基于 DDCT 表的多副本完整性审计方案

杜瑞忠[1,2],石朋亮[1,2],田俊峰[1,2]
(1. 河北大学网络空间安全与计算机学院,河北保定 071002;
2. 河北省高可信信息系统重点实验室,河北保定 071002)

摘 要: 在云存储环境下,云数据采用多副本存储已经成为一种流行的应用. 针对恶意云服务提供商威胁云副本数据安全问题,提出一种基于 DDCT(Dynamic Divide and Conquer Table)表的多副本完整性审计方案. 首先引入 DDCT 表来解决数据动态操作问题,同时表中存储副本数据的块号、版本号和时间戳等信息;接下来为抵制恶意云服务商攻击,设计一种基于时间戳的副本数据签名认证算法;其次提出了包括区块头和区块体的副本区块概念,区块头存储副本数据基于时间戳识别认证的签名信息,区块体存放加密的副本数据;最后委托第三方审计机构采用基于副本时间戳的签名认证算法来审计云端多副本数据的完整性. 通过安全性分析和实验对比,本方案不仅有效的防范恶意存储节点之间的攻击,而且还能防止多副本数据泄露给第三方审计机构.

关键词: 云存储; 完整性; 多副本; 时间戳; 数据加密; 副本区块; 签名算法
中图分类号: TP-393. 2 **文献标识码:** A **文章编号:** 0372-2112 (2020)01-0164-08
电子学报 URL: http://www. ejournal. org. cn **DOI:** 10. 3969/j. issn. 0372-2112. 2020. 01. 020

Multi-copy Integrity Audit Scheme Based on DDCT Table

DU Rui-zhong[1,2], SHI Peng-liang[1,2], TIAN Jun-feng[1,2]
(1. *Cyberspace Security and Computer, Hebei University, Baoding, Hebei* 071002, *China*;
2. *Key Lab on High Trusted Information System in Hebei Province, Baoding, Hebei* 071002, *China*)

Abstract: In the cloud storage environment, the multiple copies are more popular. However, aiming at the problems of data dynamic operation and malicious cloud service provider attacks encountered in multi-copy data integrity audit, a multi-copy integrity audit scheme based on dynamic divide and conquer table (DDCT) is proposed. Firstly, the dynamic divide and conquer table is introduced to solve the problem of dynamic data operation, and the block number, version number and timestamp of the copy data are stored in the table. In order to resist the malicious cloud service provider attacks, a time-based replica data signature authentication algorithm is designed. Secondly, it proposes the concept of replica block including block header and block body. The block header stores the authenticated signature information which is based on timestamp, and the block body stores the encrypted data. Finally, the third-party auditing agency uses a replica timestamp-based signature authentication algorithm to audit the integrity of the multi-copy data. Through security analysis and experimental comparison, this solution protects data information from third-party auditors while effectively preventing malicious cloud service provider attacks.

Key words: cloud storage; integrity; multiple copies; timestamp; data encryption; copy block; signature algorithm

1 引言

目前,云存储[1]凭借着其海量存储、资源共享和按需付费的特点迎来广大的市场,越来越多的人或者企业把数据存储到云端. 但是,用户在享受云服务带来便利的同时也面临着数据安全问题. 在云存储环境下,用户失去对数据副本的控制权,不可信的云服务商为了商业利益,会把使用频率少的副本数据删除. 因此为保证用户数据的安全性,有必要对多副本数据进行完整性审计[2,3].

文献[4,5]提出利用同态函数生成标签检测云端数据,无需下载完整的数据到客户端就可以验证数据

收稿日期:2018-12-19;修回日期:2019-04-23;责任编辑:覃怀银
基金项目:国家自然科学基金(No. 61572170);河北省自然科学基金(No. F2018201153, No. 2016205023)

持有性. 为支持云端数据动态操作,文献[6~8]通过引入可标记哈希树结构提出一种支持动态操作的数据完整性认证模型. Yang 等人[9]设计一种新的数据索引表来支持数据动态操作的同时验证数据完整性. 为进一步提高效率,M 等人[10]在文献[9]的基础上,采用分而治之的思想,将整个大的数据索引表划分为许多分表,以此减少动态操作时数据块的移动时间消耗.

为了应对目前云存储环境下多副本存储的情况,Zhang 等人[11]提出适用于多副本数据完整性检测模型,可以验证云端副本数据的完整性. 为确定副本存储数量的正确性,付艳艳等人[12]以多副本数据可恢复性为目标,借助多叉树定位损坏数据的位置并进行恢复. Cha 等人[13]利用多分支树数据结构不仅能提高副本数据块签名的效率,而且还能帮助数据更新操作.

为抵制恶意云服务器的攻击,Kang 等人[14]利用有关身份密码学技术提出一种基于身份的公共审计协议,用于优化云数据完整性审计结构、隐私保护和有效的聚合验证. Shen 等人[15]提出一种基于身份的数据完整性审计方案,通过隐藏的敏感信息实现数据共享. 为了防止身份泄露,Li 等人[16]提出一种基于模糊身份的数据完整性审计方案,在审计协议中用户身份可以视为一组描述性属性,同时与私钥绑定,有效降低了密钥管理的复杂性. Surmila 等人[17]提出一种同时支持隐私保护和公开审计的云数据审计方案,其中基于 Boneh 和 Boyen 的签名机制使第三方审核机构在进行用户完整性审计的同时保护客户的隐私数据.

因此,本文针对云存储环境下多副本安全性问题,提出一种基于 DDCT 表的多副本完整性审计方案(Dynamic Multiple Copies Integrity Audit Scheme, DMCIA). 首先为解决数据动态操作,采用 DDCT 表记录副本数据块的块号、版本号、时间戳等信息,同时借助 DDCT 表来实现数据块的修改、插入和删除操作;然后为了抵制恶意云服务商的攻击,提出了副本区块概念,同时设计一种基于时间戳的多副本数据签名认证算法,并通过敌手游戏对算法进行安全性分析. 最后实验仿真测试表明本方案与对比方案[7,9,10]相比,在保护副本数据安全性的同时提升了审计效率.

2　预备知识

2.1　双线性知识

设 p 是素数,G_T和 Gv 是阶为 p 的乘法循环群,通常称映射 $e: G_T \times G_T \rightarrow G_V$ 为一个双线性对,e 满足以下的三个性质.

(1)双线性:对于任意 $\delta, \xi \in Z_p$ 和 $\chi, \gamma \in G_T$,都有 $e(\chi^{\delta}, \gamma^{\xi}) = e(\chi, \gamma)^{\delta\xi}$.

(2)非退化性:存在 $\chi, \gamma \in G_T$,使 $e(\chi, \gamma) \neq 1_{G_v}$.

(3)可计算性:对任意的 $\chi \in G_T, \gamma \in G_V$,存在有效的算法计算 $e(\chi, \gamma)$ 的值.

2.2　符号及意义

表 1　符号与意义

符号	表示意义
g	群的基本生成元
a, η	大整数
P_s	标签私钥
ssk	文件加密私钥
spk	文件加密公钥
F	文件
t	副本数量
C	密文
n	文件分块数
m_{ij}	表示第 i 个副本第 j 个数据块
S	时间戳哈希值
r	时间戳私钥
H_1, H_2, H_3	哈希函数
w	属性集合
T_S	时间戳
I	抽样数据块集合
c	抽样数据块数
v_{ij}	随机数
P_f	数据时间戳证据
Proof	标签证据
m'	待验证数据标签值
$e(.)$	线性映射
σ_{ij}	第 i 个副本第 j 个数据块标签

2.3　CDH 困难问题

假设 G_0、G_1和 G_2是阶数为素数 p 的乘法循环群,映射 $e: G_0 \times G_1 \rightarrow G_2$,存在 $h \in G_0, H \in G_1$ 使得 $e(h, H) \neq 1$,则 e 是非退化,计算 $d_A = \Pr[A(g, H, H^a) = g^a \mid g \leftarrow G_0; H \leftarrow G_1; a \leftarrow Z_{G_1}] < \varepsilon$.

如果对所有的多项式时间 t 内敌手 A 以不可忽略的概率 ε 解决以上问题,则 CDH 问题(Computational Diffie-Hellman Problem)是(t, ε)困难问题,即映射 e 是(t, ε)安全的,当且仅当 CDH 问题是(t, ε)困难问题.

3　方案设计

3.1　整体结构

方案的整体结构主要包括数据拥有者、密钥管理中心、云服务提供商和第三方审计机构,如图 1 所示. 数据拥有者主要是上传数据的同时实现对数据的加密处

理、多副本的生成、数据块标签的生成、审计信息的下达以及接收第三方审计机构返回的审计结果；密钥管理中心的主要任务是给数据拥有者提供数据加密时的公私密钥对；云服务提供商主要是存储数据拥有者上传的数据以及响应第三方审计者的挑战信息；第三方审计机构的作用是基于挑战应答模式，采用随机抽样的方法检测云端多副本数据的完整性.

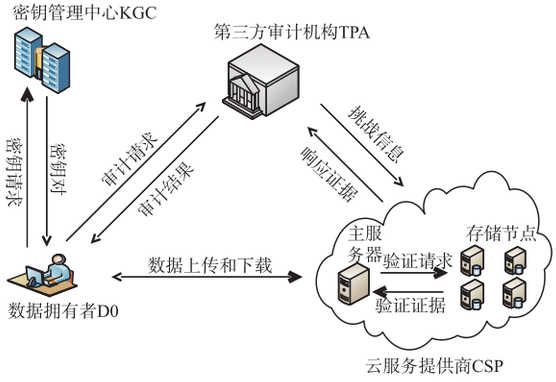

图1 系统整体结构图

3.2 数据区块的设计

针对多副本数据完整性审计情况，本文借鉴区块链思想，设计一种适合多副本数据完整性审计的副本区块，主要将加密的数据和时间戳标签存放在副本区块中，从而形成一个完整的副本数据块. 副本区块主要包括区块头和区块体两部分结构，区块头主要用来存放此副本的时间戳签名信息；区块体用来存放加密的副本数据，如图 2 所示.

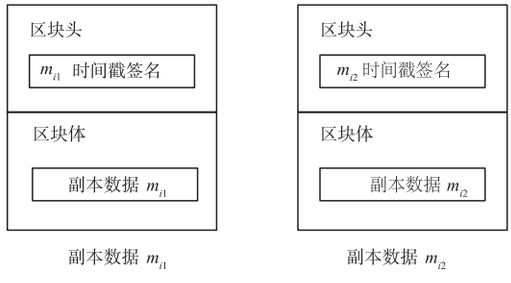

图2 数据区块结构图

3.3 DDCT 表的设计

将数据存储到云端，在节省本地存储空间的同时也失去对数据的控制权. 要想再对数据进行修改、插入、删除的动态操作就更繁琐，且动态操作完成后对存储数据的正确性和完整性进行审计也不容易实施. 面对这种情况下本方案引入 DDCT 表来存储外包到云服务器的数据信息. DDCT 表的设计如下：

(1) DDCT 表的行号，用 N_o 表示；

(2) 数据块在文件分块后的块号，用 B_{no} 表示；

(3) 数据块版本号，用 V_n 表示，初始值为 1，数据块每执行一次动态操作后版本数加 1；

(4) 数据块生成时间 Time，表示副本数据块生成时间的数据为一个二元组，例如 $F_{id}<x,y>$，F_{id} 表示文件的唯一识别标识，x 表示副本号，y 表示此副本数据块生成的时间戳，有多少副本，这列就有多少个的二元组；

(5) DDCT 表中块号的范围，用 range 表示.

3.4 敌手攻击模型

为本方案构建敌手攻击游戏，主要包括敌手 A、挑战者和模拟器 S，主要过程如下：

(1) 挑战者运行初始化算法产生公私钥对，然后将公钥发送敌手 A，私钥保存.

(2) 敌手 A 可以向挑战者进行一系列的询问，主要包括：时间戳询问和标签询问.

(a) 时间戳询问. 敌手询问用任意数据块的时间戳生成的私钥. 挑战者根据时间计算私钥并发送给敌手.

(b) 标签询问. 敌手 A 根据已获时间戳私钥询问数据块的标签. 挑战者根据收到的信息计算数据块的标签并返回给敌手 A.

(3) 敌手 A 通过获得的标签和数据块时间戳私钥，获得数据块的证据. 挑战者充当第三方审计机构，若果获取的证据可以欺骗第三方审计机构，则敌手获胜，否者敌手失败.

4 数据动态操作

4.1 数据更新

运用 DDCT 数据表结构，将存储到表结构中的第 i 个数据块 $f[i]$ 修改为 $f'[i]$，其基本步骤如下(以后不做特殊说明都只选取 DDCT 分表的一部分作为示例，副本生成时间那项数据暂且用符号 t_i 表示，$1\leqslant i\leqslant m$).

(1) 在 DDCT 表中进行搜索，找到对应的序号为 i 的数据块；

(2) 首先运行数据加密算法对数据加密，然后计算加密后数据块的标签 σ_{ij}，之后在从 DDCT 表读出副本相应的信息，分别封装到区块头和区块体，组成副本数据；

(3) 客户端生成一个修改信息集，发给云服务提供商；

(4) 云服务商接受到信息后，用新的数据块替换旧的数据块；数据拥有者然后将数据块版本信息 V_n 进行加 1 操作(如图 3 所示修改序号为 1 的数据块).

4.2 数据插入

数据插入操作是把新数据块插入到数据块 $f[i]$ 之后成为 $f[i+1]$，具体的操作步骤如下：

(1) 在 DDCT 表中搜索到数据块 i；

(2) 将所在分表 DDCT 中 i 位置以后的数据块平移，同时在 i 的位置后面插入一个空位置 $i+1$；

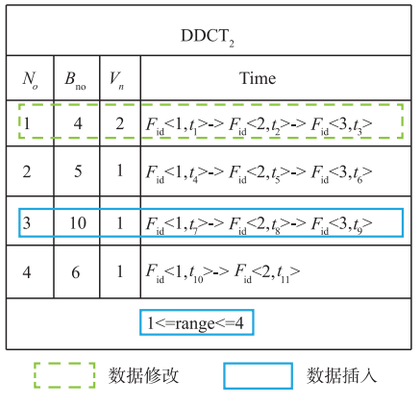

图3　数据更新和插入操作图

(3)DDCT 分表中对应的版本号 V_n 执行加 1 操作;

(4)当前 DDCT 表的最大范围和以后的 DDCT 分表的最小范围和最大范围均加 1;

(5)对文件块 $f[i+1]$ 先进行加密操作,然后产生副本之后用标签生成算法计算其标签值;

(6)将插入的信息集发送给云服务提供商.

4.3　数据删除

数据删除从操作即从数据集中删除数据块 $f[i]$,具体操作步骤如下:

(1)在 DDCT 表中查找到序号为 i 的数据块;

(2)在当前表中将 i 数据块后面的数据块向前移动一个位置;

(3)更改受到影响的数据块序号和 DDCT 分表的范围;

(4)将删除信息发送给云服务商.

5　多副本审计过程

DMICA 方案主要包括初始化阶段和验证阶段,其中初始化阶段包括密钥生成、副本生成和标签生成三个步骤;验证阶段包括挑战信息生成、证据生成和证据验证三个步骤. 初始化阶段流程如图 4 所示.

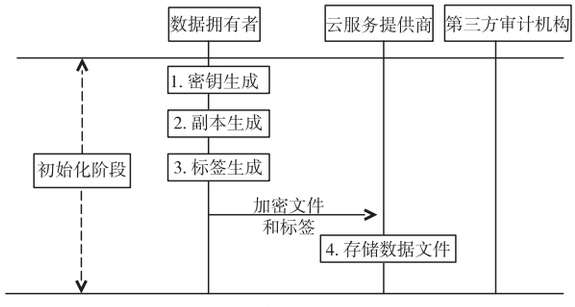

图4　初始化阶段流程图

步骤一　密钥生成

在客户端运行密钥生成算法,生成一对用于数据签名的公私钥. 首先选取双线性乘法循环群中的生成元 g,然后随机选取大整数 α,并计算

$$P_s = g^{\alpha} \tag{1}$$

接下来在密钥管理中心生成用于文件非对称加密的密钥对(ssk,spk). 其中生成的私钥为 sk(α,ssk)需要保密,公钥为 pk(P_s,g,spk)需要公开.

步骤二　副本生成

副本生成在客户端完成. 输入文件 F 和副本数量 t. 运用非对称加密算法为文件 F 加密变为 C,然后根据 t 生成副本密文$\{C_i\}(1\leqslant i\leqslant t)$之后将密文 C 分为 n 块即 $C_i=\{m_{i1},\cdots,m_{ij},\cdots,m_{in}\}$,$(1\leqslant i\leqslant t,1\leqslant j\leqslant n)$从 DDCT 表的 Time 属性列读出相对应数据块的时间戳,计算时间戳私钥 S,计算式为

$$S = H_1(\text{Time}) \tag{2}$$

步骤三　标签生成

为要上传到云端的数据块生成签名标签,首先选取 $\eta\in Z_q^*$,计算

$$r = g^{\eta} \tag{3}$$

然后计算属性集合

$$w = \{F_{\text{id}} \parallel B_{\text{no}} \parallel V_n\} \tag{4}$$

其中 F_{id} 为文件的唯一标识,B_{no} 为数据的块号,V_n 为数据的版本号. 对于每个副本数据 m_{ij} 计算数据块的标签 σ_{ij} 为

$$\sigma_{ij} = S^{m_{ij}} H_2(w \parallel i \parallel j)^{\eta} \tag{5}$$

式中$(1\leqslant i\leqslant t,1\leqslant j\leqslant n)$,将 m_{ij} 存放到数据区块 b_{ij} 的区块体中. 然后计算副本数据区块时间戳的签名

$$T_s = (r \parallel \text{Time}) \tag{6}$$

将 T_S 信息存放到数据区块 b_{ij} 的区块头中. 最后把(b_{ij},σ_{ij},r)发送给云服务商.

验证阶段的主要任务是验证云端存储的副本数据是否是完整的,其主要包括挑战信息生成,证据生成和证据验证三个步骤,主要流程图如图 5 所示.

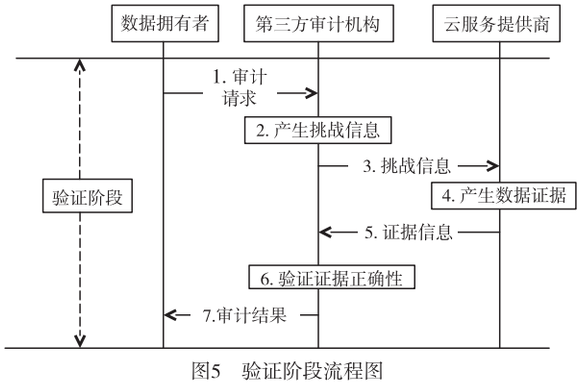

图5　验证阶段流程图

步骤四　挑战信息

根据随机抽样来检测副本的正确性. 首先随机选取 c 个数据块组成集合 I,$I\subset\{(1,t)\times(1,n)\}$,然后随

机选取整数 $v_{ij} \in Z_q^*$,为每个选取的数据块生成一个随机数则有 $Q = \{(i,j,v_{ij})\}$. 接下来选取 $\rho \in Z_q^*$,计算

$$Z = e(H_1(\text{Time}), P_s) \tag{7}$$

$$c_1 = g^{\rho} \tag{8}$$

$$c_2 = Z^{\rho} \tag{9}$$

$$p_f = \text{POK}((\rho): c_1 = g^{\rho} \wedge c_2 = Z^{\rho}) \tag{10}$$

最后生成挑战信息 $\text{chall}(c_1, c_2, Q, p_f)$ 发送给云服务提供商.

步骤五 证据生成

当云服务提供商收到挑战信息后,开始计算

$$Z = e(H_1(\text{Time}), P_s) \tag{11}$$

$$u = \sum_{ij \in I} v_{ij} m_{ij} \tag{12}$$

$$\sigma = \prod_{ij \in I} \sigma_{ij}^{v_{ij}} \tag{13}$$

$$m' = H_3(e(\sigma, c_1) \cdot c_2^{-u}) \tag{14}$$

生成证据信息 proof,$\text{proof}(m', r, T_s)$ 发送给第三方审计机构.

步骤六 证据验证

当第三方审计端收到证据信息 proof 后,开始验证返回的证据信息是否正确. 首先检测基于时间戳的签名信息 T_s 是否正确,如果不正确,审计出错;若正确,则继续验证

$$m' = H_3\left(\prod_{ij \in I} e(H_2(w \| i \| j)^{v_{ij}}, r^{\rho})\right) \tag{15}$$

是否成立,如果成立,则通知数据拥有者,审计正确;反之不成立,第三方审计者通知数据拥有者审计出现问题,及时做出决策.

6 安全性分析

6.1 正确性分析

正确性分析主要是验证上述式(15)是否成立,式(15)左边的数据是云服务提供商根据收到第三方审计者发送的挑战信息后,返回来的数据标签证据,而式(15)右边是第三方审计机构根据数据拥有者提供的副本数据信息计算的数据标签证据,如果云服务提供商没有任何不诚实的行为,即存储了数据拥有者所有的副本数据,那么式(15)左边是等于式(15)右边的. 证明如下所示:

$$\begin{aligned} m' &= H_3(e(\sigma, c_1) \cdot c_2^{-u}) \\ &= H_3\left(\frac{e(\sigma, c_1)}{e(H_1(\text{Time}), P_s)^{\rho \sum_{ij \in I} m_{ij} v_{ij}}}\right) \\ &= H_3\left(\frac{\prod_{ij \in I} e(\sigma_{ij}^{v_{ij}}, c_1)}{\prod_{ij \in I} e(S, c_1)^{m_{ij} v_{ij}}}\right) \\ &= H_3\left(\prod_{ij \in I} e\left(\frac{\sigma_{ij}}{S^{m_{ij}}}, g^{\rho v_{ij}}\right)\right) \\ &= H_3\left(\prod_{ij \in I} e(H_2(w \| i \| j)^{\eta}, g^{\rho v_{ij}})\right) \\ &= H_3\left(\prod_{ij \in I} e(H_2(w \| i \| j)^{v_{ij}}, r^{\rho})\right) \end{aligned}$$

证毕.

6.2 稳健性分析

稳健性分析主要分析基于时间戳的签名机制在多项式时间内是安全的,即云服务商必须存储数据拥有者的文件才能产生有效的证据响应第三方审计者的挑战请求,如果云服务提供商有任意不诚实的表现,则无法产生有效的回答.

定理 在 CDH 困难问题前提下,若本方案 DMCIA 是安全的,则基于时间戳的签名机制在多项式时间内存在不可伪造性.

证明 游戏主要是在敌手 A 和模拟器 S 之间进行的,敌手通过询问模拟器 S,然后根据返回信息猜测、破解方案 DMCIA 中的签名机制.

Game1 游戏 1 主要是进行初始化操作. 首先模拟器 S 运行步骤一,产生需要的密钥对(pk,sk),然后将公钥 pk 发送给敌手 A. 敌手 A 可以随机选择询问模拟器 S 数据块的标签信息或者时间戳签名信息,当模拟器 S 收到敌手 A 的请求后,利用步骤三中的式子计算相应的标签信息返回敌手 A.

Game2 游戏 2 是在游戏 1 的基础进行更加严格的信息交互. 与游戏 1 的区别主要为挑战者维护一个外包数据块列表,其中包括数据块标签信息和时间戳签名信息. 假设敌手询问的信息未在挑战者维护的数据列表中,则游戏终止.

Game3 与游戏 2 基本相同,主要区别是挑战者可以检测全部的挑战应答过程. 假设数据块 b_{ij} 的的签名为 σ_{ij} 和产生的挑战信息为 chall,当挑战者检测到敌手 A 回复证据 $P_f' = \{u', \delta'\}$ 可能存在不诚实表现时,终止游戏. 得到有效的回复证据格式应该为 $p_f = \{u, \delta\}$,其中 $u = \sum_{ij \in I} v_{ij} m_{ij}$,$\sigma = \prod_{ij \in I} \sigma_{ij}^{v_{ij}}$,验证者收到证据消息 $p_f = \{u, \delta\}$ 后,验证等式 $m' = H_3\left(\prod_{ij \in I} e(H_2(w \| i \| j)^{v_{ij}}, r^{\rho})\right)$ 是否成立. 根据上述分析,当 $\delta = \delta'$ 时,$u \neq u'$. 定义 $\Delta u = u' - u$,S 回复敌手 A 的质询,最终敌手 A 回复伪造的证据 $p'_f = \{u', \delta'\}$. 公式 $m' = H_3\left(\prod_{ij \in I} e(H_2(w \| i \| j)^{v_{ij}'}, r^{\rho'})\right)$,则可以得到 $u^{\Delta u} = 1$,即 $u' = u \bmod p$,与假设 $u' \neq u$ 不符. 经过以上分析,敌手 A 无法以可忽略的概率伪造有效的证据欺骗挑战者,则证明本方案基于时间戳的签名算法是安全的.

6.3 私有性安全分析

为了证明方案在借助第三方审计机构时可以有效的保护数据的私有性,引入模拟器 S 和验证者 V. 假设

时间戳签名为 $T_s=(\text{Time}\parallel r)$,$r$、Time 发送给模拟器 S,而 $r=e(g,g)^{\eta}$,其中 η 是数据拥有者自己随机选取,与数据块没有任何联系. 然后验证者 V 生成挑战信息 $\text{chall}(c_1,c_2.\ Q,p_f)$ 给模拟器 S.

当模拟器 S 收到挑战信息 $\text{chall}(c_1,c_2.\ Q,p_f)$ 后,模拟器 S 开始从验证者 V 提取密钥信息 ρ,计算 $c_1=g^{\rho}$,$c_2=e(H_1(\text{Time}),P_s)^{\rho}$,$m'=H_3\left(\prod_{ij\in I}e(H_2(w\parallel i\parallel j)^{v_{ij}},r^{\rho})\right)$. 模拟器 S 返回 (m',r,T_s) 作为挑战的证据信息.

对于每个挑战信息 $\text{chall}(c_1,c_2.\ Q,p_f)$,都有一个唯一的 m'是有效的,因此可以证明模拟器 S 是安全的,即本方案是可以保护数据私有性的.

7　实验仿真

实验采用的是本地的虚拟机加载开源项目 OpenStack 来进行性能测试,其中主要用到 OpenStack 中的 Hadoop 子项目来搭建所需要的实验环境. 部署的 Linux 系统为 Centos6. 7,Hadoop 版本为 Hadoop-2. 6. 0,加密函数为 PBC 提供的函数库,采用 Python 语言编程开发,所有测试的数据都是测试了 100 多次取得平均值.

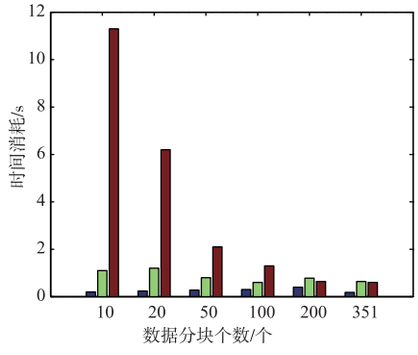

图6　不同分块数对不同文件的影响图

研究测试对不同文件从 1GB 到 100GB,探究文件分块对文件更新操作的影响. 实验结果如图 6,从实验数据上来看对不同数据文件来说,每个 DDCT 分表存储的数据块结构在 350 个左右,数据操作更新时间最佳.

7.1　计算花销对比

本方案 DMCIA 选择和文献[7]DHT-PA、文献[9]IHT-PA 和文献[10]M 进行对比分析结果如表 2 所示,其中 m 代表数据块数量,n 代表数据块的分区,c 表示挑战验证的数据块,s 是 DDCT 表的数量.

7.2　审计性能

实验主要是针对云端数据审计的仿真测试. 包括开始的初始化阶段以及后来更新和挑战审计. 准备了 1GB 的外包资源,将其分为 12500 个数据块,更新的数据块从 100 到 1000 个,通过和文献[7]DHT-PA 方案以及文献[10]IHT-PA 方案一起来对比测试. 第三方审计机构选取随机抽样的数据块的数量 c 由以下式推导可得,假设数据块为 n 个,数据块损坏的概率为 η,则有 $e=n\eta$ 个数据块损坏,检查出损坏数据块的概率为:

表 2　计算开销对比表

	支持动态操作	副本检测	修改花销	插入花销	输出花销
DHT-PA[7]	是	否	$O(c\log^m)$	$O(c\log^m)$	$O(c\log^m)$
IHT-PA[9]	是	否	$O(c)$	$O(m)$	$O(m)$
M[10]	是	否	$O(c)$	$O(m/s)$	$O(m/s)$
DMCIA	是	是	$O(c)$	$O(m/s)$	$O(m/s)$

$$P(e\in\{i\})=1-P(e\notin\{i\})=1-\frac{C_{n(1-\eta)}^{c}}{C_n^c}$$

$$=1-\frac{(n-\eta n)(n-\eta n-1)\cdots(n-\eta n-c+1)}{n(n-1)\cdots(n-c+1)}$$

由于 $\frac{n-\eta n}{n}>\frac{n-\eta n-1}{n-1}>\cdots>\frac{n-\eta n-c+1}{n-c+1}$,则有 $p(e\in\{i\})>1-(\frac{n-\eta n}{n})^c=1-(1-\eta)^c$

当取检测成功的极限为 $\varepsilon=1-(1-\eta)^c$,得最小的数据块数为 $c_{\min}=\ln(1-\varepsilon)/\ln(1-\eta)$,可以知道当 $\varepsilon=99\%$,$\eta=1\%$ 时,c 为 458.

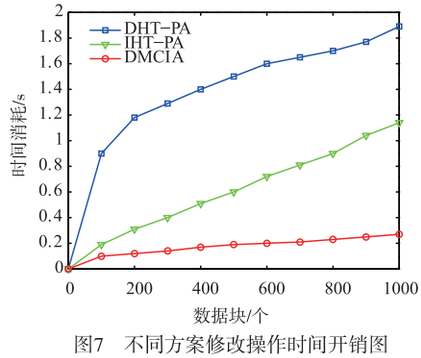

图7　不同方案修改操作时间开销图

如图 7 所示,可以看出 DHT-PA 方法的时间花销最大,其主要原因是此模式更新时必须在 MHT 树中精确找到数据块的位置,同时在计算根节点到本节点路径的新叶子节点的哈希值,导致很大的开销. 其次 IHT-PA 的方案时间花销,主要原因是此模型采用一个 DCT 表,这样查找起来时间花销较多.

图 8 是对副本标签生成消耗的测试,本次选取副本数量从 1 到 20 个,从图中可以得知,随着副本数量的增多,生成副本标签的时间也逐渐增多,大致和副本数量成正比. 而且可以看出本方案与其它方案相比,生成副本标签的消耗明显少于对比文献.

最后图 9 测试的是不同文件数据审计时,生成挑战过程、产生证据和检测挑战证据的时间消耗. 从图中

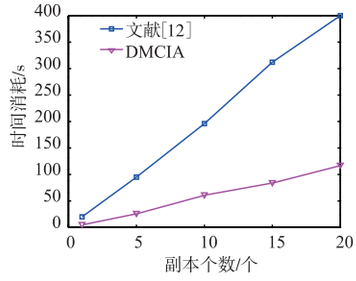

图8 副本标签生成时间消耗图

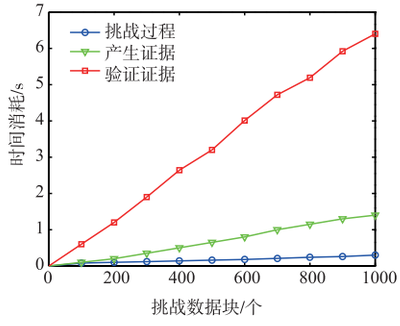

图9 不同文件数目审计时间花销图

可以看出随着抽样数据块的增多,挑战请求过程生成时间、产生证据时间和验证证据时间逐渐增多,大致呈正比趋势.

8 结束语

本文提出一种基于 DDCT 表的多副本数据完整性审计方案,通过引入 DDCT 表结构和副本区块概念,不仅支持全面的动态操作而且有效的验证存储在云端多副本数据的完整性. 然后通过敌手模拟游戏进行安全性分析,设计的基于时间戳的签名机制可以很好抵制恶意云服务提供商的攻击,同时减少数据信息泄露给第三方审计机构. 实验仿真测试显示在分布式文件存储系统中有很好的应用效果.

参考文献

[1] 王国峰,刘川意,潘鹤中. 云计算模式内部威胁综述[J]. 计算机学报,2017,40(02):296 – 316.
Wang G F,Liu C Y ,Pan H Z. Survey on insider to cloud computing[J]. Chinese Journal of Computers, 2017, 40(02):296 – 316. (in Chinese)

[2] 田俊峰,李天乐. 基于 TPA 云联盟的数据完整性验证模型[J]. 通信学报,2018,39(08):113 – 124.
Tian J F,Li T L. Data integrity verification based on model cloud federation of TPA[J]. Journal on Communications, 2018,39(08):113 – 124. (in Chinese)

[3] Feng B,Ma X,Guo C,et al. An efficient protocol with bi-directional verification for storage security in cloud computing[J]. IEEE Access,2017,99(4):7899 – 7911.

[4] Atenise G,Burns R. ,Curtmol R. ,et al. Provable data possession at untrusted stores[J]. ACM Conference on Computer and Communications Security, 2007, 14(1): 598 – 609.

[5] Fun T S, Samsudin A, Zaaba Z F. Enhanced security for public cloud storage with honey encryption[J]. Advanced Science Letters,2017,23(5):4232 – 4235.

[6] Wang Q, Wang C, et al. Enabling public verifiability and data dynamics for storage security in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2011,22(5):847 – 859.

[7] Wang C, Chow S S, WANG Q, et al. Privacy-preserving public auditing for secure cloud storage[J]. IEEE Transactions on Computers,2013,62(2):362 – 375.

[8] Tian H,Chen Z ,Chang C,et al. Enabling public auditability for operation behaviors in cloud storage[J]. Soft Computing,2016,21(8):2175 – 2187.

[9] Yang K, Jia X. An efficient and secure dynamic auditing protocol for data storage in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems,2013,24(9):1717 – 1726.

[10] Sookhak M, Gani A, et al. Dynamic remote data auditing for securing big data storage in cloud computing[J]. Information Sciences,2017,380(3):101 – 116.

[11] Zhou Y, Ni J, Tao X. Provable multiple replication data possession with full dynamics for secure cloud[J]. Concurrency and Computation,2015,28(4),1164 – 1173.

[12] 付艳艳,张敏,陈开渠,冯登国. 面向云存储的多副本文件完整性验证方案[J]. 计算机研究与发展,2014,51(07):1410 – 1416.
Fu Y Y,Zhang M,Chen K Q,et al. Proofs of data possession of multiple-copies[J]. Journal of Computer Research and Development,2014,51(07):1410 – 1416. (in Chinese).

[13] Cha Y X,Luo S S,Bian J C,Li W. Multiuser and multiple-replica provable data possession scheme based on multi-branch authentication tree[J]. Journal on Communications,2015,36(11):80 – 91.

[14] Yu Y, Ni J, Man H A, et al. Improved security of a dynamic remote data possession checking protocol for cloud storage[J]. Expert Systems with Applications, 2014, 41(1):7789 – 7796.

[15] Shen W T,Qin J,Yu J,Hao R,Hu J K:Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage [J]. IEEE Transactions Information Forensics and Security,2019,14(2):331 – 346.

[16] Li Y N,Yu Y,Min G Y,Susilo W,Ni J B. Fuzzy identi-

ty-based data integrity auditing for reliable cloud storage systems [J]. IEEE Transactions Dependable Security Computer,2019,16(1):72 -83.

[17] Surmila T, Dilip K. Privacy preserving and public auditable integrity checking on dynamic cloud data[J]. Network Security,2019,21(2):221 -229.

作者简介

杜瑞忠　男,1975 年 10 月出生,河北献县人. 博士,教授,硕士生导师,主要研究方向为可信计算与信息安全.
E-mail:drzh@ hbu. edu. cn

石朋亮　男,1992 年 10 月出生,河北唐县人. 河北大学硕士生,主要研究方向为分布式计算、云存储安全.
E-mail:782764152@ qq. com

田俊峰　男,1965 年 1 月出生,河北保定人. 博士,教授,博士生导师,主要研究方向为分布式计算与信息安全.