支持隐私保护的云存储框架设计

黄汝维[1,3]，桂小林[1]，余思[1]，张进[2]，卿杜政[2]
(1. 西安交通大学电子与信息工程学院，710049，西安；2. 北京仿真中心航天系统仿真重点实验室，100854，北京；
3. 广西大学计算机与电子信息学院，530004，南宁)

摘要：针对云存储中的隐私安全问题，设计了一个支持隐私保护的、高效且安全的云存储框架. 该框架采用多叉树结构构建数据索引，设计密钥推导算法 EKDA(Extirpation-Based Key Derivation Alogrithm)实现密钥的管理和分发，构建关键字检索算法 DLSEK(Discrete Logarithm-Based Search on Encrypted Keyword)实现对数据共享和密文检索的支持，并结合延迟更新技术解决用户访问权限变更和数据更新问题. 从 EKDA 的有效性、DLSEK 的性能和隐私安全方面进行实验评估和安全分析，结果表明：EKDA 能有效地减少通信和存储负载，DLSEK 是一种具有单向性安全的支持检索的加密技术，整个框架的设计能有效地保护用户的隐私，同时支持高效的数据访问.

关键词：云存储；密钥推导；离散对数；密文检索
中图分类号：TP393 **文献标志码**：A **文章编号**：0253-987X(2011)10-0001-06

Design of Cloud Storage Framework with Privacy-Preserving

HUANG Ruwei[1,3], GUI Xiaolin[1], YU Si[1], ZHANG Jin[2], QING Duzheng[2]
(1. Department of Electronics and Information Engineering, Xi′an Jiaotong University, Xi′an 710049, China;
2. Science and Technology on Special System Simulation Laboratory, Beijing 100854, China;
3. School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China)

Abstract: An efficient and secure framework of cloud storage is proposed to support privacy security in cloud storage. The framework adopts a multi-tree structure for indexing, designs an extirpation-based key derivation algorithm (EKDA) for key management, and constructs a discrete logarithm-based search on encrypted keyword (DLSEK) for data sharing and ciphertext retrieval. The lazy revocation is combined into the framework to deal with the changes of users' access right and dynamic operations of data. Analyzing results for the effectiveness of EKDA, the performance of DLSEK and the privacy security of the framework show that EKDA can efficiently reduce the communication and storage overheads and that DLSEK is an encryption technique which supports ciphertext retrieval and is one-way security. The proposed framework is privacy-preserving while supporting data access efficiently.

Keywords: cloud storage; key derivation; discrete logarithm; ciphertext retrieval

云存储以服务的形式向用户提供按需的、可扩展的、满足 QoS 要求的存储资源，用户可随时随地地访问这些服务. 面对云存储如此强大的优势，许多个人和企业却犹豫是否要将个人或企业的数据迁移到云存储中，其主要原因是人们觉得失去了对数据的控制. 目前发生的多起云存储中的数据泄漏和丢失情况证实了人们的担心：LinkUp 是一家提供云存储服务的公司，它由于系统管理员的失误而丢失

收稿日期：2011-03-01. 作者简介：黄汝维(1978—)，女，博士生；桂小林(联系人)，男，教授，博士生导师. 基金项目：国家自然科学基金资助项目(60873071)；国家高技术研究发展计划资助项目(2008AA01Z410)；IBM 共享高校研究计划资助项目.
网络出版时间：2011-07-22 网络出版地址：http://www.cnki.net/kcms/detail/61.1069.T.20110722.1712.001.html

了45%的用户数据从而倒闭;攻击者使用钓鱼攻击的方法,成功地盗取了云服务提供商 Salesforce.com 存储的用户 Emial 和地址信息;Google 的 Docs 遭受无授权的攻击者的访问,数据发生了泄露. 因此,为了持续深入地发展,云存储必须提供支持隐私保护的数据存储和访问策略.

关于数据外包的研究,Benaloh 等[1]基于对称密钥和非对称密钥设计了密钥推导策略,开发了一个支持隐私保护的电子健康记录系统;Thompson 等[2]提出了一个支持隐私保护的协议 PDAS,用于多个服务提供者协作完成聚集查询的计算和证明;Wang 等[3]采用二叉树结构构建索引,使用密钥推导技术管理密钥,并结合再加密和延迟更新技术来处理数据的动态变化;Liu 等[4-5]提出了基于非对称加密的关键字检索方法;Dawn 等[6]提出了基于对称加密的关键字检索方法;Yasuhir 等[7-8]提出了基于 Bloom Filter 的密文检索方法;Wong 等[9]介绍了一种对加密的外包数据进行 KNN(K-Nearest Neighbor)计算的方法; Agrawal 等[10]设计了一种保序的加密方案,可以用来实现加密的数值数据的比较. 宋伟等[11]通过分析不同查询对客户端查询误检率的影响,提出了一种面向服务的加密数据高效查询方法 AEI,可以很好地适应数据的频繁更新操作,也可以适应用户查询和数据的非均匀分布. 通过对以上工作的研究发现:①目前缺乏一个支持隐私保护的云存储框架,以从整体上考虑密钥管理和分发、数据和用户权限动态变化、加密数据检索等问题;②目前的很多方案都没有考虑到数据共享问题,往往假设数据的使用者就是数据的拥有者,因此所设计的加密数据检索方案只能由数据拥有者使用.

针对以上问题,本文设计了一个支持隐私保护的云存储框架. 该框架采用多叉树结构构建数据索引,设计密钥推导算法 EKDA (Extirpation-Based Key Derivation Algorithm)实现密钥的管理和分发,构建关键字检索算法 DLSEK (Discrete Logarithm-Based Search on Encrypted Keyword)实现支持数据共享的密文检索,并结合延迟更新技术解决用户访问权限变更和数据更新问题.

1 支持隐私保护的云存储框架

支持隐私保护的云存储框架反映了云环境中的数据拥有者(O)、数据使用者(U)和服务提供者(S)各自的功能模块以及它们之间的交互过程,如图1所示.

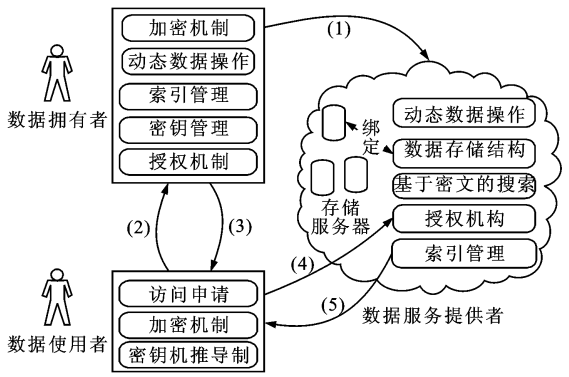


图1 支持隐私保护的云存储框架

以下介绍数据拥有者、数据使用者和服务提供者之间的交互协议。

(1)数据拥有者对第 i 个数据块 d_i 及其关键字 $\{k_1,k_2,\cdots,k_n\}$ 加密,并将加密后的结果传递给服务提供者

$$M_{OS}=\{O,S,E((O,S,t_{\mathrm{m}},M,E(d_i,k_{s,i})\parallel E_{\mathrm{KWEnc}}(k_1,k_{\mathrm{pub}})\parallel\cdots\parallel E_{\mathrm{KWEnc}}(k_n,k_{\mathrm{pub}})),k_{OS})\}$$

式中:E 表示一种对称加密算法;$k_{s,i}$ 为对称密钥;k_{pub} 为 DLSEK 的公钥;E_{KWEnc} 是 DLSEK 的外包关键字加密算法;k_{OS} 表示数据拥有者和服务提供者之间的密钥;t_{m} 用来表示 d_i 最近更新的时间;M 是信息验证码.

(2)数据使用者向数据拥有者申请访问权限.

$$M_{UO}=\{U,O,E((U,O,I,M),k_{UO})\}$$

式中:k_{UO} 表示数据使用者和数据拥有者之间的密钥; I 表示数据使用者发送请求的序号.

(3)数据拥有者验证数据使用者的身份,对访问控制列表进行查找,确定其可访问的数据块范围,返回可以产生这些数据块密钥的最小编号组 $N_{\min}$ 和最小密钥组 $K_{\min}$、证书 C.

$$C=\{E((U,I,N_{\min},t_c,N_{\mathrm{AR}},M),k_{OS})\}$$

$$M_{OU}=\{O,U,E((O,U,I,C,M,N_{\min},K_{\min}),k_{OU})\}$$

式中:k_{OS} 表示数据拥有者和服务提供者之间的密钥;t_c 表示 C 产生的时间;N_{AR} 表示该用户权限的更新次数.

(4)数据使用者向服务提供者发起查询请求,希望服务提供者返回符合请求的数据块

$$M_{US}=\{U,S,O,I,C,E_{\mathrm{TDGen}}(s,k_{\mathrm{pub}})\}$$

式中:s 表示数据使用者感兴趣的数据块的关键字;E_{TDGen} 是 DLSEK 的查询参数加密算法.

(5)服务提供者检验 C,返回在数据使用者访问权限内的以 s 为关键字的数据块

$$M_{SU}=\{S,U,I,M,\ E(d_i,k_{s,i})\parallel E(d_j,k_{s,j})\parallel\cdots\parallel E(d_t,k_{s,t})\}$$

数据使用者获得数据块，用 $K_{\min}$ 产生各数据块的密钥，从而对数据块进行解密.

关于以上模型有 2 点需要说明：①数据块的定义因数据拥有者的要求不同而不同，本文以文件为数据块单位；②对于不需要保密的文件，可以不进行加密.

2　框架设计中的关键问题

2.1　多叉树型索引结构

数据拥有者总是按照一定的逻辑来组织自己的文件，为了真实地反映这种逻辑，本框架采用多叉树结构来构建文件索引. 在用户将文件存储到服务提供者的存储空间前，云计算的客户端软件会自动地根据用户的文件组织形式生成对应的多叉树索引，如图 2 所示.

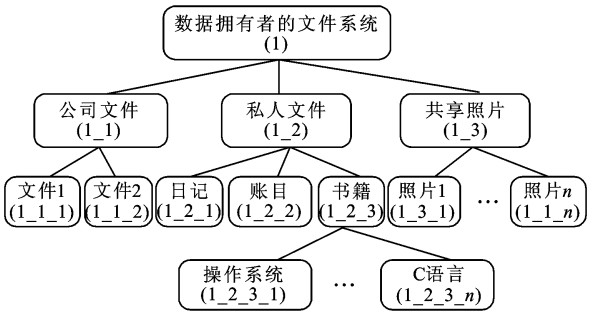


图 2　文件的索引结构

在这个索引中，只有叶子节点对应文件，非叶子节点对应文件的分类或文件夹. 当数据拥有者准备存储文件时，例如日记，首先采用对称密钥算法（例如 AES）对内容进行加密，然后将文件名改为 $1\_2\_1\$E(\text{文件名},k_{s,1\_2\_1})$. 这样，服务提供者只能知道这个文件的编号为 1_2_1，既不能知道文件的内容，也不能知道文件的名字，从而能很好地保护用户的隐私. 同时，服务提供者根据文件的编号为每个用户建立一个多叉树索引.

2.2　基于 EKDA 的密钥管理和分发

为了有一个灵活的、细粒度的数据访问控制机制，每个文件都有不同的密钥. 本框架采用对称加密算法对文件进行加密，从而减轻加解密操作带来的负担. 但是，如何管理为数众多的密钥呢？本框架采用密钥推导方法：首先随机选取一个 128 位的密钥作为根密钥；假设当前文件的编号为 n，其父节点的密钥为 k_p，通过哈希函数 h 计算文件 n 的密钥 $k_n=h(k_p\parallel n)$. 数据拥有者只需保存根密钥，这样不仅方便了密钥的管理，也极大地节约了数据拥有者的存储空间. 当一个数据使用者请求访问数据拥有者的文件时，数据拥有者首先审核数据使用者的身份，确定数据使用者的访问范围；然后，将可以推导出访问范围内所有文件密钥的最小编号组和最小密钥组发给用户. 这样的设计能有效地减少数据使用者、数据拥有者和服务提供者之间交互的通信量.

但是，密钥推导方法的有效性在一些情况下会受到损害：当数据使用者的权限发生了变化，数据拥有者为了不让数据使用者再访问到文件，就只能采用新的密钥来加密文件. 新密钥不能再通过 $K_n=h(k_p\parallel n)$ 计算，而且当使用新密钥的文件数量很多，或者每个倒数第 2 层目录下都有文件使用了新密钥，这时系统采用密钥推导方法的效果和没有采用该方法的效果是一样的，即满足数据使用者要求的文件数量为 n 时就要返回 n 个密钥.

为了解决以上问题，我们设计了一个新的密钥推导算法 EKDA：数据拥有者将受到使用者访问权限变更影响而首次使用新密钥的节点从原索引树中摘除，放到更新树中，更新树中对应节点的编号和密钥是随机选取的；当该节点要再次更换新的密钥时，直接修改更新树对应节点的密钥即可；当数据使用者请求访问一组文件时，数据拥有者使用 EKDA 产生符合要求的最小密钥组和最小编号组. 如图 3 所示，当数据拥有者更新文件 1_3_1 时，它让服务提供者删除原文件，并提示服务提供者将索引中的1_3_1 节点标识为“updated”，然后在更新树中选取节点 2_2_8_1来标识更新后的文件，随机选取新密钥 $k_{s,2\_2\_8\_1}$ 来加密文件的名字和内容，最后再把加密好的文件存到服务提供者处，并提示服务提供者将节点 1_3_1 和 2_2_8_1 关联起来. 当数据使用者被授权访问在 1_3 目录下的文件时，数据拥有者返回的最小编号和密钥组为 $1\_3:k_{s,1\_3}\ \$\ 2\_2\_8\_1:k_{s,2\_2\_8\_1}$. EKDA 的具体步骤如下.

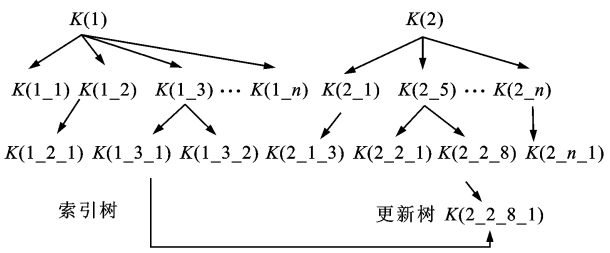


图 3　索引树到更新树的映射

(1)如果节点 N 第一次更新密钥,则:①标识 N 为已更新节点;②任意在更新树中选取一个节点 N_u, N_u 记录下随机生成的新密钥,将 N 与 N_u 关联;③用新密钥加密对应文件并存到服务提供者处.

(2)如果 N 非第一次更新密钥,则:①找到 N 在更新树上对应的 N_u, 随机生成一个新密钥存入 N_u;②用新密钥加密对应的文件并存到服务提供者处.

(3)如果要生成符合检索要求的 m 个节点的最小密钥组 $K_{\min}$,对于其中的每个节点 $N_i(i\in[1,m])$ 进行如下操作:①如果 N_i 是更新过的,则返回更新树上对应节点的密钥;②如果 N_i 没有更新过,则搜索这 m 个节点中从 N_i 开始的其他节点的共同父节点,如果有共同的父节点,返回父节点的密钥,否则只返回 N_i 的密钥.

2.3 加密关键字检索算法 DLSEK

在云环境中,数据拥有者将文件存储到服务提供者处,凡是经数据拥有者授权的数据使用者都可以向服务提供者提交查询,服务提供者会根据数据使用者的授权范围和查询条件找到相应的文件并返回.这样设计的好处是服务提供者承担了对文件的检索工作,减轻了数据拥有者的负担.为了保证数据拥有者和数据使用者的隐私,需要将文件、文件的关键字和检索参数加密.服务提供者在不知道关键字内容和检索参数的情况下能够正确地找到满足检索条件的文件. DLSEK 正是为此目的而设计的.

离散对数的定义:给定一个素数 p 和有限域 Z_p 上的一个原根 a,对 Z_p 上的整数 b 寻找唯一的整数 c,使得 $a^c\equiv b \bmod p$. 但是,计算离散对数是困难的,对于大的素数,计算出离散对数几乎是不可能的.

DLSEK 由 4 个关键算法组成,它们分别是关键字抽取、关键字加密、检索参数加密、关键字检索算法.

(1)关键字的抽取: 数据拥有者为每个存储到服务提供者处的文件创建若干个关键字.但是,当文件数量很多时,这是一项繁杂的工作.因此,可以采用基于文件名的关键字抽取客户端软件,它可以根据语言的特点自动地抽取文件名中的关键字.例如有一个文件名为“The Storage of Cloud Computing”,该软件会将“Storage”、“Cloud”和“Computing”抽取出来形成关键字.目前的工作主要是针对文件名上的关键字抽取.抽取出来的关键字 k 通过连接其每个字符的 ASCII 码值,从而转换为一个十进制数 k_d.

(2)关键字加密:数据拥有者选择一个大素数 p 和其原根 a 作为公钥对外公布.为了加密 k_d,生成随机数 $r(r\in Z^+)$,并计算 $p-1$ 的最大因子 d,接着进行如下加密计算

$$w=(a^{k_d}r^{(p-1)/d})\bmod p \tag{1}$$

最后将 w 与加密的文件一起存储到服务提供者处.

(3)检索参数加密: 数据使用者希望查找名字中包含关键字 s 的文件,他首先按照同样的方法将 s 转换为一个十进制数 s_d,接着生成一个随机数 $r_s(r_s\in Z^+)$,将 s_d 加密为 w_s 并生成一个比较值 c

$$w_s=(a^{-s_d}r_s)\bmod p \tag{2}$$

$$c=r_s{}^d \bmod p \tag{3}$$

最后将 w_s 和 c 一起发给服务提供者.

(4)关键字检索:服务提供者收到请求后,首先从数据使用者的证书中提取数据使用者的访问权限,然后对该权限范围内的每个文件的关键字进行如下操作

$$ww_s\equiv(a^{k_d}r^{(p-1)/d})(a^{-s_d}r_s)\bmod p\equiv (a^{k_d-s_d}r^{(p-1)/d}r_s)\bmod p \tag{4}$$

$$(ww_s)^d\equiv(a^{k_d-s_d}r^{(p-1)/d}r_s)^d\bmod p\equiv (a^{k_d-s_d}r_s)^d r^{(p-1)/d^2}\bmod p\equiv (a^{k_d-s_d}r_s)^d r^{p-1}\bmod p\equiv (a^{k_d-s_d}r_s)^d\bmod p \tag{5}$$

$$\text{if}\quad (w\times w_s)^d==c \tag{6}$$

$$\text{then}\quad k_d==s_d$$

$$\text{if}\quad (k_d==s_d)$$

then 该文件是满足条件的文件.

于是,服务提供者找到了名字中包含数据使用者检索参数的文件.

2.4 访问权限的变更

首先,服务提供者为每个数据拥有者建立了一个权限更新列表 L,它的每个元素都是一个链表,第 i 个元素记录了编号为 i 的数据拥有者的用户权限变化情况.链表节点属性 N_l 和 N_t 分别记录了发生权限更新的数据使用者编号及其更新次数.当编号为 i 的数据拥有者修改编号为 j 的数据使用者的权限后,向服务提供者发送更新权限消息,该消息由 2 部分组成:数据使用者的编号 j \$ 更新标识(\$ 为连接符;更新标识为 1 表示已经更新, 更新标识为 0 表示没有更新).服务提供者接到消息后查找链表 $L[i]$,看看其中是否存在 $N_l=j$ 的节点,如果存在则 N_t++,否则创建一个 N_l 为 j 的节点,使 $N_t=1$.当编号为 j 的数据使用者向服务提供者申请数据

时，服务提供者首先检查 $L[i]$ 中是否有节点 $N_l=j$，如没有，则查找符合条件的文件并返回；如有，则检查 N_t 是否等于证书中的权限更新次数 N_{AR}，如果一致则查找符合条件的文件并返回，如果不一致，则拒绝提供文件，并提醒该用户证书已经过期，需向编号为 i 的数据拥有者重新申请. 以上操作防止了权限更改的编号为 j 的数据使用者再从服务提供者处获取文件.

当然，数据使用者可以通过传输线路盗取相应的文件. 本文采用了延迟更新的方法，即当用户的访问权限发生了变更，如果文件没有发生更新，数据拥有者和服务提供者不需要做任何操作.

2.5　数据的动态操作

数据拥有者对数据的动态操作包括增加、删除和更新. 当数据拥有者要增加一个新文件时，系统会根据该文件与其他文件的逻辑关系为它生成一个编号 n，计算其对称密钥 $k_n=h(k_p \| n)$，然后对该文件进行加密并储存到服务提供者处. 当数据拥有者要删除一个文件时，他将给服务提供者发送一条删除命令，然后在索引树中该节点处标明"deleted". 当有一个新的文件希望使用被删除的文件编号时，它会被当成更新的文件处理.

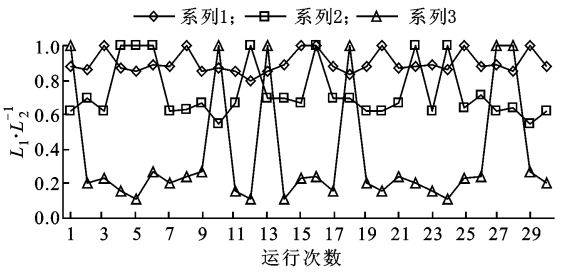
当文件被更新后，如果原来可访问该文件的数据使用者的权限没有被更新，可以使用原来的密钥进行加密，否则要进行如下处理：由于采用了延迟更新的方法，因此要想更新后的文件不被更改了访问权限的数据使用者访问，在将文件发送给服务提供者之前要采用新的密钥进行加密. 此时可以使用 EKDA 来创建新的 N_u，用其密钥 k_n 对更新后的文件和文件名加密并发送给服务提供者，让服务提供者在其索引中将更新前和更新后的节点关联起来. 这使得访问权限没有变的数据使用者仍然能够访问更新后的文件. 当已授权数据使用者向服务提供者申请访问该文件时，服务提供者将进行如下比较，如果 $t_m>t_c$ 并且 $N_{AR}==N_t$，说明该用户是一个已授权的用户，只是他的密钥已经过期，这时服务提供者在返回文件的同时会提示用户该文件的密码已经过期，因此用户要想解密这个文件，就要向编号为 i 的数据拥有者请求新的密码；如果 $t_m\leqslant t_c$ 并且 $N_{AR}==N_t$，说明该用户是一个已授权用户并且密码是最新的，这时服务提供者将返回符合条件的文件；如果 $N_{AR}<N_t$，说明该用户是一个权限被更改的用户，服务提供者将拒绝处理他的查询并提示访问权限已过期.

3　系统性能分析

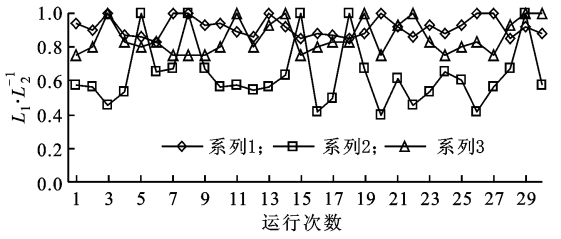
本课题组正在进行"校园云计算平台——青云"的设计与开发工作. 基于本文介绍的云存储框架，本课题组用 Java 语言初步实现了一个云存储的系统原型. 下面从 EKDA 的有效性、DLSEK 的性能和隐私安全 3 个方面对框架的可行性进行分析.

3.1　EKDA 的有效性

实验通过模拟多数据使用者、多数据拥有者和多服务提供者的交互，并让数据使用者随机地提出请求，数据拥有者随机地修改权限和更新数据来反映真实的云计算数据存储和访问的环境. 数据拥有者将 10 000 个文件按照不同的组织结构存储到服务提供者的服务器上. 通过分别测量采用 EKDA 和普通的密钥推导算法返回的符合检索要求的最小文件编号组大小 L_1 和 L_2 的比值，从而证明 EKDA 的有效性，如图 4 所示.



(a)情况 1



(b)情况 2

图 4　EKDA 有效性分析

图 4a 反映了在 3 种不同文件组织结构中更新同一文件时 EKDA 的有效性；图 4b 是在与图 4a 相同的 3 种文件组织结构中更新另一个文件时的有效性. 通过对图 4a、4b 的分析可以看到：①EKDA 是很有效的，因为 $L_1/L_2\leqslant 1$；②文件的组织结构对 EKDA 有直接的影响；③更新的文件所在的位置对 EKDA 也有影响；④在某一种文件组织结构下，其曲线除了值为 1 的点外，其他的点都围绕某一值上下波动. 其原因是如果要访问的内容为某目录下的所有文件，假设个数为 n，当更新了该目录下的 m 个文件的密钥后，$L_1/L_2=(1+m)/n$，所以曲线除值为

1 的点外会围绕$(1+m)/n$上下波动.

3.2 DLSEK 的性能

实验评估了在不同的字符串长度下、数据拥有者加密一个关键字的平均计算负载(KWEnc)、数据使用者加密一次查询参数的平均计算负载(TDGen)和服务提供者查询该数据使用者授权范围内的文件的一个关键字的平均计算负载(KWSearch)、假设数据拥有者有 10 000 个文件,每个文件有 5 个关键字,加密使用 150 位十进制素数为公钥,实验结果如图 5 所示,其中 l 表示关键字(或查询参数)的长度.

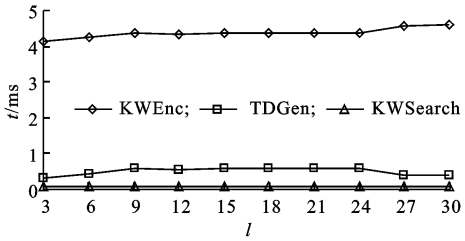


图 5 DLSEK 的运行负载

由图 5 可知:①随着关键字长度的增加,关键字加密算法的计算负载也逐渐增加;②检索参数加密算法的计算负载几乎不随其长度的增加而增长;③对同一组文件的关键字进行检索,其计算负载是不变的,不受检索参数和关键字长度增加的影响.

3.3 隐私安全性

通过图 1,可以分析出云环境中可能出现的 2 种隐私攻击.①外部攻击:数据在传递的过程中,外部攻击者可以通过窃听等方式盗取数据,或者通过攻击服务提供者盗取数据;②内部攻击:由于数据拥有者的数据存放在服务提供者的存储服务器上,内部攻击者可以访问用户的数据.针对第 1 类攻击,即使攻击者窃听到数据,由于文件是经过加密的,攻击者没有对称加密算法的密钥,无法破解加密的文件;由于离散对数的难解性,攻击者无法在多项式时间内破解加密的关键字.因此,我们主要讨论第 2 类攻击.

假设内部攻击者可以访问所有存储在云服务器上的数据,并且由于好奇,他会持续地对数据拥有者的数据和数据使用者的查询请求进行统计和分析.由于文件是用成熟的对称加密算法 AES 加密的,因此我们主要分析 DLSEK 的隐私安全性.

DLSEK 是一种具有单向安全性的加密算法.

证明 根据式(1)和(2)可知,关键字加密算法和检索参数加密算法建立在离散对数的基础上,已知密文和公钥无法在多项式时间内求出对应的明文,并且由于 r 和 r_s 是随机数,因此上述算法是不确定的加密算法,即相同的明文在公钥加密后会产生不同的密文.在服务提供者只是进行简单的比较操作的情况下,以上特点确保了数据拥有者关键字和数据使用者查询模式的隐私安全.

但是,当服务提供者对加密关键字和检索参数进行如下计算时

$$w^d = (a^{k_d} r^{(p-1)/d})^d \bmod p = a^{k_d} \bmod p$$

$$c^{-1}c \equiv 1 \bmod p \Rightarrow c^{-1}$$

$$w_s c^{-1} \equiv a^{-s_d} \bmod p$$

就会使加密的关键字和检索参数丧失不确定性.由于离散对数的难解性,攻击者还是无法求出 k_d 和 s_d,所以 DLSEK 是一种具有单向性的加密算法.

4 总 结

本文建立了一个支持隐私保护的云存储框架.该框架采用多叉树结构建立数据索引,基于 EKDA 管理和分发密钥,构建 DLSEK 实现对加密关键字的检索,并采用延迟更新策略来处理用户权限的变更和数据的更新.本文从 EKDA 的有效性、DLSEK 的性能和隐私安全 3 方面进行了分析.结果证明,本框架能很好地解决密钥管理、隐私保护和数据共享等问题,并能有效地减轻通信、存储和计算负担.

为了进一步实现隐私保护的目标,我们下一步的工作主要有 3 个方面:①改进 DLSEK,使其具有更高的安全性;②进一步研究支持模糊查询的加密关键字检索技术;③研究支持算术运算的加密技术.

参考文献:

[1] BENALOH J, CHASE M, HORVITZ E, et al. Patient controlled encryption: ensuring privacy of electronic medical records [C]// Proceedings of the 2009 ACM workshop on Cloud computing security. New York, USA: ACM, 2009: 103-114.

[2] THOMPSON B, HABER S, HORNE W G, et al. Privacy-preserving computation and verification of aggregate queries on outsourced databases [C]// Proceedings of the 9th International Symposium on Privacy Enhancing Technologies. New York, USA: ACM, 2009: 185-201.

[3] WANG Weichao, LI Zhiwei, OWENS R, et al. Secure and efficient access to outsourced data [C]// Proceedings of the 2009 ACM Workshop on Cloud Computing Security. New York, USA: ACM, 2009: 55-66.

[4] LIU Qin, WANG Guojun, WU Jie. An efficient privacy preserving keyword search scheme in cloud computing[C]// Proceedings of the 2009 International Conference on Computational Science and Engineering. New York, USA: ACM, 2009: 715-720.

[5] BONECH D, CRESCENZO D G, OSTROVSKY R, et al. Public-key encryption with keyword search [C] // Proceedings of Eurocrypt 2004. Berlin, Germany: Springer, 2004: 506-522.

[6] CHANG Yancheng, MITZENMACHER M. Privacy preserving keyword searches on remote encrypted data [EB/OL]. [2011-07-06]. http://eprint.iacr.org/2004/051.pdf

[7] OHTAKI Y. Partial disclosure of searchable encrypted data with support for Boolean queries [C]// Proceedings of the 2008 Third International Conference on Availability, Reliability and Security. New York, USA: ACM, 2008: 1083-1090.

[8] BELLOVIN S, CHESWICK W. Privacy-enhanced searches using encrypted bloom filters [EB/OL]. [2011-07-06]. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid = FDCF3E4971210E7BF4E2BAAD9939371C? doi = 10.1.1.58.6899&rep = rep1&type=pdf

[9] WONG W K, CHEUNG D W, KAO Ben, et al. Secure kNN computation on encrypted databases[C]// Proceedings of the 35th SIGMOD International Conference on Management of Data. New York, USA: ACM, 2009: 139-152.

[10] AGRAWAL R, KIEMAN J, SRIKANT R, et al. Order preserving encryption for numeric data [C]// Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. New York, USA: ACM, 2004: 563-574.

[11] 宋伟，彭智勇，程芳权，等. 可信数据库环境下面向服务的自适应密文数据查询方法[J]. 计算机学报，2010，10(8)：1324-1338.
SONG Wei, PENG Zhiyong, CHENG Fangquan, et al. Service-oriented adaptive search method over encrypted data in trusted database [J]. Journal of Computers, 2010, 10(8): 1324-1338.

（编辑　荆树蓉　赵大良）