2.5 kB，MKSE-UR 的安全索引和检索陷门长度都在 5 kB 左右，是 MRAM 的 2 倍，ABKS-UR 的安全索引长度是 8 kB，接近 MRAM 的 3 倍，而检索陷门长度则接近 16KB，超过 MRAM 的 5 倍。可以得出，MRAM 的存储开销优于 MKSE-UR 和 ABKS-UR。

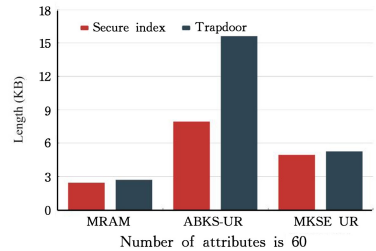

图 11　安全索引和检索陷门存储开销的比较

Fig. 11　Comparison of storage overhead between Security index and Trapdoor

结束语　现有的可搜索加密方案大多存在着仅支持单一或连接关键词检索、访问控制策略不灵活、检索效率低、计算和存储开销大、密钥管理复杂等问题。针对上述问题，文中提出了一种多云环境中基于属性加密的高效多关键词检索方案(MRAM)，相较于已有的相关工作[14-17]，MRAM 基于高性能的 CP-ABE 算法实现了任意密文多关键词检索、细粒度的访问控制，并且通过引入检索服务器有效支持多云环境中高效准确的密文检索。在未来工作中，将进一步研究多检索服务器集群的设计以及检索服务器功能的高效并行实现，以提高检索服务器的处理效率和容错性，从而实现性能和可靠性均更高的方案。

参 考 文 献

[1] JIANG Q, MA J F, WEI F S. On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services[J]. IEEE Systems Journal, 2018, 12(2): 2039-2042.

[2] LIY X, ZHOU F C, XU Z F, et al. An efficient two-server ranked dynamic searchable encryption scheme[J]. IEEE Access, 2020, 8: 86328-86344.

[3] TIAN H L, ZHANG Y, LI C, et al. A survey of confidentiality protection for cloud database[J]. Chinese Journal of Computers, 2017(10): 2245-2270.

[4] GE R J, YANG G Y, WU J S, et al. A novel chaos-based symmetric image encryption using bit-pair level process[J]. IEEE Access, 2019, 7: 99470-99480.

[5] XIE D. Public key image encryption based on compressed sensing[J]. IEEE Access, 2019, 7: 131672-131680.

[6] WANG G F, LIU C Y, DONG Y F, et al. IDCrypt: A multi-user searchable symmetric encryption scheme for cloud applications [J]. IEEE Access, 2018, 6: 2908-2921.

[7] CHEN B W, WU L B, WANG H Q, et al. A Blockchain-based searchable public-key encryption with forward and backward privacy for cloud-assisted vehicular social networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(6)5813-5825.

[8] SAHAI A, WATERS B. Fuzzy identity-based encryption [C]// Advances in Cryptology (EUROCRYPT 2005). Berlin: Springer, 2005: 457-473.
[9] DAN B, FRANKLIN M. Identity-based encryption from the Weil pairing [C]// Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology. Berlin: Springer, 2001: 213-229.
[10] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption [C]// Proceedings of the 2007 IEEE Symposium on Security and Privacy. Piscataway, NJ: IEEE, 2007: 321-334.
[11] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data [C]// Proceedings of the 13th ACM Conference on Computer and Communications Security. New York, NY: ACM, 2006: 89-98.
[12] LI S, XU M Z. Attribute-based public encryption with keyword search[J]. Chinese Journal of Computers, 2014, 37(5): 1017-1024.
[13] ZHENG Q J, XU S H, ATENIESE G. VABKS: Verifiable attribute-based keyword search over outsourced encrypted data [C]// Proceedings of the IEEE Conference on Computer Communications. Piscataway, NJ: IEEE, 2014: 522-530.
[14] LIANG K, SUSILO W. Searchable attribute-based mechanism with efficient data sharing for secure cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(9): 1981-1992.
[15] SONG Y, HAN Z, CHEN D, et al. Attribute-based encryption supporting arbitrary conjunctive key word search[J]. Journal on Communications, 2016, 37(8): 77-85.
[16] SUN W, YU S, LOU W, et al. Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(4): 1187-1198.
[17] WU Q Y, MA J F, LI H, et al. Multi-keyword search over encrypted data with user revocation[J]. Journal on Communications, 2017, 38(8): 183-193.
[18] YAN X X, MENG H. Ciphertext policy attribute-based encryption scheme supporting direct revocation[J]. Journal on Communications, 2016, 37(5): 44-50.
[19] IBRAIMI L, TANG Q, HARTEL P, et al. Efficient and provable secure ciphertext-policy attribute-based encryption schemes [C]// Proceedings ofthe 5th International Conference on Information Security Practice and Experience, Berlin: Springer, 2009: 1-12.
[20] LI J, WANG Q, WANG C, et al. Fuzzy keyword search over encrypted data in cloud computing [C]// Proceedings of the IEEE Conference on Information Communications, Piscataway, NJ: IEEE, 2010: 441-445.
[21] WANG Y, FAN K. Effective CP-ABE with Hidden Access Policy[J]. Journal of Computer Research and Development, 2019, 56(10): 2151-2159.
[22] HE H, ZHANG J, GU J G, et al. A fine-grained and lightweight data access control scheme for WSN-integrated cloud computing [J]. Cluster Computing, 2017, 20(2): 1457-1472.
[23] RUIXUAN L, CHENGLIN S, HENG H, et al. A Lightweight Secure Data Sharing Scheme for Mobile Cloud Computing[J]. IEEE Transactions on Cloud Computing, 2018, 6(2): 344-357.

HE Heng, born in 1981, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include network security, cloud computing and ciphertext retrieval.

(上接第 564 页)

[39] PAPINENI K, ROUKOS S, WARD T, et al. Bleu: a method for automatic evaluation of machine translation [C] // Association for Computa-tional Linguistics, 2002.
[40] YANG Z, JIN S, HUANG Y, et al. Automatically Generate Steganographic Text Based on Markov Model and Huffman Coding[J]. arXiv: 1811.04720.
[41] SHNIPEROV A N, NIKITINA K A. A text steganography method based on Markov chains [J]. Automatic Control & Computer Sciences, 2016, 50(8): 802-808.
[42] CHEN Z, HUANG L, MENG P, et al. Blind Linguistic Steganalysis against Translation Based Steganography[C]// International Workshop on Digital Watermarking. Springer, Berlin, Heidelberg, 2010.
[43] WEN J, ZHOU X, ZHONG P, et al. Convolutional Neural Network Based Text Steganalysis[J]. IEEE Signal Processing Letters, 2019, PP(3): 1-1.
[44] LEE J S, HSIANG J. Patent claim generation by fine-tuning OpenAI GPT-2 [J]. World Patent Information, 2020, 62: 101983.

ZHOU Xiao-shi, born in 1994, postgraduate. Her main research interests include deep transfer learning for image classification, deep learning for text information steganography, natural language processing, machine learning.

WEN Juan, born in 1982, Ph.D, associate professor. Her main research interests include artificial intelligence, information hiding, and natural language processing.