# 基于区块链和雾计算的去中心化云端数据完整性审计方案

杨小东*[①] 王秀秀[①] 李茜茜[①] 周 航[①] 王彩芬[②]

[①](西北师范大学计算机科学与工程学院 兰州 730070)

[②](深圳技术大学大数据与互联网学院 深圳 518118)

**摘 要**：针对传统云端数据完整性验证方案中存在过度依赖完全可信第三方审计者(TPA)、复杂的密钥管理和不支持数据访问者授权等问题，该文提出一种基于区块链和雾计算的去中心化数据完整性审计方案。为了实现审计方案的去中心化，使用雾节点和智能合约代替第三方审计者。利用区块链设计智能合约保障方案中各个实体的公平交易；将审计过程生成的证据存储在区块链中以防止各个实体的不诚实行为。引入无证书密码体制，解决了传统审计方案中复杂的密钥托管和证书管理问题。此外，通过加密累加器实现访问用户授权和身份认证。分析结果表明，该方案满足签名的不可伪造性，与同类方案相比具有较高的计算性能。

**关键词**：云存储；完整性验证；区块链；雾计算；无证书

**中图分类号**：TN915.08; TP309 **文献标识码**：A **文章编号**：1009-5896(2023)10-3759-08

**DOI**: 10.11999/JEIT210717

# Decentralized Integrity Auditing Scheme for Cloud Data Based on Blockchain and Edge Computing

YANG Xiaodong[①] WANG Xiuxiu[①] LI Xixi[①] ZHOU Hang[①] WANG Caifen[②]

[①](*College of Computer Science and Engineering*, *Northwest Normal University*, *Lanzhou* 730070, *China*)

[②](*College of Big Data and Internet*, *Shenzhen Technology University*, *Shenzhen* 518118, *China*)

**Abstract**: Focusing on the problems of over-reliance on Third-Party Auditors (TPA), complex key and certificate management and data leakage in traditional cloud data integrity auditing scheme, a decentralized data integrity verification audit scheme based on blockchain and edge computing is proposed. In order to achieve the decentralization of the audit scheme, fog nodes and smart contracts are used to replace the third-party auditors. Using blockchain to design smart contracts to ensure fair transactions among entities. The proofs generated by audit process are stored in the blockchain, which can prevent the dishonest behavior of each entity. The certificateless cryptosystem is introduced to solve the complex key escrow and certificate management problems in the traditional audit scheme. In addition, the access authorization and identity authentication of cloud data users is realized through encrypted accumulators. The analysis results show that this scheme satisfies the robustness of audit and the unforgeability of signatures, and has higher computing performance compared with similar schemes.

**Key words**: Cloud storage; Integrity verification; Blockchain; Fog computing; Certificateless

收稿日期：2021-07-15；改回日期：2022-07-25；网络出版：2023-08-11

*通信作者： 杨小东 y200888@163.com

基金项目：国家自然科学基金(61662069, 61562077)，中国博士后科学基金(2017M610817)，兰州市科技计划项目(2013-4-22)，西北师范大学青年教师科研能力提升计划(NWNU-LKQN-14-7)

Foundation Items: The National Natural Science Foundation of China (61662069, 61562077), China Postdoctoral Science Foundation (2017M610817), The Science and Technology Project of Lanzhou City (2013-4-22), The Foundation of Northwest Normal University (NWNU-LKQN-14-7)

## 1 引言

随着云存储技术不断发展成熟，云服务器可以通过互联网为用户提供高效快捷的外包存储服务[1]。与此同时，存储在云端的数据面临着许多安全威胁[2]，而云服务商(Cloud Service Providers, CSP)也可能为了自身利益谎称存储在云端的数据是完整的[3]。因此，数据完整性验证对云存储安全具有重要的意义。

传统的审计方案一般将审计任务外包给一个第三方审计者(Third-Party Audit, TPA)[4]，采用随机抽样的方法，在无需下载文件的情况下便可以验证其完整性。审计方案按照数据的可恢复性可以分为数据持有性证明(Provable Data Possession, PDP)[5]和数据可恢复性证明(Proofs of Retrievability, POR)[6]。这些传统的验证机制大多基于公钥基础设施(Public Key Infrastructure, PKI)，但是PKI中存在着复杂的证书的生成、存储、分发和撤销[7]等问题。为了解决传统密码体制中证书管理的问题，一些学者设计了基于身份的签名方案[8–10]。然而在基于身份的签名方案中，用户签名私钥都由一个完全可信第三方生成，即私钥生成中心(Private Key Generator, PKG)，因此存在密钥托管问题。相比基于PKI和身份的签名方案，无证书签名方案[11–13]很好地解决了证书管理和密钥托管问题。

在以上方案中，审计任务都依赖TPA并且假定TPA完全可信。事实上，TPA可能会为了自身利益与CSP串通将错误的审计结果返回给用户。因此，用户对于审计结果的正确性无从得知。区块链是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式，具有可追踪、不可篡改、匿名与开放等特点。为解决云审计方案中过度依赖完全可信第三方审计者的问题，不少学者将区块链技术应用到审计模型中。文献[14]提出了一个针对拖延审核员的无证书公开认证方案，但是用户端的计算开销过高。Wang等人[15]提出了基于区块链的数据完整性验证方案，使用一个可信的审计员代替TPA，但是方案存在密钥和证书管理的问题。

此外，这些方案都只关注对个人数据的完整性验证，并未考虑数据在多个用户之间的安全共享问题。文献[16]设计了一种组共享数据的完整性验证方案，但是方案基于群签名算法，计算开销较大。密码累加器最早由Nguyen提出[17]，可用于验证元素在集合中的成员关系。文献[18,19]将密码累加器用作共享数据的身份验证，减少了组内成员的验证延迟，受此启发，本文将使用密码累加器实现访问用户授权和身份认证。

雾节点(Fog Node, FN)作为物联网设备与云之间的中间件，具有基础的计算和存储能力，可以满足对数据处理和传输的需求。使用雾节点辅助云计算可以减轻云服务器的计算开销并且提高通信效率。Tian等人[20]提出了雾云计算中的公共审计方案，方案中使用雾节点对存储数据进行预处理。在基于属性的云端数据安全共享方案中[21,22]，雾节点经常以外包计算方式执行部分解密工作从而减轻客户端的计算开销。

在上述研究的基础上，本文设计了一个基于无证书的云端数据完整性验证方案，方案使用雾节点和智能合约代替审计端实现了去中心化。具体贡献如下：

(1) 审计模型的去中心化。使用雾节点和智能合约代替TPA的审计工作，从而解决了传统审计方案中过度依赖可信TPA的问题。

(2) 基于智能合约的公平支付。使用雾节点和智能合约进行完整性验证，只有通过完整性验证，CSP才可以获得服务报酬。

(3) 基于无证书签名的数据完整性验证。基于无证书签名体制，解决了传统审计方案中复杂的密钥托管和证书管理问题。

(4) 用户的访问授权和身份认证。针对多用户共享问题，使用加密累加器对有效用户进行访问授权，即只有通过身份认证的合法用户才能使用云端数据。

## 2 准备知识

### 2.1 复杂性假设

**定义1** CDH(Computational Diffe-Hellman)问题：已知G是一个循环乘法群，$g$是G的生成元，$(g, g^{\alpha}, g^{\beta}) \in \mathrm{G}^3$，其中$\alpha, \beta \in Z_{\mathrm{q}}^*$未知，那么群上G的CDH问题是计算$g^{\alpha\beta} \in \mathrm{G}$。

**定义2** CDH假设：如果任何一个概率多项式时间算法A，能求解G上的CDH问题的概率是可忽略，则CDH假设是成立的，可以被定义为$\mathrm{Adv}_{\mathrm{G\,A}}^{\mathrm{CDH}} = \Pr[\mathrm{A}(g, g^{\alpha}, g^{\beta}) = g^{\alpha\beta} : \alpha, \beta \xleftarrow{\mathrm{R}} Z_{\mathrm{q}}^*] \leq \varepsilon$。

### 2.2 安全模型

本文方案基于无证书签名体制，其安全模型包含两类敌手，分别是不诚实的用户和半诚实的密钥生成中心(Key Generation Center, KGC)，使用Ⅰ类敌手$\mathcal{A}_1$和Ⅱ类敌手$\mathcal{A}_2$进行模拟。

Ⅰ类敌手$\mathcal{A}_1$：无法获取系统主密钥和部分密钥，但是可以替换合法用户公钥和秘密值；

Ⅱ类敌手$\mathcal{A}_2$：可以获得系统主密钥和部分密钥，但是不能查询和替换合法用户公钥。

本文在随机预言机模型中模拟了敌手$\mathcal{A}_1$、$\mathcal{A}_2$分别与挑战者$\mathcal{C}$之间的游戏 I 、游戏 II 。该类模型在文献[1,11]已有详细的证明，此处不再赘述。

## 3　系统模型及智能合约设计

### 3.1　系统模型

本文设计了基于区块链和雾计算的去中心化云端数据完整性验证方案。如图1所示，系统模型包括如下角色，KGC, CSP，数据拥有者(Data Owner, DO)、数据使用者(Users)、FN和区块链(Block Chain, BC)。

(1) KGC：负责生成系统主密钥和部分密钥。

(2) CSP：负责存储数据并保证数据的安全性和完整性。

(3) DO：具有授权数据使用者和定期向CSP发起完整性验证挑战的权利。

(4) Users：是DO授权的数据使用者，当Users使用数据时，CSP需要对其权限进行验证。

(5) FN：负责完整性验证过程中复杂的运算以减少区块链的计算开销。

(6) BC：是一个底层不可篡改的数据库，存储了智能合约算法、交易和证据。本文利用以太坊区块链作为底层公共区块链。

### 3.2　智能合约设计

本文智能合约实现证据存储、完整性验证和公平支付。为了节约成本，智能合约的设计尽可能简短。方案设计了3种智能合约，具体如合约1～合约3。

智能合约T0：根据存储需求，实现向区块链存储数值的功能。

智能合约T1：保证完整性验证通过时，DO向CSP支付服务费用。

智能合约T2：实现数据完整性验证和公平支付。将雾节点存储的计算值进行验证，如果验证通过，激活智能合约T1，支付CSP服务金额；否则，CSP将不能得到任何报酬。

## 4　本文方案

### 4.1　系统建立

#### 4.1.1　系统初始化

给定安全参数$\lambda$，KGC选择两个阶为素数$p$的循环群$\mathrm{G}$和$\mathrm{G_T}$，一个$\mathrm{G}$的生成元$g$和一个双线性映射$\mathrm{e}:\mathrm{G}\times\mathrm{G}\to\mathrm{G_T}$；选取4个抗碰撞的哈希函数$\mathrm{H_1}:\{0,1\}^*\to\mathrm{G}$, $\mathrm{H_2}:\mathrm{G}\to Z_{\mathrm{p}}^*$, $\mathrm{H_3}:\{0,1\}^*\to\mathrm{G}$, $\mathrm{H_4}:\{0,1\}^*\to Z_{\mathrm{p}}^*$和两个伪随机函数$\pi:Z_{\mathrm{p}}^*\times\{0,1\}^*=Z_{\mathrm{p}}^*$, $\mathrm{prf}:Z_{\mathrm{p}}^*\times\{0,1\}^*=Z_{\mathrm{p}}^*$；随机选取$s\in Z_{\mathrm{p}}^*$作为系统主密钥，计算系统公钥$\mathrm{p_{pub}}=g^s$。

(1) 部分密钥生成：KGC根据数据拥有者DO的身份id，计算部分密钥$\mathrm{S_r}=\mathrm{H_1(id)}^s$并通过安全信道将部分密钥$\mathrm{S_r}$发送给DO。

(2) 秘密值生成：DO随机选取秘密值$\alpha\in Z_{\mathrm{p}}^*$并且计算私钥$\mathrm{sk_r}=\{\mathrm{sk_1}=\mathrm{S_r}=\mathrm{H_1(id)}^s,\mathrm{sk_2}=\alpha\cdot\mathrm{H_2(S_r)}\}$。

(3) 公钥生成：DO计算公钥$\mathrm{pk_r}=g^{\mathrm{sk_2}}$；KGC公开系统公共参数$\mathrm{params}=(\mathrm{G},\mathrm{G_T},p,g,\mathrm{e},\mathrm{H_1},\mathrm{H_2},\mathrm{H_3},\mathrm{H_4},\pi,\mathrm{p_{pub}})$。

#### 4.1.2　用户授权

假如数据使用者Users的成员数$N$，使用$\mathrm{id}_i(i\in[1,N])$表示其成员的唯一身份标识，DO使用密码累加器进行授权。具体如下：

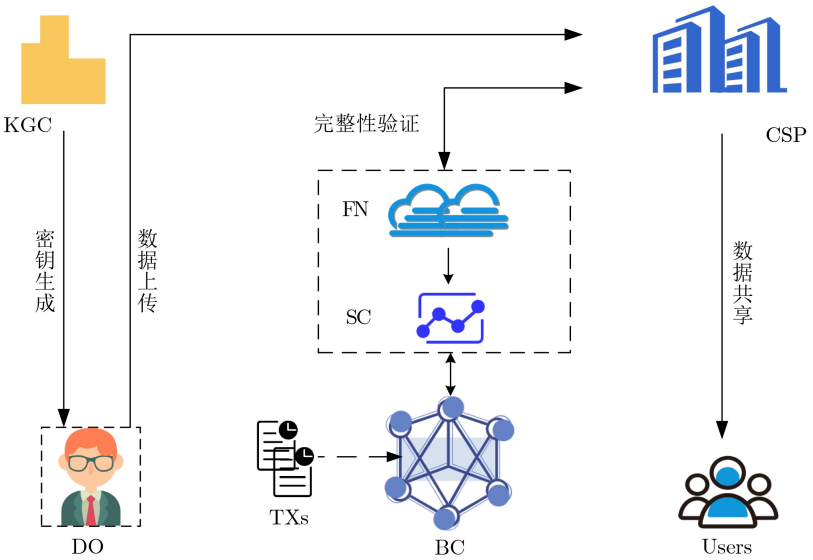


图 1 系统模型图

**合约1 智能合约T0**

输入：存储数值$d$
输出：存储结果“0/1”

```
1. Storage{
2.   If 输入值==null
3.     提示错误;
4.   Else
5.     存储d;
6.   End if
7. }
```

**合约2 智能合约T1**

输入：收款账户CSP、支付金额am、验证时间$\mathrm{ti}_1$
输出：支付时间$\mathrm{ti}_2$

```
1. Promise{
2.   If 验证结果==1
3.     DO向CSP支付金额am;
4.   End if
5. }
```

**合约3 智能合约T2**

输入：雾节点计算值$\mathrm{S}_1$和$\mathrm{S}_2$
输出：验证结果“0/1”

```
1. Promise{
2.   If S1==S2
3.     激活智能合约T1;
4.   End if
5. }
```

DO首先随机选择$\kappa$作为累加器的秘密值，随后定义累加器$\mathrm{acc(id)}=g^{\prod_{i=1}^{n}(\kappa+\mathrm{id}_i)}$和成员$\mathrm{id}_i$的见证$W_{\mathrm{id}_i}=g^{\prod_{j=1,j\neq i}^{n}(\kappa+\mathrm{id}_i)}$。最后，DO公布累加器的公共参数$\{g^{\kappa},\{g^{\mathrm{id}_i},W_{\mathrm{id}_i}\}\}$。

当任意用户请求访问该文件时，CSP验证等式(1)和(2)是否成立。

$$\mathrm{e}\left(g^{\kappa}\cdot g^{\mathrm{id}_i},\mathrm{W}_{\mathrm{id}_i}\right)=\mathrm{e}\left(\mathrm{acc}(\mathrm{id}_i),g\right) \tag{1}$$

$$\begin{aligned}\mathrm{e}(g^{\kappa}\cdot g^{\mathrm{id}_i},g^{\prod_{j=1,j\neq i}^{n}(\kappa+\mathrm{id}_i)})&=\mathrm{e}(g^{(\kappa+\mathrm{id}_i)\prod_{j=1,j\neq i}^{n}(\kappa+\mathrm{id}_i)},g)\\&=\mathrm{e}(g^{\prod_{i=1}^{n}(\kappa+\mathrm{id}_i)},g)\\&=\mathrm{e}(\mathrm{acc(id)},g)\end{aligned} \tag{2}$$

如果等式(1)和等式(2)成立，则该用户为授权用户，否则为非授权用户。

#### 4.1.3 签名生成

(1) 文件初始化：DO将文件F分为$n$块$\mathrm{F}=f_1||f_2||\cdots f_n$，对于$f_i(i\in[1,n])$，随机选取$s_i\in Z_{\mathrm{p}}^{*}$，计算对称密钥$k_i=\mathrm{prf}(s_i,f_i)$。使用密钥$k_i$分别对数据块$f_i$进行对称加密得到$m_i=\mathrm{ENC}(f_i,k_i)$，令$\mathrm{M}=\{m_i\}$。

(2) 块签名：对于密文$m_i$，DO随机选择$\beta_i\in Z_{\mathrm{p}}{}^{*}$，计算块标签$\varsigma_i=m_i\beta_i$和块签名$\tau_i=(\mathrm{sk}_1\omega_i{}^{\beta_i\mathrm{sk}_2})^{m_i}$，其中$\omega_i=\mathrm{H}_3(\mathrm{F},n,m_i)$。

(3) 签名验证：DO上传密文M、对称密钥$\{k_i\}$、块标签$\{\varsigma_i\}$和块签名$\{\tau_i\}$到CSP。CSP验证等式(3)是否成立。

$$\begin{aligned}\mathrm{e}\left(\prod_{i=1}^{n}\tau_i,g\right)=&\mathrm{e}\left(\prod_{i=1}^{n}\mathrm{H}_1(\mathrm{id})^{m_i},\mathrm{P}_{\mathrm{pub}}\right)\\&\cdot\mathrm{e}\left(\prod_{i=1}^{n}\mathrm{H}_3(\mathrm{F},n,m_i)^{\varsigma_i},\mathrm{pk}_{\mathrm{r}}\right)\end{aligned} \tag{3}$$

如果等式(3)成立，CSP存储文件，否则返回一个错误信息。

### 4.2 完整性验证

#### 4.2.1 挑战生成

(1) 数据拥有者DO从集合$\{1,2,\cdots,n\}$随机选择$c$个数生成子集Q，并选择一个随机数$a\in Z_p^{*}$，生成挑战集合$\mathrm{chal}=\{a,Q\}$发送给CSP。

(2) 数据拥有者DO部署智能合约T0，T1和T2到智能合约平台。

#### 4.2.2 证据生成

当收到来自数据拥有者的挑战信息之后，CSP计算$\{r_j\}=\{\pi(a,j)|j\in\mathrm{Q}\}$，从而计算响应证据$\xi=\prod_{j\in\boldsymbol{Q}}\tau_j{}^{r_j}$和$\mathrm{R}=\sum_{j\in\boldsymbol{Q}}m_jr_j$并发送给雾节点。最后CSP调用智能合约T0存储响应证据和R。

#### 4.2.3 证据验证

(1)雾节点FN计算$\mathrm{S}_1=\mathrm{e}(\xi,g)$和$\mathrm{S}_2=\mathrm{e}(\mathrm{H}_1(\mathrm{id})^{\mathrm{R}},\mathrm{p}_{\mathrm{pub}})\cdot\mathrm{e}(\prod_{j\in\boldsymbol{Q}}\mathrm{H}_3(\mathrm{F},n,\mathrm{m}_j)^{\varsigma_j r_j},\mathrm{pk}_{\mathrm{r}})$，调用智能合约T0存储计算值$\mathrm{S}_1$和$\mathrm{S}_2$；

(2) 当$\mathrm{S}_1$和$\mathrm{S}_2$被存储之后，DO调用智能合约T2验证$\mathrm{S}_1$和$\mathrm{S}_2$是否相等。若验证通过，激活智能合约T1支付CSP服务费用；否则，向DO返回一个验证失败的消息。

## 5 方案分析

### 5.1 正确性分析

为了验证存储在云端数据的完整性，智能合约验证$\mathrm{S}_1=\mathrm{S}_2$是否成立。如果对于挑战$\mathrm{chal}=\{a,\mathrm{Q}\}$和响应证据和R，总能返回一个正确信息，那么说明CSP完整存储了用户文件。等式的正确性为

$$
\begin{aligned}
\mathrm{S}_1 &= \mathrm{e}(\xi, g) = \mathrm{e}(\prod_{j\in \boldsymbol{Q}} \tau_j{}^{r_j}, g) \\
&= \mathrm{e}(\prod_{j\in \boldsymbol{Q}} (\mathrm{sk}_1 {\omega_j}^{\beta_j \mathrm{sk}_2})^{m_j r_j}, g) \\
&== \mathrm{e}(\prod_{j\in \boldsymbol{Q}} \mathrm{sk}_1{}^{m_j r_j}, g) \cdot \mathrm{e}(\prod_{j\in \boldsymbol{Q}} ({\omega_j}^{\beta_j})^{\mathrm{sk}_2 m_j r_j}, g) \\
&= \mathrm{e}(\mathrm{H}_1(\mathrm{id})^{s \sum_{j\in \boldsymbol{Q}} m_j r_j}, g) \cdot \mathrm{e}(\prod_{j\in \boldsymbol{Q}} {\omega_j}^{\varsigma_j r_j}, g^{\mathrm{sk}_2}) \\
&= \mathrm{e}(\mathrm{H}_1(\mathrm{id})^{\sum_{j\in \boldsymbol{Q}} r_j m_j}, g^s) \cdot e(\prod_{j\in \boldsymbol{Q}} \omega_j{}^{\varsigma_j r_j}, \mathrm{pk}_r) \\
&= \mathrm{e}(\mathrm{H}_1(\mathrm{id})^{\mathrm{R}}, \mathrm{p}_{\mathrm{pub}}) \cdot \mathrm{e}(\prod_{j\in \boldsymbol{Q}} \mathrm{H}_3(\mathrm{F}, n, m_j)^{\varsigma_j r_j}, \mathrm{pk}_r) \\
&= \mathrm{S}_2
\end{aligned}
$$

### 5.2 安全性证明

**定理1**　基于随机预言机模型和CDH困难问题，本文方案在自适应选择消息和id′攻击下可以满足标签的不可伪造性。

**证明**　当满足引理1和引理2时，定理1成立。

**引理1**　在CDH困难问题下，本文方案对Ⅰ类敌手$\mathcal{A}_1$是安全的。

**证明**　假设Ⅰ类敌手$\mathcal{A}_1$能以一个不可忽略的概率$\varepsilon$赢得游戏Ⅰ，则可以构造一个挑战者$\mathcal{C}$以一个不可忽略的概率来解决CDH问题。

**游戏Ⅰ**　给定一个CDH问题实例$(g, g^{\alpha}, g^{\beta})$，$\mathcal{C}$与$\mathcal{A}_1$通过以下查询模拟游戏。

**系统建立**：$\mathcal{C}$执行系统初始化获取参数，设置主公钥$\mathrm{msk} = g^{\alpha}$。

**$\mathrm{H}_1$询问**：$\mathcal{A}_1$对任意选定的id′适应性执行$\mathrm{H}_1$查询，$\mathcal{C}$为$\mathrm{H}_1$询问维护一个列表$\mathrm{L}_1 = \{(\mathrm{id}, h_1, D, T)\}$。如果选择的id′属于$\mathrm{L}_1$，那么$\mathcal{C}$提取相应的元组$(\mathrm{id}', h_1', D', T')$，发送$D'$给$\mathcal{A}_1$；否则，$\mathcal{C}$随机选择$h_1' \in Z_{\mathrm{p}}{}^*$并掷硬币$T \in \{0,1\}$。假设$T=0$的概率为$\lambda$，那么$T=1$的概率为$1-\lambda$。当$T=0$时，$\mathcal{C}$计算$D' = g^{h_1'}$；当$T=1$时，计算$D' = (g^{\beta})^{h_1'}$。$\mathcal{C}$将$D'$返回$\mathcal{A}_1$，同时将元组$(\mathrm{id}', h_1', D', T')$添加到$\mathrm{L}_1$中。

**部分密钥询问**：$\mathcal{A}_1$对任意选定的id′适应性执行部分密钥查询，$\mathcal{C}$维持列表$\mathrm{L}_2 = \{(\mathrm{id}, S, \mathrm{sk}_1, \mathrm{sk}_2, \mathrm{pk})\}$。$\mathcal{C}$检查$(\mathrm{id}', h_1', D', T')$是否存在于$\mathrm{L}_1$中。如果不存在，执行$\mathrm{H}_1$询问。

如果id′存在于$\mathrm{L}_2$中，$\mathcal{C}$查看$S' = \perp$是否成立。如果$S' = \perp$，$\mathcal{C}$从$\mathrm{L}_1$中询问$(\mathrm{id}', h_1', D', T')$。当$T' = 0$，计算$S' = (D')^{\alpha} = g^{\alpha h_1'}$；当$T' = 1$，终止。如果$S' \neq \perp$，直接提取$S'$。

如果id′不存在于$\mathrm{L}_2$中，那么$\mathcal{C}$将从$\mathrm{L}_1$中得到$(\mathrm{id}', h_1', D', T')$。当$T' = 0$，计算$S' = g^{\alpha h_1'}$；当$T' = 1$，终止。$\mathcal{C}$将$S'$返回给$\mathcal{A}_1$，$\mathcal{A}_1$添加$(\mathrm{id}', S', \perp, \perp)$到$\mathrm{L}_2$中。

**秘密值询问**：$\mathcal{A}_1$对任意id′适应性执行秘密值询问，$\mathcal{C}$检查$(\mathrm{id}', h_1', D', T')$是否存在于$\mathrm{L}_1$中。如果不存在，执行$\mathrm{H}_1$询问。

如果id′存在于$\mathrm{L}_2$中，$\mathcal{C}$查看$\mathrm{sk}_1' = \perp$是否成立。如果$\mathrm{sk}_1' = \perp$，$\mathcal{C}$随机选择$x \in \mathrm{Z}_{\mathrm{p}}^*$，设置$\mathrm{sk}_1' = x$并将其添加到$\mathrm{L}_2$中；如果$\mathrm{sk}_1' \neq \perp$，$\mathcal{C}$直接从$\mathrm{L}_2$中获取$\mathrm{sk}_1'$并发送给$\mathcal{A}_1$。

如果id′不存在于$\mathrm{L}_2$中，$\mathcal{C}$从$\mathrm{L}_1$中得到$(\mathrm{id}', h_1', D', T')$，然后随机选择$x \in \mathrm{Z}_{\mathrm{p}}^*$，设置$\mathrm{sk}_1' = x$并将其添加到$\mathrm{L}_2$。

**公钥询问**：$\mathcal{A}_1$对任意id′适应性执行公钥询问，$\mathcal{C}$维持列表$\mathrm{L}_2 = \{(\mathrm{id}, S, \mathrm{sk}_1, \mathrm{sk}_2, \mathrm{pk})\}$。$\mathcal{C}$检查$(\mathrm{id}', h_1', D', T')$是否存在于$\mathrm{L}_1$中。如果不存在，执行$\mathrm{H}_1$询问。

检查元组$(\mathrm{id}', S', \mathrm{sk}_1', \mathrm{sk}_2', \mathrm{pk}')$是否存在于$\mathrm{L}_2$中，如果存在，$\mathcal{C}$查看$\mathrm{pk}' = \perp$是否成立。如果$\mathrm{pk}' = \perp$，$\mathcal{C}$随机选择$x \in Z_{\mathrm{p}}^*$，设置$\mathrm{sk}_2' = x$，计算$\mathrm{pk}' = g^{\mathrm{sk}_2'}$，将$\mathrm{pk}'$发送给$\mathcal{A}_1$并且更新列表；如果$\mathrm{pk}' \neq \perp$，$\mathcal{C}$直接获取并发送给$\mathcal{A}_1$；

如果$\mathrm{L}_2$中不包含元组$(\mathrm{id}', S', \mathrm{sk}_1', \mathrm{sk}_2', \mathrm{pk}')$，$\mathcal{C}$随机选择$x \in Z_{\mathrm{p}}^*$，设置$\mathrm{sk}_2' = x$，计算$\mathrm{pk}' = g^{\mathrm{sk}_2'}$，并将$\mathrm{pk}'$发送给$\mathcal{A}_1$。之后$\mathcal{C}$添加元组$(\mathrm{id}', \perp, \mathrm{sk}_1', \mathrm{sk}_2', \mathrm{pk}')$到$\mathrm{L}_2$中。

**替换公钥询问**：$\mathcal{A}_1$对$(\mathrm{id}', \mathrm{pk}')$适应性执行公钥替换询问。

如果$(\mathrm{id}', S', \mathrm{sk}_1', \mathrm{sk}_2', \mathrm{pk}')$存在于$\mathrm{L}_2$中，$\mathcal{C}$替换元组$(\mathrm{id}', S', \mathrm{sk}_1', \mathrm{sk}_2', \mathrm{pk}')$为$(\mathrm{id}', S', \perp, \mathrm{sk}_2', \mathrm{pk}')$；如果不存，$\mathcal{C}$向$\mathrm{L}_2$中添加元组$(\mathrm{id}', \perp, \perp, \mathrm{sk}_2', \mathrm{pk}')$。

**$\mathrm{H}_2$询问**：$\mathcal{A}_1$对任意$m_i'$适应性执行$\mathrm{H}_2$询问，$\mathcal{C}$维持一个列表$\mathrm{L}_3 = (m_i, h_2, Q)$。如果$\mathrm{L}_3$包含$m_i$’，那么$\mathcal{C}$直接从$\mathrm{L}_3$中获取$h_2'$，设置$Q = g^{h_2'}$并发送给$\mathcal{A}_1$；如果$\mathrm{L}_3$不包含$m_i'$，$\mathcal{C}$随机选择$h_2' \in Z_{\mathrm{p}}{}^*$，计算$Q = g^{h_2'}$并发送给$\mathcal{A}_1$，然后将$(m_i', h_3', Q')$插入$\mathrm{L}_3$中。

**签名询问**：$\mathcal{A}_1$自适应对$(\mathrm{sk}_1', \mathrm{sk}_2', h_2', \mathrm{id}')$执行标签询问，$\mathcal{C}$查看$\mathrm{L}_1$中id′对应的$T'$的值。如果$T' = 1$，则$\mathcal{C}$停止；否则，$\mathcal{C}$从$\mathrm{L}_2$中得到$\mathrm{sk}_1'$和$\mathrm{sk}_2'$，并从$\mathrm{L}_3$中获取$h_3'$。然后，$\mathcal{C}$通过签名生成算法对$(\mathrm{sk}_1', \mathrm{sk}_2', h_2', \mathrm{id}')$生成签名并发送给$\mathcal{A}_1$。

**伪造**：$\mathcal{A}_1$输出一个伪造的元组$\mathrm{B} = (\xi', \mathrm{sk}_1', \mathrm{sk}_2', \mathrm{pk}', m_i', \mathrm{id}')$，其中伪造签名为$\xi'$。

如果$\mathcal{A}_1$可以赢得游戏Ⅰ，$\mathcal{C}$可以得到$\mathrm{e}(\xi', g) = \mathrm{e}(\mathrm{H}_1(\mathrm{id}')^{\mathrm{R}}, \mathrm{p}_{\mathrm{pub}}) \cdot \mathrm{e}(\mathrm{H}_3(\mathrm{F}, \mathrm{B}, m_i'), \mathrm{pk}')$。$\mathcal{C}$从$\mathrm{L}_1$提取元组$(\mathrm{id}', h_1', D', T')$，如果$T' = 0$，$\mathcal{C}$终止。否则，$\mathcal{C}$从$\mathrm{L}_1$中检索$\mathrm{H}_1(\mathrm{id}') = g^{\beta h_1'}$，从$\mathrm{L}_3$中检索$\mathrm{H}_3(\mathrm{F}, \mathrm{B}, m_i) = g^{h_2'}$，并通过等式$\mathrm{e}(\xi', g) = \mathrm{e}(g^{\beta h_1'}, g^{\alpha}) \cdot \mathrm{e}(g^{h_2'}, \mathrm{pk}')$可以计算出$g^{\alpha\beta} = \left(\xi' \Big/ {\mathrm{pk}'}^{h_2'}\right)^{\frac{1}{h_1'}}$。假设在时间$t$内经过$\mathrm{H}_1$询

问、部分密钥询问、秘密值询问、$H_2$询问、公钥询问、替换公钥询问、和签名询问的次数分别是$q_{H_1}$, $q_{pa}$, $q_{se}$, $q_{H_2}$, $q_{pk}$, $q_{pr}$和$q_T$，$\mathcal{A}_1$赢得游戏 I 的优势是$\varepsilon$。在$\mathcal{A}_1$没有停止的情况下，$\mathcal{C}$与$\mathcal{A}_1$完整交互的概率是$(1-\lambda)^{q_{pa}q_T}$，$\mathcal{C}$输出正确结果的概率是$\varepsilon' \geq \varepsilon\lambda(1-\lambda)^{q_{pa}q_T} \geq \varepsilon/((q_{pa}+q_T)\cdot 2e)$，对应的时间成本为$t' \leq t + O(q_{H_1}+q_{pa}+q_{se}+q_{H_2}+q_{pk}+q_{pr}+q_T)$。因此，挑战者$\mathcal{C}$不能以一个不可忽略的概率赢得游戏 I。 证毕

**引理2** 在CDH困难问题下，本文方案对 II 类敌手$\mathcal{A}_2$是安全的。

**证明** 假设 II 类敌手$\mathcal{A}_2$能以一个不可忽略的概率$\varepsilon$赢得游戏 II，则可以构造一个挑战者$\mathcal{C}$以一个不可忽略的概率来解决CDH问题。

**游戏 II** 本文对游戏 II 的设计和大多数无证书签名方案相似。游戏 II 与游戏 I 的区别在于攻击者的不同，$\mathcal{A}_2$知道系统主密钥，但是不知道秘密值并且不能替换公钥。游戏 II 和文献[11]的步骤基本相似，包括系统建立、$H_1$询问、秘密值询问、$H_2$询问、公钥询问、签名询问和伪造这几个步骤，这里不做赘述，只对游戏 II 结果进行分析。

通过游戏 II，$\mathcal{C}$可以得到$\mathrm{e}(\xi', g) = \mathrm{e}(g^{h_1'}, g^s)\cdot\mathrm{e}(g^{\beta h_3'}, (g^{\alpha})^{xh_2'\varsigma'})$，从而可以计算得：$g^{\alpha\beta} = \left(\xi'/g^{h_1's}\right)^{\frac{1}{xh_2'\varsigma'}}$。假设在时间$t$内经过$H_1$询问、密钥值询问、$H_2$询问、公钥询问、签名询问的次数分别是$q_{H_1}$, $q_{se}$, $q_{H_2}$, $q_{pk}$和$q_T$，$\mathcal{A}_2$赢得游戏 II 的优势是$\varepsilon$。在$\mathcal{A}_2$没有停止，$\mathcal{C}$与$\mathcal{A}_2$完整交互的概率是$(1-\lambda)^{q_{h1}\cdot q_T}$，$\mathcal{C}$输出正确结果的概率是$\varepsilon' \geq \varepsilon\lambda(1-\lambda)^{q_{H_1}\cdot q_T} \geq \varepsilon/((q_{H_1}+q_T)\cdot 2e)$，对应的时间成本$t' \leq t + O(q_{H_1}+q_{se}+q_{H_2}+q_{pk}+q_T)$。因此，挑战者$\mathcal{C}$不能以一个不可忽略的概率赢得游戏 II。 证毕

## 6 性能分析

### 6.1 性能比较

将本文方案与几个现有研究成果进行比较，分析方案性能的优缺点。从表1可以看出，与文献[11,14,15]相比，本文方案基于区块链，实现了多用户共享、审计去中心化、外包计算和公平支付，同时避免了密钥和证书管理的问题。

### 6.2 计算开销

文献[11,14]和本文方案均为基于无证书的完整性验证方案，本节对文献[11,14]和本文方案进行计算开销对比。实验计算机配备了i5-6 500的处理器、8 GB内存和Windows 10 64位操作系统。基于PBC (Pairing-Based Cryptography) 库，使用C编程语言仿真模拟CSP, TPA和FN。表2给出了实验所涉及的一些密码学的符号定义和运算时间，并使用$n$和$c$分别表示文件分块数和挑战数据块个数。如表3所示，可以看出，块签名生成时，这3类方案的计算开销相差较小，可以忽略；在证据生成阶段，文献[11]方案的计算开销高于本文和文献[14]方案；在签名验证和证据验证阶段，相比于文献[11,14]方案，本文方案计算开销最小，具有明显的优势。

实际环境中，本文方案完整性验证延迟主要包括CSP和雾节点的计算延迟、智能合约T2的验证延迟和网络往返时间。而往返时间主要依赖于网络环境，因此实验不予考虑。实验中，智能合约使用Solidity语言编写，通过以太坊测试网络Rinkeby进

**表 1 性能比较**

| 方案 | 无证书签名 | 多用户共享 | 区块链 | 审计去中心化 | 外包计算 | 公平支付 |
|---|---|---|---|---|---|---|
| 文献[11] | √ | √ | × | × | × | × |
| 文献[14] | √ | × | √ | × | × | × |
| 文献[15] | × | × | √ | √ | × | √ |
| 本文方案 | √ | √ | √ | √ | √ | √ |

**表 2 相关运算时间**

| 符号 | 操作 | 运算时间(ms) |
|---|---|---|
| $T_P$ | 双线性配对运算 | ≈ 7.839 506 |
| $T_E$ | 群中点的幂运算 | ≈ 6.216 714 |
| $T_M$ | 群上标量积运算 | ≈ 0.022 544 |
| $T_m$ | 模乘运算 | ≈ 0.000 322 |
| $T_H$ | 哈希函数映射到点 | ≈ 13.346 683 |

**表 3 计算开销对比(ms)**

| 方案 | 文献[11] | 文献[14] | 本文方案 |
|---|---|---|---|
| 块签名生成 | $2nT_E + nT_M + nT_H$<br>$\approx 12.46n$ | $2nT_E + 3nT_M + nT_H$<br>$\approx 12.48n$ | $2nT_E + nT_M + nT_m + nT_H$<br>$\approx 12.46n$ |
| 签名验证 | $4T_P + 2nT_E + nT_M$<br>$\approx 31.36 + 12.45n$ | $3T_P + 4nT_E + 2nT_M$<br>$\approx 23.51 + 25.32n$ | $3T_P + 2nT_E$<br>$\approx 23.51 + 12.43n$ |
| 证据生成 | $2cT_P + 4cT_E + cT_M + cT_m + T_H$<br>$\approx 13.35 + 40.56n$ | $cT_E + cT_M + cT_m$<br>$\approx 6.24c$ | $cT_E + cT_M + cT_m$<br>$\approx 6.24c$ |
| 证据验证 | $2T_P + 4cT_E + cT_M + cT_m + cT_H$<br>$\approx 15.68 + 38.23c$ | $4T_P + (3c+2)T_E + 3cT_M + 2cT_m + (c+4)T_H$<br>$\approx 97.18 + 32.07c$ | $3T_P + (c+1)T_E + cT_M + cT_m + 2T_H$.<br>$\approx 56.33 + 6.24c$ |

行部署和测试。经测试，智能合约在Rinkeby网络的交易时间大约为20 s。

### 6.3 智能合约成本测试

在以太坊中，使用gas值的大小表示每笔交易消耗的成本，本文方案的交易包括智能合约的部署和调用。通过以太坊测试网络Rinkeby，对方案中每笔交易所消耗的成本进行了测试。如表4所示，实验结果表明，智能合约部署和调用都需要消耗gas值。相比于智能合约调用，将智能合约首次部署到以太坊的消耗较高，而所有gas的消耗都在可以接受的范围之内。

**表 4　智能合约成本测试**

| 交易 | 消耗成本(gas) | 交易费用(美元) |
|---|---|---|
| 智能合约T0生成 | 178621 | 0.41 |
| 智能合约T1生成 | 124833 | 0.28 |
| 智能合约T2生成 | 89391 | 0.2 |
| 存储$S_1$和$S_2$ | 65814 | 0.15 |
| 智能合约T1调用 | 32467 | 0.07 |
| 智能合约T2调用 | 54031 | 0.12 |

## 7　结束语

本文提出了基于无证书签名体制的云端数据完整性验证方案。方案使用雾节点和智能合约代替第三方审计者从而实现去中心化，基于无证书密码体制可以避免密钥和证书管理问题，同时具有多用户共享和公平支付的功能。在随机预言机模型下证明了本文方案是安全的，能够抵御I类和II类攻击。性能分析表明，相比现有同类无证书签名方案，本文方案具有较高的计算效率和较低的存储成本。未来将扩展工作，以实现方案支持在多云环境下的批量验证和数据动态操作。

## 参 考 文 献


[1] ZHOU Lei, FU Anmin, YANG Guomin, *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. doi: 10.1109/TDSC.2020.3013927.

[2] 鲁金钿, 肖睿智, 金舒原. 云数据安全研究进展[J]. 电子与信息学报, 2021, 43(4): 881–891. doi: 10.11999/JEIT200158.
LU Jindian, XIAO Ruizhi, and JIN Shuyuan. A survey for cloud data security[J]. *Journal of Electronics & Information Technology*, 2021, 43(4): 881–891. doi: 10.11999/JEIT200158.

[3] WANG Tao, YANG Bo, LIU Hongyu, *et al*. An alternative approach to public cloud data auditing supporting data dynamics[J]. *Soft Computing*, 2019, 23(13): 4939–4953. doi: 10.1007/s00500-018-3155-4.

[4] ZHAO Haichun, YAO Xuanxia, ZHENG Xuefeng, *et al*. User stateless privacy-preserving TPA auditing scheme for cloud storage[J]. *Journal of Network and Computer Applications*, 2019, 129: 62–70. doi: 10.1016/j.jnca.2019.01.005.

[5] ATENIESE G, BURNS R, CURTMOLA R, *et al*. Provable data possession at untrusted stores[C]. The 14th ACM Conference on Computer and Communications Security, Alexandria, USA, 2007: 598–609. doi: 10.1145/1315245.1315318.

[6] BOWERS K D, JUELS A, and OPREA A. Proofs of retrievability: Theory and implementation[C]. The 2009 ACM Workshop on Cloud Computing Security, Chicago, USA, 2009: 43–54. doi: 10.1145/1655008.1655015.

[7] 张振超, 刘亚丽, 殷新春, 等. 无证书签名方案的分析及改进[J]. 密码学报, 2020, 7(3): 389–403. doi: 10.13868/j.cnki.jcr.000375.
ZHANG Zhenchao, LIU Yali, YIN Xinchun, *et al*. Analysis and improvement of certificateless signature schemes[J]. *Journal of Cryptologic Research*, 2020, 7(3): 389–403. doi: 10.13868/j.cnki.jcr.000375.

[8] 魏松杰, 李莎莎, 王佳贺. 基于身份密码系统和区块链的跨域认证协议[J]. 计算机学报, 2021, 44(5): 908–920. doi: 10.11897/SP.J.1016.2021.00908.
WEI Songjie, LI Shasha, and WANG Jiahe. A cross-domain authentication protocol by identity-based cryptography on consortium blockchain[J]. *Chinese Journal of Computer*, 2021, 44(5): 908–920. doi: 10.11897/SP.J.1016.2021.00908.

[9] 赵艳琦, 来齐齐, 禹勇, 等. 标准模型下基于身份的环签名方案[J]. 电子学报, 2018, 46(4): 1019–1024. doi: 10.3969/j.issn.0372-2112.2018.04.033.
ZHAO Yanqi, LAI Qiqi, YU Yong, *et al*. ID-based ring signature in the standard model[J]. *Acta Electronica Sinica*, 2018, 46(4): 1019–1024. doi: 10.3969/j.issn.0372-2112.2018.04.033.

[10] ZHANG Xiaojun, WANG Huaxiong, and XU Chunxiang. Identity-based key-exposure resilient cloud storage public auditing scheme from lattices[J]. *Information Sciences*, 2019, 472: 223–234. doi: 10.1016/j.ins.2018.09.013.

[11] WU Ge, MU Yi, SUSILO W, *et al*. Privacy-preserving certificateless cloud auditing with multiple users[J]. *Wireless Personal Communications*, 2019, 106(3): 1161–1182. doi: 10.1007/s11277-019-06208-1.

[12] 曾萍, 郭瑞芳, 马英杰, 等. 车载自组网中可证明安全的无证书认证方案[J]. 电子与信息学报, 2020, 42(12): 2873–2881. doi: 10.11999/JEIT190883.

ZENG Ping, GUO Ruifang, MA Yingjie, *et al*. Provable security certificateless authentication scheme for vehicular ad hoc network[J]. *Journal of Electronics & Information Technology*, 2020, 42(12): 2873–2881. doi: 10.11999/JEIT190883.

[13] 谢永, 李香, 张松松, 等. 一种可证安全的车联网无证书聚合签名改进方案[J]. 电子与信息学报, 2020, 42(5): 1125–1131. doi: 10.11999/JEIT190184.
XIE Yong, LI Xiang, ZHANG Songsong, *et al*. An improved provable secure certificateless aggregation signature scheme for vehicular ad hoc NETworks[J]. *Journal of Electronics & Information Technology*, 2020, 42(5): 1125–1131. doi: 10.11999/JEIT190184.

[14] ZHANG Yuan, XU Chunxiang, LIN Xiaodong, *et al*. Blockchain-based public integrity verification for cloud storage against procrastinating auditors[J]. *IEEE Transactions on Cloud Computing*, 2021, 9(3): 923–937. doi: 10.1109/TCC.2019.2908400.

[15] WANG Hao, QIN Hong, ZHAO Minghao, *et al*. Blockchain-based fair payment smart contract for public cloud storage auditing[J]. *Information Sciences*, 2020, 519: 348–362. doi: 10.1016/j.ins.2020.01.051.

[16] LI Jiguo, YAN Hao, and ZHANG Yichen. Certificateless public integrity checking of group shared data on cloud storage[J]. *IEEE Transactions on Services Computing*, 2021, 14(1): 71–81. doi: 10.1109/TSC.2018.2789893.

[17] NGUYEN L. Accumulators from bilinear pairings and applications[C]. Cryptographers' Track at the RSA Conference, San Francisco, USA, 2005: 275–292. doi: 10.1007/978-3-540-30574-3_19.

[18] NAIR M S and RAJASREE M S. Fine-grained search and access control in multi-user searchable encryption without shared keys[J]. *Journal of Information Security and Applications*, 2018, 41: 124–133. doi: 10.1016/j.jisa.2018.06.006.

[19] FENG Xia, SHI Qichen, XIE Qingqing, *et al*. An efficient privacy-preserving authentication model based on blockchain for VANETs[J]. *Journal of Systems Architecture*, 2021, 117: 102158. doi: 10.1016/j.sysarc.2021.102158.

[20] TIAN Hui, NAN Fulin, CHANG C C, *et al*. Privacy-preserving public auditing for secure data storage in fog-to-cloud computing[J]. *Journal of Network and Computer Applications*, 2019, 127: 59–69. doi: 10.1016/j.jnca.2018.12.004.

[21] LI Hui and JING Tao. A lightweight fine-grained searchable encryption scheme in fog-based healthcare IoT networks[J]. *Wireless Communications and Mobile Computing*, 2019, 2019: 1019767. doi: 10.1155/2019/1019767.

[22] LI Hui and JING Tao. A ciphertext-policy attribute-based encryption scheme with public verification for an IoT-fog-cloud architecture[J]. *Procedia Computer Science*, 2020, 174: 243–251. doi: 10.1016/j.procs.2020.06.080.

杨小东：男，博士后，教授，研究方向为云计算安全与代理重加密.
王秀秀：女，硕士生，研究方向为云计算安全.
李茜茜：女，硕士生，研究方向为区块链与大数据安全.
周 航：女，硕士生，研究方向为属性基加密.
王彩芬：女，博士，教授，研究方向为信息安全协议与网络安全.

责任编辑：马秀强