

具有紧凑标签的基于身份匿名云审计方案

卢晨昕¹, 陈兵², 丁宁², 陈立全¹, 吴戈¹

(1. 东南大学网络空间安全学院, 江苏 无锡 214081;

2. 宿迁市互联网信息办公室, 江苏 宿迁 223834;

摘要: 云存储技术具有效率高、可扩展性强等优点。用户可以借助云存储技术节省本地的存储开销, 并与其他人共享数据。然而, 数据存储到云服务器后, 用户失去对数据的物理控制, 需要有相应的机制保证云中数据的完整性。数据拥有证明 (PDP, provable data possession) 机制允许用户或用户委托的第三方审计员 (TPA, third party auditor) 对数据完整性进行验证。但在实际应用中, 数据通常由多个用户共同维护, 用户在进行完整性验证请求的同时泄露了自己的身份。匿名云审计支持 TPA 在完成数据完整性验证时保证用户的匿名性。在基于身份体制下, 匿名云审计方案通常需要借助基于身份的环签名或群签名技术实现, 数据标签的构成元素与用户数量相关, 使得数据标签不够紧凑, 存储效率较低。为了解决这一问题, 提出一种基于身份的匿名云审计方案通用构造, 使用一个传统体制下的签名方案和一个传统体制下的匿名云审计方案即可构造一个基于身份的匿名云审计方案。基于该通用构造, 使用 BLS 签名和一个传统体制下具有紧凑标签的匿名云审计方案设计了具有紧凑标签的基于身份匿名云审计方案。该方案主要优势在于数据标签短, 能够减少云服务器的存储压力, 提高存储效率。此外, 证明了该方案的不可欺骗性和匿名性。

关键词: 云审计; 隐私保护; 匿名; 基于身份体制; 紧凑标签

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2022087

Identity-based anonymous cloud auditing scheme with compact tags

LU Chenxin¹, CHEN Bing², DING Ning², CHEN Liquan¹, WU Ge¹

1. School of Cyber Science and Engineering, Southeast University, Wuxi 214081, China

2. Cyberspace Administration of Suqian City, Suqian 223834, China

收稿日期: 2022-03-29; **修回日期:** 2022-06-28

通信作者: 吴戈, gewu@seu.edu.cn

基金项目: 国家重点研发计划 (2020YFE0200600); 国家自然科学基金 (62002058); 江苏省自然科学基金 (BK20200391); 中央高校基本科研业务费专项资金 (2242021R40011); 宿迁市网信领域研究课题

Foundation Items: The National Key R&D Program of China (2020YFE0200600), The National Natural Science Foundation of China (62002058), Natural Science Foundation of Jiangsu Province (BK20200391), Fundamental Research Funds for the Central Universities (2242021R40011), Research Topic in the Network Communication Field in Suqian

引用格式: 卢晨昕, 陈兵, 丁宁, 等. 具有紧凑标签的基于身份匿名云审计方案[J]. 网络与信息安全学报, 2022, 8(6): 156-168.

Citation Format: LU C X, CHEN B, DING N, et al. Identity-based anonymous cloud auditing scheme with compact tags[J]. Chinese Journal of Network and Information Security, 2022, 8(6): 156-168.

Abstract: Cloud storage has the advantages of high efficiency and scalability. Users can save local storage cost and share data with others through cloud storage technology. However, when data is uploaded to cloud servers, its owner also loses the physical control, and hence there needs a corresponding mechanism to ensure the integrity of data stored in the cloud. The Provable Data Possession (PDP) mechanism allows users or a Third-Party Auditor (TPA) appointed by the user to verify data integrity. In practice, data is usually maintained by multiple users. Users may reveal their identities while making an integrity verification request in traditional auditing processes. Anonymous cloud auditing ensures anonymity of users against the TPA during auditing. Currently, in identity-based systems, anonymous cloud auditing schemes usually resort to identity-based ring signature or group signature schemes. As a result, the size of a tag is related to the number of users, which makes it not compact and causes high storage cost. In order to solve this issue, a general construction of identity-based anonymous cloud auditing scheme was proposed. With a signature scheme and an anonymous cloud auditing scheme, a concrete identity-based anonymous cloud auditing scheme based on the general construction was proposed. It combined the BLS signature and an anonymous cloud auditing scheme with compact tags. The main advantage of this solution is that the tags are compact, which can significantly reduce storage cost and improve storage efficiency. Furthermore, the uncheatability and anonymity of the scheme are proved.

Keywords: cloud auditing, privacy protection, anonymous, identity-based system, compact tags

0 引言

随着大数据时代的来临,网络中的数据呈爆发式增长,数据呈多元化趋势,数据存储安全变得尤为重要。云存储作为现代存储的一种典型方式,具有高效率、可扩展性、灵活性等优势,云用户可以随时随地访问云中各类共享资源^[1]。因此,个人、企业、政府机构等开始将数据上传至云服务器中以降低本地存储成本和管理成本^[2]。然而,数据从本地存储变为云存储,使个人或机构失去了对数据的物理控制。此外,存储在云服务器中的数据可能会因为断电、设备损坏等物理原因或人为恶意操作遭受破坏,云服务提供商(CSP, cloud service provider)并非完全可信,如何验证云中数据完整性成关键问题^[3]。

为了解决这个问题,Deswarte 等^[4]提出了远程数据完整性检验(RDIC, remote data integrity checking)这一概念。远程数据完整性检验主要包含3个实体:用户、云服务器以及第三方审计员(TPA, third party auditor)。在公开可验证的RDIC协议中,用户可以委托信任的第三方审计员对云服务器发起挑战,通过服务器对挑战的应答来验证上传至服务器中的数据是否被篡改等。Ateniese 等^[5]提出了数据拥有证明机制(PDP, provable data possession)以验证数据完整性。在

该方案中,云服务器中的每个数据块都附加一个标签。当检验云中数据完整性时,验证者随机选择一组数据块对服务器发起挑战。若服务器的响应通过验证,则证明数据被安全正确地存储在云服务器中。该方案避免了验证者下载数据块再进行验证的烦琐操作。在实际应用中,云中数据通常由组内成员共享,且在某些场合下要求验证过程中为组内成员提供匿名性。

2012年,Wang 等^[6]提出了共享数据的隐私保护云审计。在该方案中,TPA生成的挑战包含组内所有用户的公钥,能够在审计过程中保证用户的匿名性。在此之后,许多隐私保护云审计方案被提出,如文献[7-8]等,但大多采用了环签名^[9-10]或群签名^[11-12]技术实现匿名性。这种做法生成的标签大小显著增加,存储效率不高。

为了改进这一局限性,Wu 等^[13]提出了多用户环境下的匿名云审计方案。在该方案中,不采用环签名或群签名技术,消息的验证标签仅包含一个元素,在实现用户匿名性的同时降低了标签的存储成本以及审计过程中的传输和验证成本。然而,该方案为传统体制下的云审计方案,密钥管理较为复杂。基于身份的密码体制以用户身份作为公钥,无须证书机构CA授权,减少了证书管理。在基于身份体制下实现匿名云审计,类似于传统体制,可以借助基于身份的环签名或群签

名技术实现基于身份体制下的匿名审计。但这种方案产生的标签同样不紧凑,存储效率不高,增加了审计过程中的传输成本和验证成本。

在电子匿名投票系统中,用户将自己的投票结果提交至云服务器。为了保证投票结果是真实有效的,云存储中的数据应不可被篡改伪造,即应保证云中数据完整性。为了保证选举的公平,应对投票人进行匿名化操作。在选票数量较大时,为了减少存储空间,用户生成的标签应较为紧凑,且投票结果往往具有时效性,这就要求方案有较高的审计效率。此外,普通用户部署公钥基础设施(PKI, public key infrastructure)成本较高,更适合采用基于身份的密码体制。因此,基于身份体制下,标签短且审计效率高的匿名云审计方案有很大的应用场景。目前,对于这一方面的研究较少,相关工作如下。

Shacham 和 Waters^[14]提出了基于线性同态验证器(LHA, linear homomorphic authenticator)的云审计方案。该方案基于PKI证书的生成、管理以及吊销等,过程较为复杂,且计算成本较高。1984年,Shamir^[15]提出了基于身份的密码体制,使用用户的身份标识(如姓名、IP地址、电子邮件地址等)作为公钥,解决了用户的公钥与身份进行绑定的问题。近年来,许多基于身份的云审计方案被提出。Yu等^[16]提出了基于身份的远程数据完整性检验方案,该方案使用同态加密技术,降低了公钥管理的复杂性。Rabaninejad等^[3]提出了基于身份的在线/离线云审计方案。该方案将用户的计算分为两个阶段:在线阶段和离线阶段。用户在离线阶段完成复杂的计算,在线阶段仅需完成轻量级的操作,提高了运行效率。Zhang等^[17]提出了支持用户高效撤销的云审计方案。Shen等^[18]提出了隐藏用户敏感信息的条件下实现用户数据共享的云审计方案。

目前,尚未有一种基于身份体制下的匿名云审计方案的通用构造,大多数的文献是针对具体的应用场景或安全需求提出的方案。2004年,Bellare等^[19]提出了将传统体制下的签名方案转换为基于身份的签名方案的方法。2006年,Galindo等^[20]对该方法进行了拓展,提出了具有附加属性的基于身份的签名方案的通用构造,包

括基于身份的盲签名方案、代理重签名方案等。这些方案都是由传统体制下的签名和具有附加属性的签名方案构造而成。受Bellare等及Galindo等通用构造的启发,本文研究借助传统体制下的云审计方案构造基于身份体制下的云审计方案。

本文提出了一种基于身份体制下的匿名云审计方案的通用构造,并给出了安全性证明。基于该通用构造,以BLS签名^[21]和具有紧凑标签的匿名云审计方案^[13]为组件,设计了具有紧凑标签的基于身份的匿名云审计方案。该方案实现了云服务器对于TPA的不可欺骗性和信息论安全的用户匿名性,且标签紧凑,标签大小与用户的数量无关。

1 预备知识

1.1 系统模型

基于身份的匿名云审计方案包含4个实体,系统模型如图1所示。

- 1) 私钥生成中心(PKG): PKG产生主密钥以及基于用户身份的密钥。
- 2) 云服务器(CS): 云服务器中包含用户上传的索引-数据-标签元组,可以应对来自TPA的挑战。
- 3) 用户组: 用户组中的每个用户都可以使用基于其身份的密钥计算上传的数据标签,然后将索引-数据-标签元组上传至云服务器中。
- 4) 第三方审计员: TPA在接收到某一组用户对存储在云服务器中的数据检验请求后,向云服务器发出挑战来执行审计过程,检验用户数据的有效性,并将检验结果告知用户。

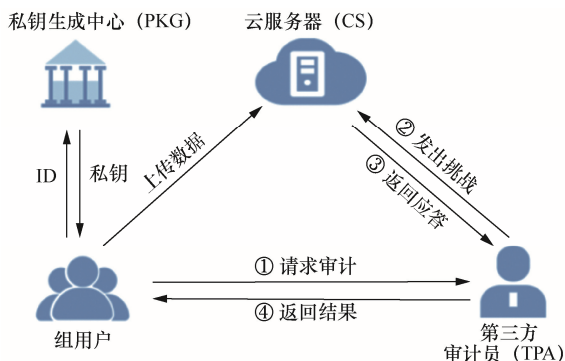


图1 系统模型
Figure 1 System model

1.2 双线性映射

令 G_1 、 G_2 为素数 p 阶的循环群, g 为 G_1 生成元, 则称 $e: G_1 \times G_1 \rightarrow G_2$ 为一个映射。如果它满足下列条件, 则为双线性映射。

- 1) 双线性: 对于 $\forall u, v \in G_1, a, b \in \mathbb{Z}_p$, 有 $e(u^a, v^b) = e(u, v)^{ab}$ 。
- 2) 非退化性: $\exists g \in G_1$, 使得 $e(g, g) \neq 1$ 。
- 3) 可计算性: 对于 $\forall u, v \in G_1, a, b \in \mathbb{Z}_p$, 存在一种概率多项式时间算法可以计算 $e(u^a, v^b)$, 并且在 G_1 上的操作是高效的。

2 方案定义

2.1 标准签名方案

定义 1 标准签名方案由 4 个算法组成, 分别为初始化算法 Setup, 密钥生成算法 KeyGen, 签名算法 Sign 以及验证算法 Verify。算法具体描述如下。

- 1) Setup(1^k): 输入安全参数 1^k , 输出公开参数 params。
- 2) KeyGen(params): 输入公开参数 params, 输出密钥对 (pk, sk)。
- 3) Sign(params, sk, m): 输入公开参数 params、私钥 sk 以及消息 m , 输出签名后的消息 σ_m 。
- 4) Verify(params, pk, m, σ_m): 输入公开参数 params、公钥 pk、消息 m 以及签名 σ_m 。若验证成功输出 1, 否则输出 0。

定义 2 (不可伪造性)。标准签名方案的不可伪造性由敌手 A 和挑战者 C 之间的游戏定义, 它包含 3 个阶段, 分别为初始化阶段、签名询问阶段以及伪造阶段, 具体描述如下。

- 1) 初始化
挑战者 C 运行 Setup 和 KeyGen 算法, 生成公开参数 params、密钥对 (pk, sk), 并将 params、pk 返回给敌手 A, 自己保存 sk。
- 2) 签名询问
敌手 A 自适应地选择需要进行签名询问的消息 m 。挑战者 C 运行 Sign 算法生成签名 σ_m , 并将结果返回给 A。
- 3) 伪造
敌手 A 输出消息 m^* 的伪造签名 σ_{m^*} 。

如果 σ_{m^*} 通过验证且 A 没有询问过 m^* 的签名, 则 A 赢得游戏。A 在上述游戏中的优势定义为

$$\text{Adv}_A^{\text{EU-CMA}} = \Pr[l' \leftarrow \text{Verify}(\text{params}, \text{pk}, m^*, \sigma_{m^*})]$$

2.2 匿名云审计方案

定义 3 匿名云审计方案由 6 个算法组成, 分别为初始化算法 Setup、密钥生成算法 KeyGen、标签生成算法 TagGen、挑战算法 Challenge、应答算法 Respond、验证算法 Verify。算法具体描述如下。

- 1) Setup(1^k): 输入安全参数 1^k , 输出公开参数 params。
- 2) KeyGen(params): 输入公开参数 params, 输出密钥对 (pk, sk)。
- 3) TagGen(params, sk, i, m_i): 输入公开参数 params、私钥 sk, 索引为 i 的消息块 m_i , 输出签名 σ_{m_i} 。之后, 将索引-信息-标签元组 (i, m_i, σ_{m_i}) 上传至云服务器。
- 4) Challenge(params, pk₁, ..., pk_n, I): 输入公开参数 params, 随机选择当前占用块的索引子集 I 以及用户的公钥 (pk₁, ..., pk_n), 输出挑战 chal 并将其发送至云服务器。
- 5) Respond(params, chal, M, Σ): 输入公开参数 params、挑战 chal、消息集合 $M = \{m_i | i \in I\}$ 以及标签集合 $\Sigma = \{\sigma_{m_i} | i \in I\}$, 计算应答 res 并将其发送给 TPA。
- 6) Verify(params, chal, s, res): 输入公开参数 params、挑战 chal、秘密值 s 以及应答 res 验证应答是否有效。若验证成功, 输出 1, 否则输出 0。

定义 4 (不可欺骗性)。匿名云审计方案的不可欺骗性由敌手 A 和挑战者 C 之间的游戏定义, 它包含 4 个阶段, 分别为初始化阶段、标签询问阶段、挑战阶段和应答阶段, 具体描述如下。

- 1) 初始化
挑战者 C 运行 Setup 和 KeyGen 算法, 生成公开参数 params、密钥对 (pk, sk), 并将 params、pk 返回给 A。
- 2) 标签询问
A 自适应地选择索引为 i 的消息块 m_i , 将 (i, m_i) 发送给 C。C 运行 TagGen 算法生成标签 σ_{m_i}

返回给 A。

3) 挑战

A 选择消息块索引集 I^* , I^* 中至少包含一个之前没有被查询过签名预言的索引 i^* 。A 将 I^* 发送给 C, C 运行 Challenge 算法, 生成挑战 chal 返回给 A。

4) 应答

A 输出应答 res^* 。如果 res^* 通过验证并且 I^* 中至少包含一个没有询问过标签的索引 i^* , 则 A 赢得游戏。A 在上述游戏中的优势定义为

$$Adv_A^{UC-CMA} = \Pr[l' \leftarrow \text{Verify}(\text{params}, \text{chal}^*, s, res^*)]$$

定义 5 (匿名性)。匿名云审计方案的匿名性由敌手 A 和挑战者 C 之间的游戏定义, 它包含 4 个阶段, 分别为初始化阶段、挑战阶段、应答阶段以及猜测阶段, 具体描述如下。

1) 初始化

C 运行 Setup 和 KeyGen 算法, 生成公开参数 params 和 n 对密钥 (pk_j, sk_j) 返回给 A。

2) 挑战

A 选择索引为 i 的消息块 m_i , 运行 Challenge 算法生成 chal, 并将 (i, m_i) , chal 返回给 C。

3) 应答

C 随机选择 $j \in \{1, \dots, n\}$, 运行 TagGen 算法生成标签 σ_{m_j} , 之后运行 Respond 算法生成应答 res, 并将 res 返回 A。

4) 猜测

A 验证 res 的有效性。如果 res 通过验证, 则 A 输出猜测 $j' \in \{1, \dots, n\}$ 。

如果 $j = j'$, 则 A 赢得游戏。A 在上述游戏中的优势定义为

$$Adv_A^{IND} = |\Pr[j = j'] - \frac{1}{n}|$$

2.3 基于身份的匿名云审计方案

定义 6 基于身份的匿名云审计方案由 6 个算法组成, 分别为初始化算法 Setup、密钥提取算法 KeyExt、标签生成算法 TagGen、挑战算法 Challenge、应答算法 Respond、验证算法 Verify。算法具体描述如下。

1) Setup(1^k): PKG 输入安全参数 1^k , 输出主密钥对 (mpk, msk) 。

2) KeyExt(mpk, msk, id_j): PKG 输入主公钥 mpk、主私钥 msk、身份 id_j , 输出基于身份的密钥 d_j 。

3) TagGen(mpk, d_j, i, m_i): 组用户输入主公钥 mpk、身份密钥 d_j 、索引为 i 以及消息 m_i , 输出标签 $\sigma_{(id_j, m_i)}$ 。

4) Challenge($mpk, id_1, \dots, id_n, I$): TPA 输入主公钥 mpk、 n 个用户身份 $id_1, \dots, id_n$ 、索引集合 I , 输出挑战 chal。

5) Respond($mpk, chal, M, \Sigma$): CS 输入主公钥 mpk、挑战 chal、消息集合 M 、标签集合 Σ , 输出应答 res。

6) Verify($mpk, chal, s, res$): TPA 输入主公钥 mpk、挑战 chal、秘密值 s 以及应答 res, 验证应答有效性。如果应答有效, 输出 1, 否则输出 0。

定义 7 (不可欺骗性)。基于身份的匿名云审计方案由敌手 A 和挑战者 C 之间的游戏定义。敌手 A 模仿云服务器的不诚实行为, 即在数据没有安全正确存储在服务器上时, 云服务器欺骗 TPA 数据仍被安全存储。游戏包含 5 个阶段, 分别为初始化阶段、密钥询问阶段、标签询问阶段、挑战阶段以及应答阶段, 具体描述如下。

1) 初始化

C 运行 Setup 算法, 生成主密钥对 (mpk, msk) 。C 选择 n 个身份 $(id_1, \dots, id_n)$, 将 $(id_1, \dots, id_n)$ 和主公钥 mpk 返回给 A, 自己保存 msk。

2) 密钥询问

A 询问它所选择的身份 id_j 的密钥, $id_j \notin \{id_1, \dots, id_n\}$ 。C 运行 KeyExt 算法, 生成基于用户身份的密钥 d_j , 并将 d_j 返回 A。如果身份 id_j 被询问两次, C 必须返回相同的密钥 d_j 。

3) 标签询问

A 可以选择任意身份 id_j 对索引为 i 的消息 m_i 的标签。C 运行 TagGen 算法, 生成标签 $\sigma_{(id_j, m_i)}$ 并返回 A。

4) 挑战

A 选择任意消息索引块 I^* , 其中至少包含一个之前没有被询问过的标签。C 运行 Challenge 算法生成挑战 $chal^*$, 并将 $chal^*$ 返回 A。

5) 应答

A 输出应答 res^* 。如果 A 输出的 res^* 通过验证且 I^* 中至少包含一个没有询问过标签的索引，则敌手 A 赢得游戏。A 赢得游戏的优势定义为：

$$Adv_A^{UC-ID-CMA} = \Pr[I' \leftarrow \text{Verify}(\text{params}, \text{chal}^*, s, res^*)]$$

定义 8 (匿名性)。除了要保证用户上传的信息被安全存储在云服务器外，安全的基于身份的匿名云审计方案还要保证 TPA 无法得知用户的身份。在审计过程中，不诚实的 TPA 可能会试图区分用户的身份。基于身份的匿名云审计方案的匿名性由敌手 A 和挑战者 C 之间的游戏定义。考虑敌手 A 拥有无限的计算能力，如果它在以下游戏中赢得游戏的优势为零，则该方案具有信息论安全的匿名性。游戏包含 4 个阶段，分别为初始化阶段、挑战阶段、应答阶段以及猜测阶段，具体描述如下：

1) 初始化

C 运行 Setup 算法生成主密钥对 (mpk, msk) ，运行 KeyExt 算法生成身份为 id_j 、密钥 d_j ， $j=1, \dots, n$ 。C 将 (mpk, msk) 、身份 $(\text{id}_1, \dots, \text{id}_n)$ 返回给 A。

2) 挑战

A 选择任意索引消息对 (i, m_i) ，运行 Challenge 算法生成挑战 chal 返回给 C。

3) 应答

C 随机选择 $j \in \{1, \dots, n\}$ ，运行 TagGen 算法，生成 (i, m_i) 的标签 $\sigma_{(\text{id}_j, m_{ij})}$ 之后，C 运行 Respond 算法生成应答 res ，将其返回给 A。

4) 猜测

A 运行 Verify 算法检查应答的有效性。如果应答通过验证，则输出 $j' \in \{1, \dots, n\}$ 。如果 $j = j'$ ，则敌手 A 赢得游戏。A 赢得游戏的优势定义为

$$Adv_A^{IND} = |\Pr[j = j'] - \frac{1}{n}|$$

3 基于身份的匿名云审计通用构造

3.1 通用构造

基于身份的匿名云审计方案的通用构造（如图 2 所示）包含一个传统签名方案 Π 和一个传统

匿名云审计方案 Ψ ，具体细节如下。

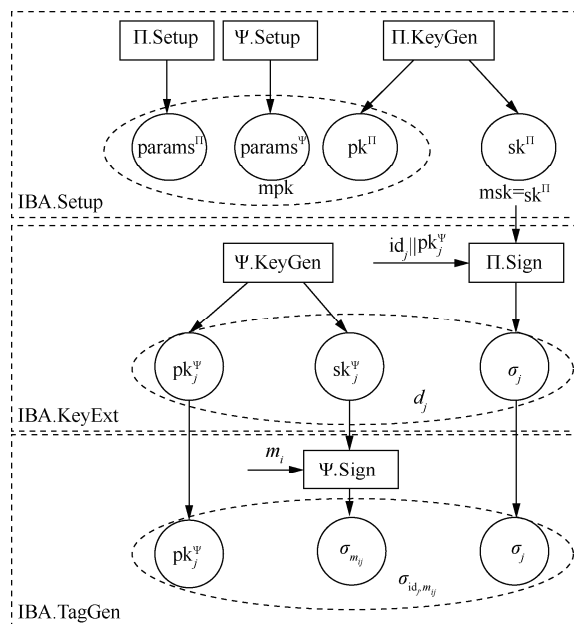


图 2 通用构造

Figure 2 General construction

设 $\Pi=(\Pi.\text{Setup}, \Pi.\text{KeyGen}, \Pi.\text{Sign}, \Pi.\text{Verify})$ ， $\Psi=(\Psi.\text{Setup}, \Psi.\text{KeyGen}, \Psi.\text{TagGen}, \Psi.\text{Challenge}, \Psi.\text{Respond}, \Psi.\text{Verify})$ ，基于身份的匿名云审计方案 $\text{IBA}=(\text{IBA}.\text{Setup}, \text{IBA}.\text{KeyExt}, \text{IBA}.\text{TagGen}, \text{IBA}.\text{Challenge}, \text{IBA}.\text{Respond}, \text{IBA}.\text{Verify})$ 构造如下。

1) $\text{IBA}.\text{Setup}(1^k)$ ：PKG 输入安全参数 1^k ，运行 $\Pi.\text{Setup}$ ， $\Psi.\text{Setup}$ 得到公共参数 params^Π 、 params^Ψ 之后，运行签名方案 Π 的密钥生成算法 $\Pi.\text{KeyGen}$ ，获得主私钥 $\text{msk} = \text{sk}^\Pi$ ，主公钥 $\text{mpk}=(\text{params}^\Pi, \text{params}^\Psi, \text{pk}^\Pi)$ 。

2) $\text{IBA}.\text{KeyExt}(\text{mpk}, \text{msk}, \text{id}_j)$ ：输入主私钥 msk ，第 j 个用户的身份 id_j ，按以下步骤生成基于身份的密钥。

a) PKG 运行匿名云审计方案 Ψ 的密钥生成算法 $\Psi.\text{KeyGen}$ 生成密钥对 $(\text{pk}_j^\Psi, \text{sk}_j^\Psi)$ 。

b) 运行签名方案 Π 的签名算法 $\Pi.\text{Sign}(\text{params}^\Pi, \text{msk}, \text{id}_j \parallel \text{pk}_j^\Psi)$ ，生成 $\text{id}_j \parallel \text{pk}_j^\Psi$ 的签名 σ_j 。

c) 身份 id_j 的密钥定义为 $d_j = (\text{pk}_j^\Psi, \text{sk}_j^\Psi, \sigma_j)$ 。

3) $\text{IBA}.\text{TagGen}(\text{mpk}, d_j, i, m_i)$ ：使用索引 i 标记消息块 m_i 。第 j 个用户按照如下步骤生成消息

m_i 的标签。

a) 运行匿名云审计方案 Ψ 的标签生成算法 $\Psi.\text{TagGen}(\text{params}^\Psi, \text{sk}_j^\Psi, i, m_i)$ ，得到标签 σ_{m_i} 。

b) 基于身份的密钥 d_j 对消息 m_i 生成的标签为 $\sigma_{(id_j, m_i)} = (\text{pk}_j^\Psi, \sigma_j, \sigma_{m_i})$ 。

c) 用户将索引-信息-标签元组 $(i, m_i, \sigma_{(id_j, m_i)})$ 上传至云服务器。

4) $\text{IBA.Challenge}(\text{mpk}, id_1, \dots, id_n, I)$ ：主要包含以下两个步骤。

a) TPA 向服务器询问身份 $id_1, \dots, id_n$ 对应的审计方案的公钥，服务器返回 $(\text{pk}_1^\Psi, \dots, \text{pk}_n^\Psi)$ 以及 $(\sigma_1, \dots, \sigma_n)$ 。

b) TPA 运行签名方案 Π 的验证算法 $\Pi.\text{Verify}$ 检验基于身份的密钥 d_j 的有效性。若验证通过，

$$\Pr \left[\begin{array}{l} \text{IBA.Verify}(\text{mpk}, \text{chal}, s, \text{res}) = 1 \\ \text{res} \leftarrow \text{IBA.Respond}(\text{mpk}, \text{chal}, M, \Sigma) \end{array} \right] \begin{array}{l} (\text{msk}, \text{mpk}) \leftarrow \text{IBA.Setup}(1^k) \\ d_j \leftarrow \text{IBA.KeyExt}(\text{mpk}, \text{msk}, id_j) \\ \sigma_{(id_j, m_i)} \leftarrow \text{IBA.TagGen}(\text{mpk}, d_j, i, m_i) \\ \text{chal} \leftarrow \text{IBA.Challenge}(\text{mpk}, id_1, \dots, id_n, I) \end{array} \quad (1)$$

3.2 安全性分析

定理 1 (不可欺骗性)。假设签名方案 Π 具有不可伪造性，匿名云审计方案 Ψ 具有不可欺骗性，则利用我们的通用构造设计的基于身份的匿名云审计方案 IBA 具有不可欺骗性。

证明 如果存在针对方案 IBA 不可欺骗性的成功的敌手 A_{IBA} ，那么存在能够成功破坏签名方案 Π 或匿名云审计方案 Ψ 的模拟者 B，即

$$\text{Adv}_{\text{IBA}, A_{\text{IBA}}}^{\text{cheat}}(k) \leq \text{Adv}_{\Pi, B}^{\text{forge}}(k) + \text{Adv}_{\Psi, B}^{\text{cheat}}(k)$$

令 B 为签名方案 Π 和匿名云审计方案 Ψ 的敌手，令 C 为 Π 和 Ψ 的挑战者，并与 B 进行交互。同时，B 作为方案 IBA 的挑战者与 A_{IBA} 进行交互。

1) 初始化

B 从 C 处获取公开参数 params^Π 、 params^Ψ 、 pk^Π ，方案 Ψ 的 n 个公钥 $(\text{pk}_1^\Psi, \dots, \text{pk}_n^\Psi)$ 以及 n 个身份 $(id_1, \dots, id_n)$ 。B 设主公钥为 $\text{mpk} = (\text{params}^\Pi, \text{params}^\Psi, \text{pk}^\Pi)$ ，并将 mpk 和

则运行匿名云审计方案 Ψ 的挑战算法 $\Psi.\text{Challenge}$ 生成一个挑战 chal ，并将其返回给云服务器。

5) $\text{IBA.Respond}(\text{mpk}, \text{chal}, M, \Sigma)$ ：云服务器在接收挑战 chal 后，运行匿名云审计方案 Ψ 的应答算法 $\Psi.\text{Respond}$ ，计算应答 res 发送给 TPA。

6) $\text{IBA.Verify}(\text{mpk}, \text{chal}, s, \text{res})$ ：TPA 在收到服务器发来的应答后，运行匿名云审计方案 Ψ 的验证算法 $\Psi.\text{Verify}$ 检验应答 res 的有效性。如果验证成功，输出 1，否则输出 0。

3.3 正确性定义

如果一个基于身份的匿名云审计方案 IBA 是正确的，那么该方案满足以下条件：对于任意安全参数 k 、任意消息块 m_i ，以及任意用户的身份 id_j ，式(1)的概率为 1。

$(id_1, \dots, id_n)$ 返回给 A_{IBA} 。

B 维护一个列表 L，用于存储用户询问的身体的密钥。表 L 的结构为 $(id, \text{pk}^\Psi, \text{sk}^\Psi, \sigma_{id})$ ，其中 id 表示身份， pk^Ψ 和 sk^Ψ 为 id 在 Ψ 方案下的密钥， σ_{id} 为 id 在 Π 方案下的签名。B 将身份 id_j 和公钥 pk_j^Ψ ($j \in \{1, \dots, n\}$) 一一对应插入表 L 中。

2) 密钥询问

A_{IBA} 可以询问它所选择的身份为 id_j ($j \in \{1, \dots, n\}$) 的密钥，B 运行 $\Psi.\text{KeyGen}$ 生成密钥对 $(\text{pk}_j^\Psi, \text{sk}_j^\Psi)$ 。之后，B 将 $id_j \parallel \text{pk}_j^\Psi$ 发送给签名预言 O_{Sign}^Π ，获得对消息 $id_j \parallel \text{pk}_j^\Psi$ 的有效签名 σ_j 。B 将 $(id_j, \text{pk}_j^\Psi, \text{sk}_j^\Psi, \sigma_j)$ 插入表 L 中，并将密钥 $d_j = (\text{pk}_j^\Psi, \text{sk}_j^\Psi, \sigma_j)$ 返回 A_{IBA} 。

3) 标签询问

A_{IBA} 可以询问任意被身份 id_j 签名的索引为 i 的消息 m_i 的标签。根据身份 id_j 的不同共有以下 3 种情况：

a) $j \notin \{1, \dots, n\}$ 且列表 L 中存在 id_j 的记录：
 B 拥有 sk_j^Ψ ，运行 $\Psi.\text{TagGen}$ 获得 σ_{m_j} 并设标签 $\sigma_{(\text{id}_j, m_j)} = (\text{pk}_j^\Psi, \sigma_j, \sigma_{m_j})$ ，将结果返回给 A_{IBA} 。

b) $j \in \{1, \dots, n\}$: B 向 C 询问 pk_j^Ψ 对应的 sk_j^Ψ ，之后将 $\text{id}_j \parallel \text{pk}_j^\Psi$ 绑定后发送给 O_{Sign}^Π ，获得签名 σ_j 。B 将 $(\text{id}_j, \text{pk}_j^\Psi, \text{sk}_j^\Psi, \sigma_j)$ 插入表 L 中，之后的操作与 a) 相同。

c) 如果 id_j 不属于以上两种情况，即列表中不存在与 id_j 有关的记录，则 B 运行 $\Psi.\text{KeyGen}$ 生成 $(\text{pk}_j^\Psi, \text{sk}_j^\Psi)$ ，通过 O_{Sign}^Π 获得 σ_j ，将 $(\text{id}_j, \text{pk}_j^\Psi, \text{sk}_j^\Psi, \sigma_j)$ 插入表 L 中，之后的操作与 a) 相同。

4) 挑战

A_{IBA} 选择任意消息索引块 I^* ，其中至少包含一个之前没有询问过的索引 i' ，B 将 I^* 发送给 C 获得挑战 chal ，并将其发送给 A_{IBA} 。

5) 伪造

A_{IBA} 输出伪造的应答 res 。

如果敌手 A_{IBA} 赢得游戏，这意味着输出的应答 res 有效。B 查询表中记录 $(\text{id}_j, \text{pk}_j^\Psi, \text{sk}_j^\Psi, \sigma_j)$ 。

1) 如果 $\text{pk}_j^\Psi \notin (\text{pk}_1^\Psi, \text{pk}_2^\Psi, \dots, \text{pk}_n^\Psi)$ ，则 $(\text{id}_j \parallel \text{pk}_j^\Psi, \sigma_j)$ 是针对签名方案 Π 的成功伪造。B 可以将其发送给 C 破坏方案 Π 。

2) 如果 $\text{pk}_j^\Psi \in (\text{pk}_1^\Psi, \text{pk}_2^\Psi, \dots, \text{pk}_n^\Psi)$ ，则 $(\text{pk}_j^\Psi, \sigma_j, \sigma_{m_j})$ 是针对匿名云审计方案 Ψ 的成功欺骗。B 可以将其发送给 C 破坏方案 Ψ 。

在这种情况下， A_{IBA} 的成功欺骗意味着标准签名方案 Π 的成功伪造或匿名云审计方案 Ψ 的成功欺骗。由于签名方案 Π 已证明具有不可伪造性，匿名云审计方案 Ψ 也具有不可欺骗性，因此方案 IBA 具有不可欺骗性。

定理 2 (匿名性)。假设传统体制下的匿名云审计方案 Ψ 具有信息论匿名性，则基于身份的匿名云审计方案 IBA 具有信息论匿名性。

证明 如果存在针对方案 IBA 匿名性的成功的敌手 A_{IBA} ，那么存在能够成功破坏方案 Ψ 成功的 B。

令 B 作为方案 Ψ 的敌手，并设 C 为 Ψ 的挑

战者与 B 进行交互。同时，B 作为方案 IBA 的挑战者与 A_{IBA} 进行交互。

1) 初始化

B 运行 $\Pi.\text{Setup}$ 和 $\Pi.\text{KeyGen}$ 算法，生成公开参数 params^Π 、密钥对 $(\text{pk}^\Pi, \text{sk}^\Pi)$ ，并从 C 处获取公开参数 params^Ψ 、 n 个公钥 $(\text{pk}_1^\Psi, \dots, \text{pk}_n^\Psi)$ 以及 n 个身份 $(\text{id}_1, \dots, \text{id}_n)$ 。B 设主公钥 $\text{mpk} = (\text{params}^\Pi, \text{params}^\Psi, \text{pk}^\Pi)$ ，主私钥 $\text{msk} = \text{sk}^\Pi$ ，将 (mpk, msk) 和 $(\text{id}_1, \dots, \text{id}_n)$ 返回 A_{IBA} 。

B 维护列表 L ，用于存储用户询问的密钥。表 L 的结构为 $(\text{id}, \text{pk}^\Psi, \sigma_{\text{id}})$ ，其中 id 表示身份， pk^Ψ 为 id 在 Ψ 方案下的密钥， σ_{id} 为 id 在 Π 方案下的签名。B 将 $\text{id}_j \parallel \text{pk}_j^\Psi$ 绑定，运行 $\Pi.\text{Sign}$ 算法计算 σ_j ，将身份 id_j 、公钥 pk_j^Ψ 以及签名 σ_j ($j \in \{1, \dots, n\}$) 一一对应插入表 L 中。

2) 挑战

A_{IBA} 选择存储的一个索引消息对 (i, m_i) ，运行 $\Psi.\text{Challenge}$ 算法生成挑战 $\text{chal} = (i, s_i, \text{pk}_{\text{chal}})$ 。 A_{IBA} 将索引消息对 (i, m_i) 和挑战 chal 一起发送给 B。

3) 应答

B 首先验证挑战的有效性。若验证成功，B 随机选择 $j \in \{1, \dots, n\}$ ，运行 $\Psi.\text{TagGen}$ 生成 (i, m_i) 对应的标签 σ_{m_j} 。之后，B 运行 $\Psi.\text{Respond}$ 生成应答 $\text{res} = (\mu, \sigma_{\text{res}})$ ，并将其返回给敌手。

4) 猜测

A_{IBA} 首先判断应答有效性。若应答有效，则输出猜测 $j' \in \{1, \dots, n\}$ 。

如果 $j = j'$ ，则 A_{IBA} 赢得游戏，这意味着 B 能够成功猜测出 Ψ 方案中的身份。由于 Ψ 方案已证明具有信息论上的匿名性，因此 IBA 方案具有匿名性。

4 基于身份的匿名云审计具体实例

BLS (boneh-lynn-shacham) 签名算法是一种可以实现签名聚合和密钥聚合的算法。使用 BLS 签名算法构造多用户匿名云审计方案，能够降低计算开销和通信开销。此外，Wu 等^[13]提出的匿名云审计方案标签紧凑，仅包含一个元素，能够降低标签的存储成本，也能使审计效率和验证效

率大大提高。因此,具体实例以这两个方案为模块,构造具有紧凑标签的基于身份的匿名云审计方案。

4.1 标准签名方案构造

Boneh 等^[21]提出了 BLS 签名方案 $\Pi = (\Pi.Setup, \Pi.KeyGen, \Pi.Sign, \Pi.Verify)$, 具体细节如下。

1) $\Pi.Setup(1^k)$: 初始化算法以安全参数 1^k 作为输入, 输出公开参数 $params = (p, G_1, G_2, g \in G_1, e: G_1 \times G_1 \rightarrow G_2, H_1: \{0,1\}^* \rightarrow G_1)$ 。

2) $\Pi.KeyGen(params)$: 密钥生成算法以公开参数 $params$ 作为输入, 随机选择 $\alpha \in Z_p^*$, 令 $sk = \alpha$, $pk = g^\alpha$, 输出密钥对 (pk, sk) 。

3) $\Pi.Sign(params, sk, m)$: 签名算法以公开参数 $params$, 私钥 $sk = \alpha$ 以及消息 m 为输入, 计算 $H_1(m) \in G_1$ 和 $H_1(m)^\alpha$ 。之后, 输出消息 m 的签名 $\sigma_m = H_1(m)^\alpha$ 。

4) $\Pi.Verify(params, pk, m, \sigma_m)$: 验证算法以公开参数 $params$ 、公钥 $pk = g^\alpha$ 、消息 m 以及签名 σ_m 为输入。如果 $e(\sigma_m, g) = e(H_1(m), g^\alpha)$, 输出 1, 否则输出 0。

引理 1 (不可伪造性)。设哈希函数 H_1 是一个随机预言。如果 CDH 问题是困难问题, 那么 BLS 签名在 EU-CMA 安全模型中是安全的, 规约损失 $L = q_H$, 其中 q_H 为随机预言机的哈希查询数量。

证明 详见参考文献[21]。

4.2 匿名云审计方案

Wu 等^[13]提出了具有紧凑标签的匿名云审计方案 $\Psi = (\Psi.Setup, \Psi.KeyGen, \Psi.TagGen, \Psi.Challenge, \Psi.Respond, \Psi.Verify)$, 具体细节如下。

1) $\Psi.Setup(1^k)$: 初始化算法以安全参数 1^k 作为输入, 输出公开参数 $params = (p, G_1, G_2, e: G_1 \times G_1 \rightarrow G_2, \mu, g \in G_1, H_2: \{0,1\}^* \rightarrow G_1)$ 。

2) $\Psi.KeyGen(params)$: 密钥生成算法以公开参数 $params$ 作为输入, 随机选择 $x \in Z_p^*$, 令 $sk = x$, $pk = g^x$, 输出密钥对 (pk, sk) 。

3) $\Psi.TagGen(params, sk, i, m_i)$: 标签生成算法以公开参数 $params$ 、私钥 sk 以及索引为 i 的消息块 $m_i \in Z_p$ 作为输入, 计算标签

$\sigma_{m_i} = (H_2(i) \cdot u^{m_i})^{1/x}$ 。之后将索引-消息-标签元组 (i, m_i, σ_{m_i}) 上传至云服务器。

4) $\Psi.Challenge(params, pk_1, \dots, pk_n, I)$: 挑战算法以公开参数 $params$ 、 n 个用户的公钥 $(pk_1, \dots, pk_n)$ 、标签子集 I 为输入, 进行如下操作:

a) 随机选择 $s \in Z_p^*$, $h \in G_1$, 计算 $pk_{chal} = (pk_1^s, \dots, pk_n^s, h, h^s)$;

b) 对于 $\forall i \in I$, 随机选择 $s_i \in Z_p^*$, 令 $Q = \{(i, s_i) | i \in I\}$, 输出挑战 $chal = (Q, pk_{chal})$ 。

5) $\Psi.Respond(params, chal, M, \Sigma)$: 应答算法以公开参数 $params$ 、挑战 $chal$ 、消息集合 $M = \{m_i | i \in I\}$ 、标签集合 $\Sigma = \{\sigma_{m_i} | i \in I\}$ 为输入, 按如下操作:

a) 检查 $e(pk_j^s, h) = e(pk_j, h^s), j \in \{1, \dots, n\}$;

b) 计算

$$\mu = \sum_{(i, s_i) \in Q} s_i \cdot m_i, \sigma_{res} = \prod_{(i, s_i) \in Q} e(\sigma_{m_i}^{s_i}, pk_j^s),$$

输出 $res = (\mu, \sigma_{res})$ 。

6) $\Psi.Verify(params, chal, s, res)$: 验证算法以公开参数 $params$ 、挑战 $chal$ 、秘密值 s 以及应答 res 为输入, 计算下式是否成立:

$$\sigma_{res} = e(\prod_{(i, s_i) \in Q} H_2(i)^{s_i} \cdot u^\mu, g^s)$$

若成立, 输出 1, 否则输出 0。

引理 1 (不可欺骗性)。如果有一个在时间 t 内运行的敌手, 最多询问 q 个哈希预言, 并且可以自适应地询问签名预言, 则它在对某个挑战输出一个有效的应答方面具有优势 ϵ , 那么, 有一个在多项式时间运行的仿真算法通过与敌手的交互解决 vBDH 问题的优势至少为 ϵ/q 。

证明 详见参考文献[13]。

引理 2 (匿名性)。匿名多用户云审计方案达到信息论上的匿名性。

证明 详见参考文献[13]。

4.3 参照通用构造的基于身份匿名云审计实例方案

4.3.1 方案构造

结合上述标准签名方案 Π 、具有紧凑标签的匿名云审计方案 Ψ , 具有紧凑标签的基于身份的匿名云审计方案 $IBA = (IBA.Setup, IBA.KeyExt,$

IBA.TagGen, IBA.Challenge, IBA.Respond, IBA.Verify), 具体构造如下。

1) IBA.Setup(1^k): PKG 输入安全参数 1^k , 选择两个素数阶为 p 的循环群 G_1, G_2 , 双线性映射 $e: G_1 \times G_1 \rightarrow G_2$, $u, g \in G_1$, 抗碰撞哈希函数 $H_1, H_2: \{0,1\}^* \rightarrow G_1$, 则公开参数 $\text{params} = (p, G_1, G_2, e, u, g, H_1, H_2)$ 。随机选择 $\alpha \in Z_p^*$, 主私钥 $\text{msk} = \alpha$, 主公钥 $\text{mpk} = (g^\alpha, \text{params})$ 。

2) IBA.KeyExt($\text{mpk}, \text{msk}, \text{id}_j$): PKG 输入第 j 个用户的身份 id_j , 主私钥 msk , 按以下步骤生成基于身份的密钥。

a) 随机选择 $x_j \in Z_p^*$, 则 $sk_j = x_j, pk_j = g^{x_j}$;

b) 计算 $\sigma_j = H_1(\text{id}_j \parallel g^{x_j})^\alpha$;

c) 基于身份的密钥为 $d_j = (g^{x_j}, x_j, \sigma_j)$ 。

3) IBA.TagGen(mpk, d_j, i, m_i): 索引 i 中存储的内容为 m_i , 第 j 个用户按如下步骤对 m_i 签名:

a) 计算 $\sigma_{m_{ij}} = (H_2(i) \cdot u^{m_i})^{1/x_j}$;

b) 使用 d_j 对消息 m_i 进行计算, 产生的标签为 $\sigma_{(\text{id}_j, m_{ij})} = (g^{x_j}, \sigma_j, \sigma_{m_{ij}})$;

c) 将索引-信息-标签元组 $(i, m_i, \sigma_{(\text{id}_j, m_{ij})})$ 上传至云服务器中。

4) IBA.Challenge($\text{mpk}, \text{id}_1, \dots, \text{id}_n, I$): 为了检查用户数据的完整性。

a) TPA 发送身份 $(\text{id}_1, \dots, \text{id}_n)$ 给服务器, 服务器将 $(pk_1, \dots, pk_n)$ 和 $(\sigma_1, \dots, \sigma_n)$ 返回给 TPA。

b) TPA 检查下式是否成立。

$$e(\sigma_j, g) \stackrel{?}{=} e(H_1(\text{id}_j \parallel g^{x_j}), g^\alpha)$$

若成立, TPA 选择一个随机的具有 c 个元素的子集 $I \subset [1, n]$ 作为审计过程中挑战的索引块。对于 $\forall i \in I$, 选择随机值 $s_i \in Z_p^*$, 令 $Q = \{(i, s_i) \mid i \in I\}$ 。

c) 选择 $s \in Z_p^*$, 计算 $\text{pk}_{\text{chal}} = (pk_1^s, \dots, pk_n^s, h, h^s)$ 。TPA 将挑战 $\text{chal} = (Q, \text{pk}_{\text{chal}})$ 发送给服务器。

5) IBA.Respond($\text{mpk}, \text{chal}, M, \Sigma$): 服务器在收到 TPA 发来的挑战 chal 时, 首先检查以下公式是否成立。

$$e(pk_j^s, h) \stackrel{?}{=} e(pk_j, h^s)$$

之后, 按照以下步骤生成证明。

a) 根据 $M = \{m_i \mid i \in I\}$, 计算挑战块的合集。

$$\mu = \sum_{(i, s_i) \in Q} s_i \cdot m_i$$

b) 根据对应的标签 $\Sigma = \{\sigma_{(\text{id}_j, m_{ij})} \mid i \in I\}$, 计算应答标签。

$$\sigma_{\text{res}} = \prod_{(i, s_i) \in Q} e(\sigma_{m_{ij}}^{s_i}, pk_j^s)$$

c) 将应答 $\text{res} = (\mu, \sigma_{\text{res}})$ 发送给 TPA。

6) IBA.Verify($\text{mpk}, \text{chal}, s, \text{res}$): TPA 收到服务器发来的应答后, 如果

$$\sigma_{\text{res}} = e(\prod_{(i, s_i) \in Q} H_2(i)^{s_i} \cdot u^\mu, g^s)$$

则验证成功, TPA 认为用户上传至云服务器中数据的完整性得到保证。

4.3.2 正确性分析

假设数据被安全地存储在云服务器中, 数据存储格式为 $(i, m_i, \sigma_{(\text{id}_j, m_{ij})})$, 其中

$$\sigma_{(\text{id}_j, m_{ij})} = (g^{x_j}, \sigma_j, \sigma_{m_{ij}})$$

$$\sigma_j = H_1(\text{id}_j \parallel g^{x_j})^\alpha, \quad \sigma_{m_{ij}} = (H_2(i) \cdot u^{m_i})^{1/x_j}$$

为了检验云中数据的完整性, TPA 首先通过 $e(\sigma_j, g) \stackrel{?}{=} e(H_1(\text{id}_j \parallel g^{x_j}), g^\alpha)$ 验证密钥正确性。若密钥正确, 则随机选择大小为 c 的集合 $I \subset [1, n]$ 作为挑战块。对于 $\forall i \in I$, 选择 $s_i \in Z_p^*$, 令 $Q = \{(i, s_i) \mid i \in I\}$ 。选择 $s \in Z_p^*$, 令

$$\text{pk}_{\text{chal}} = (pk_1^s, \dots, pk_n^s, h, h^s),$$

挑战 $\text{chal} = (Q, \text{pk}_{\text{chal}})$ 。

云服务器在接收到挑战 chal 后, 通过

$$e(pk_j^s, h) \stackrel{?}{=} e(pk_j, h^s)$$

验证秘密值 s 的正确性。之后, 计算

$$\mu = \sum_{(i, s_i) \in Q} s_i \cdot m_i, \quad \sigma_{\text{res}} = \prod_{(i, s_i) \in Q} e(\sigma_{m_{ij}}^{s_i}, pk_j^s), \quad i \in I$$

其中,

$$\begin{aligned}\sigma_{\text{res}} &= \prod_{(i,s_i) \in Q} e(\sigma_{m_{ij}}^{s_i}, \text{pk}_j^s) = \\ & \prod_{(i,s_i) \in Q} e((H_2(i) \cdot u^{m_i})^{1/x_j \cdot s_i}, g^{x_j \cdot s}) = \\ & \prod_{(i,s_i) \in Q} e((H_2(i) \cdot u^{m_i})^{s_i}, g^s) = \\ & \prod_{(i,s_i) \in Q} e(H_2(i)^{s_i} \cdot u^{m_i \cdot s_i}, g^s) = \\ & \prod_{(i,s_i) \in Q} e(H_2(i)^{s_i}, g^s) \cdot \prod_{(i,s_i) \in Q} e(u^{m_i \cdot s_i}, g^s)\end{aligned}$$

TPA 在接收到云服务器发来的应答 σ_{res} 后, 验证

$$\sigma_{\text{res}} \stackrel{?}{=} e\left(\prod_{(i,s_i) \in Q} H_2(i)^{s_i} u^\mu, g^s\right)$$

TPA 计算如下:

$$\begin{aligned}\sigma_{\text{res}} &= \prod_{(i,s_i) \in Q} e(H_2(i)^{s_i}, g^s) \cdot \prod_{(i,s_i) \in Q} e(u^{m_i \cdot s_i}, g^s) = \\ & \prod_{(i,s_i) \in Q} e(H_2(i)^{s_i}, g^s) \cdot e(u^{\sum_{(i,s_i) \in Q} m_i \cdot s_i}, g^s) = \\ & e\left(\prod_{(i,s_i) \in Q} H_2(i)^{s_i}, g^s\right) \cdot e(u^\mu, g^s) = \\ & e\left(\prod_{(i,s_i) \in Q} H_2(i)^{s_i} \cdot u^\mu, g^s\right)\end{aligned}$$

易得, 如果数据被安全存储在云中, 没有被篡改, 则 $\sigma_{\text{res}} = e\left(\prod_{(i,s_i) \in Q} H_2(i)^{s_i} \cdot u^\mu, g^s\right)$ 。

因此, 有:

$$\text{Verify}(\text{mpk}, \text{chal}, s, \text{res}) = 1$$

4.3.3 安全性分析

引理 3 (不可欺骗性)。第 3 节中证明了如

果一个传统签名方案具有不可伪造性, 一个匿名云审计方案具有不可欺骗性, 则由这两个方案构造的基于身份的匿名云审计方案具有不可欺骗性。由于 BLS 签名方案 Π 具有不可伪造性, 紧凑标签的匿名云审计方案 Ψ 具有不可欺骗性, 因此, 该方案具有不可欺骗性。

引理 4 (匿名性)。第 3 节中证明了如果一个传统签名方案具有不可伪造性, 一个匿名云审计方案具有匿名性, 则由这两个方案构造的基于身份的匿名云审计方案具有匿名性。由于 BLS 签名方案 Π 具有不可伪造性, 紧凑标签的匿名云审计方案 Ψ 具有匿名性, 因此, 该方案具有匿名性。

5 讨论

本节从标签大小、挑战计算开销、应答计算开销以及验证计算开销 4 个方面, 对本文提出的基于身份的匿名云审计方案 IBA 和传统体制下的匿名云审计方案^[13]、基于身份体制下采用环签名的云审计方案^[22]进行比较。表 1 列出了比较中所使用的各个符号定义。表 2 列出了 3 个方案的在以上 4 个方面的性能比较。

表 1 符号定义
Table 1 Notations definition

符号	意义
$\ G\ $	群中元素二进制长度
$\ Z_p\ $	整数集合中二进制长度
BP	双线性对计算开销
E	指数计算开销
H	哈希计算开销

表 2 性能比较
Table 2 performance comparison

方案	标签大小	挑战计算开销	应答计算开销	验证计算开销
文献[13]方案	$\ G\ $	$(n+1)E$	$(c+2)BP + cE$	$(c+2)E + BP$
文献[22]方案	$\ Z_p\ + n\ G\ $	\	\	$2nBP + nH + nG$
本文方案	$\ G\ $	$2BP + (n+1)E$	$(c+2)BP + cE$	$(c+2)E + BP$

注: 文献[22]主要采用基于身份的环签名技术和承诺方案实现审计, 因此主要的开销在标签的大小和验证标签的计算上, 挑战和应答过程并未进行说明, 表中用“\”表示; 本文的构造中, 单独的标签由 3 个元素构成, 但每个标签均含有公共部分 pk_j 和 σ_j , 故针对大量的数据标签而言, 每个数据块的平均存储成本可近似认为一个群元素。

从表 2 可以看出, 与文献[13]方案相比, 本文提出的基于身份的匿名云审计方案仅在标签大小和挑战计算开销上有所增加, 在应答计算开销和验证计算开销上与文献[13]方案保持一致。与文献[22]方案相比, 本文方案在标签大小上显著减少, 在验证计算开销方面, 效率也高于文献[22]方案。

6 结束语

本文提出了一种基于身份的匿名云审计方案的通用构造, 并使用具体的签名方案和匿名云审计方案实现了这一构造。在该方案中, 用户可以将文件上传至云服务器, 与组中其他用户共享。用户可以委托 TPA 检验存储的数据是否被篡改或删除, 确保上传数据的完整性。由于采用的匿名云审计方案具有紧凑标签这一优势, 该方案产生的标签也是紧凑的, 大大降低了存储成本。此外, 该方案证明了在 TPA 不诚实的情况下, 无法辨认出上传文件是组中的哪一个用户, 保证了用户的匿名性。综上, 该方案具有理想的安全性。

参考文献:

- [1] BHANJANTRI L B, MUJAWAR T.A survey of cloud computing security challenges, issues and their countermeasures[C]// 2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud). 2019: 376-380.
- [2] GUDEME J R, PASUPULETI S, KANDUKURI R. Certificateless privacy preserving public auditing for dynamic shared data with group user revocation in cloud storage[J]. Journal of Parallel and Distributed Computing, 2021, (156): 163-175.
- [3] RABANINEJAD R, RAJABZADEH ASAAR M, AHMADIAN ATTARI M, et al. An identity-based online/offline secure cloud storage auditing scheme[J]. Cluster Computing: The Journal of Networks, Software Tools and Applications, 2020, 23(2): 55-68.
- [4] DESWARTE Y, QUISQUATER J J, SAÏDANE A. Remote integrity checking[C]//Integrity and Internal Control in Information Systems VI-IICIS 2003. 2003: 1-11.
- [5] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//14th ACM Conference: Computer & Communications Security. 2007: 598-609.
- [6] BOYANG W, BAOCHUN L, HUI L. Oruta: privacy-preserving public auditing for shared data in the cloud[J]. IEEE Transactions on Cloud Computing, 2014, 2(1): 43-56.
- [7] FENG Y, MU Y, YANG G, et al. A new public remote integrity checking scheme with user privacy[C]//20th Australasian Conference on Information Security and Privacy (ACISP 2015). 2015: 377-394.
- [8] BOYANG W, HUI L, MING L. Privacy-preserving public auditing for shared cloud data supporting group dynamics[C]//2013 IEEE International Conference on Communications-ICC. 2013: 1946-1950.
- [9] BONEH D, GENTRY C, LYNN B, et al. Aggregate and verifiably encrypted signatures from bilinear Maps[C]//Advances in Cryptology—EUROCRYPT 2003. 2003: 416-432.
- [10] RIVEST R L, SHAMIR A, TAUMAN Y. How to leak a secret[C]//Advances in Cryptology—ASIACRYPT 2001. 2001: 552-565.
- [11] FERRARA A L, GREEN M, HOHENBERGER S, et al. Practical short signature batch verification[C]// Topics in Cryptology—CT-RSA 2009. 2009: 309-324.
- [12] BONEH D, BOYEN X, SHACHAM H. Short group signatures[C]//Advances in Cryptology—CRYPTO 2004. 2004: 41-55.
- [13] WU G, MU Y, SUSILO W, et al. Privacy-preserving cloud auditing with multiple uploaders[C]// Information Security Practice and Experience-ISPEC 2016. 2016: 224-237.
- [14] SHACHAM H, WATERS B. Compact proofs of retrievability[J]. Journal of Cryptology, 2013, 26(3): 442-83.
- [15] SHAMIR A. Identity-based cryptosystems and signature schemes[C]//Advances in Cryptology—CRYPTO 1984. 1984: 47-53.
- [16] YU Y, AU M H, ATENIESE G, et al. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(4): 767-78.
- [17] ZHANG Y, YU J, HAO R, et al. Enabling efficient user revocation in identity-based cloud storage auditing for shared big data[J]. IEEE Transactions on Dependable and Secure Computing, 2020, 17(3): 608-619.
- [18] WENTING S, JING Q, JIA Y, 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): 331-46.
- [19] BELLARE M, NAMPREMPRE C, NEVEN G. Security proofs for identity-based identification and signature schemes[J]. Journal of Cryptology, 2009, 22(1): 1-61.
- [20] GALINDO D, HERRANZ J, KILTZ E. On the generic construction of identity-based signatures with additional properties[C]//Advances in Cryptology—ASIACRYPT 2006. 2006: 178-193.

[21] BONEH D, LYNN B, SHACHAM H. Short signatures from the Weil pairing[J]. Pairings and Their Use in Cryptology, 2004, 17(4): 297-319.

[22] SINGH S, THOKCHOM S. Public integrity auditing for shared dynamic cloud data[C]//6th International Conference on Smart Computing and Communications (ICSCC) 2018: 698-708.



丁宁（1990—），女，江苏宿迁人，主要研究方向为网络安全、可信体系建设。

[作者简介]



卢晨昕（1998—），女，福建宁德人，东南大学硕士生，主要研究方向为密码学。



陈立全（1976—），男，广西玉林人，东南大学教授、博士生导师，主要研究方向为信息安全、密码学和网络安全协议。



陈兵（1970—），男，江苏泗阳人，宿迁市互联网信息办公室副主任，主要研究方向为网络安全、可信体系建设。



吴戈（1990—），男，江苏徐州人，东南大学副研究员，主要研究方向为密码学。