基于改进索引散列树的动态大数据审核方法

王向华[1] 刘 颖[2]

[1](天津职业大学电子信息工程学院 天津 300410)

[2](天津科技大学计算机科学与信息工程学院 天津 300222)

摘 要 为了解决公共审核和数据动态性问题,以及降低计算成本,提出一种改进的索引散列树的数据审核方法。该方法在经典 Merkle 散列树 MHT(Merkle Hash Tree)和 BLS 签名的基础上,提出一种改进的 MHT。增加节点字段,修改 MHT 树的节点信息,使每个节点信息均包含存储数据块的哈希值和相对索引。通过时间戳字段与 MHT 的根节点相关联,以提供数据新鲜性。数据分析表明,该方法的不当操作检测概率较高,具有较好的安全性。与其他方法相比,该方法的计算成本较低,服务器和审核者的总体时间复杂度为 $O(n)$,验证了对 MHT 的改进行之有效。

关键词 数据动态性 索引散列树 BLS 签名 安全性 计算成本

中图分类号 TP309 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2019.01.053

DYNAMIC BIG DATA AUDIT METHOD BASED ON IMPROVED INDEX MERKLE HASH TREE

Wang Xianghua[1] Liu Ying[2]

[1](*College of Electronic Information Engineering, Tianjin Vocational Institute, Tianjin 300410, China*)

[2](*College of Computer Science and Information Engineering, Tianjin University of Science and Technology, Tianjin 300222, China*)

Abstract To solve the problem of public audit and data dynamics, and to reduce the computational cost, we proposed a data audit method based on improved index Merkle hash tree. We put forward an improved MHT on the basis of the classical Merkle hash tree (MHT) and the BLS signature. We added the node field, and modified the node information of MHT tree so that each node information contained the hash value and relative index of the stored data block. The timestamp field was associated with the root node of MHT to provide data freshness. Data analysis shows that the proposed method has higher detection probability and better security. Compared with other methods, the proposed method has lower computational cost, and the overall time complexity of server and reviewer is $O(n)$, which verifies the effectiveness of the MHT improvement.

Keywords Data dynamics Index Merkle hash tree BLS signature Safety Computational cost

0 引 言

云计算[1]作为一个基础设施,提供数据存储服务,并进行常用安全维护工作,取代了用户计算机的硬盘驱动器或其他本地私有的存储设施。由此,企业和个人保护数据安全的压力,以及日益增长的容量需求都转由云服务供应商来负担。用户使用基于 Web 的接口或一些特定的应用程序接口,通过互联网连接,方便地访问数据[2]。虽然云提供了存储便利,但大数据安全性是用户的主要担忧。一旦用户数据从本地系统转移到云端,则该数据就暴露在云中,云中的数据可能面临外部和内部的威胁。例如,当一个云服务供应商 CSP[3](Cloud Service Provider)面临不定时的复杂故障

收稿日期:2018-07-06。天津市基础研究计划项目(14JCTPJC00553);天津市高等学校科技发展基金计划项目(20130711)。
王向华,副教授,主研领域:数据分析处理,软件工程。**刘颖**,副教授。

时,其可能会为了私利而选择向用户隐瞒数据异常情况[4]。为了获得经济利益或更大的存储空间,CSP 可能会故意删除用户不经常访问的数据。云中的数据也可能由于外部攻击者的入侵而被破坏或删除。传统的完整性审核方法,例如散列函数和数字签名,并不适用于云环境,因为其没有可用的本地数据副本。此外,由于用户的资源限制,为进行完整性验证而下载整个数据不具备实践意义。

为此,有很多研究者对大数据审核进行了研究,提出了一些远程数据审核 RDA(Remote Data Audit)协议,以确保云系统中的数据完整性。如文献[5]提出的方法在审核程序的时间复杂度方面具有高效性,使用伪随机函数和 BLS 签名提供安全保障。但是该协议的通信复杂度一般正比于外包数据块大小,使用仅限于静态数据。文献[6]在可证明的数据占有模式下进行扩展,提出了一种动态数据处理方法,即:使用秩信息定义动态可证明数据占有模式,以支持数据动态更新存储。然而,该方法依赖于身份验证的跳跃表。文献[7]提出了一个通用框架,可检索动态证据。该方法使用日志存储所有的更新信息,使得数据更新失败时确保其可检索性。但服务器端必须对数据进行两次存储,一次用于数据真实性确认,另一次用于日志存储,增加了处理负担。文献[8]提出的协议中给出了一个涉及到移动攻击者的框架,可执行遗传算法对动态数据进行最优分解,但该研究仅限于静态数据。文献[9]基于网络编码设计出了一个用于分布式数据存储的框架。该框架的主要缺点是不支持公共审核,代码缺少系统性,且该框架不适用于在线存储。此外,其在访问一个文件的一小部分时需要重新生成整个数据文件。

现有的协议试图为云系统提供数据完整性审核,但是均存在一定的缺陷:有些方法针对静态数据设计,但无法处理动态数据;部分方法虽然具有分布式框架,但受限于处理程序框架,其数据动态性表现不佳。对此,本文在 MHT 树的基础上进行改进,提出一种改进的索引散列树 IIMHT(Improved Index Merkle Hash Tree)的数据审计方法。本文主要工作总结如下:1)通过分析云计算中的数据审核模型,降低了完整性验证的计算复杂度。2)将时间戳集成到持有权证据的一个字段,以确保检索到最近的数据副本,较好地保证了数据新鲜性。

1　修改的 MHT

1.1　BLS 签名

BLS 签名[10-11]一般用于对消息的签名者进行身份验证,使用双线性配对进行签名认证,其中签名是一些椭圆曲线的群元素。其主要特性是:多个消息生成的带有多个公共密钥的多个签名可以被聚合成单个签名。由此,文件“F”的 n 个数据块($d[1],d[2],\cdots,d[n]$)的 BLS 签名可生成如下:

$$\psi_{agg} = \prod_{i=1}^{n} (H(d[i]))^{k} \tag{1}$$

式中:$H:\{0,1\}^{*}\rightarrow G$ 为散列函数,G 为包含生成器 g 的一个素数阶群。$k\in\mathbb{Z}_p$ 表示密钥,其是一个整数模 p 环。BLS 签名法由三个函数组成:密钥生成、消息签名和验证:

1)密钥生成:生成密钥 $k\in\mathbb{Z}_p$。将 g^k 作为公共密钥发布并保存私有密钥;

2)消息签名$(F,k):\psi\rightarrow H(F)^k$;

3)验证$(F,\psi):e(\psi,g) =^{?} e(H(F),g^k)$。

1.2　双线性配对

考虑$(G_1,+)$、$(G_2,\cdot)$为包含素数阶 r 的两个循环群,其中$(G_1,+)$为加法群,$(G_2,\cdot)$为乘法群。一次配对即定义为 $e:G_1\times G_1\rightarrow G_2$ 的一个映射,其满足以下条件:

1)双线性:$\forall P\in G_1,\forall Q\in G_2,\forall a,b\in Z_p$,若 $e(Pa,Qb)=e(P,Q)ab$,则映射“e”被认为是双线性的;

2)非退化性:$\forall P,Q\in G_1, e(P,Q)\neq 1$;

3)可计算性:映射“e”是可计算的。

这些配对可以应用到密码方案中,且允许用户创建新的方案,例如 BLS 签名。在密码技术的所有签名方法中,BLS 签名的长度最短。

1.3　CDH 问题

在 CDH 假设[12]中,循环群中一个特定的计算问题是 NP-hard 问题。假定一个阶为 g 的一个循环群 G。对于给定的(g,g^a,g^b),其中 $a,b\in Z_p$ 为未知,则确立数值 g^{ab} 是计算难题。

1.4　经典 Merkle 散列树

MHT[13]是一个知名的二叉树数据结构,其中每个非叶节点均是其对应叶节点串联字符串的哈希 hash,最上方的节点被称为根节点。其中,叶节点代表着数据块的 hash。MHT 的优点包括了对较大数据的安全验证。根节点的认证机制能够确保所有叶节点的完整性。图 1 给出了包括 8 个叶节点的一棵 Merkle 散列树。

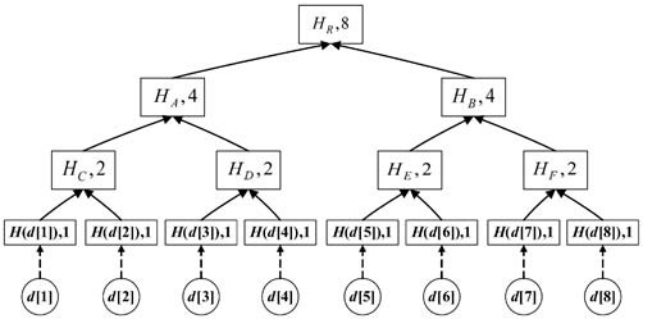

图 1 经典 Merkle 散列树

举例来说,如果审核者要求位于位置 3 的数据节点处的完整性验证的证明,则该证明程序与审核者共享辅助信息(AI),即:

$AI(d[3]):\{(H_C,L),H(d[3]),(H(d[4],R),(H_B,R)\}$。则审核者可以生成根 H_R 如下:

1) 计算 $H_D \leftarrow (H(d[3]) \| H(d[4]))$;

2) 计算 $H_A \leftarrow (H_C \| H_D)$;

3) 计算根 $H_R \leftarrow (H_A \| H_B)$。

通过验证根节点的真实性,自动完成对所有数据块的认证。但经典 MHT 的主要缺点是节点的信息难以满足大量动态数据的更新和检索,因此,签名所用的计算量较大。另外,根节点很难提供数据副本的最新证明。

1.5 经典 MHT 的修改

为了解决经典 MHT 的不足,本文对 MHT 进行了修改,形成了 IIMHT 协议,使其在云计算中能够较好地执行数据完整性审核。在本文 IIMHT 协议中,每个节点包含两条信息:hash 数值和相对索引。修改后的 MHT 中,根节点包含了树建立的日期和时间,将时间戳集成到持有权证据的一个字段,以确保检索到最近的数据副本,保证了数据的新鲜性。首先,对 MHT 的每个节点进行了修改,增加两个字段以存储两条信息:数据块的 hash 值和节点的相对索引。与一个节点 P 相关联的相对索引,指定了属于 P 子树的叶节点数量。将一个时间戳字段与 MHT 的根节点相关联。该时间戳为串联到根哈希值的 MHT 的根节点创建的时间和日期。针对图 1,此处,$H_R = (H_a \| H_b \| d_t)$,其中,$d_t$ 为树建立的日期和时间。由于对于树中任何数据块进行的任何改动均会更新 H_R,因此通过 H_R,可以反映出最后依次修改的时间和日期,数据新鲜性得以保证。

下面就重要的 3 个算法,即:块标签生成算法、节点搜索算法和文件标签认证算法进行描述,如算法 1 - 算法 3 所示。

算法 1 块签名生成算法

1. procedure GENERATE_BLOCK_SIG

2. 输入:块数量 n、私钥 k、文件块 hash($H(d[i])$)、日期和时间 d_t

输出:块签名集合 ψ,根签名 ρ

3. 对 $G(\forall i \in n, u \in G)$ 的随机元素数组 $h[i]$、$temp[i]$、$roothash$ 进行初始化

4. element_init_$G(h[i]$, pairing)

5. element_init_$G(temp[i]$, pairing)

6. element_init_$G(u$, pairing)

7. element_init_$G(roothash$, pairing)

8. 在 $hash[i]$ 中读取块的 $hash$, $\forall i \in n$

9. 将 $hash[i]$ 转换为一个元素 $h \in G$, $\forall i \in n$

10. $h[i] \leftarrow$element_from_hash($hash[i]$)

11. 生成 $u^{d[i]}$

12. $temp[i] \leftarrow$element_pow_$Z_P(temp[i], u, d[i])$

13. 生成 $h[i].u^{d[i]}$

14. $h[i] \leftarrow$element_mul($h[i], h[i], temp[i]$)

15. 生成块签名 ψ_i

16. $\psi_i \leftarrow$element_pow_$Z_P(\psi_i, h[i], k)$

17. 使用算法 4 生成 MHT

18. $roothash \leftarrow$create_tree($hash$)

19. 将 d_t 串联到 $roothash$

20. $roothash \leftarrow roothash \| d_t$

21. 生成根签名 ρ

22. $\rho \leftarrow$element_pow_($\rho, roothash, k$)

算法 2 节点搜索算法

1. procedure SEARCH_NODE(ROOT, KEY, SIB)

输入:F,MHT 的根(Root),要搜索的节点(key)

输出:i 的认证信息 AAI(key),key 的兄弟节点数量 SIB(key)

2. 初始化左索引和右索引变量:

3. left_index←0; right_index←0; SIB←0

4. **if** key 为 1 **then**

5. 返回(root)

6. **else if** key 为 0 **then**

7. 返回(NULL)

8. **else**

9. left_index = root→left → index;
right_index = root→right→index

10. **if** key≤left_index **then**

11. SEARCH_NODE(root→left, key, SIB)

12. **else**

13. SIB = SIB + left_index

14. key = key - left_index

15. SEARCH_NODE(root→right, key, SIB)

算法 3 文件标签认证

1. **procedure** FILE_TAG_AUTHENTICATION

输入:系统参数 g,公共密钥 η,本地文件 hash h

输出:若文件标签被认证,则恢复 d_1 和 u 并输出 TRUE;否则输出 FALSE

2. 初始化配对:

```
3.    pbc_demo_pairing_init(pairing, count, param)
4.   初始化 G 中的随机元素 temp1,temp2
5.       element_init_G(temp1, pairing);
6.       element_init_G(temp2, pairing);
7.     从服务器读取文件标签 τ,在 τ 上应用配对 τ
8.     pairing_apply(temp2, τ, g, pairing);
9.   在 h 上应用配对
10.      pairing_apply(temp1, h, η, pairing);
11. 验证文件标签
12.   result = element_cmp(temp1, temp2)
13.   if 结果不为 0 then 输出 TRUE
14.        恢复 d_t,恢复 u;
15. else 输出 FALSE
```

2　数据动态操作

数据动态操作的支持对于数据完整性审核协议非常重要。当前大部分协议中不包含这一特征,而 IIMHT 完全支持动态操作,这些动态操作主要包括数据修改、数据插入、数据附加和数据删除程序。文中假设 F 为数据文件,θ 为文件块签名集合,ρ 为存储在服务器的根签名。

2.1　数据修改程序(DMP)

DMP 是 IIMHT 的一个重要功能,其允许 DP 在不下载全部数据块的前提下对选择的文件块进行更新。修改第 i 个文件块时,DP 启动 DMP 如下:

1) DP 首先为新的块生成签名 $\psi' = (H(d[i]' \cdot u^{d[i]'})^k$。

2) DP 生成新的文件标签 $\tau' \leftarrow sig_k(\text{fname} \| n \| u \| d_t)$。该标签被用于确认修改的日期和时间,以确保数据新鲜性。

构建数据修改消息 $(M, i, d[i]', H(d[i]'), \psi', \tau')$ 并将该消息传递给 CSP。此处 M 表示 DMP,i 表示要使用 i' 修改的原文件块。在接收到该消息时,CSP 执行以下替换:

1) 将 d_i 替换为 d';

2) 将 ψ 替换为 ψ';

3) 将 τ 替换为 τ';

4) 将 $H(d[i])$ 替换为 $H(d[i]')$,并生成新的根哈希 hash R'。

CSP 将修改程序的证据发送给 DP 进行验证,即 $P_f = \{\psi, H(d[i]), AI(i), \rho, SIB(i), R'\}$。接收到来自 CSP 的 DMP 的证据后,DP 首先验证 τ。然后,使用 $(H(d[i]), AI(i), \rho)$ 验证根哈希。若验证通过,则 DP 使用 $(H(d[i]'), AI(i)_i)$ 计算新的根,并将新生成的根与 R' 相比较,若比较通过,则 DP 以私钥 k 对 R' 进行签名并由此生成 $\rho' = H(R')^k$,并将其发送到 CSP 进行存储。最后,DP 为新文件块运行默认完整性审核协议。如果结果为 TRUE,则 DP 可以从其本地系统中删除 $\{d[i]', \tau', \rho'\}$。

2.2　数据插入程序(DIP)

若 DP 希望在特定位置处插入一个数据块 d^*,其需要执行以下 DIP:

1) DP 为新文件块生成签名,$\psi^* = (H(d^*) \cdot u^{d^*})^k$;

2) DP 生成一个新的文件标签 $\tau^* \leftarrow sig_k(fname \| n \| u \| d_t)$。

DP 需要构建一个数据插入消息 $(I, V, i, d^*, \psi', \tau')$,并将该消息传递到 CSP,此处 I 表示 DIP。字段"V"表示要插入新数据块的位置,$V \leftarrow A$ 表示第 i 个位置之后插入,$V \leftarrow B$ 表示第 i 个位置之前插入。在接收到插入消息后,CSP 保留 d^* 及其相应的叶节点 $H(d^*)$。这样,CSP 找到 MHT 中的 $H(d[i])$,保留 $AI(i)$ 并插入叶节点 $H(d^*)$。若"V"字段为集合"A",则将 hash 值为 $(H(d[i]) \| H(d^*))$ 的一个内部节点加入原始树;若"V"字段为集合"B",则将 hash 值为 $(H(d^*) \| H(d[i]))$ 的一个内部节点加入原始树,并将其索引设为 2。然后,对第 i 个节点至最高节点(根节点)轨道上的每个节点进行修改。同时,CSP 为再生的 MHT 生成一个新的根节点 R。接着,CSP 向 DP 提交用于插入操作的证据:$P_f = \{\psi, H(d[i]), AI(i), \rho, SIB(i), R'\}$,其中 $AI(i)$ 表示在之前树中 $d[i]$ 的附加信息。收到插入程序的证据时,DP 首先验证 τ。若通过,则使用 $\{AI(i), H(d[i])\}$ 生成根节点,对新生成的根节点进行验证。若验证通过,在 DP 能够通过 $\{AI(i), H(d[i]), H(d^*)\}$ 进一步生成根节点的一个新鲜值,并将其与 R' 相比较,验证 CSP 是否正确完成插入操作。若结果是成功的,则 DP 将 R' 认证为 $(H(R'))^k$,并将其传输到 CSP,用于更新。最后,DP 为新数据块运行默认完整性审核协议。若结果为 TRUE,则 DP 可以从本地存储中擦除 $\{\rho', d^*, \tau'\}$。

2.3　数据删除程序(DDP)

DDP 的前几步与 DIP 相似。数据删除消息的结构为 (D, V, i, τ')。若"V"字段为集合"A",那么从原始树中移除 hash 值为 $H(d[i+1])$ 的一个节点;若"V"字段为集合"B",则从原始树中移除 hash 值为 $H(d[i-1])$ 的一个节点。父节点的相对索引值和 hash 值也被更新(见图 2)。在此之后的 DDP 步骤与 DIP 相似。

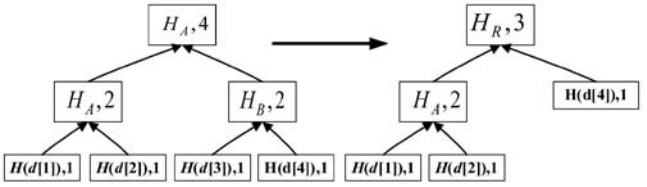

图 2 数据删除操作示例

3 测试结果与分析

为了体现所提大数据审核方法的优势,将文献[6]和文献[8]视为对照组,测试了这三种方法的计算成本,并对本文方法的服务供应商(CSP)不当操作的检测概率进行证明和分析。

3.1 CSP 不当操作的检测概率分析

为了证明所提方法可以降低计算负担,本文对 CSP 不当操作的检测概率进行分析和证明。

本文采样方法是将 DP 的数据文件"F"分割为"n"个数据块,然后从中随机选择"t"个数据块对可能存在的不当操作进行检查。考虑云服务供应商(CSP)从总计"n"个块中修改 x_a 个块。则随机检测到的数据块为损坏块的概率为:$p_0=\frac{x_a}{n}$。

现在,假定在至少一个数据块中检测到不当操作的概率为 P。由此,P 等于 1 减去未检测到不当操作的概率。

在第 1 个块中未检测到不当操作的概率:

$$checked=\frac{n-x_a}{n}=1-\frac{x_a}{n} \tag{2}$$

在第 2 个块中未检测到不当操作的概率:

$$checked=\frac{n-x_a-1}{n-1}=1-\frac{x_a}{n-1} \tag{3}$$

在第 3 个块中未检测到不当操作的概率:

$$checked=\frac{n-x_a-2}{n-2}=1-\frac{x_a}{n-2} \tag{4}$$

在第 t 个块中未检测到不当操作的概率:

$$checked=\frac{n-x_a-t+1}{n-t+1}=1-\frac{x_a}{n-t+1} \tag{5}$$

因此,

$$P=1-\left(1-\frac{x_a}{n}\right)\left(1-\frac{x_a}{n-1}\right)\cdots\left(1-\frac{x_a}{n-t+1}\right)=1-\prod_{i=0}^{t-1}\left(1-\frac{x_a}{n-i}\right) \tag{6}$$

为找到 P 的最小数值,考虑:

$$\left(1-\frac{x_a}{A}\right)\left(1-\frac{x_a}{B}\right)\leqslant\left(1-\frac{x_a}{\left(\frac{A+B}{2}\right)}\right)\quad A,B\geqslant 0 \tag{7}$$

$$1-x_a\left(\frac{1}{A}+\frac{1}{B}\right)+\frac{x_a^2}{AB}\leqslant$$

$$1-\frac{4x_a}{A+B}+\frac{4x_a^2}{(A+B)^2}\cdot\frac{4}{A+B}-\left(\frac{1}{A}+\frac{1}{B}\right)\leqslant$$

$$x_a\left(\frac{4}{(A+B)^2}-\frac{1}{AB}\right)\frac{4AB-(A+B)^2}{(A+B)AB}\leqslant$$

$$x_a\left(\frac{4AB-(A+B)^2}{AB(A+B)^2}\right)\Rightarrow\frac{-(A-B)^2}{(A+B)AB}\leqslant x_a\left(\frac{-(A-B)^2}{AB(A+B)^2}\right) \tag{8}$$

为寻找 P 的最大值,考虑:

$$\left(1-\frac{x_a}{n-i}\right)\geqslant\left(1-\frac{x_a}{n-t+1}\right)\Rightarrow\prod_{i=0}^{t-1}\left(1-\frac{x_a}{n-i}\right)\geqslant\left(1-\frac{x_a}{n-t+1}\right)\Rightarrow\prod_{i=0}^{t-1}\left(1-\frac{x_a}{n-i}\right)\leqslant\left(1-\frac{x_a}{n-t+1}\right)^t \tag{9}$$

由此:$P\leqslant\left(1-\frac{x_a}{n-t+1}\right)^t$。

现在,总结出 CSP 处不当操作的检测概率为:

$$\left(1-\frac{x_a}{n-\frac{t-1}{2}}\right)^t\leqslant P\leqslant\left(1-\frac{x_a}{n-t+1}\right)^t \tag{10}$$

假定 DP 将 1 GB 文件分割为 62 500 个数据块,其中每个数据块大小为 16 KB,从集合 $P_x=\{0.7,0.8,0.9,0.99,0.999\ 99\}$ 中考虑不当检测概率 P。在损坏数据块的不同百分比下,为检测到不当操作,质询消息中所需的块数量为 t。若 CSP 改动了外包数据块总数(n)的 10%,那么第三方审核者(TPA)在质询集合中仅需 68 个随机块,即能够以 0.999 的概率检测到服务器不当操作。表 1 给出了受损数据块占比 90% 时,不同数量的受损数据块下,为检测到不当操作质询消息中所需的块数量。随着损坏块的增加,所需的质询数据块的数量会减少。举例来说,若 CSP 改动了外包块总数(n)的 40%,则第三方审核者 TPA(Third Party Auditor)在质询集中仅需 18 个随机块,即可以 0.999 的概率检测到服务器不当操作。

表 1 检测到不当操作与质询消息中所需的块数量

受损数据块的数量 x_a	质询数据块的数量 t
6 250	22
12 500	11
18 750	9
25 000	7
31 250	5
37 500	4
43 750	3
50 000	3
56 250	2

3.2　综合成本比较分析

针对第三方审核者(TPA),数据动态操作与常规数据完整性审核形式不同,其计算成本也是不同的。为了体现所提大数据审核方法的优势,将文献[6]和文献[8]视为对照组,以测试比较三种方法的计算成本。所用的表示符号如表 2 所示。

表 2　符号和描述

符号	描述
Add(G)	加法任务
Multip(G)	乘法任务
Exp(G)	求幂任务
Match(G)	配对任务

三种方法的计算成本如表 3 所示。本文算法增加了每个节点的字段信息:hash 值和相对索引,使得动态操作的可检索性增强,根节点所包含的信息更多,数据新鲜性得以保障,协议的效率最高,其复杂度为 $O(\lg n)$。文献[6]依赖能够提供基于排名的身份验证的跳跃表来支持动态操作,其复杂度为 $O(n)$,因此,数据块的搜索计算时间很长。文献[8]执行遗传算法对动态数据进行最优分解,分解速度基本正比于数据块数量,其复杂度约为 $O(n)$。

表 3　各算法的综合计算成本比较

算法	服务器	审核者	时间复杂度
文献[6]	tEXP(G) + (2t − 1)Multip(G)	2Match + (t + 2)Exp(G) + (t + 1)Multip(G)	$O(n)$
文献[8]	2tExp(G) + (2t − 2)Multip(G)	3Match + (t + 1)Exp(G) + tMultip(G)	$O(n)$
IIMHT	tExp(G) + (t − 1)Multip(G)	4Match + tExp(G) + tMultip(G)	$O(\lg n)$

本文提出的算法中,对经典 MHT 树进行了改进,采用了索引字段,搜索一个树节点复杂度明显降低了,为 $O(\lg n)$,与二叉搜索树相似。

数据块改动时,文献[6]、文献[8]、经典 MHT 和本文方法的计算成本的比较如图 3 所示。对于大约 1 GB 的文件进行不停更新,数据块数量在 100 至1 000 间,且都是针对动态操作的算法。由表 3 可以看出,本文方法的成本增加速率最低,得益于 MHT 树的改进,即每个节点增加了哈希值和相对索引两条信息,便于动态操作和搜索,使得计算成本降低。相比于 MHT 树,文献[6]使用秩信息定义动态数据的占有模式,文献[8]使用遗传算法对数据进行分解。当数据块改动时,这两种方法的搜索时间都是线性的,复杂度为 $O(n)$,明显低于 MHT 树类型 $O(\lg n)$,而本文的 IIMHT 通过修改节点信息,使得搜索等动态操作更加便利,效率更高。

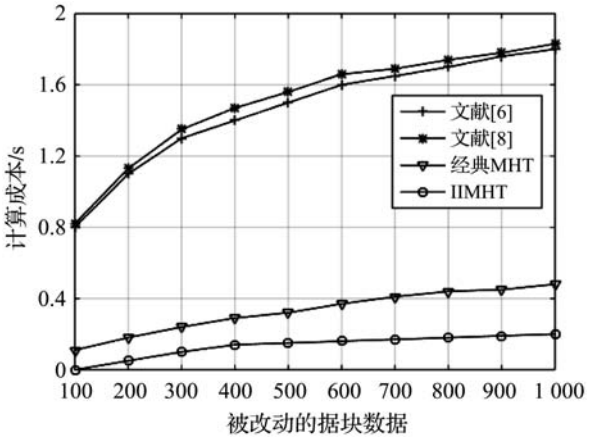

图 3　计算成本与改动数据块数的关系

4　结　语

本文在经典 MHT 和 BLS 签名的基础上,提出一种改进的索引 MHT,用于远程大数据的完整性审计。通过集成的哈希值和索引集降低了节点的搜索复杂度,保证了数据的新鲜性。同时,操作不当的概率表明,所提方法具有较高的安全性。未来将考虑单个远程数据的审核方法,这样可以产生更低开销和成本。同时也将对安全性进行考证。

参　考　文　献

[1] 单莘, 祝智岗, 张龙,等. 基于流处理技术的云计算平台监控方案的设计与实现[J]. 计算机应用与软件, 2016, 33(4): 88 - 90.

[2] 李凌. 云计算服务中数据安全的若干问题研究[D]. 合肥:中国科学技术大学, 2013.

[3] 张云勇, 李素粉, 吴俊,等. 面向云服务提供商的服务选择方法研究[J]. 通信学报, 2012(9): 66 - 76.

[4] Garg N, Bawa S. Comparative analysis of cloud data integrity auditing protocols[M]. Academic Press Ltd. 2016.

[5] Shacham H, Waters B. Compact proofs of retrievability[J]. Journal of Cryptology, 2013, 26(3): 442 - 483.

[6] Erway C C, Papamanthou C, Tamassia R. Dynamic provable data possession[J]. Acm Transactions on Information & System Security, 2015, 17(4): 15 - 26.

[7] 毛瑞雪. 基于数据血缘的审计证据追踪技术研究及应用[D]. 哈尔滨:哈尔滨工程大学, 2012.

[8] 罗剑明, 程良伦. 一个移动云计算中的数据流应用分解与运行框架[J]. 计算机应用研究, 2014, 31(6): 1731 - 1735.

(下转第 333 页)

美”、“安乐”、“汉源”;相关人员:“小堡乡解放村党支部书记王兴陆”、“民兵连长张甫全”、“地灾监测员锁不子、”“河南乡盐井村 1 组组长王天明”、“盐井村副主任胡洪强”;记者:“郑荣”、“李萌”、“王亚军”;相关部门:“县气象局”、“县国土资源局”、“乡镇人民政府”、“国土资源部门”、“汉源县民政局”、“汉源县化工总厂党委”、“安乐乡党委政府”、“顺河乡党委政府”、“省国土资源厅”;时间:“7 月 2 日上午 11 时 30 分”、“19 时许”;刊登日期:“2012 - 07 - 13”;来源:“中国国土资源报”;标题:“奇迹源于对生命的尊重——四川汉源突发泥石流抢险救灾纪实”。

假设各个类的实例的权重为 0.1。

应急案例 A 与应急案例 B 相同的非空类有 8 个,应急案例 A 与应急案例 B 所有类有 10 个,A 与 B 的结构相似度为 $S(A,B)=8/10=0.8$,事件分类的相似度 $Sim(A,B)=0.1\times1=0.1$,相关链接的相似度 $Sim(A,B)=0$,…。A 与 B 的全局相似度为 $Sim(A,B)=0.8\times0.1=0.08$。

应急案例 A 与应急案例 C 相同的非空类有 9 个,应急案例 A 与应急案例 C 所有类有 10 个,A 与 C 的结构相似度为 $S(A,C)=9/10=0.9$,事件分类的相似度 $Sim(A,C)=0$,相关链接的相似度 $Sim(A,C)=0$,…。A 与 C 的全局相似度为 $Sim(A,C)=0$。

实验证明,本体中相同类的实例越相似,两个应急案例就越相似。

6　结　语

本文介绍了本体的有关知识,探讨了应急案例库系统设计,通过示例完成应急案例的本体表示。通过相似度计算来进行应急案例的排序。应急案例本体库的建立是一个不断完善的过程,需要投入更多的精力。接下来将进一步完善本体库,寻找合适的属性权重设置方法,并将本体理论应用于应急预案。

参　考　文　献

[1] 王能干,王坚,凌卫青. 基于本体的突发事件应急服务模型研究[J]. 计算机与现代化,2014(1):192 - 195,200.

[2] 王文俊,杨鹏,董存祥. 应急案例本体模型的研究及应用[J]. 计算机应用,2009,29(5):1437 - 1440,1445.

[3] 何岩新,倪丽萍,曹琳,等. 基于本体的股票主题事件案例推理系统研究[J]. 计算机技术与发展,2016,26(1):33 - 38.

[4] 瞿琼丹. 基于本体的群体性突发事件案例推理研究[D]. 兰州:兰州大学,2012.

[5] 于秀丽,王旭坪,张娜娜. 应急领域本体的构建方法研究[J]. 电子科技大学学报(社科版),2015,17(3):20 - 23,59.

[6] 谢娟娜. 本体技术在知识管理系统中的应用研究[D]. 南京:南京航空航天大学,2007.

[7] 王姣. 本体驱动的信息系统开发中的本体建模研究[D]. 吉林:吉林大学,2005.

[8] 吴珊燕. 基于领域本体的案例推理优化及其应用研究[D]. 上海:华东师范大学,2014.

[9] 刘晓文,胡克勤. 基于本体和 CBR 的电子政务项目审批决策支持系统[J]. 计算机应用,2009,29(3):896 - 899.

[10] 李锋刚. 基于优化案例推理的智能决策技术研究[D]. 合肥:合肥工业大学,2007.

[11] 张巍,张绚,陈俊杰. 基于本体的高血压诊疗系统推理模型研究[J]. 计算机工程与设计,2013,34(11):4016 - 4020.

[12] 陈桂芬,汪江,杨志刚. 基于本体的规则推理和案例推理结合的糖尿病诊疗专家系统研究[J]. 长春大学学报,2016,26(6):19 - 25.

[13] Liu W,Hu G,Li J. Emergency resources demand prediction using case-based reasoning[J]. Safety Science,2012(3):530 - 534.

[14] Cook R L. Case-based reasoning systems in purchasing: applications and development[J]. Journal of Supply Chain Management, 1997, 33(4):32 - 39.

[15] Brandt S C, Morbach J, Miatidis M, et al. An ontology-based approach to knowledge management in design processes[J]. Computers & Chemical Engineering, 2008, 32(1):320 - 342.

[16] Yuan L,Zhang H,Chen J,et al. Design pattern of knowledge base based on ontology knowledge model[J]. Computer Engineering & Applications, 2006, 42(30):65.

(上接第 307 页)

[9] Chen B, Curtmola R, Ateniese G, et al. Remote data checking for network coding-based distributed storage systems[C]//ACM Cloud Computing Security Workshop, Ccsw 2010, Chicago, Il, Usa, October. DBLP, 2010: 31 - 42.

[10] 沈志东, 林晨, 佟强. 面向移动云计算的轻量级数据完整性验证方法[J]. 东北大学学报(自然科学版), 2015, 36(11): 1562 - 1566.

[11] Gao W, Li F, Xu B. An abuse-free optimistic fair exchange protocol based on BLS signature[C]//International Conference on Computational Intelligence and Security. IEEE, 2008: 278 - 282.

[12] 皮若男. 有限域上 d-th CDH 问题的研究[D]. 济南:山东大学, 2016.

[13] 谢飞. 基于 Merkle 散列树的可信云计算信息安全证明方法[J]. 激光杂志, 2016, 37(11): 122 - 127.