软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(12):4758–4770 [doi: 10.13328/j.cnki.jos.006360]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于身份的组用户数据完整性验证方案*

袁艺林[1,2], 张建标[1,2], 徐万山[1,2], 李 铮[1,2]

[1](北京工业大学 信息学部, 北京 100124)

[2](可信计算北京市重点实验室, 北京 100124)

通信作者: 张建标, E-mail: zjb@bjut.edu.cn

摘 要: 云存储系统为用户提供大容量、高访问效率、价格合理的存储服务. 然而, 使用云存储服务的用户, 一旦将文件上传至CSP (cloud server provider), 便失去了数据的绝对控制权. 众所周知, CSP并不可靠. 因此, 云上存储的数据是否完整, 成为值得深入探讨的问题. 在公共云存储环境中, 将公司、机构或组织定义为一个组, 组内由负责人进行管理. 组内用户为便于使用云存储服务, 可借助于组负责人进行统一操作. 这种场景下, 为解决位于同一组内的用户数据完整性验证问题, 提出了一个组用户数据完整性验证方案. 为协助组内用户进行一系列操作, 方案提出了代理这一实体. 方案基于IBE (identity-based encryption)进行标签的设计, 摆脱了复杂的证书管理问题. 在数据完整性验证阶段, 通过采用随机抽样的方式, 减少了系统的性能开销. 借助于随机预言机模型, 该方案被证明是安全的. 且通过的一系列的性能分析与评估, 验证了该方案是可行的.

关键词: 云存储; 数据完整性验证; 基于身份的密码学; 组用户

中图法分类号: TP306

中文引用格式: 袁艺林, 张建标, 徐万山, 李铮. 基于身份的组用户数据完整性验证方案. 软件学报, 2022, 33(12): 4758–4770. http://www.jos.org.cn/1000-9825/6360.htm

英文引用格式: Yuan YL, Zhang JB, Xu WS, Li Z. Identity-based Group User Data Integrity Verification Scheme. Ruan Jian Xue Bao/Journal of Software, 2022, 33(12): 4758–4770 (in Chinese). http://www.jos.org.cn/1000-9825/6360.htm

Identity-based Group User Data Integrity Verification Scheme

YUAN Yi-Lin[1,2], ZHANG Jian-Biao[1,2], XU Wan-Shan[1,2], LI Zheng[1,2]

[1](Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China)

[2](Beijing Key Laboratory of Trusted Computing, Beijing 100124, China)

Abstract: Cloud storage systems provide users with storage services with large capacity, high access efficiency, and reasonable prices. Nevertheless, the users who use cloud storage services will lose absolute control over the data once they upload files to the CSP. As it is well known, CSP (cloud server provider) is not reliable. Whether the data on the cloud is with integrity has become a problem worth considering. Under the public cloud storage environment, this study defines a company, organization or organization as a group, and the group is managed by the person in charge who can help the users in the group using the cloud storage service conveniently. In this scenario, to solve the problem of user data integrity verification in the same group, a data integrity verification scheme is proposed for group users in this study. To assist users in one group to carry out a series of operations, an entity named Agency is proposed. In this scheme, the design of the tag is based on IBE (identity-based encryption), which frees the users from complicated certificate management. In the integrity verification process, by adopting random sampling, the performance overhead of the system is greatly reduced. With the help of the random oracle model, the security of the proposed scheme is proved. A practical experiment validates the feasibility of the scheme in the end.

Key words: cloud storage; data integrity verification; identity-based encryption; user group

* 基金项目: 北京市自然科学基金(M21039)

收稿时间: 2020-11-18; 修改时间: 2021-03-02; 采用时间: 2021-04-26

1 引 言

云存储作为云计算技术概念的延伸与应用, 自提出以来便受到了广泛的关注. 用户将自己的私有数据存储到云端, 实现了数据外包, 极大地减轻了本地存储负担, 并节省了存储空间. CSP (cloud server provider)为用户提供大容量、价格合理、存储快速的存储服务, 而用户仅需根据自身需求选择存储服务类型即可. 云存储的按需付费、位置独立、用户无需参与日常维护等优点, 令其备受欢迎. 然而, 由于 CSP 并非完全可靠, 因此云上用户数据的安全性受到质疑[1]. 例如: CSP 为节省存储空间, 丢弃用户不常使用的数据; 出于好奇, CSP 非法访问用户数据; 出于利益, CSP 甚至恶意破坏用户数据. 云存储的安全性包括 3 方面: 机密性、完整性与可用性(confidentiality, integrity, availability, CIA). 机密性保证非授权用户无法访问数据; 完整性保证非授权用户无法篡改数据; 可用性保证合法用户可随时访问数据. 本文从安全性出发, 对公共云存储环境下的用户数据安全性与完整性进行探讨.

选择使用云存储的用户, 一旦将数据上传至 CSP, 便失去了外包数据的控制权, 数据不再绝对保密, 存在被篡改的风险. 为了验证存储在 CSP 上的用户数据是否完整, 适用于不同场景的数据完整性验证方案被提出. 在数据完整性验证方案中, 用户在将数据外包到云端之前, 通常需要为自己的文件生成便于后续索引数据的标签——它不仅是衡量方案安全性与可靠性的重要指标, 还能协助用户完成远程数据完整性验证. 因此, 在数据完整性验证方案中, 设计安全且高效的标签相当必要.

1.1 动 机

在公共云存储环境下, 使用云存储服务的用户可能是个人、公司或组织, 他们根据文件的安全级别, 为自己挑选符合需求的云存储服务器. 云存储服务器安全等级不同, 收费标准不同. 通常情况下, 安全级别越高, 收费相对越贵. 例如: 对于一些公共广告数据, 存储在廉价的公共服务器上即可; 而对于一些私密的金融、教育等数据, 往往需存储在安全级别高的私有服务器上. 因此, 为自己的数据界定安全级别并挑选不同类型的存储服务器, 也是用户需要慎重考虑的问题.

在公共存储环境下, 本文提出一个新的场景: 假设一个公司、机构或组织称为一个组, 位于该公司、机构或组织内的用户称为组用户. 若位于同一组内的用户需使用云存储服务, 假设他们可以选择两种方式:

(1) 自行加密数据并完成上传, 后续定期执行数据完整性验证工作;

(2) 对自己的文件进行加密, 计算标签, 然后将⟨加密后的文件,标签⟩打包发送给组负责人, 并授权他们进行后续操作(包括文件上传、完整性验证等工作).

方法(1)操作简便且直观, 但由于所有工作都由用户执行, 存在一些问题: 一方面, 对一些手持手机、笔记本等低容量、低性能设备的用户而言, 一系列的操作不仅消耗了大量的网络带宽, 且极大地增加了用户的负担; 另一方面, 确定自己文件的安全级别后, 用户需完成挑选存储服务器、上传文件等一系列后续工作. 方法(2)中, 由于增加了组负责人, 用户仅需完成数据加密、标签计算、告知文件安全等级这些工作, 而后续操作由组负责人全权负责. 这种方法减轻了用户的操作负担, 且便于系统化地统一管理组用户数据. 因此, 在此场景下, 研究组内用户的数据完整性验证方案很有价值.

1.2 相关工作

传统的数据完整性验证方案要求用户将存储在云上的所有数据下载至本地, 然后进行校验. 这对于持有低容量设备或低计算性能的用户而言, 代价是致命的. 2007 年, Ateniese 等人[2]提出了一种称为可证数据持有(provable data prossession, PDP)的方案. 在 PDP 方案中, 用户无需从云端下载所有外包数据, 仅需借助于基于 RSA 设计的同态验证标签, 采用随机抽样的方式, 便可验证远程数据的完整性. 因此, 该模型是一个有效的概率模型. PDP 方案不仅实现了无阻塞验证, 且极大地降低了 I/O 开销. 同年, Juels 等人[3]提出了一种性能更佳的数据完整性验证方案——可检索证明(proof of retrievability, PoR). 该方案基于双线性配对实现, 它不仅能为用户提供远程数据完整性验证, 还能实现数据检索. 随后, 基于 PoR 方案, Shacham 等人[4]提出了一个更加完善的方案. 在文献[4]中给出了构造同态验证器的两种方式: 当基于伪随机函数(pseudorandom functions,

PRFs)构造同态认证器时, 方案支持私有验证, 并证明在标准模型中安全; 当使用 BLS 签名构造同态认证器时, 方案支持公开验证, 并证明在随机预言模型中安全. 此外, 该方案是第 1 个针对任意敌手的安全性进行了证明的方案.

但 PDP 方案与 PoR 方案均只针对静态数据, 并未对数据块的动态操作进行研究. 为了实现数据块的修改、删除和追加等动态操作, Ateniese 等人[5]提出了一种基于对称密钥密码学的、扩展的 PDP 方案. 通过操纵等级认证信息表, Erway 等人[6]提出了一种改进的动态 PDP 方案. 基于 Merkle 哈希树, Wang 等人[7]提出了一种公开数据完整性审计方案, 该方案通过构造块标记认证来实现动态性. 借助于无阻塞 Merkle 树的认证结构, He 等人[8]提出了一种能够实现全动态的、面向动态组的 PDP 方案. 针对文件的多个副本的同时更新问题, Barsoum 等人[9]给出了具体的方案. 在公开完整性验证过程中, 当用户的密钥被泄露时, Yu 等人[10]提出了对密钥进行实时更新的策略. 借助一种新的结构——分治表, Sookhak 等人[11]研究了数据块动态操作的实现. 此外, 更多的数据完整性方案被提出[12−14].

以上提及的所有方案均基于公钥基础设施(public key infrastructure, PKI)实现. 在使用 PKI 分发密钥的完整性验证方案中, PKI 是不可缺少的实体, 它主要负责公私钥的分发与管理. 然而, 由于证书的存在, 极大地增加了验证负担. 例如: 用户在验证远程数据完整性时, 除需验证数据完整外, 还需检查证书. 此外, 系统需承受证书生成、转发、验证、更新等负担, 对于一些计算性能较低的终端设备, 如手机, 这些开销往往难以承受. 现实中, 证书管理的效率相当低下且繁琐. 为了摆脱复杂的证书管理, 基于身份的密码学(identity-based encryption, IBE)[15,16]被提出. 在此基础上, 第 1 个公共数据完整性审计方案被提出[17]. Wang 等人[18]讨论了当用户被限制访问 CSP 时, 如何执行公开完整性验证. 在实体——sanitizer 的协助下, Shen 等人[19]提出了一种支持敏感信息隐藏的数据共享方案. 为了抵御恶意审计员的攻击, Zhang 等人[20]提出了一种基于身份的公开诚信审计方案, 并借助区块链实现. Yu 等人[21]提出了一个可支持隐私保护的完整性验证方案. Li 等人[22]以用户的生物特征数据作为加密密钥, 结合属性基加密, 研究了一种基于模糊身份加密的新方案. He 等人[23]讨论了一个可在无线体域网中使用的高效的无证书公开审计方案. 此外, 在不同的应用场景下, 更多的基于身份的数据完整性验证方案被提出[24−35].

1.3 贡 献

基于以上考虑, 在公共云存储环境下, 为解决位于同一组内的用户数据完整性验证问题, 在本文提出的新的场景下, 基于 IBE, 提出了一个组用户数据完整性验证方案, 贡献如下:

(1) 解决了位于同一组内的用户数据完整性验证问题. 在该方案中, 标签基于 IBE 进行设计. 由于用户私钥由私钥生成中心直接计算并分发, 令系统免于复杂的证书管理, 用户仅需利用该密钥进行标签的计算即可. 数据完整性验证阶段, 在无需下载全部外包数据的前提下, 方案采用随机抽样的方式, 极大地减少了系统的计算与通讯开销;

(2) 方案给出了详细的安全性分析: 针对不可信的 CSP, 方案基于 CDH 难题, 设计随机预言机模型, 并借助该模型, 验证了方案的可靠性;

(3) 通过一系列的性能分析与仿真实验, 验证了方案的可行性.

1.4 组织结构

本文第 2 节给出相关的预备知识. 第 3 节中给出定义与模型. 第 4 节介绍方案的具体算法. 第 5 节进行安全性分析. 第 6 节给出性能评估. 第 7 节进行总结.

2 预备知识

2.1 双线性对

G_1, G_2 是阶均为 p 的乘法循环群, g 是 G_1 的生成元. 定义双线性对: e:$G_1\times G_1\rightarrow G_2$, 且满足以下条件.

(1) 双线性: 对 $\forall u,v \in G_1$, $a,b \in Z_P^*$, 有 $e(u^a,v^b)=e(u,v)^{ab}$;

(2) 非退化性: $e(g_1,g_2)\neq 1$;

(3) 可计算性: 存在一个有效的算法, 可计算出 e.

2.2 安全性假设

Computational Diffie-Hellman 假设. 对于未知的 $a,b \in Z_P^*$, 输入三元组$(g,g^a,g^b)\in G_1$, 输出 $g^{ab}\in G_1$.

定义 1(CDH 问题). 下列公式中, 概率多项式时间(PPT)算法$\mathcal{A}$在 G_1 中解决 CDH 问题的优势可忽略不计:

$$AdvCDH_{\mathcal{A}} = \Pr[\mathcal{A}(g,g^a,g^b)=g^{ab} : a,b \xleftarrow{R} Z_P^*].$$

Discrete Logarithm 假设. 对于未知的 $x \in Z_P^*$, 给定$(g,g^x)\in G_1$, 输出 x.

定义 2(DL 问题). 下列公式中, PPT 算法$\mathcal{A}$在 G_1 中解决 DL 问题的优势可忽略不计:

$$AdvDL_{\mathcal{A}} = \Pr[\mathcal{A}(g,g^x)=x : x \xleftarrow{R} Z_P^*].$$

3 定义与模型

3.1 系统模型与设计目标

方案的系统模型如图 1 所示, 共包含 5 个实体.

- 数据持有者(data owner, DO): 持有私有数据的实体, 云存储服务使用者. 在我们的方案中, 称位于同一个公司、机构或组织中的用户为组用户;
- 云服务提供商(cloud server provider, CSP): 提供云存储服务的实体. 用户可以根据需求选择不同的 CSP 来存储数据;
- 代理: 方案提出的新实体, 主要工作为管理组用户, 并根据用户数据的安全级别为每个用户分配 CSP 的实体;
- 私钥生成中心(private key generator, PKG), 一个被其他实体完全信任的实体. PKG 根据用户的标识符 ID 为用户生成私钥;
- 验证者: 为用户执行完整性验证的实体.

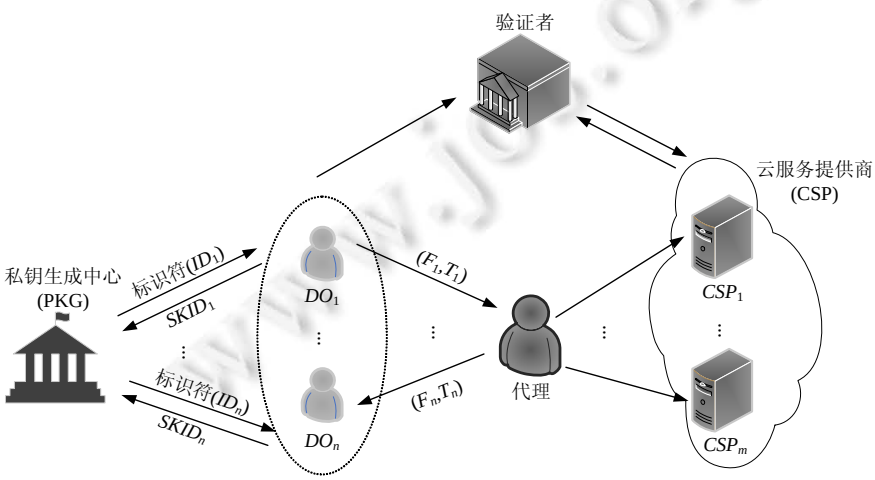


图 1 系统模型图

基于身份的组用户数据完整性验证方案, 旨在完成以下目标.

(1) 正确性

a) 标签集的正确性: 代理验证用户产生的标签集的正确性. 若标签集能够通过代理的验证, 代理

接受它并为其分配 CSP;

b) 审计正确性: CSP 只有完整地、正确地存储了用户的文件, 在数据完整性验证阶段产生的证据才能通过验证者的验证;

(2) 审计可靠性: 若 CSP 没有真正地存储用户的全部外包数据, 那么它一定不能通过验证者的验证.

3.2 符号定义

方案中涉及到的符号含义见表 1.

表 1 符号表

符号	描述	符号	描述
p	大素数	G_1, G_2	阶为 p 的乘法循环群
g	G_1 的生成元	e	双线性对 $e:G_1\times G_1\to G_2$
Z_p^*	非零元素的素数域	H	密码哈希函数 $H:\{0,1\}^*\to G_1$
n	用户个数	DO_i	第 i 个用户, $1\leqslant i\leqslant n$
ID_i	第 i 个用户的标识符, $1\leqslant i\leqslant n$	F_i'	第 i 个用户的源文件, $1\leqslant i\leqslant n$
F_i	第 i 个用户的加密文件, $1\leqslant i\leqslant n$	pp	系统公开参数
mpk_i	用户 DO_i 的主公钥	msk_i	用户 DO_i 的主私钥
SK_{ID_i}	用户 DO_i 的私钥	T_i	第 i 个用户文件的标签集, $1\leqslant i\leqslant n$
$\hat{n}$	CSP 的个数	CSP_l	第 l 个 CSP, $1\leqslant l\leqslant \hat{n}$

3.3 方案框架

定义 2. 方案包含 7 个算法: Setup, Extract, TagGen, Allocation, Challenge, ProofGen, ProofVerify.

1) Setup: 该算法由 PKG 执行. 输入安全参数 k, 输出系统公开参数 pp、主公钥 mpk_i、主私钥 msk_i;

2) Extract: 该算法由 PKG 执行. 以主公钥 mpk_i、主私钥 msk_i、DO_i 身份标识 ID_i 为输入, 输出 DO_i 的私钥 SK_{ID_i};

3) TagGen: 该算法由 DO_i 执行. DO_i 根据 SK_{ID_i} 处理上传文件. 该算法以 DO_i 的加密文件 F_i、身份标识 ID_i、私钥 SK_{ID_i} 为输入, 输出加密文件的标签集 T_i;

4) Allocation: 该算法由代理执行. 代理在本地维护一张用户存储服务器表, 并根据该表为 DO_i 分配 CSP;

5) Challenge: 该算法由验证者执行;

6) ProofGen: 该算法由 CSP 执行. 在接收到挑战信息 *chal* 后, CSP 执行该算法产生完整性证据 P;

7) ProofVerify: 该算法由验证者执行. 验证者验证 CSP 返回的完整性证据 P: 若通过验证, 则输出“1”; 否则, 输出“0”.

3.4 用户存储服务器表(user cloud server provider table, 简称UCT)

代理在本地维护一张 UCT, 并依据用户文件的安全级别, 为每个用户分配存储服务器. UCT 是一个小型的动态数据结构, 包含 3 列: 用户 ID (user ID, uid)、存储服务器 ID (cloud server provider ID, cid)、添加次数 (append number, *an*). uid 指示第 i 个用户 DO_i, cid 指示第 l 个服务器 CSP_l. *an* 记录 DO_i 的文件的追加次数, 初始设置为 0. 当 DO_i 需要追加上传的文件时, *an* 加 1.

注: 在我们的方案中, 为便于后续操作, 定义来自同一用户的文件具有相同的安全级别. 定义 DO_i 本地的文件达到一定大小(例如 20TB)时, 可执行一次追加操作. 代理将追加的⟨文件,标签⟩存储到相应的 CSP 中, 并更新 UCT 表 DO_i 对应的 *an*. 图 2 给出了 UCT 的一个示例.

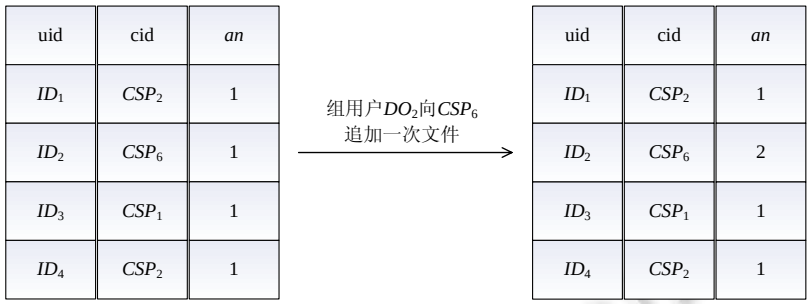


图 2 UCT 表的一个示例

3.5 安全模型

为了形式化安全模型, 在我们的方案中, 提出了一个挑战者 C 和敌手 A 之间的游戏, 假设 DO 为挑战者 C, 不可信的 CSP 为敌手 A.

定义 3. 挑战者 C 与敌手 A 之间的查询——伪造游戏定义如下.

(1) Setup 阶段. 挑战者 C 运行 Setup 算法, 获得公开参数 pp、主公钥 mpk、主私钥 msk, 并将(pp,mpk,msk)发送给敌手 A;

(2) Query 阶段. 在 Query 阶段, 针对挑战者 C, 敌手 A 做两种类型的查询:

a) Extract 查询: 敌手 A 查询用户的私钥. 挑战者 C 运行 Extract 算法获得私钥 SK_{ID}, 并将其发送给敌手 A;

b) TagGen 查询: 挑战者查询用户文件 F 的标签集. 在获得私钥 SK_{ID} 后, 挑战者 C 运行 TagGen 算法, 获得文件 F 的标签集{T}, 并将其发送给敌手 A;

(3) Challenge 阶段. 在此阶段, 敌手 A 充当证明者(prover), 挑战者 C 充当验证者(verifier). 挑战者 C 向敌手 A 发起完整性挑战, 并要求敌手 A 针对挑战信息 chal, 提供完整性证据 P;

(4) Forgery 阶段. 在接收到来自挑战者 C 的完整性挑战后, 针对挑战信息 chal, 敌手 A 计算完整性证据 P, 并回复挑战者 C. 若证据 P 能够以不可忽略的概率、通过挑战者 C 的验证, 则认为敌手 A 取得了游戏的胜利.

在上述的安全模型中, 若敌手 A 没有正确地、完整地存储挑战信息 chal 中涉及到的相关数据块, 那么该敌手 A 无法给出能够通过挑战者 C 验证的完整性证据 P.

定义 4. 若方案是安全的, 则满足以下条件: 当敌手 A 产生的完整性证据 P 能够以不可忽略的概率通过挑战者 C 的验证, 则存在一个知识提取器(knowledge extractor): 它能以可忽略的概率, 提取出与挑战信息 chal 中下标相对应的数据块.

定义 5. 若 CSP 损坏了文件的ρ个数据块, 那么这些损坏的数据块能以至少δ的概率被检测到.

4 具体算法

方案包含 7 个算法, 具体描述如下.

(1) Setup 算法

a) PKG 选择两个阶均为 p 的乘法循环群 G_1, G_2, g 是 G_1 的生成元. 定义密码哈希函数: H:{0,1}*→G_1, 双线性对: e:G_1×G_1→G_2, 加密算法 E 及其密钥 Key;

b) PKG 随意选择 n 个来自 Z_p^* 的元素 $x_1, x_2, ..., x_n \in Z_p^*$ 以及来自 G_1 的元素$\mu_1,\mu_2,\ldots,\mu_n \in G_1$ 和$\zeta_1,\zeta_2,\ldots,\zeta_n \in G_1$;

c) PKG 产生主公钥 $mpk_i=g^x$ 以及主私钥 $msk_i=x_i(1\leqslant i\leqslant n)$;

d) PKG 公开参数 $pp=(G_1,G_2,g,p,H,e,\mu_1,\mu_2,\ldots,\mu_n,\zeta_1,\zeta_2,\ldots,\zeta_n)$及主公钥$\{mpk_i\}_{1\leqslant i\leqslant n}$, 主私钥$\{msk_i\}_{1\leqslant i\leqslant n}$

保密;

(2) Extract 算法

a) 根据 DO_i 的 ID_i, PKG 为 DO_i 计算私钥. PKG 选择随机值 $r_i \in Z_P^*$,并计算:

$$SK'_{ID_i} = g^{r_i},\ SK''_{ID_i} = g^{x_i}(\prod_{t=1}^{u}\zeta_t^{ID_t})^{r_i};$$

b) PKG 为 DO_i 生成的私钥为 $SK_{ID_i} = (SK'_{ID_i}, SK''_{ID_i}) = (g^{r_i}, g^{x_i}(\prod_{t=1}^{u}\zeta_t^{ID_t})^{r_i})$. 通过安全信道, PKG 将私钥发送给 DO_i;

(3) TagGen 算法: DO_i 根据自己的私钥 SK_{ID_i}, 为自己的文件计算标签集. 生成标签集之前, DO_i 先加密源文件 F_i', $F_i = E_{key}(F_i')$, 并将加密后的文件 F_i 分割为 m 个数据块 $F_i=\{b_{ij}\}_{1\leqslant i\leqslant n,1\leqslant j\leqslant m}$. 然后, 再将每个数据块分割为 s 个数据片 $\{b_{ij1},...,b_{ijk},...,b_{ijs}\}_{1\leqslant k\leqslant s} \in Z_P^*$:

a) 对每个数据块 b_{ij}, DO_i 计算 $\sigma_{ij} = g^{x_i}(\prod_{t=1}^{u}\zeta_t^{ID_t})^{r_i}\cdot H(name\|i)\prod_{k=1}^{s}\mu_i^{b_{ijk}}$, 而 $T_{ij}=\{\sigma_{ij}\}_{1\leqslant j\leqslant n}$, 其中, $name \in Z_P^*$ 为文件标识符;

b) DO_i 为自己的文件设置安全级别 $sec_i \in Z_P^*$;

c) DO_i 将$\{F_i,T_i,sec_i\}$发送给代理, 并删除本地文件存储;

(4) Allocation 算法

a) 在接收到 DO_i 的标签集后, 代理先根据公式(1)验证标签集的正确性:

$$e(\sigma_{ij},g) = e(mpk_i,g)\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t}, g^{r_i})\cdot e(H(name\|i)\prod_{k=1}^{s}\mu_i^{b_{ijk}}, g) \tag{1}$$

若公式(1)成立, 则代理接受 DO_i 的$\{F_i,T_i,sec_i\}$, 并继续后续的操作; 否则, 代理通知 DO_i 重新传送$\{F_i,T_i,sec_i\}$;

b) 代理将与 DO_i 相关的信息添加到 UCT 中(包括用户标识符 ID_i、文件安全级别 sec_i);

c) 根据用户文件的安全级别 sec_i, 代理为其分配 CSP_l, $1\leqslant l\leqslant \hat{n}$, 并将相应信息添加到 UCT 中.

在我们提出的方案中, 代理机构定期代替组内用户发起完整性验证挑战. 通过定期向代理询问完整性验证结果, 组内用户不仅可以从验证的重担中解脱出来, 还可以督促代理履行职责. 我们以 DO_i 为例, 给出完整性验证过程(如图 3 所示).

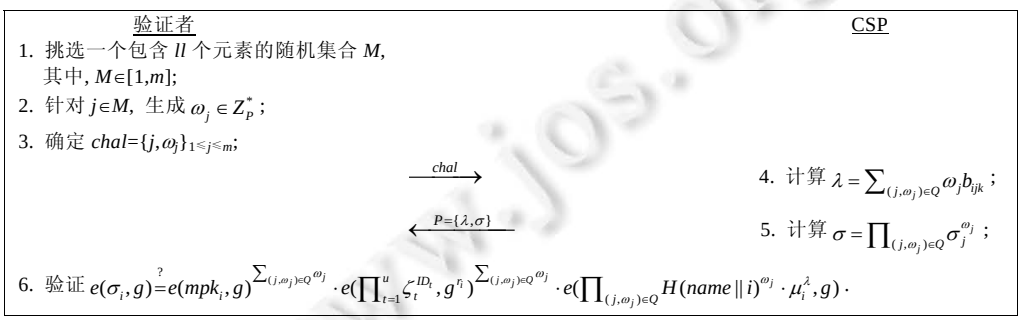


图 3 完整性验证过程

假设 DO_i 的文件存储在 CSP_l 上, 详细过程如下.

(5) Challenge 算法

a) 验证者随机挑选包含 ll 个元素的集合 M, 且 $M\in[1,m]$, 作为参与挑战的随机下标集;

b) 验证者针对每一个 $j\in M$, 产生 $\omega_j \in Z_P^*$;

c) 验证者将挑战信息 $chal=\{j,\omega_j\}_{1\leqslant j\leqslant m}$ 发送给 CSP_l;

(6) ProofGen 算法: 接收到挑战信息 chal 后, CSP_l 产生完整性证据 P:

a) CSP_l 计算 $\sigma = \prod_{(j,\omega_j)\in Q}\sigma_j^{\omega_j}$, $\lambda = \sum_{(j,\omega_j)\in Q}\omega_j b_{ijk}$;

b) CSP_l将证据 $P=\{\lambda,\sigma\}$返回给验证者;

(7) ProofVerify 算法: 验证者在接收到 CSP_l 返回的完整性证据 P 后, 验证以下公式是否成立:

$$e(\sigma_i,g)\overset{?}{=}e(mpk_i,g)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t},g^{r_i})^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{(j,\omega_j)\in Q}H(name\|i)^{\omega_j}\cdot\mu_i^{\lambda},g) \tag{2}$$

若公式(2)成立, 则输出"1", 表示 CSP_l 通过了完整性验证; 否则, 输出"0".

5 安全性分析

在本节中, 我们将从正确性、审计可靠性两方面讨论方案的安全性.

定理 1(正确性). 方案的正确性包括标签集的正确性以及审计正确性两方面.

证明:

(1) 标签集的正确性

在 Allocation 算法中, 当代理接受到 DO_i 的标签集 T_i 后, 通过公式(1)是否成立来验证标签集的正确性:

$$\begin{aligned}e(\sigma_{ij},g)&=e(g^{x_i}(\prod_{t=1}^{u}\zeta_t^{ID_t})^{r_i}\cdot H(name\|i)\prod_{k=1}^{s}\mu_i^{b_{ijk}},g)\\&=e(g^{x_i}(\prod_{t=1}^{u}\zeta_t^{ID_t})^{r_i},g)\cdot e(H(name\|i)\prod_{k=1}^{s}\mu_i^{b_{ijk}},g)\\&=e(g^{x_i},g)\cdot e((\prod_{t=1}^{u}\zeta_t^{ID_t})^{r_i},g)\cdot e(H(name\|i)\prod_{k=1}^{s}\mu_i^{b_{ijk}},g)\\&=e(mpk_i,g)\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t},g^{r_i})\cdot e(H(name\|i)\prod_{k=1}^{s}\mu_i^{b_{ijk}},g)\end{aligned}$$

若公式(1)成立, 则代理继续后续的操作; 否则, 代理通知 DO_i 重新传送$\{F_i,T_i,sec_i\}$.

(2) 审计正确性

在 ProofVerify 算法中, 验证者通过公式(2)是否成立来验证完整性证据 P 的正确性:

$$\begin{aligned}e(\sigma_i,g)&=e(\prod_{(j,\omega_j)\in Q}\sigma_{ij}^{\omega_j},g)\\&=e(\prod_{(j,\omega_j)\in Q}(g^{x_i}(\prod_{t=1}^{u}\zeta_t^{ID_t})^{r_i}\cdot H(name\|i)\prod_{k=1}^{s}\mu_i^{b_{ijk}})^{\omega_j},g)\\&=e(\prod_{(j,\omega_j)\in Q}(g^{x_i})^{\omega_j},g)\cdot e(\prod_{(j,\omega_j)\in Q}((\prod_{t=1}^{u}\zeta_t^{ID_t})^{r_i})^{\omega_j},g)\cdot e(\prod_{(j,\omega_j)\in Q}(H(name\|i)\prod_{k=1}^{s}\mu_i^{b_{ijk}})^{\omega_j},g)\\&=e\left((g^{x_i})^{\sum_{(j,\omega_j)\in Q}\omega_j},g\right)\cdot e\left((\prod_{t=1}^{u}\zeta_t^{ID_t})^{\sum_{(j,\omega_j)\in Q}\omega_j},g^{r_i}\right)\cdot e(\prod_{(j,\omega_j)\in Q}(H(name\|i))^{\omega_j}\cdot\prod_{(j,\omega_j)\in Q}(\prod_{k=1}^{s}\mu_i^{b_{ijk}})^{\omega_j},g)\\&=e(mpk_i,g)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t},g^{r_i})^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{(j,\omega_j)\in Q}(H(name\|i))^{\omega_j}\cdot\prod_{(j,\omega_j)\in Q}\prod_{k=1}^{s}\mu_i^{b_{ijk}\omega_j},g)\\&=e(mpk_i,g)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t},g^{r_i})^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{(j,\omega_j)\in Q}(H(name\|i))^{\omega_j}\cdot\prod_{k=1}^{s}\prod_{(j,\omega_j)\in Q}\mu_i^{b_{ijk}\omega_j},g)\\&=e(mpk_i,g)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t},g^{r_i})^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e\left(\prod_{(j,\omega_j)\in Q}(H(name\|i))^{\omega_j}\cdot\prod_{k=1}^{s}\mu_i^{\sum_{(j,\omega_j)\in Q}b_{ijk}\omega_j},g\right)\\&=e(mpk_i,g)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t},g^{r_i})^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{(j,\omega_j)\in Q}(H(name\|i))^{\omega_j}\cdot\prod_{k=1}^{s}\mu_i^{\lambda},g)\\&=e(mpk_i,g)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t},g^{r_i})^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{(j,\omega_j)\in Q}(H(name\|i))^{\omega_j}\cdot\mu_i^{\lambda},g)\end{aligned}$$

若公式(2)成立, 则输出"1", 表示 CSP 通过了本次完整性验证; 否则, 输出"0"

定理 2(审计可靠性). 假设 CDH 问题与 DL 问题在双线性群 G_1 中难解, 且标签存在不可伪造. 在我们的方案中, 对于敌手 A 或不可信的 CSP, 若存储在 CSP 上的数据块已经被破坏, 那么, 伪造出能够通过验证者验证的完整性证据 P 在计算上是不可行的. 即, 敌手伪造出的证据 P 不能以不可忽略的概率通过验证者的验证, 除非他持有本次验证所涉及的所有数据块.

证明: 我们构造一个知识提取器, 其作用为: 通过与方案进行多次重复交互, 提取出参与挑战的所有数据块. 换言之, 若敌手 A 能够通过验证者的验证, 但他并未持有所有的数据块, 那么就可以通过知识提取器与方案之间的重复交互, 提取出所有被挑战的数据块. 我们假设 CSP 作为对手 A, 验证者作为挑战者 C. A 与 C

之间多次重复地进行交互, 并通过执行以下一系列的游戏, 方案的审计可靠性可被证明.

- **Game 0**. Game 0 的定义已在第 3 节中给出;
- **Game 1**. Game 1 与 Game 0 类似, 但只有一处不同. 挑战者 C 保留一个标签查询列表, 记录敌手 A 曾查询过的所有标签信息. 当敌手 A 做 TagGen 查询时, 挑战者 C 将该条记录添加至列表中;
- **Game 2**. Game 2 与 Game 1 类似, 只有一处不同. 挑战者 C 保留一个查询回复列表, 记录它针对敌手 A 的查询, 做出的所有回复信息.

考虑以下情形: 敌手 A 通过了验证者的验证, 但他却并未持有所有被挑战的数据块的完整内容. 一旦出现此情形, 游戏将会被终止, 即使敌手 A 已经赢得了本次游戏的胜利. 此处存在两种情形: a) 聚合标签$\sigma\neq\sigma'$; b) 聚合消息$\lambda\neq\lambda'$. 以上两种情形均可导致游戏终止. 针对以上两种情形, 我们分别进行分析.

- 分析

假设敌手 A 以不可忽略的概率赢得 Game 2. 那么给定一个可靠的完整性证据 P, 公式(3)必定能够成功地通过验证者的验证:

$$e(\sigma,g)=e(mpk,g)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t},g^r)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{(j,\omega_j)\in Q}H(name\|i)^{\omega_j}\cdot\mu^{\lambda},g) \tag{3}$$

假设敌手 A 输出伪造的证据 $P'=\{\lambda',\sigma'\}$, 并回复挑战者 C. 验证者借助于公式(4)验证 $P'=\{\lambda',\sigma'\}$的正确性:

$$e(\sigma',g)=e(mpk,g)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{t=1}^{u}\zeta_t^{ID_t},g^r)^{\sum_{(j,\omega_j)\in Q}\omega_j}\cdot e(\prod_{(j,\omega_j)\in Q}H(name\|i)^{\omega_j}\cdot\mu^{\lambda'},g) \tag{4}$$

a) 若存在聚合标签$\sigma\neq\sigma'$.

在此情形下, 针对该敌手 A, 我们构建一个能够解决 CDH 难题的模拟器. 给定 $g,g^{\alpha},h\in G_1$, 模拟器的目标是成功输出 h^{α}. 模拟器与 Game 1 中的挑战者 C 相似, 但存在不同: 模拟器随机挑选一个元素 $x\in Z_P^*$, 设主私钥为 $msk=x$, 主公钥为 $mpk=g^x$. 然后, 它挑选一个随机值 $b\in Z_P^*$, 选取 $\overline{x}\in Z_P^*$, 并设$\mu=(g^{\alpha})^{\overline{x}}h^b$.

随后, 模拟器与敌手 A 进行多次交互. 定义$\Delta\lambda=\lambda'-\lambda$,并用公式(4)除以公式(3), 可得:

$$e(\sigma'/\sigma,g)=e(\mu^{\Delta\lambda},g)=e(((g^{\alpha})^{\overline{x}}h^b)^{\Delta\lambda},g) \tag{5}$$

进一步地, 通过公式(5), 可得 $h^{\alpha}=(\sigma\cdot\sigma^{-1}\cdot g)^{1/\Delta\lambda\overline{x}b}$. 若能够成功地解决 CDH 难题, 那么模拟器必定能够成功地计算出 h^{α}. 我们可通过计算 $\Delta\lambda\overline{x}b\neq 0 \bmod p$ 的概率来间接求出成功计算出 h^{α}的概率.

可知 $\Delta\lambda\overline{x}b=0 \bmod p$ 成立的概率是 $1/p$, 但由于 p 是一个大素数, 所以 $1/p$ 是可忽略的. 因此, 我们能以 $1-1/p$ 的概率求出 h^{α}; 换言之, 我们能以 $1-1/p$ 的概率解决 CDH 难题, 但这与 CDH 难题在 G_1 中计算不可行是相矛盾的.

b) 若存在聚合消息$\lambda\neq\lambda'$.

在此情形下, 针对该敌手 A, 我们构建一个能够解决 DL 难题的模拟器. 给定 $g,h=g^{\alpha}$, 模拟器的目标是成功输出α. 模拟器与 Game 1 中的挑战者 C 相似, 但存在不同: 模拟器随机挑选两个随机值 $a,b\in Z_P^*$, 并设 $\mu=g^ah^b$.

随后, 模拟器与敌手 A 进行多次交互. 此处我们有$\sigma'=\sigma$, 但$\lambda'\neq\lambda$. 定义$\Delta\lambda=\lambda'-\lambda$, 并用公式(4)除以公式(3), 可得:

$$1=\mu^{\Delta\lambda}=(g^ah^b)^{\Delta\lambda}=g^{a\Delta\lambda}\cdot h^{b\Delta\lambda} \tag{6}$$

显然, $\Delta\lambda\neq 0 \bmod p$. 否则有$\lambda'=\lambda \bmod p$, 这与先前的假设相矛盾.

进一步地, 通过公式(6)可得 $h=g^{-a\Delta\lambda/b\Delta\lambda}=g^{a/b}$. 那么, 在 G_1 中解决 DL 问题的概率相当于 $b\neq 0$ 的概率. 而 $b\neq 0$ 的概率为 $1/p$, 但由于 p 是一个大素数, 所以 $1/p$ 是可忽略的. 因此, 我们能以 $1-1/p$ 的概率计算出 DL 难题, 但这与 DL 难题在 G_1 中计算不可行是相矛盾的.

综上, 若敌手 A 能够以不可忽略的概率赢得 Game 1 和 Game 2 的胜利, 那么构造出的模拟器必定能够解决 CDH 难题与 DL 难题, 但这与先前的假设相矛盾. 故, 若 CSP 想要通过验证者的验证, 那么他必须正确存储用户的完整文件.

定理 3(可侦测性). 假设文件 F_i 被划分为 m 个数据块, d 表示文件 F_i 中的坏块(指一些被 CSP 删除或毁坏

的数据块), c 表示在完整性验证过程中被查询的数据块, 则我们的方案能以 $\left(\frac{d}{m},1-\left(\frac{m-1}{m}\right)^c\right)$ 的概率侦测出文件 F_i 中存在的坏块.

证明: 对文件 F_i, 定义一个离散随机变量 X, 表示被挑战的块恰好匹配文件 F_i 中坏块的数量, PX 表示在挑战信息中至少检测到一个坏块的概率, 故:

$$PX=P\{X\geqslant 1\}=1-P\{X=0\}=1-\frac{m-d}{m}\times\frac{m-1-d}{m-1}\times\ldots\times\frac{m-c+1-d}{m-c+1}.$$

由于 $1-\left(\frac{m-1}{m}\right)^c\leqslant PX\leqslant\left(1-\frac{m-c+1-d}{m-c+1}\right)^c$, 因此可得 $PX\geqslant 1-\left(\frac{m-1}{m}\right)^c$. 故, 我们的方案至少能够以 $\left(\frac{d}{m},1-\left(\frac{m-1}{m}\right)^c\right)$ 的概率侦测出 CSP 的错误行为.

6 性能评估

6.1 性能分析

在这一部分, 我们对方案的性能进行了评估. 针对 ProofGen 与 ProofVerify 算法的计算代价, 将我们的方案与 Shacham 等人[4]、He 等人[23]的方案进行了比对, 结果见表 2.

表 2 计算开销的比较

方案	ProofGen 算法的计算开销	ProofVerify 算法的计算开销
Shacham 等人的方案[8]	$cExp_{G_1}+cMul_{G_1}+cMul_{Z_p}+cAdd_{Z_p}$	$2Pair+(c+1)Exp_{G_1}+cMul_{G_1}+cH_{G_1}$
He 等人的方案[23]	$cExp_{G_1}+cMul_{G_1}+cMul_{Z_p}+cAdd_{Z_p}$	$2Pair+(c+3)Exp_{G_1}+(c+3)Mul_{G_1}+(c+1)H_{G_1}+2H_{Z_p}$
我们的方案	$cExp_{G_1}+cMul_{G_1}+cMul_{Z_p}+cAdd_{Z_p}$	$4Pair+2Exp_{G_1}+2Exp_{G_2}+cMul_{G_1}+cH_{G_1}+cAdd_{Z_p}$

其中, c 表示被挑战的数据块的个数, H_{G_1} 表示在 G_1 上执行单次哈希运算的开销, H_{Z_p} 表示在 Z_P^* 上执行单次哈希运算的开销, Mul_{G_1} 表示在 G_1 上执行单次乘运算的开销, Mul_{Z_p} 表示在 Z_P^* 上执行单次乘运算的开销, Exp_{G_1} 表示在 G_1 上执行单次幂运算的开销, Exp_{G_2} 表示在 G_2 上执行单次幂运算的开销, $Pair$ 表示执行单次配对运算的开销, Add_{Z_p} 表示在 Z_P^* 上执行单次加运算的开销.

从表 2 可看出: 3 个方案在证据产生阶段所花费的计算开销是一样的, 均需要 c 次 G_1 上的幂运算和 c 次 G_1 上的乘运算, 用于生成 σ; c 次的 Z_P^* 上的乘运算和 c 次 Z_P^* 上的加运算生成, 用于 λ. 而在证据验证阶段, 我们的方案需要 4 次配对运算, 而文献[4,23]仅需两次; 文献[4]所需要的运算的种类最少, 文献[23]次之. 但由于标签设计存在差异, 因此每个方案所花费的计算开销均不相同.

6.2 仿真实验

在这一部分, 通过仿真实验, 我们对方案的可行性进行了评估. PC 硬件配置 Intel Core i5 处理器, 8 G 内存,操作系统为 Linux 18.04 LTS 64 位, 利用 PBC 库、GMP 库. 实验使用 PBC (pairing-based cryptography)库中的 *a.param* 参数设置双线性对. 设置基域大小 512 bits, Z_P^* 大小 160 bits, 换言之, $|p|$=160 bits.

为了测试方案中各个算法的性能, 在实验的第 1 阶段, 我们以单个用户为例进行评估. 假设该用户初始文件大小为 20 MB, 由 1 000 个数据块构成, 而在完整性验证阶段, 参加挑战的数据块的个数为 460. 表 3 给出了 Extract 算法、TagGen 算法、ProofGen 算法以及 ProofVerify 算法的时间开销. 由表 3 可知, Extract 算法耗时约 700.629 6 ms, 该算法为用户生成私钥, 便于用户后续进行标签生成等操作. TagGen 阶段最为耗时, 约需 1 1038.3472 ms, 该算法的性能与文件的分块个数直接相关, 两者成正比. 即: 数据块个数越多, 生成标签所需时间开销越大. 另外, 需注意的是: Extract 算法仅需在文件的初始化阶段执行一次, 而 TagGen 算法仅需针

对某一文件执行一次. ProofGen 算法与 ProofVerify 算法的时间开销与被挑战的数据块的个数直接相关，当 c=460 时，分别耗时 546.163 2 ms 与 2035.639 5 ms.

表 3 算法的时间开销

算法名称	Extract 算法	TagGen 算法	ProofGen 算法	ProofVerify 算法
时间开销(ms)	700.629 6	11 038.347 2	546.163 2	2 035.639 5

TagGen 算法作为方案中最耗时且最昂贵的算法，在实验的第 2 阶段，我们对其性能进行了测试. 为了更纯粹地展示算法的性能，我们假设组内多个用户同时进行标签集的计算. 假设组内各用户的初始文件大小均为 20 MB，由 100 个数据块构成. 图 4 给出了组内用户个数不同时，执行 TagGen 算法所需的时间开销. 由图 4 可知: 当组内用户个数为 1 时，执行 TagGen 算法大约需 8 768.495 5 ms. 而随着组内用户个数的越多，执行 TagGen 算法所需时间越多，呈现逐渐增长的趋势. 另外，TagGen 算法的耗时与数据块的个数直接相关. 虽然执行该算法在整个方案中最为耗时，但 TagGen 算法对用户而言，是初始化阶段的一次性任务.

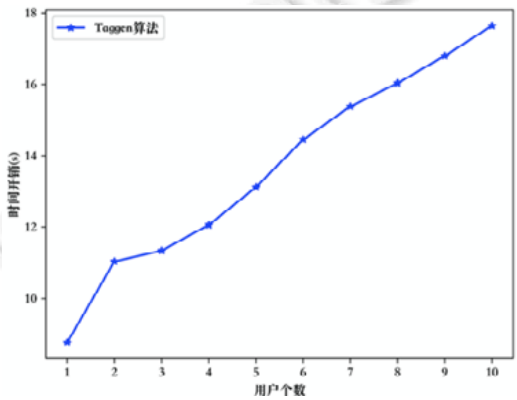
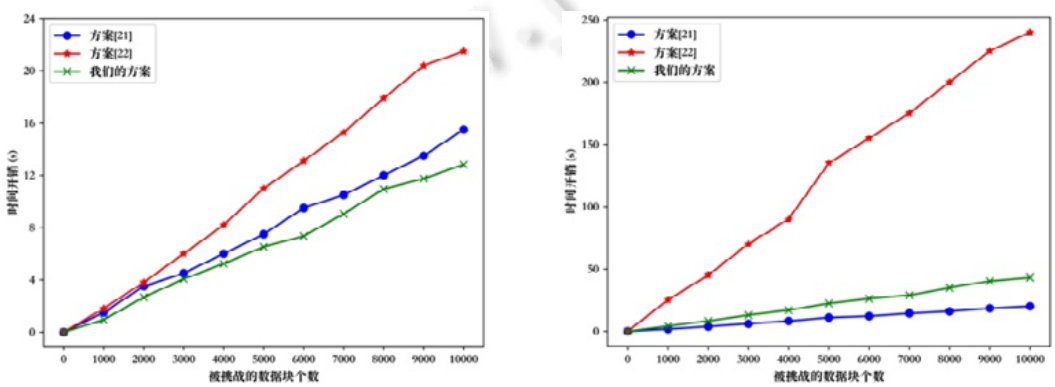


图 4 组内用户个数不同时，运行 TagGen 算法的时间开销

在实验的第三部分，我们对方案中的 ProofGen 算法与 ProofVerify 算法分别进行了性能测试. 为了方便测试,我们以组内某一用户 DO_i 为例，分别给出了其执行 ProofGen 算法与 ProofVerify 算法的时间开销；且为了更好地对方案进行评估，将结果与文献[21,22]进行了比对.

假设被参与挑战的数据块的个数从 1 000 开始，以 1 000 位间隔，逐步增长至 10 000. 图 5(a)与图 5(b)分别给出了随着被挑战的数据块个数的增加，执行 ProofGen 算法与 ProofVerify 算法的时间开销.



(a) 在证据生成阶段的时间开销对比 (b) 在证据验证阶段的时间开销对比

图 5 我们的方案与文献[21,22]在证据生成阶段以及证据验证阶段的时间开销对比

由图 5(a)可知: 在证据产生阶段, 3 个方案在该阶段的耗时并无较大的差异. 但由于各个方案的设计细节不同, 与文献[21,22]提出的方案相比, 我们的方案整体耗时最少. 当被挑战的数据块个数为 10 000 时, 耗时约 12.834 s, 而方案[21,22]分别耗时 15.547 s 与 21.599 s. 从图 5(b)可知: 在证据验证阶段, 我们的方案与文献[21]提出的方案耗时较为接近, 但我们的方案耗时相对较多. 由于各个方案的设计细节不同, 文献[22]提出的方案耗时最多, 当被挑战的数据块个数为 10 000 时, 耗时约为 240.324 s, 而我们的方案与方案[21]分别耗时为 43.075 s 与 20.659 s.

7 结束语

本文研究了云存储环境下的组用户完整性验证方案. 方案定义了一个新的场景——组, 并展开了相关问题讨论. 方案利用 IBE 实现标签的构造, 与传统的基于 PKI 分发公钥的方案相比, 减少了处理证书的时间与通信开销. 在完整性验证过程中, 通过采用随机抽样, 而非下载全部的外包数据的方式, 极大地节省了系统的性能开销. 借助于随机预言机模型, 方案的正确性和安全性得到了证明; 并通过性能评估与仿真实验, 验证了方案的有效性与可行性.

References:

[1] Feng CS, Qin ZG, Yuan D. Techniques of secure storage for cloud data. Chinese Journal of Computers, 2015, 38(1): 150−163 (in Chinese with English abstract).

[2] Ateniese G, Burns R, Curtmola R, *et al*. Provable data possession at untrusted stores. In: Ning P, De Capitani di Vimercati S, Syverson PF, eds. Proc. of the ACM Conf. on Computer and Communications Security. New York: ACM, 2007. 598−609. [doi: 10.1145/1315245.1315318]

[3] Juels A, Kaliski BS. PORs: Proofs of retrievability for large files. In: Proc. of the 14th ACM Conf. on Computer and Communications Security (CCS). New York: ACM, 2007. 584−597. [doi: 10.1145/1315245.1315317]

[4] Shacham H, Waters B. Compact proofs of retrievability. Journal of Cryptology, 2013, 26(3): 442−483. [doi: 10.1007/s00145-012-9129-2]

[5] Ateniese G, Dipietro R, Mancini LV, *et al*. Scalable and efficient provable data possession. In: Proc. of the 4th Int'l Conf. on Security and Privacy in Communication Networks. New York: ACM, 2008. 1−10. [doi: 10.1145/1460877.1460889]

[6] Erway C, Kupcu A, Papamanthou C, *et al*. Dynamic provable data possession. In: Al-Shaer E, Jha S, Keromytis AD, eds. Proc. of the ACM Conf. on Computer and Communications Security. New York: ACM, 2009. 213−222. [doi: 10.1145/2699909]

[7] Wang Q, Wang C, Ren K, *et al*. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. on Parallel and Distributed Systems, 2011, 22(5): 847−859. [doi: 10.1109/TPDS.2010.183]

[8] He K, Chen J, Yuan Q, *et al*. Dynamic group-oriented provable data possession in the cloud. IEEE Trans. on Dependable and Secure Computing, 2019. [doi: 10.1109/TDSC.2019.2925800]

[9] Barsoum AF, Hasan MA. Provable multicopy dynamic data possession in cloud computing systems. IEEE Trans. on Information Forensics and Security, 2015, 10(3): 485−497. [doi: 10.1109/TIFS.2014.2384391]

[10] Yu J, Ren K, Wang C, *et al*. Enabling cloud storage auditing with key-exposure resistance. IEEE Trans. on Information Forensics and Security, 2016, 10(6): 1167−1179. [doi: 10.1109/TIFS.2015.2400425]

[11] Sookhak M, Yu FR, Zomaya AY. Auditing big data storage in cloud computing using divide and conquer tables. IEEE Trans. on Parallel and Distributed Systems, 2018, 29(5): 999−1012. [doi: 10.1109/TPDS.2017.2784423]

[12] Xu Y, Ren J, Zhang Y, *et al*. Blockchain empowered arbitrable data auditing scheme for network storage as a service. IEEE Trans. on Services Computing, 2020, 13(2): 289−300. [doi: 10.1109/TSC.2019.2953033]

[13] Shen WT, Yang GY, Yu J, *et al*. Remote data possession checking with privacy-preserving authenticators for cloud storage. Future Generation Computer System, 2017, 76: 136−145. [doi: 10.1016/j.future.2017.04.029]

[14] Wang YJ, Wu QH, Qin B, *et al*. Online/Offline provable data possession. IEEE Trans. on Information Forensics and Security, 2017, 12(5): 1182−1194. [doi: 10.1109/TIFS.2017.2656461]

[15] Shamir A. Identity-based cryptosystems and signature schemes. In: Proc. of the Advances in Cryptology (CRYPTO'84). Santa Barbara, 1984(8): 47−53. [doi: 10.1007/3-540-39568-7_5]

[16] Boneh D, Franklin M. Identity-based encryption from the Weil pairing. In: Proc. of the CRYPTO 2001. Berlin, Heidelberg: Springer-Verlag, 2001. 213−229. [doi: 10.1007/3-540-44647-8_13]

[17] Wang HQ, Wu QH, Qin B, *et al*. Identity-based remote data possession checking in public clouds. IET Information Security, 2014, 8(2): 114−121. [doi: 10.1049/iet-ifs.2012.0271]

[18] Wang HQ, He DB, Tang SH. Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud. IEEE Trans. on Information Forensics and Security, 2016, 11(6): 1165−1176. [doi: 10.1109/TIFS.2016.2520886]

[19] Shen WT, Qin J, Yu J, *et al*. Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage. IEEE Trans. on Information Forensics and Security, 2019, 14(2): 331−346. [doi: 10.1109/TIFS.2018. 2850312]

[20] Zhang Y, Xu CX, Lin XD, *et al*. Blockchain-based public integrity verification for cloud storage against procrastinating auditors. IEEE Trans. on Cloud Computing, 2019. [doi: 10.1109/TCC.2019.2908400]

[21] Yu Y, Au MH, Ateniese G, *et al*. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. IEEE Trans. on Information Forensics and Security, 2017, 12(4): 767−778. [doi: 10.1109/ TIFS.2016.2615853]

[22] Li YN, Yu Y, Min GY, *et al*. Fuzzy identity-based data integrity auditing for reliable cloud storage systems. IEEE Trans. on Dependable and Secure Computing, 2019, 16(1): 72−83. [doi: 10.1109/TDSC.2017.2662216]

[23] He DB, Zeadally S, Wu LB. Certificateless public auditing scheme for cloud-assisted wireless body area networks. IEEE Systems Joural, 2018, 12(1): 64−73. [doi: 10.1109/JSYST.2015.2428620]

[24] Xue L, Yu Y, Li YN, *et al*. Efficient attribute-based encryption with attribute revocation for assured data deletion. Information Sciences, 2019, 479: 640−650. [doi: 10.1016/j.ins.2018.02.015]

[25] Tao S, Zhang F, Chen XY, *et al*. Identity-based dynamic data auditing for big data storage. IEEE Trans. on Big Data, 2019. [doi: 10.1109/TBDATA.2019.2941882]

[26] Wang HQ. Identity-based distributed provable data possession in multicloud storage. IEEE Trans. on Services Computing, 2015, 8(2): 328−340. [doi: 10.1109/TSC.2014.1]

[27] Ateniese G, Burns R, Curtmola R, *et al*. Remote data checking using provable data possession. ACM Trans. on Information & System Security, 2011, 14(1): 1−34. [doi: 10.1145/1952982.1952994]

[28] Wang HQ. Proxy provable data possession in public clouds. IEEE Trans. on Services Computing, 2013, 6(4): 551−559. [doi: 10.1109/TSC.2012.35]

[29] Shen J, Shen J, Chen XF, *et al*. An efficient public auditing protocol with novel dynamic structure for cloud data. IEEE Trans. on Information Forensics and Security, 2017, 12(10): 2402−2415. [doi: 10.1109/TIFS.2017.2705620]

[30] Zhu Y, Wang HX, Hu ZX, *et al*. Zero-knowledge proofs of retrievability. Science China Information Sciences, 2011, 54(8): 1608−1617. [doi: 10.1007/s11432-011-4293-9]

[31] Yan H, Li JG, Han JG, *et al*. A novel efficient remote data possession checking protocol in cloud storage. IEEE Trans. on Information Forensics and Security, 2017, 12(1): 78−88. [doi: 10.1109/TIFS.2016.2601070]

[32] Xu ZY, Wu LB, Khan MK, *et al*. A secure and efficient public auditing scheme using RSA algorithm for cloud storage. The Journal of Supercomputing, 2017, 73(12): 5285−5309. [doi: 10.1007/s11227-017-2085-8]

[33] Boneh D, Lynn B, Shacham H. Short signatures from the weil-pairing. Journal of Cryptology, 2004, 17(4): 297−319. [doi: 10.1007/3-540-45682-1_30]

[34] Xue JT, Xu CX, Zhao JN, *et al*. Identity-based public auditing for cloud storage systems against malicious auditors via blockchain. Science China Information Sciences, 2019, 62(3). [doi: 10.1007%2Fs11432-018-9462-0]

[35] Zhang XJ, Wang HX, Xu CX. Identity-based key-exposure resilient cloud storage public auditing scheme from lattices. Information Sciences, 2019, 472: 223−234. [doi: 10.1016/j.ins.2018.09.013]

附中文参考文献:

[1] 冯朝胜, 秦志光, 袁丁. 云数据安全存储. 计算机学报, 2015, 38(1): 150−163.

袁艺林(1991－), 女, 硕士, 主要研究领域为信息安全, 云计算, 云存储.

徐万山(1988－), 男, 硕士, 主要研究领域为信息安全, 可信计算, 云安全.

张建标(1969－), 男, 博士, 教授, 博士生导师, 主要研究领域为可信计算, 网络安全, 区块链技术.

李铮(1992－), 女, 博士, 讲师, 主要研究领域为信息安全, 密码分析.