边缘环境下基于无证书公钥密码的数据完整性审计方案

王子园[1,2]，杜瑞忠[2,3]

（1. 河北大学管理学院，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；
3. 河北大学网络空间安全与计算机学院，河北 保定 071002）

摘 要：边缘环境下，当数据传输到云端时需途经边缘节点这一新的实体，这使数据安全问题变得更加复杂，数据的机密性和完整性很难得到保证，传统的数据完整性审计方案不适用于设备繁多的边缘环境。基于此，提出了一种边缘环境下基于无证书公钥密码的数据完整性审计方案，结合在线/离线签名思想，在边缘节点半可信的情况下，用户设备只需在上传数据时进行轻量级的计算，其余计算量交由离线阶段执行。该方案利用边缘节点进行审计工作，同时支持不同存储状态下的审计和隐私保护等特性。安全性分析表明，所提方案在随机预言模型下能有效应对三类敌手攻击，证明该方案是安全的。与其他方案进行实验对比，结果显示所提方案时间开销最低。

关键词：边缘计算；数据完整性审计；在线/离线签名；无证书公钥密码

中图分类号：TP309

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2022130

Certificateless public key cryptography based provable data possession scheme in edge environment

WANG Ziyuan[1,2], DU Ruizhong[2,3]

1. School of Management, Hebei University, Baoding 071002, China
2. Key Lab on High Trusted Information System in Hebei Province, Baoding 071002, China
3. School of Cyber Security and Computer, Hebei University, Baoding 071002, China

Abstract: In the edge environment, data transmission to the cloud needs to pass through a new entity, the edge node, which makes the data security problem more complicated, the confidentiality and integrity of data are difficult to be guaranteed, and the traditional provable data possession scheme is not suitable for the edge environment with a large number of devices. Based on this, a certificateless public key cryptography based provable data possession scheme was proposed for the edge environment, combining the online/offline signature idea, where the user device only needed to perform light computation when uploading data in the case of semi-trusted edge nodes, leaving the rest of the computation to be performed in the offline phase. The scheme used edge nodes for auditing work while supporting auditing in different storage states, as well as privacy protection and other features. The security analysis shows that the proposed scheme is proven to be secure by being able to effectively combat three types of adversary attacks under a stochastic prediction model. Experimental comparisons with other schemes show that the proposed scheme has lowest time overhead.

Keywords: edge computing, provable data possession, online/offline signature, certificateless public key cryptography

0 引言

外包数据的存储安全一直是云环境下的重要安全问题。为保证远程数据的安全与完整，Ateniese 等[1]首次提出了数据完整性审计（PDP, provable data possession）机制，将验证外包数据

收稿日期：2022−02−09；**修回日期**：2022−06−02

通信作者：杜瑞忠，durz@hbu.edu.cn

基金项目：国家自然科学基金资助项目（No.61572170）；河北省自然科学基金重点资助项目（No.F2019201290）

Foundation Items: The National Natural Science Foundation of China (No.61572170), Key Project of Natural Science Foundation of Hebei Province (No.F2019201290)

完整性的解决方案系统化、规范化，在此基础上许多行之有效的数据完整性审计方案被提出[2]。

随着万物互联的飞速发展和广泛应用[3]，以及边缘计算[4]的逐步成熟，越来越多的设备被接入边缘环境中，现有的云计算集中式处理环境逐渐向边缘与云计算协同处理的形势发展，用户侧产生了边缘节点这一新的实体。而边缘用户设备在计算能力和存储能力等方面的局限性使用户数据的计算和存储安全等问题[5]变得更加严重。受以上因素的影响，传统云环境下的安全性方案已经不适于新的边缘环境。

此外，数据完整性审计方案通常会将审计工作交由可信的第三方处理，然而实际应用环境下第三方并不完全可信，再加上边缘环境下数据存储的形式越来越多样化，实体间的信任模型越来越复杂。因此，如何有效利用边缘节点的存储与计算能力，设计一种面向边缘环境的低复杂度的数据完整性验证方案是本文研究的重点。

为了解决上述问题，本文在假定边缘节点半可信的情况下，致力于在降低用户计算开销的同时保障外包数据的完整性与安全性。本文的贡献总结如下。

1) 基于边缘环境资源受限的特性，将无证书公钥密码思想与在线/离线标签思想相结合，设计了一种适用于边缘环境的完整性审计方案。

2) 利用边缘节点的存储与计算能力，在保证隐私的情况下使不同的边缘节点分别执行审计者与存储者的职能。

3) 针对数据的不同存储状态进行分析，保证边缘环境下数据存储的正确性以及确定性删除。

4) 通过理论与实验分析，基于计算性 Diffie-Hellman（CDH, computational Diffie-Hellman）问题假设，证明本文方案在随机预言模型下是安全的，且与其他低复杂度审计方案相比，本文方案效率更高。

1　相关工作

自 Ateniese 等[1]首次提出数据完整性验证机制开始，之后的数据完整性验证方案基本遵循该审计结构。近年来，诸多数据完整性审计方案被提出[6-9]，且在安全性与计算开销上有了很好的提升。

PDP 方案通常需花费计算成本和存储成本来进行证书的管理，为设备带来较高的负担。为解决这一问题，有学者提出基于身份的数据完整性审计方案，其能够避免传统 PDP 方案系统建立和管理密钥生成中心（KGC, key generation center）的困难。在这类审计方案中，用户的公钥由用户自身的身份信息产生，私钥由 KGC 生成。Tian 等[10]提出的方案能够保证验证者在验证用户完整性的同时不会获取用户信息。Shen 等[11]使用清理程序来清理与文件敏感信息相对应的数据块，并将这些数据块的签名转换为已清理文件的有效签名。Li 等[12]提出了基于模糊身份的数据完整性审计方案，利用秘密共享技术，使用生物识别信息作为模糊身份。Wang 等[13]支持在多云环境下实现数据完整性审计，解决了分布式数据存储审计困难。Yu 等[14]支持审计单个服务器中存储的多个数据副本，同时支持部门级的动态操作。Wang 等[15]利用索引逻辑表为云存储构建基于身份的不可否认的动态可证明数据占有方案，有效防止恶意用户的攻击。

基于身份的加密体制使 KGC 可以获取所有用户的私钥，若 KGC 被攻破，则攻击者能够使用任意用户的数据标签，数据完整性验证就失去意义。针对这一问题，无证书公钥密码机制被提出。在无证书公钥密码学中，用户的私钥由两部分组成，一部分由 KGC 产生，另一部分则由用户自己生成。因此 KGC 无法得到用户的完整私钥，即增强了密钥托管的安全性。

He 等[16]提出了一个隐私保护的无证书数据完整性审计（PP-CLPDP, privacy-preserving certificateless PDP）方案，大大减少了用户上传数据时的计算开销。在其基础上，Ji 等[17]重新定义了安全性模型，提出了 tHKWWC（twisted HKWWC）方案，该方案将 CSP（cloud service provider）与 KGC 实体分开，更接近真实的云环境。随后 Gao 等[18]提出了无证书公共审计方案（CL-PAS, certificateless public auditing scheme），在标签生成阶段对数据块进行哈希处理，能够确保恶意云服务器无法伪造审计证据，并防止半可信的第三方审计直接获取用户数据信息，加强了隐私保护。

边缘环境下，也有一些数据完整性审计方案被提出。Wang 等[19]在可信的边缘计算环境下，为给资源有限的终端提供计算能力，将数据预处理任务卸载到边缘，降低了计算负荷，同时支持第三方验证平台进行数据完整性验证。Liu 等[20]提出了针对企业多媒体数据边缘环境下的数据完整性审计方案，采用同态验证码，交由完全可信的第三方进行数据审计，从而降

低了用户计算开销。Li 等[21]允许应用程序供应商检查其应用的缓存数据的完整性，并能高效地定位损坏数据的位置。由于边缘计算设备位于网络边缘，缺少有效的审计措施，导致攻击者可能修改或删除用户在边缘节点上的数据来销毁某些证据。又因为边缘计算场景下参与的实体类型多、数量大，所以信任情况非常复杂。攻击者可能将恶意边缘节点伪装成合法边缘节点，使终端用户连接到恶意边缘节点，隐秘地收集用户数据[22]。为此，如何在真实环境中进行数据完整性审计是本文要解决的问题之一。

综上所述，传统的数据完整性方案并不完全适于边缘环境，因为边缘环境对计算及存储有很严格的资源限制，而已有的面向边缘环境的完整性审计方案计算复杂度依然偏高，所以本文提出了边缘环境下基于无证书公钥密码的完整性审计方案，在省去复杂的证书管理的同时，尽可能降低用户侧的计算开销，审计时预先对数据区块进行处理，不直接传输真实数据，从而保护用户的数据隐私不被泄露。

2 相关知识

2.1 双线性映射

设 q 是素数，G_T 和 G_V 是阶为 q 的乘法循环群，通常称映射 $e:G_T\times G_T\to G_V$ 为一个双线性对，e 满足以下 3 个性质。

1) 双线性：对于任意 $\delta,\xi\in Z_q$ 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 困难型假设

1) CDH 问题。给定三元组 (P,aP,bP)，对任意 $a,b\in Z_q^*$，直接计算 abp 是困难的。

2) 离散对数困难性问题。给定 $P,Q\in G_1$，求满足 $Q=xP$ 的整数 x 是困难的，其中 $x\in Z_q^*$。

2.3 在线/离线签名

本文方案允许签名者在分离线阶段和在线阶段生成签名。在给出要签名的数据之前，执行离线阶段，当用户设备接通电源时，离线阶段可以作为后台计算持续执行。产生数据后，将执行在线阶段，通常在线阶段的执行时间很短，即使具有弱处理器的设备也可以有效执行。

3 方案设计

3.1 整体框架

本文方案采用边云混合的方式，分为用户、边缘以及云三层，涉及终端、边缘节点、密钥生成中心、云端四类实体，各实体分别负责不同的功能，具体介绍如下。系统模型如图 1 所示。

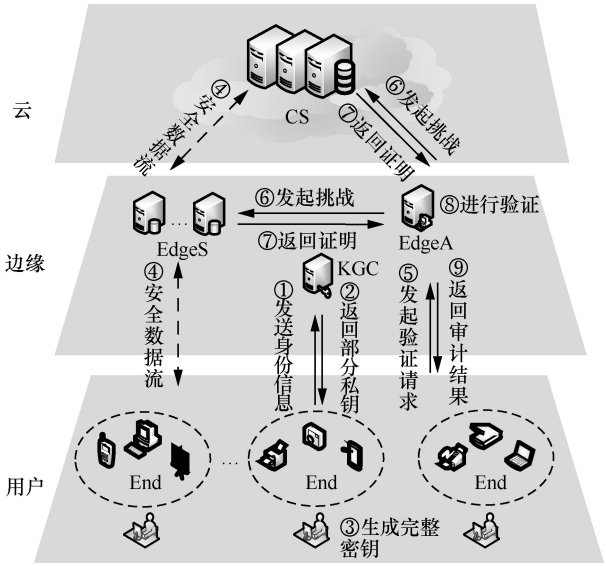

图 1 系统模型

1) 终端（End）。每个用户可以拥有一台或多台终端设备，其数量众多，性能参差不齐，自身的存储空间往往不能满足所产生的数据量，所以选择上传数据到边缘节点或云端，并将数据完整性验证工作外包。

2) 边缘节点（Edge）。接近用户侧的边缘节点负责一定范围内设备的数据交互，具有存储与计算功能。本文假定边缘节点是半可信的，拥有一定的用户数据管理权限，不会违背用户要求，但可能会对数据产生好奇。

为避免单一数据节点损坏或受到攻击而产生数据损坏的问题，本文规定不同边缘节点具有不同的职能。本文将进行存储数据的边缘节点命名为 EdgeS，进行审计数据的边缘节点命名为 EdgeA。

3) 密钥生成中心（KGC）。KGC 部署在边缘层，被所有用户信任，负责在初始化阶段使用主密钥为其他实体生成基于身份的部分私钥。

4) 云端（CS）。云端为边缘用户提供云存储服务。在本文中，云端不可信，且其需要接收边缘节点发来的数据完整性验证请求，以确认数据是否完整存储在服务器上。

3.2　具体方案

当终端接入网络后，由 KGC 利用用户身份及设备身份生成部分私钥，交由终端生成完整私钥。传输数据时，用户在离线阶段预先处理计算量较大的运算，在线阶段只需进行轻量级运算即可上传数据至 EdgeS，随后根据用户需求存储在本地或选择上传至云端。音视频等数据量较大的数据，经由边缘节点上传至云端；用户密码、智能家居配置信息等数据量小、隐私性强的数据，存储在边缘节点本地。审计阶段，用户委托 EdgeA 代为审计，以减轻用户计算压力，最后得到审计结果。

系统包括 6 个阶段，下面对每个阶段进行说明。具体符号说明如表 1 所示。

表 1　　符号说明

参数	含义
H_1, H_2	安全的哈希函数
z	KGC 主密钥
$P_{\text{pub}} = zP$	KGC 系统公钥
ID_u	设备 u 的身份信息
$D_u = zQ_u$	设备 u 的部分私钥
PK_u, SK_u	设备 u 的完整公私钥对
$\text{SK}_{\text{EdgeS/CS}}$	存储节点的私钥
m_i	数据块
δ_i	数据块签名
θ	用户选择的数据 F 存储状态

3.2.1　初始化阶段

初始化阶段包括系统参数生成、部分密钥提取和完整密钥生成。

系统参数生成。KGC 执行以下操作，输入安全参数 k，生成大素数 q、乘法循环群 G_1 和 G_2、生成元 $P \in G_1$ 和双线性对 $e: G_1 \times G_1 \to G_2$。选择随机数 $z \in Z_q^*$ 作为 KGC 的主密钥并保密，设置 $P_{\text{pub}} = zP$ 为系统公钥。定义 2 个安全的 Hash 函数 $H_1: \{0,1\}^* \to G_1$，$H_2: \{0,1\}^* \to Z_q^*$。随后 KGC 公开其系统参数 $\{q, G_1, G_2, e, P, P_{\text{pub}}, H_1, H_2\}$。

部分密钥提取。当用户设备 ID_u 要注册到 KGC 时，计算该设备部分私钥的步骤如下。

1) ID_u 选择随机数 $x_u \in Z_q^*$ 作为 ID_u 的秘密值，计算 $X_u = x_u P$。

2) 用户设备将发送 (ID_u, X_u) 给 KGC，KGC 计算 $Q_u = H_1(\text{ID}_u, X_u)$ 和 $D_u = zQ_u$，其中 z 是 KGC 的主密钥。

3) 输出 D_i 为部分私钥，并发送部分私钥给用户。

完整密钥生成。当终端收到 KGC 返回的部分私钥后，执行以下步骤完成密钥的生成。

1) 终端计算 $\text{SK}_u = x_u D_u$ 为用户的完整私钥。

2) 计算 $Y_u = x_u P_{\text{pub}}$，得到 $\text{PK}_u = (X_u, Y_u)$ 为对应的公钥。

每台用户设备在接入网络后需执行初始化阶段生成密钥，考虑到密钥有泄露风险，在怀疑密钥泄露或一定周期后，应再次执行初始化阶段产生新的密钥。

3.2.2　标签生成阶段

离线阶段，终端还未生成数据，而是预先随机生成一组数据块标号，每一块的标号为 $\text{id}_1, \text{id}_2, \cdots, \text{id}_n$，离线计算标号与系统公钥的哈希值 $H_2(\text{id}_1), H_2(\text{id}_2), \cdots, H_2(\text{id}_n)$ 以及 $T = H_1(P_{\text{pub}})$。计算部分签名 $\alpha_i = \text{SK}_u H_2(\text{id}_i)$，得到用户的部分签名集合 $\Phi' = \{\alpha_i\}_{1 \leqslant i \leqslant n}$ 并离线存储在本地。

在线阶段，终端准备上传数据 F，将数据 F 平均分成 n 块，$F = m_1 // m_2 // \cdots // m_n, m_i \in Z_q^*$。假设数据 F 以 ID_F 为标识，单一数据块的标识对应预生成的 $\text{id}_1, \text{id}_2, \cdots, \text{id}_n$，生成完整签名 $\delta_i = \alpha_i + TH_2(m_i) = \text{SK}_u H_2(\text{id}_i) + TH_2(m_i)$，完整签名集合 $\Phi = \{\delta_{m_i}\}_{1 \leqslant i \leqslant n}$，同时利用私钥 SK_u 生成数据 F 的签名 $\text{Tag}_F = e(\text{SK}_u H_2(\text{ID}_F), P)$。

当数据发生变化需要更新时，标签集合 ϕ 与文件签名 Tag_F 也会对应更新，随后重新上传。

3.2.3　数据传输阶段

外包数据存在 3 种状态，即只保存在云端、只保存在边缘节点以及既保存在云端又保存在边缘节点，状态分别用 $\theta = (0,1)$、$(1,0)$和$(1,1)$ 来表示，且状态的选择权交由用户。

终端首先选择上传 $\{\Phi, F, \text{Tag}_F, \text{ID}_u, \theta\}$ 到 EdgeS，并删除本地数据，随后发送 $\{\text{ID}_u, \text{ID}_{\text{Edge}}, \text{ID}_F, \text{Tag}_F, \theta\}$ 至 EdgeA。EdgeS 收到数据后，首先对进行 Tag_F 验证，验证过程为

$$e(H_2(\text{ID}_F), Y_u H_1(\text{ID}_F, X_u)) = \text{Tag}_F \tag{1}$$

对式(1)进行证明，即

$$\begin{aligned}&\mathrm{Tag_F}=e\left(\mathrm{SK}_u H_2(\mathrm{ID}_F),P\right)=\\&e\left(x_u z H_1(\mathrm{ID}_F,X_u)H_2(\mathrm{ID}_F),P\right)=\\&e\left(H_1(\mathrm{ID}_F,X_u)H_2(\mathrm{ID}_F),Y_u\right)=\\&e\left(H_2(\mathrm{ID}_F),Y_u H_1(\mathrm{ID}_F,X_u)\right)\end{aligned}\tag{2}$$

若验证失败，则返回失败结果，拒绝存储，以防恶意用户伪造数据欺骗存储数据者；若验证成功，则说明设备上传数据正确。然后依据用户选择 θ 对数据进行操作，数据上传流程如图 2 所示。

存储数据时，建议将敏感、短期、数据量较小的数据存储于边缘节点，而长期、数据量较大的数据上传至云端。

当数据不存储于某些节点，或达到一定存储期限需要进行删除时，设备通过随机生成脏数据 f 替换原有节点中的文件 F，以确保原有数据正确删除。

3.2.4　挑战阶段

数据上传后，当用户需要对数据完整性进行审计时，向 EdgeA 发起验证请求，EdgeA 收到请求后，根据状态对相应存储位置发起挑战。首先存储数据处的节点利用数据 F 与节点私钥生成签名 $\mathrm{Tag}_F^{'}=e\left(\mathrm{SK}_{\mathrm{EdgeS/CS}}H_2(\mathrm{ID}_F),P\right)$，并将其发送给 EdgeA 进行验证。若 EdgeS 未存储数据，则生成脏数据 f 的文件标签 $\mathrm{Tag}_f^{'}=e\left(\mathrm{SK}_{\mathrm{EdgeS}}\xi H_2(\mathrm{ID}_f),P\right)$，验证过程均同式(1)。若验证不成立，则说明审计数据错误，需终止程序，向用户返回结果；若验证成立，则说明存储数据正确，且不存储于错误的存储节点，继续挑战。数据审计流程如图 3 所示。

1) EdgeA 从全集 $\Omega=\{1,2,\cdots,n\}$ 中随机选择 i 个不同元素，并为每一个 $i\in I$ 生成随机值 v_i。

2) EdgeA 生成挑战 $\mathrm{Chall}=\{i,v_i\}_{i\in I}$ 并发送给存储数据者。

由于存储状态 θ 不同，挑战发送位置也不同，当 $\theta=(0,1)$ 时，数据只保存在云端，EdgeA 只需向 CS 发送挑战；当 $\theta=(1,0)$ 时，EdgeA 只需向 EdgeS 发送挑战；当 $\theta=(1,1)$ 时，EdgeA 需向 EdgeS 和 CS 都发送挑战，以验证 2 个副本的数据完整性。

3.2.5　证据生成阶段

收到挑战请求 $\mathrm{Chall}=\{i,v_i\}_{i\in I}$ 后，EdgeS/CS 执行以下步骤。

1) 计算 $\delta=\sum_{i\in I}v_i\delta_{m_i}$。

2) 计算 $\mu=\sum_{i\in I}v_i H_2(m_i)$。

先对证据信息进行哈希处理后再发送 $\mathrm{Proof}=\{\delta,\mu\}$ 给 EdgeA，以免证据信息在传输过程中被恶意窃取，从而推导出用户的数据信息。

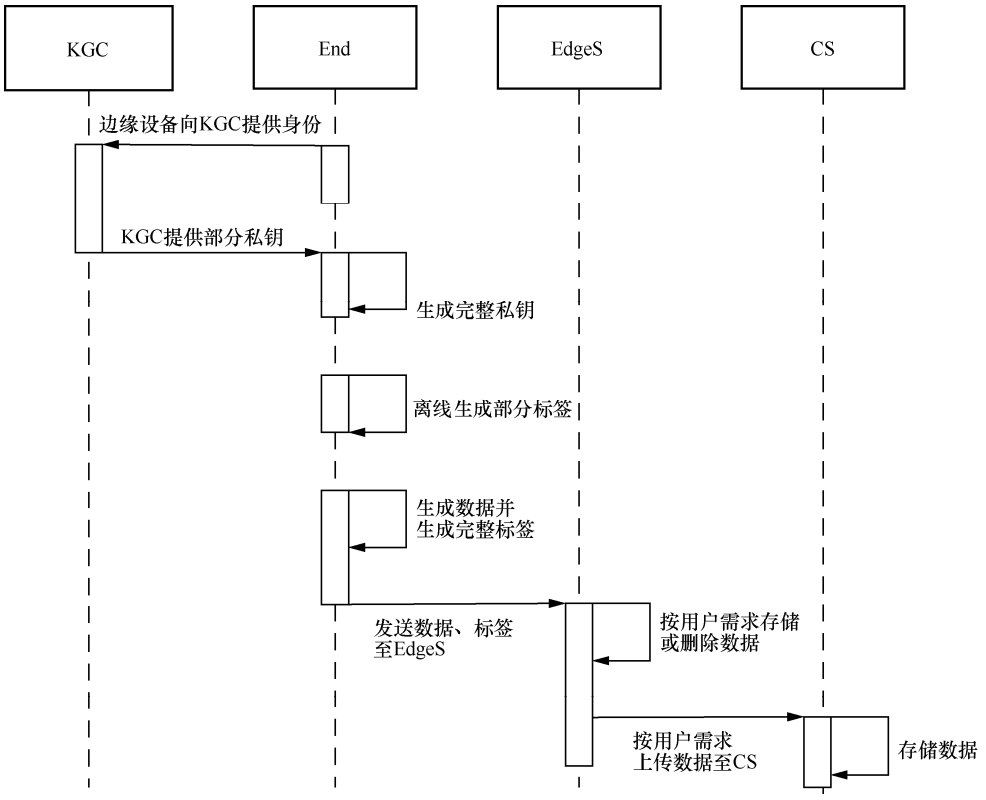

图 2　数据上传流程

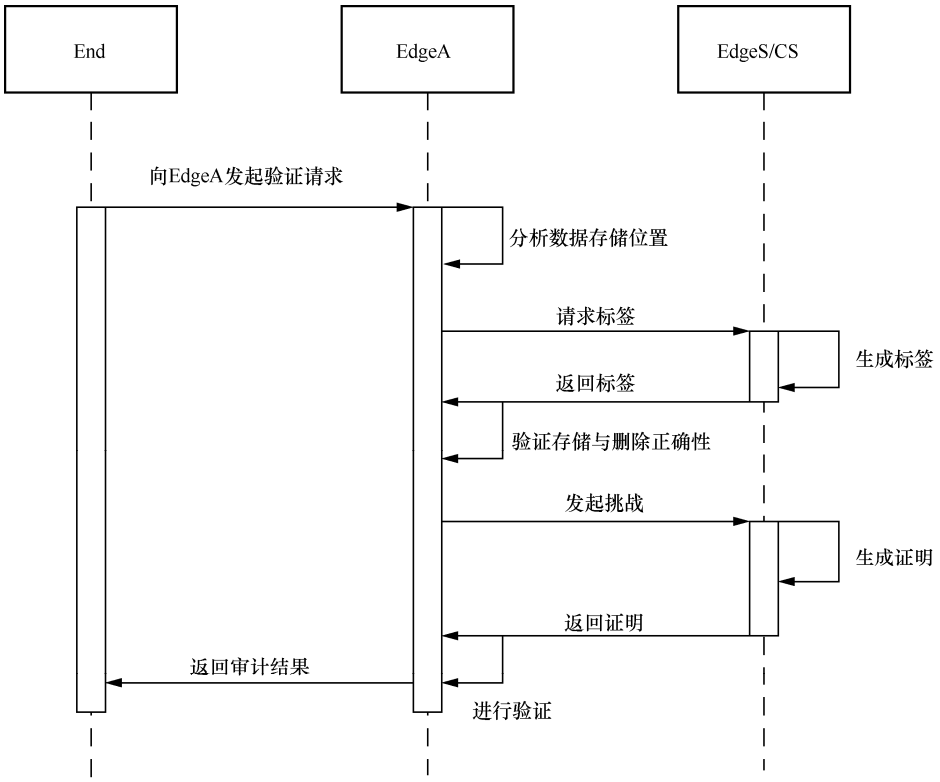

图 3　数据审计流程

3.2.6　证据验证阶段

EdgeA 收到存储位置发来的证据信息后，执行下面步骤验证数据完整性。

首先 EdgeA 计算 $h_i = H_2(\mathrm{ID}_u)$、$R = \sum_{i \in I} v_i h_i$、$Q' = H_1(\mathrm{ID}_u, X_u)$ 和 $T = H_1(P_{\mathrm{pub}})$。验证等式(3)是否成立，若成立则返回验证成功，否则返回验证失败。

$$e(\delta, P) == e(Q'R, Y_u)e(T'\mu, P) \tag{3}$$

对式(3)进行正确性分析

$$e(\delta, P) = e\left(\sum_{i \in 1} v_i \delta_{m_i}, P\right) =$$

$$e\left(\sum_{i \in I} v_i \left(\mathrm{SK}_u H_2(\mathrm{id}_i) + TH_2(m_i)\right), P\right) =$$

$$e\left(\mathrm{SK}_u \sum_{i \in I} v_i H_2(\mathrm{id}_i), P\right) e\left(T \sum_{i \in 1} v_i H_2(m_i), P\right) =$$

$$e(\mathrm{SK}_u R, P)e(T'\mu, P) =$$

$$e\left(x_u z H_1(\mathrm{ID}_u, X_u)R, P\right)e(T'\mu, P) =$$

$$e\left(H_1(\mathrm{ID}_u, X_u)R, x_u zP\right)e(T'\mu, P) =$$

$$e(Q'R, Y_u)e(T'\mu, P) \tag{4}$$

4　安全性分析

4.1　攻击模型分析

基于无证书公钥密码机制，本文提出了三类敌手攻击：Type-Ⅰ类型敌手 A_1，一般用户攻击，不能获取 KGC 主密钥但能替换任意用户私钥；Type-Ⅱ类型敌手 A_2，恶意 KGC 攻击，可以获取 KGC 主密钥但不能替换用户私钥；Type-Ⅲ类型敌手 A_3，恶意 EdgeS/CS 攻击，EdgeS/CS 可能会破坏数据并对审计者伪造证明。

首先，挑战者 C 执行本文方案的初始化阶段，生成主密钥及系统参数 $\{q, G_1, G_2, e, P, P_{\mathrm{pub}}, H_1, H_2\}$。

H_1 询问。C 管理 $\mathrm{Table}_{H_1} = \{\mathrm{ID}_i, X_i, h_1\}$。当接收到 A 发送的 $\{\mathrm{ID}_i, X_i\}$ 时，若 $\{\mathrm{ID}_i, X_i\}$ 在 Table_{H_1} 中存在，则 C 返回 h_1 给 A；否则，C 选择随机数 $h_1^{'} \in Z_q^*$，且 $h_1^{'}$ 与 L 中其他元素不相等，将 $h_1^{'}$ 返回给 A，并添加 $\{\mathrm{ID}_i, X_i, h_1^{'}\}$ 到 Table_{H_1} 中。

H_2 询问。挑战者 C 管理 $\mathrm{Table}_{H_2} = \{m_i, h_2\}$。当接收到 A 发送的 $\{m_i\}$ 时，C 查询 Table_{H_2}，若 $\{m_i, h_2\}$ 在 Table_{H_2} 中存在，返回 h_2 给 A；否则，C 选择随

机数 $h_2^{'} \in Z_q^*$，且 $h_2^{'}$ 与 Table_{H_2} 中其他元素不相等，将 $h_2^{'}$ 返回给 A，并添加 $\left\{m_i, h_2^{'}\right\}$ 到 Table_{H_2} 中。

替换公钥询问。当接收到 A 的询问 $(\text{ID}_i, \text{PK}_i^{'})$ 时，C 用 $\text{PK}_i^{'}$ 替换现存的公钥 PK_i。

确定删除询问。挑战者 C 管理 $\text{Table}_{\text{EdgeS/CS}} = \left\{\text{ID}_i, \text{ID}_F, \text{Tag}_F\right\}$，当接收到 A 发送的 $\left\{\text{ID}_i, \text{ID}_F\right\}$ 时，C 查询 $\text{Table}_{\text{EdgeS/CS}}$，若 Tag_F 已删除，返回 null 给 A。

Type-Ⅰ敌手。基于 CDH 问题困难性假设，在随机预言模型下，若本文方案对敌手 A_1 是不可伪造的，则本文方案是安全的。

Proof。对于这类攻击，A_1 可以随意替换用户公钥 (X_U, Y_U)，故挑战者 C 设 $Y_U = Q_2 = bP$。在部分私钥提取中，对两次 H_1 询问设 $Q_U^{'} = s_j\ Q_1 = s_j \cdot aP, j = 1,2$，其中 s_j 是 C 选择的随机数。因为 CS 在接收数据时已经收到所有的数据签名，所以不再进行标签生成询问。在证据生成阶段，挑战者 C 利用哈希重放向 H_2 询问同一个挑战，从而生成 2 个不同的证据 (δ_1, μ_1) 和 (δ_2, μ_2)，那么式(5)和式（6）成立

$$e(\delta_1, P) = e(Q_1^{'}R, Y_1)e(T'\mu_1, P) \tag{5}$$

$$e(\delta_2, P) = e(Q_2^{'}R, Y_2)e(T'\mu_2, P) \tag{6}$$

那么有

$$\begin{aligned} &e(\delta_2 - \delta_1, P) = \\ &e\left((s_2 - s_1)apR, bP\right)e\left(T'(\mu_2 - \mu_1), P\right) = \\ &e\left((s_2 - s_1)abpR, P\right)e\left(T'(\mu_2 - \mu_1), P\right) \end{aligned} \tag{7}$$

其中，

$$\delta_2 - \delta_1 = (s_2 - s_1)abpR + T'(\mu_2 - \mu_1) \tag{8}$$

整理可得

$$abP = \left((s_2 - s_1)R\right)^{-1}\left((\delta_2 - \delta_1) - T'(\mu_2 - \mu_1)\right) \tag{9}$$

如果 A_1 能够攻破，说明式(9)有解，与假设矛盾。

Type-Ⅱ敌手。基于 CDH 问题困难性假设，在随机预言模型下，若本文方案对敌手 A_2 是不可伪造的，则该方案是安全的。

Proof。对于恶意 KGC 攻击，A_2 可以随意替换用户主密钥，故设置系统公钥为 $P_{\text{pub}} = Q_1 = aP$，其中 a 为系统主密钥。C 猜测 $Q_u^{'} = bP$，用户秘密值为 s_i，计算 $Y_U = s_i Q_1 = s_i ap, i = 1,2$。在证据生成阶段，挑战者 C 利用哈希重放向 H_2 询问同一个挑战，从而生成 2 个不同的证据 (δ_1, μ_1) 和 (δ_2, μ_2)，那么式(10)和式(11)成立

$$\begin{aligned} &e(\delta_1, P) = \\ &e(bPR, s_1 aP)e(T'\mu_1, P) = \\ &e(s_1 abPR, P)e(T'\mu_1, P) \end{aligned} \tag{10}$$

$$\begin{aligned} &e(\delta_2, P) = \\ &e(bPR, s_2 aP)e(T'\mu_2, P) = \\ &e(s_2 abPR, P)e(T'\mu_2, P) \end{aligned} \tag{11}$$

那么有

$$\begin{aligned} &e(\delta_2 - \delta_1, P) = \\ &e\left(bpR, (s_2 - s_1)aP\right)e\left(T'(\mu_2 - \mu_1), P\right) = \\ &e\left((s_2 - s_1)abPR, P\right)e\left(T'(\mu_2 - \mu_1), P\right) \end{aligned} \tag{12}$$

其中，

$$\delta_2 - \delta_1 = (s_2 - s_1)abPR + T'(\mu_2 - \mu_1) \tag{13}$$

整理可得

$$abP = \left((s_2 - s_1)R\right)^{-1}\left((\delta_2 - \delta_1) - T'(\mu_2 - \mu_1)\right) \tag{14}$$

如果 A_2 能够攻破，说明式(14)有解，与假设矛盾。

Type-Ⅲ敌手。基于 CDH 问题困难性假设，在随机预言模型下，若本文方案对敌手 A_3 是不可伪造的，则本文方案是安全的。

Proof。对于这类攻击，敌手 A_3 根据挑战者 C 发起的挑战信息 $\text{Chall} = \left\{i, v_i\right\}_{i \in I}$，返回伪造的证明 $\text{Proof}^* = \left\{\delta^*, \mu^*\right\}$，并且该证明满足

$$e(\delta^*, P) = e(Q'R, Y_u)e(T'\mu^*, P) \tag{15}$$

诚实的审计者返回真实的证明 $\text{Proof} = \left\{\delta^*, \mu\right\}$

$$e(\delta^*, P) = e(Q'R, Y_u)e(T'\mu, P) \tag{16}$$

整理式(15)和式(16)，可得 $\mu^* = \mu$，与假设矛盾。

综上所述，本文方案能够抵抗三类敌手攻击，因此本文方案是安全的。

4.2 安全特性

本文方案满足以下几个性质。

1) 公开可验证。通过证据生成和证据验证阶段，审计者 EdgeA 可以使用公共参数来验证数据的完整性，而不需要终端提供秘密值。

2) 存储正确性。只有当数据正确时，产生的证明才能通过三类敌手游戏模型。

3) 隐私保护性。验证边缘节点 EdgeA 无法从

挑战证明中获取数据信息。审计过程中，EdgeA 接收到 CS 或 EdgeS 发送的证明，$(\mathrm{ID}_u,\mathrm{ID}_{\mathrm{Edge}},\mathrm{ID}_{\mathrm{CS}},\mathrm{Tag}_F,\mathrm{Tag}_f)$部分不包含具体的数据信息，下面对 μ 进行分析。EdgeA 能够忠实地完成用户审计任务，但会对数据产生好奇，CS 或 EdgeS 发送 $\mu=\sum_{i\in I}v_iH_2(m_i)$ 给 EdgeA，EdgeA 可以通过多次挑战求解线性方程组，从而获取 $H_2(m_i)$。根据哈希函数的不可逆性质，EdgeA 无法推导出数据块 m_i 的值，从而有效对数据进行了隐私保护。

5　实验与分析

5.1　实验环境配置

本文方案共涉及 4 种类型的实体，在本文的实验中，4 种实体由 4 种具有不同功能和配置的机器模拟。具体来说，云端采用阿里云服务器 ECS 部署，拥有 2 GHz 处理器、2 GB 内存，边缘节点采用阿里云边缘节点服务 ENS 部署，拥有 2 GHz 处理器、1 GB 内存，终端与 KGC 部署在本地计算机上，采用 Intel i7-8086K 4 GHz 处理器、16 GB 内存。以上均基于 Ubuntu14.04 版本系统搭建，openssl1.0.1 版本。

在实验中，本文使用 C++和 Python 进行代码编写，GMP 库和 PBC 库实现密码操作。采用类型 A 的配对曲线，参数类型 A 提供在所有默认设置中具有最快速度的对称配对参数，哈希函数为 SHA_1（160 bit）。所得实验结果均为多次计算后的平均值。

5.2　效率分析

在边缘环境中，对于资源有限的边缘设备用户来说，计算复杂度越小越好。如何在有限的资源限制下合理分配资源、缩短数据传输时的计算时间、提高计算效率是本文亟须解决的问题。本节将本文方案与另外 5 种审计方案进行对比，并将审计算法放在相同的实验条件下，因此具有一定的可比性。

表 2 给出了数据上传时的时间复杂度分析。在用户层，CLPDP 与 tHKWWC 方案需 n 次哈希运算（H）与 $2n$ 次点乘运算（Mul），CL-PAS 方案需 $2n$ 次哈希运算与 $2n$ 次点乘运算，以上 3 种方案属于轻量级审计方案。边缘环境下，文献[19]方案将用户侧计算任务全部卸载到边缘，因此用户侧有较低时延，边缘侧进行了 n 次哈希运算、n 次点乘运算以及 $2n$ 次指数运算（Exp）；文献[20]方案将边缘侧视为半可信，因此计算均在本地进行，时间复杂度同样为 n 次哈希运算、n 次点乘运算以及 $2n$ 次指数运算；本文方案由于将部分计算移至离线阶段，在线阶段只需进行 n 次哈希运算与 n 次点乘运算即可，时间复杂度最低，显著提高了数据上传时的计算效率。

表 2　数据上传时的时间复杂度分析

方案	用户层	边缘层	边缘节点
CLPDP	$n\mathrm{H}+2n\mathrm{Mul}_{Z_p}$	—	—
tHKWWC	$n\mathrm{H}+2n\mathrm{Mul}_{Z_p}$	—	—
CL-PAS	$2n\mathrm{H}+2n\mathrm{Mul}_{Z_p}$	—	—
文献[19]	$n\mathrm{H}+n\mathrm{Mul}_{Z_p}+2n\mathrm{Exp}$	—	可信
文献[20]	$n\mathrm{H}+n\mathrm{Mul}_{Z_p}+2n\mathrm{Exp}$	—	半可信
本文	$n\mathrm{H}+n\mathrm{Mul}_{Z_p}$	—	半可信

表 3 为初始化阶段生成操作用时。每台用户设备在接入网络后需执行初始化阶段生成密钥，考虑到密钥有泄露风险，在怀疑密钥泄露或一定周期后，应再次执行初始化阶段产生新的密钥，但每次执行的时间开销与标签生成的时间开销相比可以忽略不计。

表 3　初始化阶段生成操作用时

角色	系统参数/ms	部分私钥/ms	完全密钥/ms
KGC	2.341	0.859	—
用户	—	—	2.652

生成标签步骤分为离线标签生成与在线标签生成 2 个阶段，为保证用户设备能够以最低的开销进行计算，本文首先对上传文件块大小进行了评估。以本文方案与 CLPDP、tHKWWC、CL-PAS 作为评估方案，对其在文件数据总量分别为 200 MB、400 MB、600 MB、800 MB 和 1 000 MB 的在线标签生成实验进行了模拟，对比了文件块大小对标签生成时间的影响，结果如图 4 所示。

从图 4 可以看出，随着文件块大小的增长，标签生成时间呈指数形式递减。

考虑到文件传输以及安全性，文件块越大，给数据传输带来的困难就越大，完整性验证就越不能达到预期效果，因此并不能无限制增大传输文件块的大小，分析实验结果后，以文件块大小等于 4 096 bit 为基准进行后续的对比实验。

由于标签生成阶段计算开销在总计算开销中占比较大，执行步骤较多，下面将对标签生成阶段总执行时间和数据上传时用户执行时间分别进行说明。

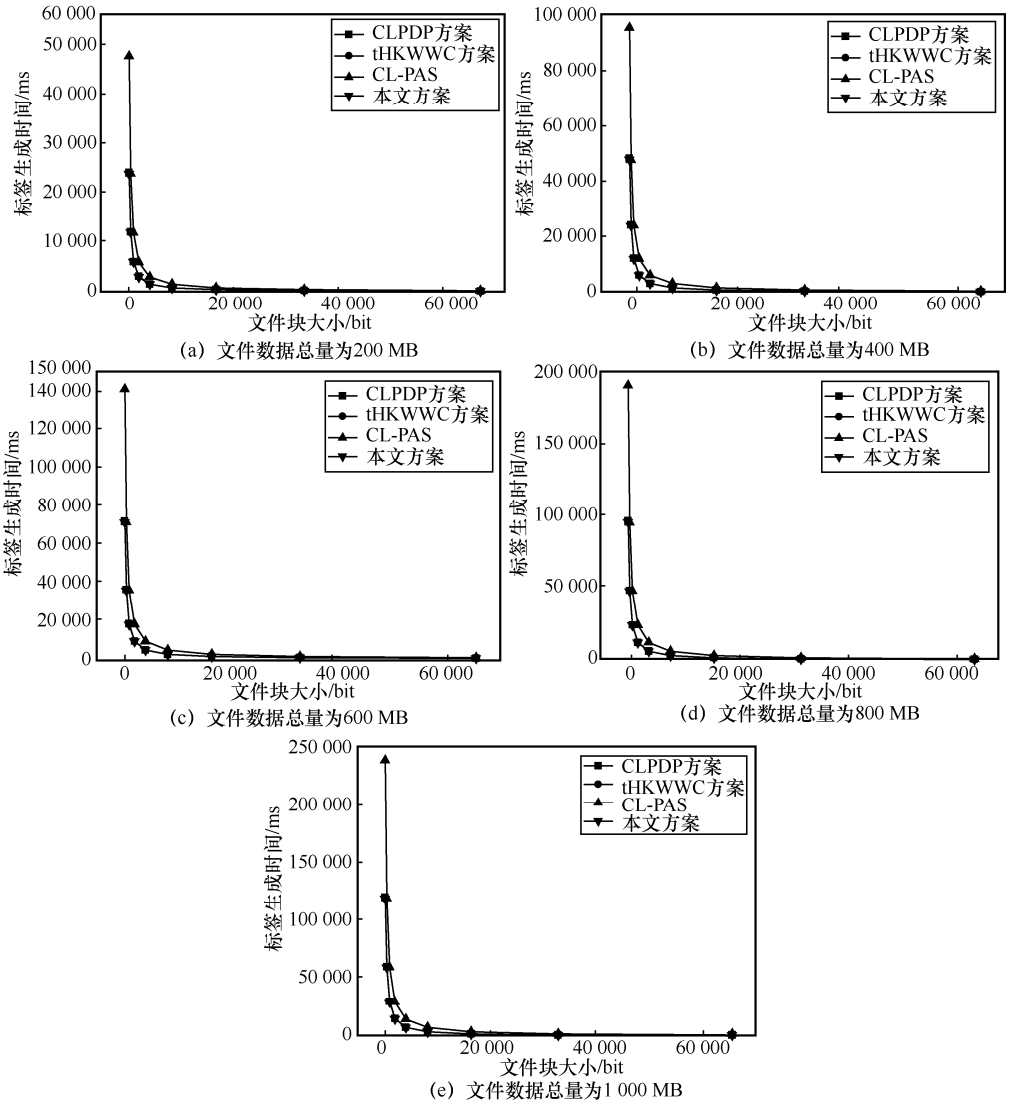

图 4 文件块大小对标签生成时间的影响

标签生成阶段总执行时间对比如图 5 所示。从图 5 中可以看出，随着文件块数量的增加，生成在线标签的时间呈线性增长，其中 CLPDP 与 tHKWWC 方案需 n 次哈希运算与 $2n$ 次点乘运算；CL-PAS 方案与本文方案需 $2n$ 次哈希运算与 $2n$ 次点乘运算，用时相对较高，这是因为 CL-PAS 方案与本文方案对数据进行了哈希处理，以防审计者从中获取数据信息；文献[19]方案与文献[20]方案均进行了 n 次哈希运算、n 次点乘运算以及 2 次指数运算，虽然计算位置不同，但时间开销最高。

数据上传时用户执行时间如图 6 所示。从图 6 中可以看出，随着文件块数量的增加，生成离线标签的时间呈线性增长。本文方案效率最高，其中文献[19]方案是在上传数据后再进行的数据预处理，因此时间开销最低。

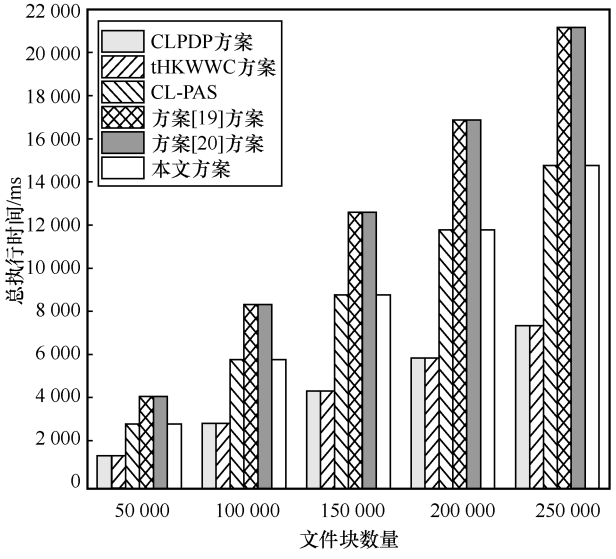

图 5 标签生成阶段总执行时间对比

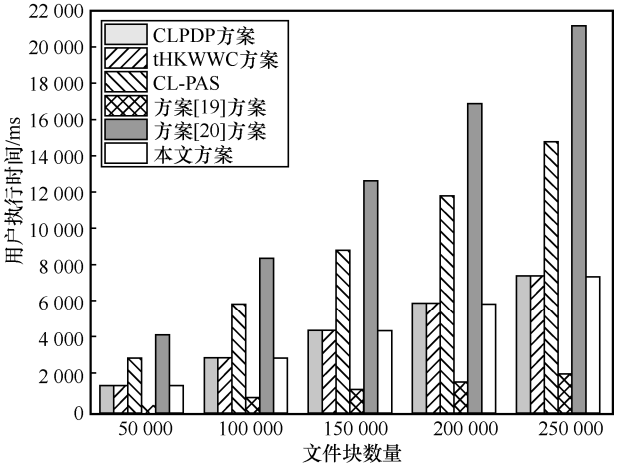

图 6　数据上传时用户执行时间

随着文件数据总量的增加，标签生成的计算时间呈线性增长。另外，在相同大小的文件数据总量下，文件块数量越多，文件块大小越小，需要计算的标签数相应增加，因此计算时间相应增长。虽然文件数据总量以及计算时间相对较大，但是只需要进行单次计算即可生成标签，并且可以在挑战验证过程中重复使用，因此实验结果可以接受。

在挑战和验证阶段，EdgeA 在接收到用户的审计请求后，向 CS 或 EdgeS 发起挑战，CS 或 EdgeS 生成证明并将之返回给 EdgeA，然后 EdgeA 根据持有的公钥对证明进行验证。

根据 Ateniese 等[1]提出的方案，假设模拟的数据块数量为 100 000，如果云服务器污染了 1%，验证者可以挑战 460 个区块使检测服务器错误的概率达到 99%。当验证的区块越小时，验证所花费的时间就越短，同时验证检测率也会下降，当验证区块数为 300 时，检测率在 95%以上。接下来，给出挑战阶段取样 300 与 460 个区块时各审计方案的时间对比。

对于不同的文件数据总量，在固定了文件块大小的情况下，挑战与验证时间不变，说明证据生成与证明验证的计算时间与文件数据总量无关。用户选择的存储状态不同，审计的目标与大小也不同，当数据在 EdgeS 与 CS 中均进行存储时，证据生成与证明验证时间也会成倍增长，为表示统一，实验阶段均以 $\theta=(0,1)$ 状态进行比较。

证据生成阶段时间如表 4 所示。证明验证阶段时间如表 5 所示。各方案生成证据时间相差不大，均进行了双线性对运算。与整体时间开销相比，证据生成、证明验证用时占比并不大，因此实验结果是可以接受的。

5.3　存储开销

本文方案的额外存储开销集中在 EdgeS 与 CS，主要是上传文件时伴随的文件块标签集合，而用户自身上传数据后只需保留文件哈希值。

单个文件块标签为 $\delta_i=\mathrm{SK}_U H_2(\mathrm{ID}_i)+TH_2(m_i)$，而存储开销的多少与划分的文件块数量 n，以及每个文件块的长度大小 $|q|$ 相关。由此可知，单个存储位置下的存储开销为 $2n|q|$。

6　结束语

本文提出的边缘环境下基于无证书公钥密码学的在线/离线数据完整性审计方案具备保证用户安全、复杂度低的特点。基于无公钥密码学思想，在离线阶段进行部分签名的生成操作，使用户设备上传数据时的效率有了显著提升。针对数据不同存储状态，本文方案能够合理审计数据的正确性和完整性。在随机预言模型下，基于 CDH 问题证明了本文方案的安全性。模拟分析审计方案中各阶段的主要运算开销，结果表明本文方案在边缘环境下的用户设备上传数据时间开销最低。

表 4　证据生成阶段时间

取样数/个	CLPDP 方案/ms	tHKWWC 方案/ms	CL-PAS/ms	文献[19]方案/ms	文献[20]方案/ms	本文方案/ms
300	7.92	7.84	11.46	14.32	13.10	10.98
460	12.23	11.98	16.84	17.33	16.87	16.45

表 5　证明验证阶段时间

取样数/个	CLPDP 方案/ms	tHKWWC 方案/ms	CL-PAS/ms	文献[19]方案/ms	文献[20]方案/ms	本文方案/ms
300	17.43	16.93	25.42	36.45	37.58	24.37
460	26.68	26.33	38.96	52.39	53.18	37.21

参考文献：

[1] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//Proceedings of the 14th ACM Conference on Computer and Communications Security. New York: ACM Press, 2007: 598-609.

[2] 谭霜, 贾焰, 韩伟红. 云存储中的数据完整性证明研究及进展[J]. 计算机学报, 2015, 38(1): 164-177.
TAN S, JIA Y, HAN W H. Research and development of provable data integrity in cloud storage[J]. Chinese Journal of Computers, 2015, 38(1): 164-177.

[3] SHI W S, PALLIS G, XU Z W. Edge computing scanning the issue[J]. Proceedings of the IEEE, 2019, 107(8):1474-1481.

[4] 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望[J]. 计算机研究与发展, 2019, 56(1): 69-89.
SHI W S, ZHANG X Z, WANG Y F, et al. Edge computing: state-of-the-art and future directions[J]. Journal of Computer Research and Development, 2019, 56(1): 69-89.

[5] 张佳乐, 赵彦超, 陈兵, 等. 边缘计算数据安全与隐私保护研究综述[J]. 通信学报, 2018, 39(3): 1-21.
ZHANG J L, ZHAO Y C, CHEN B, et al. Survey on data security and privacy-preserving for the research of edge computing[J]. Journal on Communications, 2018, 39(3): 1-21.

[6] OQAILY M, JARRAYA Y, MOHAMMADY M, et al. SegGuard: segmentation-based anonymization of network data in clouds for privacy-preserving security auditing[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(5): 2486-2505.

[7] YANG Y, CHEN Y J, CHEN F. A compressive integrity auditing protocol for secure cloud storage[J]. IEEE/ACM Transactions on Networking, 2021: doi.org/10.1109/TNET.2021.3058130.

[8] CHEN X Y, SHANG T, ZHANG F, et al. Dynamic data auditing scheme for big data storage[J]. Frontiers of Computer Science, 2020, 14(1): 219-229.

[9] KONSTA A, MYTILINIS I, DOKA K, et al. Clouseau: blockchain-based data integrity for HDFS clusters[C]//Proceedings of 2021 IEEE 37th International Conference on Data Engineering. Piscataway: IEEE Press, 2021: 2725-2728.

[10] TIAN M, YE S, ZHONG H, et al. Identity-based proofs of storage with enhanced privacy[C]//Proceedings of International Conference on Algorithms and Architectures for Parallel Processing. [S.l.:s.n.], 2018: 461-480.

[11] SHEN W T, 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, 2019, 14(2): 331-346.

[12] LI Y N, YU Y, MIN G Y, et al. Fuzzy identity-based data integrity auditing for reliable cloud storage systems[J]. IEEE Transactions on Dependable and Secure Computing, 2019, 16(1): 72-83.

[13] WANG H Q. Identity-based distributed provable data possession in multicloud storage[J]. IEEE Transactions on Services Computing, 2015, 8(2): 328-340.

[14] YU H Y, CAI Y Q, SINNOTT R O, et al. ID-based dynamic replicated data auditing for the cloud[J]. Concurrency and Computation: Practice and Experience, 2019, 31(11): 1-12.

[15] WANG F, XU L, WANG H Q, et al. Identity-based non-repudiable dynamic provable data possession in cloud storage[J]. Computers & Electrical Engineering, 2018, 69: 521-533.

[16] HE D B, KUMAR N, WANG H Q, et al. Privacy-preserving certificateless provable data possession scheme for big data storage on cloud[J]. Applied Mathematics and Computation, 2017, 314: 31-43.

[17] JI Y Y, SHAO B L, CHANG J Y, et al. Privacy-preserving certificateless provable data possession scheme for big data storage on cloud, revisited[J]. Applied Mathematics and Computation, 2020, 386: 125478.

[18] GAO G M, FEI H X, QIN Z F. An efficient certificateless public auditing scheme in cloud storage[J]. Concurrency and Computation: Practice and Experience, 2020, 32(24): e5924.

[19] WANG T, MEI Y X, LIU X X, et al. Edge-based auditing method for data security in resource-constrained Internet of things[J]. Journal of Systems Architecture, 2021, 114: 101971.

[20] LIU D Z, SHEN J, VIJAYAKUMAR P, et al. Efficient data integrity auditing with corrupted data recovery for edge computing in enterprise multimedia security[J]. Multimedia Tools and Applications, 2020, 79(15/16): 10851-10870.

[21] LI B, HE Q, CHEN F F, et al. Auditing cache data integrity in the edge computing environment[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(5): 1210-1223.

[22] 边缘计算产业联盟(ECC)与工业互联网产业联盟(AII)联合发布.边缘计算安全白皮书[R]. 2019.
Edge Compting Consortium (ECC) and Alliance of Industrial Internet (AII) Jointly Publish. Edge compting security white paper [R]. 2019.

[作者简介]

王子园（1996− ），男，河北保定人，河北大学博士生，主要研究方向为信息安全、数据完整性审计。

杜瑞忠（1975− ），男，河北献县人，博士，河北大学教授、博士生导师，主要研究方向为可信计算、信息安全等。