EMCA:一种新的多云存储中高效的多副本审计方案

胡雨晴[1,2],金　瑜[1,2]
(1. 武汉科技大学 计算机科学与技术学院,湖北 武汉 430065;
2. 湖北省智能信息处理与实时工业重点实验室,湖北 武汉 430065)

摘　要:云数据审计是一项允许数据所有者在不下载数据的情况下检查数据是否在远程云服务器上完整存储的技术。目前,云数据审计可分为单副本审计和多副本审计,针对多副本审计方案中计算开销大以及无法审计多云存储数据的不足,提出了一种在多云情景下高效的多副本云数据审计方案。所提方案使用较轻量级的模幂加密技术代替传统上计算成本较高的双线性配对技术,以降低在审计过程中的计算开销,提高了方案的效率。为每个副本数据块生成一个同态验证标签(HVT)作为元数据,当用户发出审计要求,每个云服务提供商生成对应的证据,由云服务器管理者聚合证据,实现了在多云存储中的审计。安全分析表明,所提方案可抵挡恶意云服务提供商的伪造攻击,以及恶意云服务提供商与第三方审计者的合谋攻击。性能分析表明,所提方案在计算开销上低于现有多副本审计方案,实现了高效的多云存储多副本审计。

关键词:云存储;数据审计;模幂加密;多副本存储;多云存储
中图分类号:TP309.2　　**文献标识码**:A　　**文章编号**:1673-629X(2024)03-0015-07
doi:10.3969/j.issn.1673-629X.2024.03.003

EMCA:A New Efficient Multi-replica Auditing Scheme in Multi-cloud Storage

HU Yu-qing[1,2],JIN Yu[1,2]
(1. Dept. of Computer Science & Technology,Wuhan University of Science & Technology,Wuhan 430065,China;
2. Hubei Provincial Key Laboratory of Intelligent Information Processing & Real-time Industry,Wuhan 430065,China)

Abstract:Cloud data auditing is a technology that allows data owners to check whether data is stored intact on remote cloud servers without downloading the data. At present,cloud data auditing can be divided into single-replica auditing and multi-replica auditing. Aiming at the shortcomings of high computational overhead and inability to audit multi-cloud storage data in the multi-replica audit scheme,an efficient multi-replica cloud data audit scheme in multi-cloud scenarios is proposed. The proposed scheme uses a lightweight modular exponentiation encryption technique to replace the traditional bilinear pairing technique with high computational cost,so as to reduce the computational overhead in the audit process and improve the efficiency of the scheme. A homomorphic verification tag (HVT) is generated for each copy data block as metadata. When the user issues an audit request,each cloud service provider generates corresponding evidence,and the cloud server manager aggregates evidence to realize auditing in multi-cloud storage. The security analysis shows that the proposed scheme can withstand the forgery attacks of malicious cloud service providers and the collusion attacks of malicious cloud service providers and third-party auditors. The performance analysis shows that the computational overhead of the proposed scheme is lower than that of the existing multi-replica audit scheme,and the efficient multi-replica audit of multi-cloud storage is realized.

Key words:cloud storage;data audit;modular power encryption;multi-replica storage;multi-cloud storage

0　引　言

近年来,云存储[1]得到了广泛应用,它允许用户将其海量本地数据,以低廉的价格外包给远程存储服务器。然而,云存储也存在缺点:一旦用户将其数据外包

收稿日期:2023-06-04　　**修回日期**:2023-10-08
基金项目:国家自然科学基金资助项目(61802286)
作者简介:胡雨晴(1999-),女,硕士,研究方向为云计算和信息安全;通信作者:金　瑜(1973-),女,博士,副教授,CCF 会员(14140M),研究方向为云计算、对等计算和信任模型。

给不可靠的远程存储服务器,且没有完整的本地备份,用户的数据就会面临许多安全威胁,例如,远程存储服务器会因为自然或人为事故、黑客攻击丢失数据,甚至会为了减少维护费用,故意删除用户很少使用的数据等等。学术界和工业界已经为解决这个问题付出了大量努力。云数据审计也称为可证明数据占有(Provable Data Possession, PDP)[2],是数据所有者在不下载数据的情况下检查数据是否在远程云服务器上正确存储的一项重要技术[3]。

尽管 PDP 可以帮助数据所有者检查数据完整性,但是一旦数据被破坏,数据所有者也会永远丢失数据。为了提高数据的持久性和可用性,数据所有者可以生成多个副本,如果一个副本被篡改,数据所有者可以利用其他副本恢复数据。然而,大多数现有方案只考虑检查单个数据的完整性[4-5],对于多个副本,必须运行 N 次以检查 N 个副本的完整性,这种方案的效率很低。为了克服这个问题,已有学者提出了一些用于多副本的 PDP 方案[6-7],而上述方案都只考虑了一个简单的情况,即所有副本都存储在一个云存储服务器上,一旦服务器出现故障,用户仍然面临数据丢失的风险。且传统的仅限于单一提供商数据中心的云基础设施正在不断发展,多云是下一代云系统的趋势[8]。因此,设计一种高效的多副本、多云存储服务器 PDP 方案是很有意义的。

根据上述多副本方案存在的问题,该文提出了 EMCA(Efficient Multi-replica and multi-Cloud Audit),一种新的高效的多副本、多云数据审计方案。EMCA 具有以下特性:(1)为多云存储服务提出了一个安全的数据审计方案,使用第三方审计者(Third Party Auditor, TPA)协助审计,实现了多云情景下的数据完整性验证。(2)能够一次验证不同云服务器上的所有抽样副本,采用模幂加密技术[9]计算审计结果,实现了高效的多副本批量数据审计。

1 相关工作

1.1 单副本数据审计

Ateniese 等人[2]首次提出了 PDP 的概念,该方案利用基于 RSA(Rivest-Shamir-Adleman)的同态验证标签(Homomorphic Verifiable Tag, HVT),将生成的标签聚合为单个值,大大减少了审计过程中的存储和通信开销。然而,该方案仅适用于验证静态文件。Erway 等人[10]扩展了 PDP 模型,并首次提出了一种完全动态的 PDP 解决方案,称为 DPDP,然而,该方案的执行效率仍然存在疑问。Wang 等人[11]设计了另一种动态 PDP 方法,使用 Merkle 哈希树(Merkle Hash Tree, MHT)支持数据更新。此外,Zhu 等人[12]使用索引哈希表(Index Hash Table, IHT)实现了全数据动态的公共审计,大大降低了计算和通信成本。

为了有效支持用户撤销,Wang 等人[13]提出了 Panda,通过使用代理重签名技术,半可信云服务提供商可以将被撤销用户生成的签名转移到目标用户私钥下。为了实现用户身份的隐私性,Wang 等人[14]提出了第一个针对共享数据的隐私保护公共审计方案,称为 Oruta。此方案将环签名和 HVT 相结合,对被质询块的签名进行聚合,实现了签名者身份保密的目的。此外,Shen 等人[15]提出了一种基于身份加密的数据共享方案,简化了复杂的证书管理。

1.2 多副本数据审计

Curtmola 等人[16]提出了一种基于 RSA 签名的多副本审计方案,这是首次尝试创建多个副本并对其进行检查的方案。通过先加密然后随机化来生成文件的多个副本,已实现副本的可区分性以及数据隐私。针对证书管理开销,Zhou 等人[17]提出了一个无证书的多副本动态完整性审计方案,通过改进经典 MHT 实现副本的批量更新,同时提供可变副本数存储。然而,该方案并不支持多云环境。为了解决这个问题,不少学者提出了多云多副本的审计方案[18-19]。Yang 等人[20]提出使用区块链中随机数的不可预测性来抵制恶意审计员,使用动态哈希表实现多云多副本的审计方案。然而,上述方案[18-20]均采用计算复杂的双线性配对完成完整性验证,导致计算成本较高。

基于此,设计一种在多云环境中高效的多副本数据审计方案是很有意义的。该文通过使用模幂加密技术来实现审计,其中最复杂的运算是模幂运算,这只会消耗很少的计算成本,同时,可一次验证不同云服务器上的所有副本,实现高效的多云多副本审计。

2 具体方案

2.1 预备知识

2.1.1 同态验证标签

数据块的同态可验证标签是一种不可伪造的验证元数据,它具有同态的特征,这是由 Ateniese 等人首次引入的。HVT 的特性如下:

(1)无块验证。通过使用 HVT,用户可以在不访问数据块的情况下验证 CSP 生成的完整性证明。

(2)同态标签。对于任意两个数据块 b_i 和 b_j 以及相应的同态验证标签 Tag(b_i)和 Tag(b_j),($b_i + b_j$)的 HVT 可以通过 Tag(b_i)和 Tag(b_j)生成。

该文使用以下基于 RSA 的安全哈希函数来构造 HVT。$N = pq$ 是一个公共 RSA 模,其中 $p = 2p' + 1$ 和 $q = 2q' + 1$ 是两个足够大的秘密素数。让 QR_N 为模 N 的二次剩余集,g 为 QR_N 的生成元。因此,数据块 b_i

的基于 RSA 的 HVT 可以定义为:

$$\mathrm{Tag}(b_i) = g^{b_i} \bmod N \tag{1}$$

HVT 的同态特性由以下等式给出:

$$\mathrm{Tag}(b_i + b_j) = g^{b_i+b_j} \bmod N = (g^{b_i} \bmod N) * (g^{b_j} \bmod N) = \mathrm{Tag}(b_i) * \mathrm{Tag}(b_j) \tag{2}$$

另外,如果 r 和 b_i 是两个足够大的整数,可以构造以下等式,该等式将用于后续的完整性验证:

$$\mathrm{Tag}(b_i)^s = (g^{b_i})^s \bmod N = g^{sb_i} \bmod N = (g^s)^{b_i} \bmod N \tag{3}$$

2.1.2　副本记录表

如表 1 所示,副本记录表[20]记录了文件副本与物理存储位置的映射关系。表项为文件号、副本号和 $\mathrm{CSP_{ID}}$,根据文件号和副本号可锁定唯一副本,根据 $\mathrm{CSP_{ID}}$ 可确定该副本的存储位置。其中,x_i 表示存储 ID(F)第 i 个副本的 CSP 的 ID。

表 1　副本记录表

文件号	副本号	$\mathrm{CSP_{ID}}$
ID(F)	1	CSP_{x_1}
ID(F)	2	CSP_{x_2}
…	…	…
ID(F)	i	CSP_{x_i}
…	…	…
ID(F)	r	CSP_{x_r}

2.2　系统模型

多云多副本的审计系统模型如图 1 所示,系统中包含四个实体:用户(User)、云服务提供商(Cloud Service Provider,CSP)、云服务管理者(Cloud Service Organizer,CSO)和第三方审计者(Third Party Auditor,TPA)。

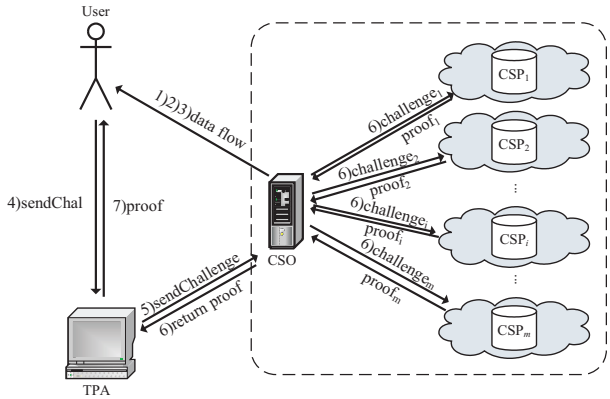

图 1　系统结构

用户(User)可以是个人或企业,它将数据存储在多个 CSP 上,还将检查外包数据的完整性。

云服务提供商(CSP)为用户提供网络存储和计算服务,但 CSP 可能会丢失或损坏用户数据。

云服务管理者(CSO)是一家特殊的云服务提供商,负责在多云环境下管理多个 CSP。

第三方审计者(TPA)在用户的授权下管理或监控外包数据。

方案的大致过程如下:

(1)用户对数据进行初始化,将文件分块,并生成副本,将副本发送至 CSP 存储。

(2)用户生成数据标签并发送至 TPA 存储。

(3)CSP 验证收到的数据是否正确。

(4)用户生成随机数,发送给 TPA。

(5)TPA 生成挑战信息并发送给 CSO。

(6)CSO 将挑战信息分派给 CSP,CSP_i 生成证据 Proof_i,CSO 将证据聚合成 $\mathrm{Proof_{CSP}}$,并发送给 TPA。

(7)TPA 收到证据后,根据存储的标签计算 $\mathrm{Proof_{TPA}}$。TPA 将 $\mathrm{Proof_{CSP}}$ 和 $\mathrm{Proof_{TPA}}$ 一同发送给用户。

(8)用户利用第四步生成的随机数审计证据。

2.3　安全性假设

假设用户和 CSO 是可信的。

假设 CSP 是半可信的,恶意或粗心大意的 CSP 可能会丢失数据,也可能会删除用户很少使用的数据以减轻其存储负担。

假设 TPA 是半可信的。TPA 可能通过与云服务器串通向用户隐瞒数据损坏事件。

2.4　关键过程

多云多副本的审计系统方案由三部分组成:初始化阶段、设置阶段和公开审计阶段。一些必要的符号定义如表 2 所示。

表 2　相关符号定义

符号	定义
p	足够大的安全的素数
q	足够大的安全的素数
g	QR_N 的生成元
ssk	用户私钥
N	模,$N = pq$
n	文件分块数
k	对称加密密钥
r	副本数
s	用户生成的秘密随机数
e	审计的数据块数,$e < n$
m	CSP 个数

文中的一些密码变换如下(以 k 表示密钥长度):

$E(\cdot):\{0,1\}^k \times \{0,1\}^* -> \{0,1\}^k$ 为对称加密算法,用户对文件进行加密,如 AES。

$H(\cdot):\{0,1\}^* ->\mathrm{Zp}$ 为普通无密钥 Hash 函数,如 SHA-256。

2.4.1 初始化阶段

选择两个足够大的素数 p 和 q 来生成 RSA 模 $N = pq$，并生成 QR_N 的生成元 g，将 g，N 公开。

用户发送 ID 给密钥生成中心，密钥生成中心选择两个随机数 $\alpha \in Z_p$，$\beta \in Z_p$，计算用户私钥(ssk)并将其发送给用户保存，ssk 计算如式 4 所示：

$$ssk = \alpha + \beta H(ID) \tag{4}$$

2.4.2 设置阶段

在设置阶段，用户将文件数据分块，并生成各数据块副本及标签，将副本发送给 CSP 存储，标签发送至 TPA 中存储，CSP 确认数据一致无误后，用户可删除本地文件。设置阶段的具体流程如下：

(1) ReplicaGen：为了减少验证开销并适应多云存储的分布式环境，用户将文件 F 分成 n 块，$F = \{b_1, b_2, \cdots, b_n\}$，$b_i$ 可以看作是一个足够大的整数。为得到 r 个副本，用户使用对称加密算法对原始数据块加密 $b_{ij} = E_k(i \| j \| b_j)$，其中 $i = 1, 2, \cdots, r, j = 1, 2, \cdots, n$，第 i 个副本 $F_i = \{b_{i1}, b_{i2}, \cdots, b_{in}\}$。同样，用户使用密钥 k 可对 b_{ij} 进行解密得到原始数据块 b_j。用户对加密数据块计算 R 如式 5 所示，随后用户将 r 个副本文件 $\{F_1, F_2, \cdots, F_r\}$ 以及 R 发送给 CSP 存储。

$$R = \prod_{i=1}^{r} \prod_{j=1}^{n} g^{b_{ij}} \bmod N \tag{5}$$

(2) TagGen：对于 $r \times n$ 个副本数据块，用户计算每个数据块的标签 Tag_{ij} 如式 6 所示，其中 $i = 1, 2, \cdots, r, j = 1, 2, \cdots, n$。$Tag_i = \{Tag_{i1}, Tag_{i2}, \cdots, Tag_{in}\}$，用户将 r 个副本标签 $\{Tag_1, Tag_2, \cdots, Tag_r\}$ 发送给 TPA 存储。

$$Tag_{ij} = g^{b_{ij} * ssk} \bmod N \tag{6}$$

(3) CSP confirm data：CSP 收到用户发送的数据后，计算以下等式是否成立：

$$R = \prod_{i=1}^{r} \prod_{j=1}^{n} g^{b_{ij}'} \bmod N \tag{7}$$

若相等，则确认数据无误；否则，它将表明数据存在问题。此时，CSP 将拒绝确认数据，并终止存储服务。此时用户并未删除本地文件，所以用户不会遭受任何损失。如果 CSP 确认所有数据无误，服务顺利进行，用户将删除本地文件。同时，CSO 在副本记录表中记录副本的存储地址。

2.4.3 公开审计阶段

在公开审计阶段，用户检查多云存储服务，以确保外包数据安全存储在 CSP 中。为了减少开销，用户将批量审计部分副本数据块，以概率性检测检查整个外包数据的完整性。具体审计流程如下：

(1) ChalGen：用户秘密生成一个足够大的随机数 s，计算挑战 chal 并将其发送给 TPA。挑战 chal 的计算如式 8 所示：

$$chal = g^s \bmod N \tag{8}$$

(2) SendChallenge：TPA 收到用户发送过来的 chal 后，生成随机序列 I 和二维数组 C：

$$I = \{1, 3, \cdots, i\}$$
$$C = \begin{pmatrix} c_{11} & \cdots & c_{1e} \\ \vdots & \ddots & \vdots \\ c_{r1} & \cdots & c_{re} \end{pmatrix} \tag{9}$$

I 表示 e 个待审计的数据块编号序列，C 表示对应 e 个待审计数据块及 r 个副本的 $r \times e$ 个系数。TPA 将 $\{chal, I, C\}$ 发送给 CSO，等待 CSO 返回证据。

(3) $Proof_{CSP}Gen$：CSO 分派挑战，各 CSP 计算证据返回，CSO 聚合证据返回。具体步骤如下：

(a) CSO 收到 TPA 发送过来的审计请求后，查询副本记录表找出存储了待审计副本的 CSP，将对应的 $\{I, C_t, chal\}$ 分别发送给对应 CSP。

(b) CSP 收到 CSO 发送的 $I = \{1, 3, \cdots, j\}$，$C_t = \{c_{t1}, c_{t2}, \cdots, c_{te}\}$ 与 chal 后，计算其存储的副本 F_t 的完整性证明 $proof_t$ 并将其发送给 CSO，$proof_t$ 的计算如式 10 所示。

$$proof_t = \prod_{i=1}^{e} chal^{c_{ti} b_{ti}} \bmod N \tag{10}$$

(c) 所有 CSP 返回证据后，CSO 聚合证据计算 $proof_{CSP}$ 如式 11 所示，其中 r 为副本个数，CSO 将 Hash($proof_{CSP}$)返回给 TPA。

$$proof_{CSP} = \prod_{i=1}^{r} proof_i \bmod N \tag{11}$$

$Proof_{CSP}Gen$ 算法流程如下所示：

算法 1：$Proof_{CSP}Gen()$

输入：chal，I，C，$\{F_1, F_2, \cdots, F_r\}$

输出：Hash($proof_{CSP}$)

(1) CSO 从副本记录表中找到存储了待审计副本的 r 个 CSP_{id}

(2) for t in range(1, r)

(3) $CSP_t \leftarrow \{I, C_t, chal\}$

(4) for i in range(1, e)

(5) $proof_i \leftarrow chal^{c_{ti} b_{ti}} \bmod N$

(6) $proof_{csp} \leftarrow \prod_{i=1}^{r} proof_i \bmod N$

(7) return Hash($proof_{CSP}$)

(4) $Proof_{TPA}Gen$：在 CSO 计算审计证据时，TPA 同时依据存储的标签以及待审计数据块编号序列 I，和系数数组 C 计算 $proof_{TPA}$ 如式 12 所示，TPA 返回证据 $\{Hash(proof_{CSP}), proof_{TPA}\}$ 给 User。

$$proof_{TPA} = \prod_{j=1}^{e} \prod_{i=1}^{r} Tag_{il_j}^{c_{ij}} \bmod N \tag{12}$$

(5) Audit：用户收到 TPA 返回的证据后，根据用户秘密保存的随机数 s 以及 ssk 计算等式 13 是否成

立。若成立,则用户认为 CSP 完好地存储了数据;否则,认为 TPA 或 CSP 是恶意的,存储在 CSP 中的数据可能遭到了损坏。

$$\mathrm{Hash}(\mathrm{proof}_{\mathrm{CSP}}^{\mathrm{ssk}} \bmod N) = \mathrm{Hash}(\mathrm{proof}_{\mathrm{TPA}}^{s} \bmod N) \quad (13)$$

3　安全性分析

根据上述安全性假设,假定用户和 CSO 是可信的,他们会诚实地执行审计步骤,TPA 和 CSP 是半可信的。在这种背景下,理论上分析所提方法的安全性。

在文中方案,有以下两种可能恶意破坏公平的云存储服务的行为:(1)CSP 存储的数据丢失,试图使用伪造的证明通过验证;(2)CSP 串通 TPA 伪造证明试图通过验证。

定理 1(正确性):如果 CSP 和 TPA 都诚实地遵守规定的程序,那么证据可以通过 User 的验证。

证明:以下等式通过从左到右的推导,证明了 Audit 算法中的等式是正确的。

$$\begin{aligned}
&\mathrm{Hash}(\mathrm{proof}_{\mathrm{CSP}}^{\mathrm{ssk}} \bmod N) = \\
&\quad \mathrm{Hash}\Big(\prod_{t=1}^{r} \mathrm{proof}_{t}^{\mathrm{ssk}} \bmod N\Big) = \\
&\quad \mathrm{Hash}\Big(\prod_{t=1}^{r}\prod_{i=1}^{e} \mathrm{chal}^{c_{ti} b_{ti}\mathrm{ssk}} \bmod N\Big) = \\
&\quad \mathrm{Hash}\Big(\prod_{t=1}^{r}\prod_{i=1}^{e} g^{sc_{ti} b_{ti}\mathrm{ssk}} \bmod N = \\
&\quad \mathrm{Hash}\Big(\prod_{t=1}^{r}\prod_{i=1}^{e} (g^{b_{ti}\mathrm{ssk}})^{sc_{ti}} \bmod N\Big) = \\
&\quad \mathrm{Hash}\Big(\prod_{t=1}^{r}\prod_{i=1}^{e} (\mathrm{Tag}_{ti}^{c_{ti}})^{s} \bmod N\Big) = \\
&\quad \mathrm{Hash}(\mathrm{proof}_{\mathrm{TPA}}^{s} \bmod N) \qquad (14)
\end{aligned}$$

假设 1(二次剩余问题):给定一个整数 $N = p \times q$,其中 p 和 q 是机密的,并且整数 $g < N$,没有 x 能够满足 $x^2 \equiv g(\bmod N)$。

根据假设 1,推导出定理 2:

定理 2:对于已知 g,$g^s \bmod N$ 和 N 的敌手,没有多项式时间算法可以找到任何满足 $s' \neq s, N \mid g^s - g^{s'}$ 的 s'。

证明:假设存在多项式时间算法 A_1,对于给定 s, g 和 $N = p \times q$,总能找到一个 s' 满足 $s' \neq s, N \mid g^s - g^{s'}$,得到以下等式:

$$s \equiv s' \bmod \mathrm{ord}(g) \qquad (15)$$

其中,$\mathrm{ord}(g)$ 是满足 $N \mid g^{\mathrm{ord}(g)} - 1$ 的最小正整数,并且是一个真值与 $s - s' \mid \mathrm{ord}(g)$ 相同的奇数。若给定 g 可以满足 $\mathrm{ord}(g) \bmod 2 = 1$,可以得到以下等式:

$$g^{\mathrm{ord}(g)+1} \equiv g(\bmod N) \qquad (16)$$

相应的,也可以得到以下等式:

$$(g^{\frac{\mathrm{ord}(g)+1}{2}})^2 \equiv g(\bmod N) \qquad (17)$$

令 $x = g^{\frac{\mathrm{ord}(g)+1}{2}}$,得到:

$$x^2 \equiv g(\bmod N) \qquad (18)$$

如上所示,在算法 A_1 的帮助下,可以生成一个多项式时间算法 A_2 来找出满足 $x^2 \equiv g(\bmod N)$ 的 x,而它与假设 1 相矛盾,由此,定理 2 的证明已完成。

定理 3:CSP 无法通过伪造的 b_i' 通过验证。

证明:恶意 CSP 可能在 $\mathrm{Proof}_{\mathrm{CSP}}\mathrm{Gen}$ 步骤中,使用伪造的 b_{ij}' 生成证据,试图通过验证。然而,根据定理 2,没有多项式时间算法可以找到一个 b_{ij} 满足 $b_{ij}' \neq b_{ij}$ 且 $\mathrm{chal}^{b_{ij}} \equiv \mathrm{chal}^{b_{ij}}(\bmod N)$,也就是说,$\mathrm{chal}^{b_{ij}}(\bmod N)$ 只能是从数据块 b_{ij} 得到的哈希值,CSP 无法通过伪造通过验证。同时,只要用户在多次验证中没有选择同一个随机数 s 来检查同一个数据块,恶意 CSP 就无法通过重播攻击欺骗用户。因此,该程序可以有效地发现 CSP 的任何恶意行为。

定理 4:CSP 串通 TPA 伪造数据无法通过验证。

证明:当 CSP 损坏或丢失数据后,可能会串通 TPA 伪造数据来通过验证。CSP 在设置阶段将用户发送的 b_{ij} 发送给 TPA,企图获取用户私钥,CSP 伪造数据块 b_{ij}',并告知 TPA,若想利用伪造的 b_{ij}' 通过验证,TPA 必须获取用户私钥(ssk)才能伪造 HVT。然而根据定理 2,TPA 无法在已知 g,$g^{b_{ij} * \mathrm{ssk}}(\bmod N)$,$b_{ij}$ 的情况下得到用户的 ssk,也就无法通过伪造的标签完成验证。

综上所述,文中方案能够有效地检测数据破坏或恶意伪造行为,从而确保数据审计方案能够安全实施。

4　性能分析

该文实现了 EMCA 和文献[17,20]的原型系统,实验证明,EMCA 大大减少了审计时间及计算开销。实验硬件环境为 macOS 操作系统,2.6 GHz 六核 Intel Core i7 处理器,16 GB 内存。

4.1　理论分析

表 3 比较了 EMCA 与文献[17,20]在标签生成、证据生成和验证证据阶段的计算开销。为了简化表达式,使用 T_{mul},T_{exp},T_{p},T_{hash} 和 T_{add} 分别代表一次乘法运算、一次幂运算、一次双线性对运算、一次 Hash 运算和一次加法运算所需的时间。其中,n 为数据块总数,r 为副本数,e 为批量审计的数据块数。对比得出,在标签生成和验证证据阶段,EMCA 都是具有优势的,在整个审计过程中,与文献[17,20]相比,文中方案具有更低的计算开销。

4.2　实验分析

该文选择了 0 ~ 200 个数据块,5 个副本,10 个云存储服务器来比较 EMCA 与文献[17,20]在标签生成、证据生成和验证证据时的计算开销。

表 3 EMCA 与文献[17,20]的计算成本比较

操作	文献[17]	文献[20]	EMCA
生成标签	$nr(3T_{mul}+T_{hash}+2T_{exp})$	$nr(2T_{exp}+T_{mul}+T_{add}+T_{hash})$	nrT_{exp}
生成证据	$e(T_{exp}+T_{mul})+(er-1)T_{add}$	$er(T_{exp}+2T_{mul}+T_{add})$	$2erT_{exp}+T_{hash}+(3er-2)T_{mul}$
验证证据	$3T_p+erT_{mul}+(e-1)T_{add}+(er+1)(T_{hash}+T_{exp})$	$3T_p+er(T_{exp}+T_{mul}+T_{hash})$	$2(T_{exp}+T_{hash})$

图 2 显示了标签生成的比较结果。当数据块数为 200 时,EMCA 需要 44 ms,而文献[17]需要 1 010 ms,文献[20]需要 2.5 s。很显然,文中方案的时间消耗要远小于文献[17,20]的时间消耗。

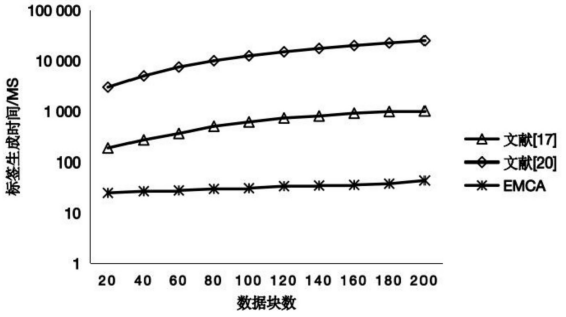

图 2 标签生成的计算成本比较

图 3 显示了证据生成的比较结果。当审计数据块不断增多时,EMCA 的计算成本比文献[17]的高,原因在于 EMCA 的计算成本集中在证据生成阶段,审计阶段所用开销极小;而文献[17]的计算开销集中在审计阶段,证据生成阶段开销较小。

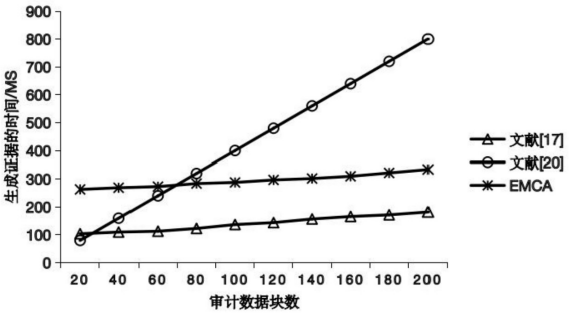

图 3 证据生成的计算成本比较

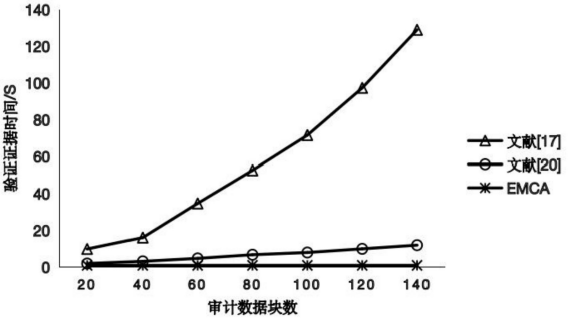

图 4 验证证据的计算成本比较

图 4 显示了验证证据阶段的比较结果。EMCA 开销最小,当数据块数为 100 时,EMCA 只需不到 1 s,而文献[17]需要 71 s,文献[20]需要 8 s。很显然,在审计阶段,文中方案的计算成本远低于文献[17,20]的计算成本。原因在于 EMCA 验证阶段只需进行两次模幂和两次哈希,计算成本与审计数据块数无关,而文献[17,20]需要三次双线性对配对和多次哈希、幂运算、加法和乘法,且计算成本随审计数据块数增加而增加。

图 5 显示了一次完整的审计过程的计算成本比较。从一次完整的审计过程来看,EMCA 的计算开销远小于文献[17,20]的计算开销,如当审计数据块为 100 时,文献[17]一次审计需要 71.8 s,文献[20]需要 21 s,而 EMCA 只需 1.3 s。原因在于 EMCA 舍弃了计算昂贵的双线性映射运算,采用了模幂加密技术来保证效率与安全。

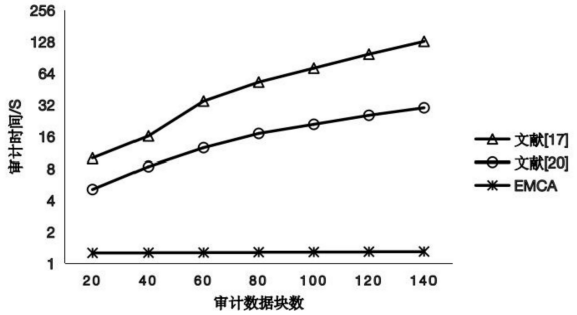

图 5 一次审计的计算成本比较

5 结束语

针对现有的多副本数据审计方案的不足,该文提出了一种在多云情景下的多副本数据完整性验证方案,副本数据可存储在不同的云存储服务器中,采用第三方审计者协助审计,通过同态可验证标签可一次批量审计所有抽样副本,大大提高了审计效率。实验分析与评估表明,该方案是安全和高效的。未来的工作是改进方案使用户不必参与审计过程,以减轻用户端负担。

参考文献:

[1] XU Y,WANG G,REN J,et al. An adaptive and configurable protection framework against android privilege escalation threats[J]. Future Generation Computer Systems,2019,92:

210-224.

[2] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//ACM conference on computer & communications security. New York: ACM, 2007:598-609.

[3] LEI Z, FU A, SHUI Y, et al. Data integrity verification of the outsourced big data in the cloud environment: a survey[J]. Journal of Network & Computer Applications, 2018, 122: 1-15.

[4] 杨海滨,李瑞峰,李秀广,等. 高效的无证书云数据审计方案[J]. 工程科学与技术,2022,54(3):72-79.

[5] XU Y, REN J, ZHANG Y, et al. Blockchain empowered arbitrable data auditing scheme for network storage as a service [J]. IEEE Transactions on Services Computing, 2020, 13 (2):289-300.

[6] 王秀秀. 云环境下基于无证书的数据完整性审计方案研究[D]. 兰州:西北师范大学,2022.

[7] 杨小东,裴喜祯,陈桂兰,等. 支持用户撤销的多用户多副本数据公开审计方案[J]. 计算机工程,2020,46(12):150-156.

[8] VARGHESE B, BUYYA R. Next generation cloud computing[J]. Future Generations Computer Systems, 2018, 79 (3):849-861.

[9] YANG Y, CHEN Y, CHEN F, et al. An efficient identity-based provable data possession protocol with compressed cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 1359-1371.

[10] ERWAY C C, KUPCU A, PAPAMANTHOU C, et al. Dynamic provable data possession[J]. ACM Transaction on Information and System Security, 2015, 17(4): 15. 1-15. 29.

[11] WANG Q, WANG C, REN K, et al. Enabling public auditability and data dynamics for storage security in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(5): 847-859.

[12] YAN Z, AHN G J, HU H, et al. Dynamic audit services for outsourced storages in clouds[J]. IEEE Transactions on Services Computing, 2013, 6(2): 227-238.

[13] WANG B, LI B, LI H. Public auditing for shared data with efficient user revocation in the cloud[J]. IEEE Transactions on Services Computing, 2015, 8(1): 92-106.

[14] WANG B, LI B, HUI L. Oruta: privacy-preserving public auditing for shared data in the cloud[J]. IEEE Transactions on Cloud Computing, 2014, 2(1): 43-56.

[15] SHEN W, QIN J, YU J, et al. Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2018, 14(2): 331-346.

[16] CURTMOLA R, KHAN O, BURNS R, et al. MR-PDP: multiple-replica provable data possession[C]//28th international conference on distributed computing systems. Beijing: IEEE, 2008: 411-420.

[17] ZHOU L, FU A, YANG G, et al. Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(2): 1118-1132.

[18] 侯高攀. 云存储中数据安全审计关键技术研究[D]. 西安:西安电子科技大学,2021.

[19] 裴喜祯. 云存储中多副本数据公开审计方案研究[D]. 兰州:西北师范大学,2021.

[20] YANG X, PEI X, WANG M, et al. Multi-replica and multi-cloud data public audit scheme based on blockchain[J]. IEEE Access, 2020, 8: 144809-144822.