南京师范大学

云数据完整性验证

姓名：　濮世杰

学号：　212243038

指导老师：　陆阳 教授

培养单位：　计算机与电子信息学院/
人工智能学院

专业：　电子信息（计算机技术）

摘要

远程数据完整性检查(RDIC)或远程数据占有检查(RDPC)协议可以使数据存储服务器，比如云服务器，能够向验证者证明它实际上是在诚实地存储数据所有者的数据。到目前为止，文献中已经提出了许多 RDIC 协议，但大多数结构都存在复杂的密钥管理问题，即它们依赖于昂贵的公钥基础设施(PKI)，这可能会阻碍协议在实践中的部署。

Yong Yu 等人在文献《Identity-Based Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage》中提出了一种新的基于身份的 RDIC 协议，通过利用基于密钥同态的密码原语，以降低系统复杂性和基于 PKI 的 RDIC 方案中建立和管理公钥认证框架的成本。他们形式化了基于身份的 RDIC 及其安全模型，包括针对恶意云服务器的安全和针对第三方验证者的零知识隐私。其提出的基于身份的 RDIC 协议在远程数据完整性检查过程中不会向验证者泄露所存储数据的信息。这个新构造被证明在一般群模型中针对恶意服务器安全，并且对验证者实现零知识隐私。广泛的安全性分析和实现结果表明，提出的协议在实际应用中是可证明安全且实用的。

但基于身份的协议具有密钥托管的固定缺陷。为了解决这些问题，Jiguo Li 等人在《Certificateless Public Integrity Checking of Group Shared Data on Cloud Storage》中，利用无证书签名技术提出了一种新的 RDPC 协议，用于检查一个群之间共享的数据的完整性。在此案中，用户的私钥包括两个部分：一个由群管理员生成的部分密钥和一个由他自己选择的秘密值。为了确保在数据完整性检查期间选择了正确的公钥，每个用户的公钥都与他的唯一身份相关联，例如名称或电话号码。因此，我们不再需要证书，同时也消除了密钥托管的问题。同时，数据的完整性仍然可以由公共验证者进行审计，而无需下载整个数据。此外，此方案还支持有效的用户撤销。他们将方案的安全性规约到 CDH 和 DL 问题的假设。实验结果表明，该方案具有一定的有效性和可行性。

本文是对上述基于身份和无证书的两个方案的阅读报告，我们分别使用基于身份的 RDIC 方案和无证书的 RDPC 方案来表述上述两个方案。我们将分别介绍这两个方案，包括系统模型、安全模型、方案构造和安全证明。

基于身份的 RDIC 方案

1 介绍

云计算[1]，是一个分布式计算模型，覆盖大量共享虚拟化计算资源，如存储能力、应用程序和服务，受到了学术界和工业界的广泛关注。云用户可以在云计算环境中随意提供和发布资源。这种新的计算模型代表了作为水电等公用事业提供计算服务的新愿景。云计算为云用户带来了许多好处。例如，（1）用户可以减少在硬件、软件和服务上的资本支出，因为他们只为他们使用的东西付费；（2）用户可以享受低管理开销和立即访问各种应用程序；（3）用户可以在任何他们有网络的地方访问他们的数据，而不必呆在他们的计算机附近。

然而，在云计算被广泛应用之前，存在着各种各样的障碍。甲骨文公司最近的一项调查引用了来自国际数据公司企业小组的数据源，结果显示，安全性占云用户担忧的 87%。云用户的主要安全问题之一是外包文件的完整性，因为他们不再实际上拥有自己的数据，因此失去了对自己数据的控制。此外，云服务器不完全受信任，云服务器也不必须报告数据丢失事件。事实上，为了确定云计算的可靠性，云安全联盟(CSA)发布了一份关于云漏洞事件的分析报告。文献[2]的调查显示，数据丢失和泄漏事件占所有事件的 25%，仅次于“不安全接口和 api”。以亚马逊的云计算崩溃灾难为例。2011 年，亚马逊巨大的 EC2 云服务崩溃，永久摧毁了一些云用户的数据。相对于存储的总数据而言，数据丢失显然很小，但任何运营网站的人都能立即理解任何数据丢失有多可怕。有时，在访问数据时检测数据损坏是不够的，因为恢复损坏的数据可能太晚了。因此，云用户有必要经常检查他们的外包数据是否被存储正确。

云数据的大小是巨大的，下载整个文件来检查完整性可能在带宽成本方面令人望而却步，因此，非常不切实际。此外，用于数据完整性检查的传统密码原语，如哈希函数、授权代码(MAC)不能直接适用于这里，因为在验证中缺少原始文件的副本。总之，远程对安全云存储的数据完整性检查是一个非常理想的研究课题，也是一个具有挑战性的研究课题。

Blum 首次提出了一个审计问题，使数据所有者能够在不明确了解整个数据的情况下验证远程数据的完整性。近年来，由于分布式存储系统和在线存储系统的发展，远程数据完整性检查变得越来越重要。由 Ateniese 等人引入的对不可信存储的可证明的数据占有(PDP)，是一种在远程服务器上“无块验证”数据完整性的新技术。在 PDP 中，数据所有者为一个文件生成一些元数据，然后将他的数据文件与元数据一起发送到远程服务器，并从其本地存储中删除该文件。为了生成服务器正确存储原始文件的证明，服务器将计算对来自验证者的挑战的响应。验证者可以通过检查响应的正确性来验证该文件是否保持不变。PDP 采用了抽查技术，是一种检测云数据完整性的实用方法。具体来说，文件被分成块，验证者只挑战一组随机选择的块进行完整性检查。根据 Ateniese 等人给出的例子，对于一个有 10,000 个块的文件，如果服务器删除了 1%的块，那么验证者可以通过询问仅对 460 个随机选择的块的占有证明来检测服务器的概率大于 99%的错误行为。Ateniese 等人利用基于 rsa 的同态线性身份验证器，提出了两种具体的 PDP 结构。由于远程数据完整性检查的必然性和实用性，近年来在[7]-[10]、[12]-[22]、[26]等方面引起了广泛的研究兴趣。 Shacham 和 Waters[7]利用来自 BLS 签名[37]的公开可验证同态身份验证器，提出了可检索性紧证明的概念。该方案还依赖于同态性质，将一个证明聚合为一个小的身份验证器值，从而可以实现公共的可检索性。

Ateniese 等人。[23]首次考虑了基于哈希函数和对称密钥加密的动态 PDP 方案，这意味着数据所有者在将数据存储在云服务器上后，可以动态地更新其文件。动态操作包括数据的

插入、修改、删除和添加。该方案[23]是有效的，但只有有限数量的查询，并且不能显式地支持块插入。Erway 等人[25]利用基于秩的身份验证跳过列表，将 PDP 模型扩展到动态 PDP 模型。Wang 等人[15]通过操作 Merkle 哈希树(MHT)进行块标记身份验证，改进了以前的 PDP 模型。Liu 等人[26]最近的一项研究表明，MHT 本身不足以验证块索引，这可能会导致替代攻击。他们给出了自上而下的基于 MHT 的数据审计方案，用于云上的动态大数据存储。

在具有公共可验证性的数据完整性检查中，外部审计员（或任何人）能够验证云数据的完整性。在这种情况下，针对第三方验证者的数据隐私非常重要，因为云用户可能会存储机密或敏感文件，如业务合同或医疗记录到云中。然而，这个问题尚未得到充分的研究。在之前的隐私保护公共审计方案[13]中，“隐私”的定义要求验证者不能从云服务器生成的响应中恢复整个块。但是，这个定义还不够强，例如，它很容易受到字典攻击。Wang 等人[16]提出了“零知识公共审计”的概念来抵制离线猜测攻击。但是，在本工作中没有提供一个正式的安全模型。Yu 等人[27]最近增强了远程数据完整性检查协议的隐私性，但他们的模型仅在基于公钥基础设施(PKI)的场景中工作，而不是基于身份的框架。在外包之前对文件进行加密可以部分解决数据隐私问题，但会导致失去协议的灵活性，因为保护隐私的 RDIC 协议可以用作其他原语的构建块。例如，Ateniese 等人[28]提出了一个框架，用于在有界检索模型中构建泄漏弹性识别协议，从计算上为零知识的公开可验证的存储证明，但在识别方案中，即使是加密的文件也必须保持隐私。

目前，大多数现有的 RDIC 结构都依赖于 PKI，其中使用数字证书来保证用户的公钥的真实性。由于证书生成、证书存储、证书更新和证书撤销耗时且昂贵，这些结构需要复杂的密钥管理程序。有各种各样的标准，比如 IntertX.509PKI 认证政策和认证实践框架(RFC2527)，它们涵盖了 PKI 的各个方面。然而，它缺乏执行这些标准的主要管理机构。尽管证书颁发机构(CA)通常被认为是可信任的，但各种 CAs 的安全程序中的缺陷已经损害了对互联网所依赖的整个 PKI 的信任。例如，在发现 500 多份假证书后，网络浏览器供应商被迫在 2011 年将荷兰 CA 公司 DigiNotar 颁发的所有证书列入黑名单。使用证书来验证公钥的另一种方法是基于身份的密码学[29]，其中用户的公钥只是他的身份，比如他的名字、电子邮件或 IP 地址。受信任密钥分发中心(KGC)为每个用户生成对应的密钥。当所有用户都拥有由同一 KGC 颁发的密钥时，单个公钥就会过时，因此不需要显式认证和所有相关成本。这些特性使得基于标识的范式与面向组织的 PDP 结合使用特别有吸引力。例如，一所大学为员工和学生购买云存储服务，他们有一个由大学的 IT 部门发布的有效的电子邮件地址。这个大学的所有成员都可以有一个由 KGC，即 IT 部门，提供的密钥。该大学的成员可以将他们的数据和文件的元数据一起存储到云端。为了确保数据的正确存储，IT 部门的工作人员只能通过其电子邮件地址检查任何成员的完整性，这可以缓解 PKI 造成的复杂密钥管理。第一个基于 id 的 PDP 在[30]中提出了，它将 Gentry 和 Ramzan[31]提供的基于 id 的聚合签名转换为基于 id 的 PDP 协议。Wang[32]提出了另一种基于身份的多云存储中可证明数据占有方法。然而，他们的安全模型称为基于身份的 PDP 的不可伪造，不足以捕获稳健性的属性，因为在这个模型中不允许 TagGen 查询，这表明对手无法访问这些块的标签。这显然与真正的云存储不一致，即云服务器实际上存储所有数据块的标签。此外，[32]中具体的基于身份识别的 PDP 协议未能实现可靠性，这是 PDP 方案的一个基本安全要求。原因是，每个块的散列值用于生成块的标记，因此，恶意云服务器可以只保留块的散列值，用于生成对挑战的有效响应。

我们的贡献：本文的贡献总结如下。

1、在基于身份的签名方案中，任何能够访问签名者身份的人都可以验证签名者的签名。类似地，在基于身份的 RDIC 协议中，任何知道云用户身份的人，比如第三方审计员(TPA)，都能够代表云用户检查数据的完整性。因此，在基于身份的 RDIC 中，公共可验证性比私有

验证更可取，特别是对于资源受限的云用户。在这种情况下，零知识隐私的特性对于基于 id 的 RDIC 协议中的数据机密性非常重要。我们的第一个贡献是首次在基于身份的 RDIC 协议中形式化了针对 TPA 的零知识隐私的安全模型。

2、我们填补了迄今为止没有一个安全和新颖的基于身份的 RDIC 方案的空白。具体地说，我们提出了一个具体的基于身份的 RDIC 协议，这是一种新的结构，不同于以前的结构，利用一个新的原语称为非对称群密钥协议[33]，[34]。更具体地说，我们的挑战-响应协议是 TPA 和云服务器之间的两方密钥协议，在生成共享密钥时必须使用被挑战的块，这是云服务器对来自 TPA 的挑战的响应。

3、我们提供了新协议的详细安全证明，包括存储数据的可靠性和零知识隐私。我们的安全性证明是在一般群模型[35]中进行的。这是基于 id 的 RDIC 协议的第一个正确的安全证明。因此，新的安全证明方法本身可能是独立的利益所在

4、我们通过开发一个协议的原型实现来展示该方案的实用性。

2 预备知识

在本节中，我们回顾了本文中使用的一些预备知识，包括双线性配对和零知识证明。

A 双线性配对

一个双线性配对 [29] 将一对群元素映射到另一个群元素。 具体来说，令 G_1, G_2 是两个 p阶循环群。$g_1$and g_2 分别表示 G_1 和G_2 的生成元。一个函数 $e: G_1 \times G_1 \to G_2$ 被称为双线性配对，如果他有以下属性：

双线性： 对所有 $u, v \in G_1$和$x, y \in Z_p$， $e(u^x, v^y) = e(u, v)^{xy}$ 成立。

非退化： $e(g, g) \neq 1_{G_2}$ 其中1_{G_2} 是 G_2的单位元。

高效计算：对 $u, v \in G_1$， $e(u, v)$可以在多项式时间内被高效计算出来。

B 离散对数等式

令G 是一个有限循环群且有$|G| = q$ ，q是素数， g_1, g_2 是G的生成元。下列协议能使一个证明者P 向一个验证者 V 证明两个元素 Y_1, Y_2 分别以g_1 和g_2具有相同的离散对数。

承诺： P 随机选择 $\rho \in Z_q$，计算$T_1 = g_1^{\rho}, T_2 = g_2^{\rho}$ 并发送 T_1, T_2 给 V。

挑战：V 随机选择一个挑战 $c \in \{0,1\}^{\lambda}$ 并把 c 发送回 P。

响应： P 计算 $z = \rho - cx (\mathrm{mod}\, q)$ 并返回 z 给 V。

验证： V 接受证据当且仅当 $T_1 = g_1^z Y_1^c \wedge T_2 = g_2^z Y_2^c$ 成立。

这个协议可以被转换成一个更有效率的非交互版本，表示为$POK\{(x): Y_1 = g_1^x \wedge Y_2 = g_2^x\}$，通过用承诺的哈希来代替挑战，即 $c = H(T_1 \| T_2)$, 其中 H 是一个安全哈希函数。

C 基于身份的签名

一个基于身份的签名(IDS)方案[39]，[40]由四个多项式时间概率算法组成，描述如下

Setup (k) : 该算法输入安全参数 k 并输出主私钥msk 和主公钥 mpk。

Extract(msk, ID): 输入用户的身份ID, 主私钥 msk ，生成一个用户私钥usk。

Sign (ID, usk, m) : 输入用户的身份 ID, 消息 m 和用户私钥usk，生成对消息m 的签名σ。

Verify (ID, m, σ, mpk) : 输入签名 σ, 消息m, 身份 ID 和主公钥 mpk, 输出签名是否有效。

3 系统模型和安全模型

在本节中，我们将描述基于身份的 RDIC 协议的系统模型和安全模型。

A 基于身份的 RDIC 系统

通常，数据所有者自己可以通过运行两方 RDIC 协议来检查其云数据的完整性。然而，在双方场景中，来自数据所有者或云服务器的审计结果可能被认为是有偏见的。具有公共可验证性的 RDIC 协议使任何人都能够审计外包数据的完整性。为了清楚地描述可公开验证的 RDIC 协议，我们假设有一个具有验证工作能力的第三方审计师(TPA)。考虑到这一点，基于 id 的 RDIC 体系结构如图 1 所示。

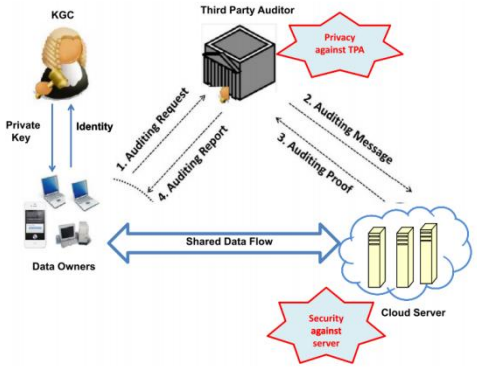

Fig. 1. The system model of identity-based RDIC.

系统涉及 KGC、云用户、云服务器和 TPA 四个不同的实体。KGC 会根据所有用户的身份为他们生成密钥。云用户有大量的文件需要存储在云上，云服务器有大量的存储空间和计算资源，为云用户提供数据存储服务。TPA 具有云用户所不具备的专业知识和能力，并且它可以被信任来代表云用户根据要求检查云数据的完整性。每个实体都各自有自己的义务和利益。云服务器可能是自私的，为了他自己的利益，比如保持良好的声誉，云服务器甚至可能决定向云用户隐藏数据损坏事件。然而，由于法规和财务激励，我们假设云服务器没有动机向 TPA 显示托管数据。TPA 的工作是代表云用户执行数据完整性检查，但 TPA 也很好奇，因为他愿意在数据完整性检查过程中了解用户数据的一些信息。

B．系统组成和它的安全性

一个基于身份的 RDIC 包括六个算法，Setup, Extract, TagGen, Challenge, ProofGen 和 ProofCheck

- Setup (1^k) 是一个由 KGC 运行的概率算法。它输入一个安全参数 k，输出系统参数 param 和主密钥 msk。
- Extract(param, msk, ID) 是一个由 KGC 运行的概率算法。它输入系统参数 param，主密钥msk ，和一个用户的身份$ID \in \{0,1\}^*$ ， 输出身份ID对应的私钥sk_{ID} 。
- TagGen(param, F, sk_{ID}) 是一个由身份为ID的数据拥有者运行的概率算法。输入系统参数 param，用户的私钥sk_{ID}和一个要存储的文件$F \in \{0,1\}^*$ ，输出每个文件块m_i的标签集合 $\sigma = (\sigma_1, \cdots, \sigma_n)$ ，它会和文件一起保存在云上。

- Challenge (param, Fn, ID) 是一个由 TPA 运行的随机算法。输入系统参数 param，数据拥有者的身份 ID，和一个唯一的Fn ，输出代表身份ID对文件名的挑战$chal$。

- ProofGen(param, ID, $chal$, F,σ) 是一个由云服务器运行的概率算法。输入系统参数 param，挑战信息$chal$，数据拥有者的身份ID，标签 σ，文件 F 和文件名 Fn ，输出一个被挑战块的数据拥有证据P。

- ProofCheck (param, ID, $chal$, P,Fn) 是一个由 TPA 运行的确定性算法。输入系统参数 param，挑战信息 $chal$，数据拥有者的身份ID，文件名 Fn 和一个声称的数据拥有证据 P ，输出 1 或 0 来表明文件是否完整。

在基于身份的远程数据完整性检查协议中，我们考虑了三个安全属性，即完整性、针对恶意服务器的安全性（可靠性）和针对 TPA 的隐私（完美的数据隐私）。根据 Shacham 和 Waters 提出的安全概念，基于身份的 RDIC 方案对服务器称为安全方案，如果不存在多项式时间算法，可以以不可忽略的概率欺骗 TPA 并且存在一个多项式时间提取器，它可以通过多次运行挑战响应协议来恢复文件。完整性表明，当与有效的云服务器交互时，验证检查算法将接受证明。可靠性说明，一个可以说服 TPA 它正在存储数据文件的作弊证明者，实际上是在存储那个文件。我们现在在下面形式化了基于身份的远程数据完整性检查的可靠性安全模型，其中涉及到扮演不受信任服务器角色的敌手和代表数据所有者的挑战者。
对于服务器的安全性：这个安全游戏捕获到，如果一个敌手无法与给定挑战信息对应的用户ID的所有块，就无法成功生成有效的完整性证明，除非它猜测所有被挑战的块。游戏由一下阶段组成[36]：

- Setup 阶段: 挑战者运行 Setup 算法获得系统参数 param 和主密钥msk, 并把 param 传给敌手，而msk自己保密。

- Queries 阶段: 敌手向挑战者做一系列询问，包括自适应的提取询问和标签询问。

  1 提取询问：敌手可以询问任意身份 ID_i的私钥。挑战者通过 Extract 算法计算私钥sk_i 并发送给敌手。

  2 标签询问： 敌手可以在身份ID_i下询问任意文件F 的标签。挑战者运行 Extract 算法获得私钥sk_i，并运行 TagGen 算法生成F 的标签。最后，挑战者返回标签的集合给敌手。

- ProofGen 阶段: 对于一个已经被做过标签询问的文件F ，敌手可以通过指定数据所有者的身份 ID 和文件名Fn来执行 ProofGen 算法。证明过程中挑战者扮演 TPA 的角色而敌手$\mathcal{A}$ 扮演证明者。最后，当协议执行完成时，敌手可以从挑战者那里获得 P 的输出。

- Output 阶段:：最后，敌手选择一个文件名$Fn^{\dagger}$ 和一个用户身份 $ID^{\dagger}$。$ID^{\dagger}$必须没有被做过私钥询问，并且存在一个使用 $F^{\dagger}$ 和$ID^{\dagger}$作为输入的标签询问。敌手输出对一个ϵ 可接受的证明者$P^{\dagger}$ 的描述，其是如下定义的。

我们说一个仿冒的证明者$P^{\dagger}$是ϵ 可接受的，如果他令人信服地回答了完整性挑战的ϵ部分。即，$\Pr\left[\left(\mathcal{V}\left((param, ID^{\dagger}, Fn^{\dagger}) \rightleftharpoons \mathcal{P}^{\dagger}\right) = 1\right] \geq \epsilon\right.$ 。这里的概率超过验证者和证明者两者的硬币。敌手赢得游戏，如果如果他能成功输出一个ϵ 可接受的证明者$P^{\dagger}$。

一个 ID-RDIC 方案被称为$\epsilon-$可靠的，如果存在一个提取算法 Extr，对于每一个敌手$\mathcal{A}$，每当$\mathcal{A}$进行可靠性安全游戏，都会基于身份$ID^{\dagger}$和文件名$Fn^{\dagger}$输出一个ϵ 可接受的证明者$P^{\dagger}$，Extr 算法从$P^{\dagger}$中恢复出$F^{\dagger}$，例如 $Extr(param, ID^{\dagger}, Fn^{\dagger}, P^{\dagger}) = F$，除非可能性可以忽略不计。

针对 TPA 的完美的数据隐私：“完美数据隐私”是指 TPA 没有获得外包数据的任何信息，也就是说，无论 TPA 学习到什么，TPA 都可以自己学习，而无需与云服务器进行任何交互。我们使用模拟器将此模型形式化，如下所示

定义 1：一个基于身份的 RDIC 协议实现完美的数据隐私，如果每一个假冒的验证者TPA^{*}，存在一个多项式时间非交互式模拟器S 对TPA^{*}，对于每一个有效的公共输入ID, chal, Tag 和私人输入F，以下两个随机变量计算不可区分：

1 $view_{TPA^{*}}(\text{Server}_{R}, \text{chal}, \text{F}, ID, Tag, TPA^{*})$，其中$R$ 表示该协议使用的随机硬币。

2 $S(\text{chal}, ID)$。

也就是说，模拟器S只获取公共输入的信息，并且与服务器没有交互，但仍然设法输出一个与TPA^{*}在交互过程中学习到的任何无法区分的响应。

4 方案构造

在本节中，我们提供了一个基于身份的安全远程数据完整性检查协议的具体构建，支持完美的数据隐私保护。我们的方案的工作原理如下。在密钥提取中，我们采用 Boneh 等人[37]提出的短签名算法对用户的身份$ID \in \{0,1\}^*$进行签名，获得用户的密钥。在 TagGen 中，我们发明了一种新的算法来生成文件块的标记，当计算对挑战的响应时，这些标记可以聚合为单个元素。在挑战阶段，TPA 通过选择块的一些索引和随机值来挑战云。在证明生成过程中，云服务器使用被挑战的块计算响应，获得相应的明文并将其转发给 TPA。协议的细节如下：

Setup：输入一个安全参数sp，KGC 选择两个素数q阶的乘法循环群G_1和G_2，g是G_1的生成元。存在一个双线性映射$e: G_1 \times G_1 \to G_2$。KGC 随机选择$\alpha \in Z_q^*$作为主密钥，并设置$P_{pub} = g^{\alpha}$。最后，KGC 选择三个哈希函数$H_1, H_2$ ：$\{0,1\}^* \to G_1$和$H_3: G_2 \to \{0,1\}^l$。系统参数为$\left(G_1, G_2, e, g, P_{pub}, H_1, H_2, H_3, l\right)$，

Extract：输入主密钥α 和一个用户身份 $ID \in \{0,1\}^*$，这个算法输出用户的私钥 $s = H_1(ID)^{\alpha}$。

TagGen：给定一个文件M ，文件名为 fname，数据拥有者首先将它分成n 个文件块$m_1, \cdots, m_n$，其中$m_i \in Z_q$ 并选择一个随机数 $\eta \in Z_q^*$ ，计算 $r = g^{\eta}$。对于每一个块m_i，数据拥有者计算

$$\sigma_i = s^{m_i} H_2(\text{ fname } \| \, i)^{\eta}.$$

文件块 m_i的标签就是 σ_i。数据拥有者把文件 M 和$\left(r, \{\sigma_i\}, \text{IDS } (r \| f \text{ name })\right)$一起存储到云中，其中 IDS $(r \| f$ name)是一个由数据拥有者对$r \| f$ name 做的基于身份的签名。

Challenge：为了检查 M 的完整性，验证者随机选择一个集合$[1, n]$的 c 个元素的子集I，对于每个$i \in I$ ，选择一个随机元素$v_i \in Z_q^*$。令 Q 为$\{(i, v_i)\}$的集合。为了生成挑战信息，验证者选择一个随机的 $\rho \in Z_q^*$，计算 $Z = e\left(H_1(ID), P_{pub}\right)$ 并如下操作：

1 计算$c_1 = g^{\rho}, c_2 = Z^{\rho}$。

2 生成一个证据：

$$pf = POK\{(\rho): c_1 = g^{\rho} \wedge c_2 = Z^{\rho}\}$$

验证者发送挑战信息 $chal = (c_1, c_2, Q, pf)$ 给服务器。

GenProof：一旦收到挑战信息$chal = (c_1, c_2, Q, pf)$，服务器首先计算出$Z = e\left(H_1(ID), P_{pub}\right)$。然后，它验证完整性证据$pf$。如果它是无效的，审计终止。否则，服务器计算$\mu = \sum_{i \in I} v_i m_i, \sigma = \prod_{i \in I} \sigma_i^{v_i}$ and $m^{'} = H_3\left(e(\sigma, c_1) \cdot c_2^{-\mu}\right)$和$m^{'} = H_3\left(e(\sigma, c_1) \cdot c_2^{-\mu}\right)$并把$\left(m^{'}, r, \text{IDS } (r \| f\text{name})\right)$返回给验证者。

CheckProof：一旦收到服务器发来的$m^{'}$ ，验证者检查是否 IDS $(r \| f$ name)是否是数据拥有这对消息 $r \| f$name 的有效签名。如果不是，完整性证据是无效的。否则，验证者检查是否

$$m^{'} = H_3\left(\prod_{i\in I} e(H_2(fname \parallel i)^{v_i}, r^{\rho})\right)$$

如果等式成立，验证者接受证据；否则证据是无效的。Challenge, CheckProof 和 GenProof 的过程总结在 Fig. 2 中。

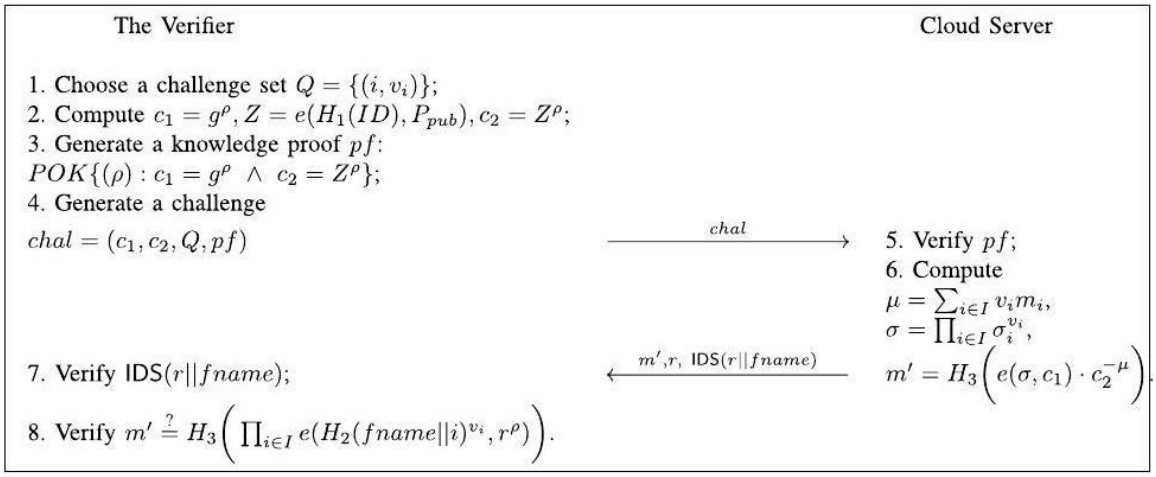

Fig. 2. Identity-based remote data integrity checking protocol.

5 安全性分析

在本节中，我们证明了该方案实现了完整性、可靠性和完美的数据隐私保护的特性。完整性保证了协议的正确性，而可靠性表明该协议对不受信任的服务器是安全的。完美的数据隐私表明，该协议不会向验证者泄露所存储文件的信息。

A 完整性

如果数据所有者和云服务器都是诚实的，那么对于每个有效的标签σ_i和一个随机的挑战，云服务器总是可以通过验证。该协议的完整性可以详细阐述如下。

$$
\begin{aligned}
m' &= H_3\left(e(\sigma, c_1)\cdot c_2^{-\mu}\right) \\
&= H_3\left(\frac{e(\sigma, c_1)}{e\left(H(ID), P_{pub}\right)^{\rho\sum_{i\in I} m_i v_i}}\right) \\
&= H_3\left(\frac{e\left(\prod_{i\in I}\sigma_i^{v_i}, c_1\right)}{e\left(s, g^{\rho\sum_{i\in I} m_i v_i}\right)}\right) \\
&= H_3\left(\frac{\prod_{i\in I} e\left(\sigma_i^{v_i}, c_1\right)}{\prod_{i\in I} e(s, c_1)^{m_i v_i}}\right) \\
&= H_3\left(\prod_{i\in I} e\left(\frac{\sigma_i}{s^{m_i}}, g^{\rho v_i}\right)\right) \\
&= H_3\left(\prod_{i\in I} e(H_2(fname \parallel i)^{\eta}, g^{\rho v_i})\right) \\
&= H_3\left(\prod_{i\in I} e(H_2(fname \parallel i)^{v_i}, g^{\rho\eta})\right) \\
&= H_3\left(\prod_{i\in I} e(H_2(fname \parallel i)^{v_i}, r^{\rho})\right)
\end{aligned}
$$

B 可靠性

先前协议的响应通常包括$\mu = \sum m_i v_i$项，这有助于构建一个提取器来提取被挑战的块m_i。例如，在 Ateniese 等人[4]的协议中，他们利用指数假设的知识构建了这样的提取器。然而，在我们的新协议中没有这部分。在这一部分中，我们证明了任何一般算法在打破我们的协议的可靠性方面的概率都是可以忽略不计的。一般算法是一种不利用底层有限循环群的代数结构的算法。一般群模型旨在捕获任何一般算法的行为。我们简要回顾一下这个想法。

设p是一个素数，而$\xi_i: \mathbb{Z}_p \to \{0,1\}^{\lceil \log_2 p \rceil}$对于$i \in \{1,2\}$是两个独立的随机编码函数。一般群$\mathbb{G}_i$表示为$\mathbb{G}_i = \{\xi_i(x) | x \in \mathbb{Z}_p\}$。由于一般算法不利用群的结构，我们说给定一个元素$\xi_i(x) \in \mathbb{G}_i$，除了相等之外，不能推断出任何关于该元素的东西。两个喻言机，$\mathcal{O}_i, i \in \{1,2\}$用于模拟群组的动作：对于任何元素$\xi_i(a)$和$\xi_i(b)$，输入$\xi_i(a), \xi_i(b)$查询$\mathcal{O}_i$，返回元素$\xi_i(a+b)$（用于乘法）或$\xi_i(a-b)$（用于除法）。另一个喻言机，$\mathcal{O}_E$，用于表示$\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$的配对操作。具体来说，输入$\xi_1(a)$和$\xi_1(b)$，$\mathcal{O}_E$返回$\xi_2(a*b)$。

证明：在讨论了设置之后，我们将证明我们的构造对于随机模型中的任何通用算法都是安全的。假设$\mathcal{A}$是一个通用对手，分别对H_1, H_2, H_3做q_1, q_2, q_3次查询，我们展示如何构造一个模拟器$\mathcal{S}$来模拟一般群模型中$\mathcal{A}$的视图，并从$\mathcal{A}$中提取消息的集合。

- 初始化。在一般群模型和随机喻言模型中，公开参数可以表示为

  $$\ell, q, \mathcal{O}_E, \mathcal{O}_1, \mathcal{O}_2, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \xi_1, \xi_\alpha,$$

  其中，前三个喻言机表示双线性配对运算和群运算，后三个喻言机表示哈希运算，ξ_1, ξ_α分别表示元素g和P_{pub}的编码。它们是由$\mathcal{S}$选择的随机比特串。$\mathcal{S}$将编码ξ_1与常数多项式 1 关联，将ξ_α与多元多项式α关联。我们可以看到，在任何阶段，任意群元素都将与一个多元多项式相关联。具体来说，呈现给$\mathcal{A}$的元素都将与变量$\left(\alpha, \{I_i\}_{i=1}^{q_I}, \{f_i\}_{i=1}^{n*q_f}, \{\eta_i\}_{i=1}^{q_t}, \rho\right)$中的一个多元多项式相关联。我们将讨论多项式关联的规则。
- 哈希询问：对于任意输入ID_k对$\mathcal{H}_1$的询问，$\mathcal{S}$返回一个随机比特串ξ_{I_k}并将其与一个多元多项式I_k关联。对于任意输入$_j \| i$对$\mathcal{H}_2$的询问，$\mathcal{S}$返回一个随机比特串$\xi_{f_{j,i}}$并将其与一个多元多项式 $f_{j,i}$关联。对于任意对$\mathcal{H}_3$的询问，$\mathcal{S}$返回一个随机比特串（不是一个群元素）。

- 群$\mathbb{G}_1$的运算（喻言机$\mathcal{O}_1$）。$\mathcal{A}$提交给$\mathcal{S}$两个元素ξ_{F_1}, ξ_{F_2}和两个整数e_1, e_2。$\mathcal{S}$搜索它的历史记录来定位和ξ_{F_1}, ξ_{F_2}相关的多项式（假设它们分别是F_1 和F_2）。如果这些元素不存在，则意味着ξ_{F_1}或ξ_{F_2}不是有效的群元素，$\mathcal{S}$拒绝该运算。$\mathcal{S}$计算$e_1 F_1 + e_2 F_2$，得到的多项式记为F_3。$\mathcal{S}$搜索它的历史记录来检查是否$\mathbb{G}_1$中是否存在与F_3关联的元素ξ。如果是，则返回ξ。否则，$\mathcal{S}$选择一个随机比特串ξ_{F_3}并将它与F_3关联。$\mathcal{S}$返回ξ_{F_3}给$\mathcal{A}$。注意元素ξ_{F_3}表示群运算ξ_{F_1}的e_1次方乘ξ_{F_2}的e_2次方的结果。

- 群$\mathbb{G}_2$的运算（喻言机$\mathcal{O}_2$）。其处理方式与 O1 查询的处理方式相同。

- 双线性配对运算（喻言机$\mathcal{O}_E$）。$\mathcal{A}$提交给$\mathcal{S}$两个元素ξ_{F_1}, ξ_{F_2}。$\mathcal{S}$搜索它的历史记录来定位和ξ_{F_1}, ξ_{F_2}相关的多项式（假设它们分别是F_1 和F_2）。如果这些元素不存在，则意味着ξ_{F_1}或ξ_{F_2}不是有效的群元素，$\mathcal{S}$拒绝该运算。$\mathcal{S}$计算 $F_1 \cdot F_2$，得到的多项式记为F_3。$\mathcal{S}$搜索它的历史记录来检查是否$\mathbb{G}_2$中是否存在与F_3关联的元素ξ。如果是，则返回ξ。否则，$\mathcal{S}$选择一个随机比特串ξ_{F_3}并将它与F_3关联。$\mathcal{S}$返回ξ_{F_3}给$\mathcal{A}$。注意元素ξ_{F_3}表示元素ξ_{F_1}和ξ_{F_3}的双线性配对的结果。

- 提取询问。$\mathcal{A}$提交给$\mathcal{S}$一个身份ID。$\mathcal{S}$定位一个具有相同输入的$\mathcal{H}_1$询问。如果没有，

它将隐式地进行一次$\mathcal{H}_1$询问。它得到表示$H(ID)$的元素，记为ξ_{I_k}并把它和多项式I_k关联。如果$\mathbb{G}_1$中存在一个元素ξ，其已经和多项式αI_k关联，$\mathcal{S}$返回ξ。否则，$\mathcal{S}$随机选择一个随机比特串$\xi_{\alpha I_k}$将它与多项式αI_k关联并将它返回给$\mathcal{A}$。

- 标签询问。$\mathcal{A}$提交给$\mathcal{S}$一个身份ID和文件$M=(m_1,\ldots,m_n)$，文件名为 $fname_j$。$\mathcal{S}$用ID定位一个提取询问。如果没有，它将隐式地用ID作为输入进行一次提取询问。假设元素$H(ID)$与多项式I_k关联。$\mathcal{S}$还会获得一个元素的集合 $\{H_2(fname_j \| i)\}$。（如果不存在，$\mathcal{S}$进行$\mathcal{H}_2$询问 ）。假设元素$H_2(fname_j \| i)$跟多项式$f_{j,i}$相关联，$\mathcal{S}$选择一个随机比特串ξ_{η_j}并将它和多项式η_j关联。对于$i=1$到n，$\mathcal{S}$计算多项式$F_{j,i}=\alpha I_k m_i + f_{j,i}\eta_j$。如果一个元素$\xi_{j,i}$已经和多项式$F_{j,i}$关联，$\mathcal{S}$返回$\xi_{j,i}$作为文件名 $fname_j$ 的文件块m_i的标签$\sigma_{j,i}$。否则，$\mathcal{S}$选择一个随机比特串$\xi_{j,i}$并将它和$F_{j,i}$关联，并返回$\xi_{j,i}$作为文件块m_i的标签$\sigma_{j,i}$。比特串ξ_{η_j}在这个阶段也交给敌手$\mathcal{A}$。注意，$\left(\xi_{\eta_j},\xi_{j,1},\ldots,\xi_{j,n}\right)$表示$(r,T)$对于一个文件$M$，文件名为 $fname_j$。一个对$r \| fname_j$基于身份的签名 IDS $(r \| fname_j)$ 也被交给敌手$\mathcal{A}$。

在某个阶段，IDA 和某个文件决定在一个 M=(m1，...，mn)上被挑战，但 A 从未对 I 进行过提取查询 D. 我们可以安全地假设 A 已经对 I 进行了哈希查询 D. 我们假设元素 H1(ID)与多项式 I 相关联。s 使用 ID 和 M 作为输入，使用 a 进行 TagGen 查询。假设返回到 A 的元素为(ξ η j， ξ j，1，...， ξ j，n)。为了清晰起见，下面我们将去掉下标 j。现在，S 扮演着挑战者的角色，而 a 在协议中扮演着云服务器的角色。

在某个阶段，$\mathcal{A}$决定被一个确定的身份ID和一个确定的名为$fname$的文件$M=(m_1,\ldots,m_n)$挑战，限制为$\mathcal{A}$没有对ID做过提取询问。我们可以安全地假设$\mathcal{A}$对ID做过哈希询问。我们假设元素 t $H_1(ID)$与多项式I关联。$\mathcal{S}$实施一次标签查询以$\mathcal{A}$使用ID和M作为输入。假设返回给$\mathcal{A}$的元素是$\left(\xi_{\eta_j},\xi_{j,1},\ldots,\xi_{j,n}\right)$。为了清晰起见，下面我们将去掉下标$j$。在，$\mathcal{S}$扮演着挑战者的角色，而$\mathcal{A}$在协议中扮演着云服务器的角色。

- （挑战）。$\mathcal{S}$选择一个子集$\mathcal{I}\subset[1,n]$，且对于每一个$i\in\mathcal{I}$，$\mathcal{S}$选择v_i。$\mathcal{S}$进一步选择一个随机比特串ξ_{c_1}并将它与多项式ρ关联。它还会选择另一个随机比特串ξ_Z并将它与多项式αI关联。ξ_{c_1}和ξ_Z分别表示群$\mathbb{G}_1$元素c_1和群$\mathbb{G}_2$元素Z。$\mathcal{S}$还会选择一个随机比特串ξ_{c_2}并将它与多项式$\alpha I\rho$关联。ξ_{c_1},ξ_{c_2}和一个模拟证据pf一起交给$\mathcal{A}$。

- （多项式计算）。回想一下，I是与$H_1(ID)$关联的多项式。令η是与元素ξ_η关联的多项式。另外，令f_i是与元素$H_2(fname_j \| i)$关联的多项式。因此，m_i的标签会与多项式$F_i \equiv m_i\alpha I + f_i\eta$相关联。在这个阶段，$\mathcal{S}$随机选择除以下变量之外的所有变量的值：

$$\alpha, I, \{f_i\}_{i\in\mathcal{I}}, \eta, \rho.$$

  换句话说，所有给$\mathcal{A}$的多项式都是$c+4$变量中的多元多项式。实例化不会影响前一个模拟的视图，只要它不会创建任何“不正确”的关系。

- （生成证据）。最后$\mathcal{A}$返回一个值m'，和一个群元素r以及它的基于身份的签名 IDS $(r \| fname_j)$。

在随机喻言模型中，$\mathcal{A}$成功返回$m^{'}=H_3\left(\prod_{i\in I}e(H_2(\text{ fname }\|\ i)^{v_i},r^{\rho})\right)$的唯一方式就是做一个$\mathbb{G}_2$中元素$\xi$的$H_3$查询。由于基于身份的签名的不可伪造性，我们可以安全地假设r（代表验证过程中使用的元素ξ_η）是在标签询问中返回给$\mathcal{A}$的值。

为了使$\mathcal{A}$以不可忽略的概率获胜，ξ必须与一个多项式F关联起来：

$$F\equiv\sum_{i\in\mathcal{J}}v_i f_i\rho\eta$$

仍然需要证明，为了生成一个关联多项式满足上述等价关系的元素ξ，$\mathcal{A}$必须进行一组群运算查询，从而允许$\mathcal{S}$提取常数集$\{m_i\}$。此外，这些询问是在挑战阶段之后进行的。首先，请注意，对于每个询问，$\mathcal{A}$最多获得一个新的群元素，因此由$\mathcal{A}$获得的元素总数是有限的。在挑战阶段之前，$\mathcal{A}$没有这样的元素，其多项式与变量ρ或$I\alpha\rho$相关联。
在挑战阶段之前，每一个$\mathbb{G}_1$中的元素都与下列形式的多项式相关联：

$$A_0+A_1\alpha+A_2\eta+A_3I+\sum_{i\in\mathcal{J}}A_{4,i}f_i+\sum_{i\in\mathcal{J}}A_{5,i}(m_iI\alpha+f_i\eta)$$

其中A是对$\mathcal{A}$已知的系数。同样，群$\mathbb{G}_2$中的每个元素都与以下形式的多项式相关联：

$$\sum_{j=1}^{k}c_jF_j*F_j^{'}$$

其中F_j和$F_j^{'}$是与群$\mathbb{G}_1$中的元素相关联的多项式，且c_j是一个常数。很容易看出，$\mathcal{A}$在挑战前获得的多项式将不能满足所需的关系，因为缺少变量ρ。

在挑战阶段之后，$\mathcal{A}$可以访问两个额外的元素，它们的多项式分别是$\mathbb{G}_1$中的ρ和$\mathbb{G}_2$中的$I\alpha\rho$。由此可见，由$\mathcal{A}$得到的所有$\mathbb{G}_2$的元素都与以下形式的多项式相关联：

$$A_0 + A_1\rho + A_2\alpha + A_3\eta + A_4 I + \sum_{i\in\mathcal{J}} A_{5,i} f_i + \sum_{i\in\mathcal{J}} A_{6,i}(m_i I\alpha + f_i\eta) + A_7 I\alpha\rho + B_1\rho^2 + B_2\rho\alpha + B_3\rho\eta + B_4\rho I + \sum_{i\in\mathcal{J}} B_{5,i} f_i\rho + \sum_{i\in\mathcal{J}} B_{6,i}(m_i I\alpha\rho + f_i\eta\rho) + B_7 I\alpha\rho^2 + C_1\alpha^2 + C_2\alpha\eta + C_3\alpha I + \sum_{i\in\mathcal{J}} C_{4,i} f_i\alpha + \sum_{i\in\mathcal{J}} C_{5,i}(m_i I\alpha^2 + f_i\eta\alpha) + C_6 I\alpha^2\rho + D_1\eta^2 + D_2 I\eta + \sum_{i\in\mathcal{J}} D_{3,i} f_i\eta + \sum_{i\in\mathcal{J}} D_{4,i}(m_i I\alpha\eta + f_i\eta^2) + D_5 I\alpha\rho\eta + E_1 I^2 + \sum_{i\in\mathcal{J}} E_{2,i} f_i I + \sum_{i\in\mathcal{J}} E_{3,i}(m_i I^2\alpha + f_i I\eta) + E_4 I^2\alpha\rho + \sum_{i\in\mathcal{J}}\sum_{j\in\mathcal{J}} F_{1,i,j} f_i f_j + \sum_{i\in\mathcal{J}}\sum_{j\in\mathcal{J}} F_{2,i,j}\left(m_i I\alpha f_j + f_i f_j\eta\right) + \sum_{i\in\mathcal{J}} F_{3,i} I\alpha\rho f_i + \sum_{i\in\mathcal{J}}\sum_{j\in\mathcal{J}} G_{1,i,j}(m_1 I\alpha + f_i\eta)\left(m_j I\alpha + f_j\eta\right) + \sum_{i\in\mathcal{J}} G_{2,i}(m_i I^2\alpha^2\rho + f_i\eta I\alpha\rho) + H_1 I^2\alpha^2\rho^2$$

其中所有的系数对$\mathcal{A}$和$\mathcal{S}$都是已知的。将这些系数等于F，除了$B_{6,i}$ 和 A_7所有的系数都是 0。特别地，对于所有$i \in \mathcal{I}$，$B_{6,i} = v_i$且$A_7 = \sum_{i\in\mathcal{J}} m_i v_i$。

换句话说，$\mathcal{S}$可以从$\mathcal{A}$中提取项A_7，也就是$\sum_{i\in\mathcal{J}} m_i v_i$。使用$|\mathcal{I}|$交互作用，$\mathcal{S}$可以计算$\mathcal{A}$的所有$m_i$值。此外，观察到导致这个系数$A_7$的运算必须在挑战后发布，因为它们与多项式$I\alpha\rho$有关。这就完成了证明。

C 完美的数据隐私保护

为了证明该方案保持了数据隐私，我们展示了如何构建一个能够访问验证器V的模拟器$\mathcal{S}$，可以在不知道数据文件块$\{m_i\}$也不知道相应的$\{\sigma_i\}^3$的情况下模拟远程数据完整性检查协议。

我们假设 IDS$\left(r \parallel fname_j\right), r, fname$被发送给$\mathcal{S}$。换句话说，我们的协议既不保存文件名也不保存参数$r$的隐私。因为$r = e(g,g)^{\eta}$，其中$\eta$是一个由数据拥有者选择的随机值，我们可以合理地说$r$不包含文件库的信息。下面我们展示$\mathcal{S}$怎么回答由$V$提交的挑战。

一旦收到来自V的挑战信息$chal$，$\mathcal{S}$将$chal$解析为(c_1, c_2, Q, pf)。接下来，$\mathcal{S}$从 V 中提取值ρ。由于pf的可靠性，$\mathcal{S}$得到了ρ，$c_1 = g^{\rho}$和$c_2 = e(H_2(\text{ fname } \parallel i)^{v_i}, r^{\rho})$。$\mathcal{S}$将$Q$解析为$\{(\mathrm{i}, v_i)\}$，并计算$m^{'} = H_3\left(\prod_{i\in I} e(H_2(\text{ fname } \parallel i)^{v_i}, r^{\rho})\right)$。$\mathcal{S}$输出$(m^{'}, r, \text{IDS}(r \parallel fname))$作为对这一挑战

的响应。

对于每个挑战(c_1, c_2, Q, pf)，都有一个有效的唯一值m'。因此，上述模拟是完美的。

参考文献

[1] P. Mell and T. Grance. (Jun. 3, 2009). Draft NIST Working Definition of Cloud Computing. [Online]. Available: http://csrc.nist.gov/groups/SNC/cloud-computing/index.html

[2] Cloud Security Alliance. (2010). Top Threats to Cloud Computing. [Online]. Available: http://www.cloudsecurityalliance.org

[3] M. Blum, W. Evans, P. Gemmell, S. Kannan, and M. Naor, "Checking the correctness of memories," Algorithmica, vol. 12, no. 2 , pp. 225-244, Sep. 1994.

[4] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 598-609.

[5] G. Ateniese et al., "Remote data checking using provable data possession," ACM Trans. Inf. Syst. Security, vol. 14, no. 1, May 2011, art. no. 12.

[6] A. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, Pp. 584 − 597.

[7] H. Shacham and B. Waters, "Compact proofs of retrievability"' in $Proc$. Adv. Cryptol. -ASIACRYPT, 2008, pp. 90-107. [8] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in Proc. ASIACRYPT, 2009, pp. 319-333.

[9] A. F. Barsoum and M. A. Hasan, "Provable multicopy dynamic data possession in cloud computing systems," IEEE Trans. Inf. Forensics Security, vol. 10, no. 3, pp. 485-497, Mar. 2015 .

[10] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," IEEE Trans. Inf. Forensics Security, vol. 10, no. 6, pp. 1167-1179, Jun. 2015.

[11] J. Liu, K. Huang, H. Rong, H. Wang, and M. Xian, "Privacypreserving public auditing for regenerating-code-based cloud storage," IEEE Trans. Inf. Forensics Security, vol. 10, no. 7, pp. 1513-1528, Jul. 2015

[12] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing,' in Proc. Comput. Security-ESORICS, vol. 5789. 2009, pp. 355-370.

[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1-9.

[14] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage

services," IEEE Netw., vol. 24, no. 4, pp. 19-24, Jul./Aug. 2010

[15] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 22 , no. 5 , pp. $847-859$, May 2011.

[16] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage" IEEE Trans. Comput., vol. 62, no. 2, pp. 362-375, Feb. 2013 .

[17] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel Distrib. Syst. vol. 24 , no. 9 , pp. $1717-1726$, Sep. 2013 .

[18] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu, "Dynamic audit services for outsourced storages in clouds," IEEE Trans. Services Comput., vol. 6, no. 2, pp. 227-238, Apr./Jun. 2013.

[19] Y. Zhu, H. Hu, G.-J. Ahn, and S. S. Yau, "Efficient audit service outsourcing for data integrity in clouds," J. Syst. Softw., vol. 85, no. 5 , pp. $1083-1095$,2012

[20] H. Wang and Y. Zhang, "On the knowledge soundness of a cooperative provable data possession scheme in multicloud storage", IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. $264-267$, Jan. 2014.

[21] J. Wang, X. Chen, X. Huang, I. You, and Y. Xiang, "Verifiable auditing for outsourced database in cloud computing," IEEE Trans. Comput., vol. 64, no. 11 , pp. $3293-3303$, Nov. 2015

[22] Y. Yu, Y. Li, J. Ni, G. Yang, Y. Mu, and W. Susilo, "Comments on 'Public integrity auditing for dynamic data sharing with multiuser modification'," IEEE Trans. Inf. Forensics Security, vol. 11, no. 3 , pp. 658-659, Mar. 2016.

[23] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. SecureComm, 2008, art. no. 9.

[24] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage secruty in cloud computing," in Proc. $IWQoS$, vol. 2009. 2009, pp. 1-9.

[25] C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," ACM Trans. Inf. Syst. Security, vol. 17, no. 4, 2015 , art. no. 15.

[26] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "MuR-DPA: Top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud," IEEE Trans. Comput., vol. 64, no. 9, pp. $2609-2622$, Sep. 2015 .

[27] Y. Yu et al., "Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage," Int. J. Inf. Security, vol. 14, no. 4, pp. 307-318, 2015

[28] G. Ateniese, A. Faonio, and S. Kamara, "Leakage-resilient identification schemes from zero-knowledge proofs of storage", in Proc. IMA Int. Conf., 2015 , pp. 311 − 328

[29] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Proc. CRYPTO, vol. 2139. 2001, pp. 213-229.

[30] J. Zhao, C. Xu, F. Li, and W. Zhang, "Identity-based public verification with privacy-preserving for data storage security in cloud computing," IEICE Trans. Fundam. Electron., Commun. Comput. Sci., vol. E96-A, pp. 2709 − 2716, Dec. 2013.

[31] C. Gentry and Z. Ramzan, "Identity-based aggregate signatures," in Proc. Public Key Cryptography, vol. 3958. 2006, pp. 257-273.

[32] H. Wang, "Identity-based distributed provable data possession in multicloud storage," IEEE Trans. Service Comput., vol. 8, no. 2, pp. 328 − 340 , Mar./Apr. 2015 [33] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer, "Asymmetric group key agreement," in Proc. EUROCRYPT, vol. 5479. 2009, pp. 153 − 170.

[34] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and Z. Dong, "Roundefficient and sender-unrestricted dynamic group key agreement protocol for secure group communications," IEEE Trans. Inf. Forensics Security, vol. 10 , no. 11 , pp. 2352 − 2364, Nov. 2015

[35] V. Shoup, "Lower bounds for discrete logarithms and related problems," in Proc. EUROCRYPT, 1997, pp. 256-266.

[36] Y. Yu, Y. Zhang, Y. Mu, W. Susilo, and H. Liu, "Provably secure identity based provable data possession," in Proc. Provable Security, vol. 9451. 2015 , pp. 310 − 325

[37] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing", J. Cryptol., vol. 17, no. 4, pp. 297-319, 2004.

[38] D. Chaum and T.P. Pedersen, "Wallet databases with observers"' in Proc. CRYPTO, 2001, pp. 89-105

[39] J. C. Cha and J. H. Cheon, "An identity-based signature from gap DiffieHellman groups", in Proc. PKC, vol. 2567. 2003, pp. 18-30.

[40] F. Hess, "Efficient identity based signature schemes based on pairings," in Proc. Sel. Areas Cryptography, vol. 2595. 2003 , pp. 310 − 324.

[41] B. Lynn. (2013). The Pairing-Based Cryptography Library (0.5.13). [Online]. Available: http://crypto.stanford.edu/pbc/

[42] A. Kate. The Pairing-Based Cryptography (PBC) Library- $C++Wrap-$ per Classes (0.8.0) , accessed on Sep. 8, 2015. [Online]. Available: http://crysp.uwaterloo.ca/software/PBCWrapper/

[43] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing

computation to untrusted workers," in Proc. CRYPTO, vol. 6223. 2010, pp. 465 – 482.

[44] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," IEEE Trans. Depend. Sec. Comput. vol. 12 , no. 5, pp. 546-556, Sep./Oct. 2015

[45] J. Lai, R. H. Deng, H. Pang, and J. Weng, "Verifiable computation on outsourced encrypted data," in Proc. ESORICS, 2014, pp. 273-291.

[46] X. Liu, R. Choo, R. Deng, R. Lu, and J. Weng, "Efficient and privacypreserving outsourced calculation of rational numbers," IEEE Trans. Depend. Sec. Comput., to be published.

无证书的 RDPC 方案

1 介绍

云存储服务为用户提供了一种共享数据和作为一个团队进行工作的有效方式。一旦团队中的某个人将一个文件上传到服务器上，其他成员就可以通过互联网访问和修改该文件。许多真实的应用程序，如 Dropbox for Business[1]和 TortoiseSVN[2]在许多公司被用于他们的员工的协同工作。这类应用程序最重要的问题是云服务器提供商(CSP)是否可以确保数据保持完整的[3]。事实上，CSP 并不完全可靠，软件或硬件的故障在某种程度上是不可避免的，因此随时都可能发生严重的数据损坏事故。因此，用户需要审核 CSP，以确认云服务器上的数据是原始的。

为了保证存储数据的完整性，提出了大量的 RDPC 方案[4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32]。在这些方案中，每个数据块都会生成一个与该块绑定的身份验证标记。通过检查标签的正确性，验证者能够了解数据的状态。然而，这些方案大多只关注于检查个人数据的完整性 [4], [5], [6], [7], [8], [9], [10], [11],[12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [29], [30],[31], [32]，这在一个群中共享数据的情况下是无效的。当数据在多个用户之间共享时，会出现一些新的挑战，这些挑战在 RDPC 方案中没有得到很好的解决。此外，当群组用户更新一个块时，它应该再次重新生成标记。在审计数据完整性时，需要聚合单独生成的所有身份验证标签，包括这些标签的所有生成器的信息。它给检查方案带来了极大的复杂性。此外，该群是动态的，任何群成员都可以在任何时候主动离开或被开除该群，因此用户撤销也是一个必须解决的重要问题。更具体地说，一旦用户被撤销，就不应该允许他访问或修改数据，而且他的所有公钥/私钥都无效。在这种情况下，不可能检查由被撤销的用户所做的标签的正确性。因此，由被撤销的用户创建的所有标记都应该由其他正常用户进行更新。传统的方法是从 CSP 中下载由被撤销的用户签署的块，计算新的标签，并将新的标签再次上传到云中。它将为正常用户增加沉重的计算和通信成本。因此，这个任务应该由 CSP 而不是普通用户来执行。如何设计一种高效、安全的方法来外包任务一直是一个难题。此外，公共验证是一种数据完整性检查工作的诱人特点。此外，共享数据的完整性不仅可以由数据所有者来验证，还可以由所有对云数据感兴趣的人来验证。在当前的开放环境下，RDPC 协议支持公共验证是非常重要的。

到目前为止，已经提出了许多方案[22]、[23]、[24]、[25]、[25]、[26]、[27]、[28]用于群共享数据的完整性验证。然而，现有的 RDPC 方案[22]、[23]、[24]、[25]、[26]、[28]大多都是基于 PKI 的。PKI 虽然在公钥密码学中得到了广泛的应用并占有重要地位，但仍存在一定的安全威胁。例如，PKI 的安全性是基于证书颁发机构(CA)的可信度，但要确保 CA 的可信度并不容易。此外，对证书的分发、存储、撤销、验证等方面的管理也是一个很大的负担。为了避免这些问题，提出了一些基于身份的 RDPC 方案[27]、[28]。不幸的是，基于身份的 RDPC 方案存在密钥托管问题。即，私钥生成器(PKG)为用户生成所有的私钥。如果 PKG 不受信任，则该方案也不安全。因此，基于身份的 RDPC 方案可能被限制在小的、封闭的设置中。与 PKI 和 IBC 相比，无证书密码学[33]同时解决了证书管理和密钥托管的问题。构建无证书的 RDPC 方案是一种很好的云数据完整性检查方法。

1.1 动机和贡献

在本文中，我们主要关注在一个群内共享的数据的完整性检查。假设有这样一个场景：软件工程师启动一个开源项目，并呼吁来自世界各地的志愿者加入该项目。他们作为一个临时的团队工作。项目的所有代码都存储在特定的云服务器上，以便所有团队成员通过互联网上传和修改源代码。这个团队可能非常大，所以应该有效地设置和管理它。志愿者可以随时离开团队，因此应该考虑用户退出团队的问题。最重要的是，需要有一些方法来保证云服务器上的源代码的完整性。

基于这样的需求，我们提出了一个新的 RDPC 方案来实现在一个群中共享的数据。与以往的工作不同，我们的方案是基于无证书的签名技术，避免了证书管理和密钥托管的问题。在我们的方案中，群创建者代表密钥生成中心为每个群用户生成部分密钥。每个用户都会私下选择一个秘密值。每个群用户的私钥包含两个部分：一个部分密钥和一个秘密值。所有的数据块都由群用户进行签名，以获得相应的身份验证标记。在数据验证过程中，对所有标签进行聚合，以降低计算和通信成本。基于 CDH 和 DL 假设，证明了该方案的安全性。此外，我们的方案支持公共验证和有效的用户撤销。我们实现了我们的方案，并进行了一些实验。实验结果表明，该方案具有良好的应用效率。

1.2 相关工作

第一个用于远程数据检查的 RDPC 协议是由 Deswarte 等人[4]提出的，其中使用基于 RSA 的哈希函数来生成数据的身份验证标签。在此基础上，提出了大量的可证明数据占有(PDP)[5]和可检索性证明(POR)[29]方案，以解决数据完整性验证的问题。

Ateniese 等人[5]首先提出了 PDP 模型，并首先引入了对远程数据的概率完整性检查技术。然而，第一个 PDP 方案只适用于静态数据。为了满足数据块的动态操作，Ateniese 等人[6]提出了另一种可伸缩的、高效的对称加密 PDP 方案，支持块的附加、更新和删除。Seb 等人提出了一种基于大整数分解困难问题的 PDP 协议。Erway 等人[8]利用经过身份验证的跳过列表提供了一个完全动态的 PDP 方案，该方案支持数据所有者插入、附加、修改和删除数据块。

Wang 等人[9]基于随机掩蔽技术和同态线性认证器，提出了一种具有隐私保护特性的公共验证 PDP 方案。为了支持公共的可审计性和数据动态，Wang 等人[10]利用默克哈希树(MHT)提出了一种云数据检查的动态方案。该方案是完全动态的，并允许任何人使用公钥来验证文件的完整性。在方案[11]、[12]中也使用了 MHT 来实现数据动态化。然而，由于 MHT 的计算复杂性，该方案造成了高昂的计算成本和通信成本。为了克服这一缺点，Yang 和 Jia[13]引入了一个线性索引表来支持数据动态。Yan 等人[14]进一步优化了线性索引表的实现，提供了一个有效的 RDPC 方案。Feng 等[15]提出了一种公共远程完整性检查方案，可以在文件级别上保护用户身份，降低存储和通信成本。

Zhu 等人提供了多云设置的合作 PDP 方案，其中数据块存储在不同的云服务器上。为了提高安全性，Wang[17]提出了另一种用于多云设置的基于身份的 PDP 方案，没有证书管理。最近，Wang 等人[18]提出了一种激励和无条件匿名的基于身份的公共 PDP 方案。为了降低数据所有者的计算成本，Wang[19]等人[19]提出了一种面向代理的 PDP 方案，该方案将标签生成工作从数据所有者转移到代理。针对解决密钥托管和证书管理的问题，分别提出了两种基于无证书[20]和基于证书的密码学[21]的 PDP 方案。

上述所有方案均集中于个人数据的完整性验证。2012 年，Wang 等人[22]提出了一种检查群中共享数据完整性的协议。他们利用群签名技术来生成每个身份验证标签，以保护标签生成器的隐私。Wang 等人[23]提出了另一种针对群数据的 PDP 方案，支持群用户的加入和离开。Liu 等[24]基于广播加密和群签名技术，提出了一种群数据的 PDP 方案。为了提高效

率，Wang 等人[25]提出了另一种基于环签名技术的方案。但是，这两种方案并没有解决用户撤销的问题。为了解决这个问题，Wang 等人[26]使用代理重签名技术提出了一种新的用户撤销方案。Yu 等人[27]提出了一个没有配对的 PDP 方案，也支持动态群。Yuan 和 Yu[28]提出了一种基于基于多项式的认证标签的 PDP 方案，旨在解决块的多用户修改问题。这些方案都依赖于传统的 PKI 机制，存在安全风险，证书管理负担大。此外，[27]、[28]的方案还存在密钥托管的问题。因此，该方案在实际应用中仍存在很大的局限性。

PoR[29]是审计云服务器上远程数据完整性的另一个方向。为了提高效率，Shacham 和 Waters[30]提出了两种基于短签名[34]技术的紧凑 PoR 方案。此外，许多 PoR 方案[31]，[32]具有更高的效率或更好的安全性。

随着云计算的发展，如何从云服务器共享数据受了越来越多的关注。为了保证系统的安全性和隐私性，获得灵活的文件访问控制，提出了基于属性的加密(ABE)方案[35]、[36]、[37]、[38]、[39]、[40]、[41]、[42]，并将其应用于云存储系统中。在 ABE 方案中，加密器将密文与一组属性关联起来。该权限发布用户与属性访问策略关联的不同私钥。Li 等人[35]，[36]建模了现有用户与被撤销用户执行的共谋攻击，构建了具有用户撤销的高效 CP-ABE 方案。最近，Wang 等人[37]提供了一种匿名分布式细粒度访问控制方案，在公共云中使用可验证的外包解密。在 ABE 方案中，处于隐私要求，外包前应对敏感文档进行加密，这阻碍了基于关键字的文档检索等有效的查询处理。为了解决这个问题，Li 等人[38]，[39]提出了具有关键字搜索功能的 ABE 方案。ABE 方案可能具有敏感信息，并泄露加密器的隐私，因为访问策略与密文一起被发送到解密器。具有隐藏访问策略[40]的 ABE 方案和保护隐私的 ABE 方案[41]可以克服上述问题。为了保证云服务提供商能够正确地存储密文，提出了一些具有有效的可验证的外包解密的 ABE 方案[42]、[43]。最近的研究主要集中在对授权用户的外包解密的可验证性上。如何对未授权用户保证外包解密的正确性仍然是一个具有挑战性的问题。最近 Li 等人[44]在 ABE 中提出了外包解密的完全可验证性，可以同时为未授权用户和授权用户验证转换后的密文的正确性。

2 储备知识

在本节中，我们将介绍在本文中使用的储备知识。

2.1 双线性映射

略

2.2 困难型假设

定义 1 (计算性 Diffie-Hellman (CDH) 问题). 假设$\mathbb{G}_1$ 是一个乘法循环群。 g 是 $\mathbb{G}_1$的一个生成元。 给定元组 (g, g^a, g^b) 其中 $a, b \in Z_q^*$未知，CDH 问题就是要计算出 g^{ab}。

定义 2 (CDH 假设). 对于任意的概率多项式时间(PPT) 算法$\mathcal{A}$, $\mathcal{A}$ 在$\mathbb{G}_1$中解决CDH 问题的优势是可忽略的，定义为: $Adv_{\mathbb{G}_1\mathcal{A}}^{CDH} = \Pr\left[\mathcal{A}(g, g^a, g^b) = g^{ab} : a, b \overset{R}{\leftarrow} Z_q^*\right] \leq \varepsilon$。

定义 3 (离散对数 (DL) 问题). 假设$\mathbb{G}_1$ 是一个乘法循环群。 g 是 $\mathbb{G}_1$的一个生成元。 G 给定 (g, g^a) 的值， 其中$a \in Z_q^*$未知， DL 问题就是要计算出 a。

定义 4 (DL 假设). 对于任意的概率多项式时间(PPT) 算法$\mathcal{A}$, $\mathcal{A}$ 在$\mathbb{G}_1$中解决DL问题的优势是可忽略的， 定义为: $Adv_{\mathbb{G}_1\mathcal{A}}^{DL} = \Pr\left[\mathcal{A}(g, g^a) = a : a \overset{R}{\leftarrow} Z_q^*\right] \leq \varepsilon$

注意, ε 在上述定义在代表一个可忽略的值。

2.3 系统模型

参考论文[25]、[26]、[27]、[28]，我们的方案的系统模型由三个主要实体组成：用户群组、云服务提供商(CSP)和公共验证者。用户群组包括许多用户，他们可以上传、访问和更新群组内共享的数据，并诚实地执行协议。在不失一般性的情况下，群组的原始创建者扮演着管理员的角色，他设置系统并为一般群组用户生成部分密钥。CSP 拥有强大的存储和计算能力，可以为云用户提供数据存储服务。在我们的方案中，共享数据被划分为多个块，每个块都带有一个身份验证标签。因此，CSP 为云用户存储了所有的块和相应的标签。数据验证者是一个检查 CSP 上数据的完整性的人。由于公共验证的特点，任何人都可以成为我们方案的验证者。图 1 显示了系统的三个实体之间的关系和相互作用。正如之前的大多数作品[4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32],一样，我们假设 CSP 是半可信的。也就是说，CSP 可以诚实地执行协议，但可能会欺骗验证者关于数据的不正确性，以保持其声誉或获得额外的利益。

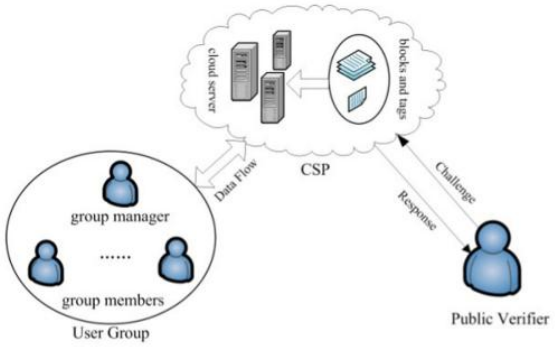

Fig. 1. System model of our scheme.

2.4 RDPC 方案大纲

定义 5：RDPC 方案由八种算法组成：

Setup: 输入安全参数k，输出主密钥msk，和系统公开参数 $params$。该算法由群组的管理员执行。

PartialKeyGen: 由群组管理员执行，为群组用户生成部分私钥。以主私钥msk和用户u_i的身份ID_i作为输入，输出u_i的部分私钥D_i。

SecretValueGen: 群组用户执行这个算法来生成秘密值。算法随机选择S_i作为用户u_i的秘密值。因此，群组用户的私钥由两部分组成：秘密值和部分私钥。

PublicKeyGen：该算法由群组用户执行来生成公钥。输入用户u_i的秘密值S_i，输出u_i的公钥PK_i。

TagGen：群组用户执行此算法，为一个数据块生成认证标签。输入u_i的部分私钥D_i，秘密值S_i，和数据块m_j，输出m_j的标签T_j。

Challenge：该算法由验证者执行。输入挑战的块数c，输出挑战消息$chal$。

ProofGen：CSP 执行该算法，获得占有证据。以被挑战的文件F，所有数据块的标签集合T，以及挑战信息$chal$作为输入，输入完整性证据P。

Verify：验证者执行该算法来验证完整性证据P。它以证据P，挑战信息$chal$和所有群组用户的公钥集合PK作为输入。如果P是正确的，输出 1，否则输出 0。

2.5 安全模型

由于我们的新方案引入了无证书密码学[33]的思想，因此我们在我们的安全模型中考虑了三个对手，即$\mathcal{A}_I$、$\mathcal{A}_{II}$和$\mathcal{A}_{III}$。$\mathcal{A}_I$和$\mathcal{A}_{II}$都试图伪造数据块的标签。但不同之处在于，$\mathcal{A}_I$不能访问系统的主密钥，但可以用任何其他值替换用户的公钥，而$\mathcal{A}_{II}$能够获得主密钥，但不能替换用户的公钥。$\mathcal{A}_{III}$的目的是伪造数据完整性证据来欺骗验证者。参考[20]，[26]，我们通过三个游戏来定义我们的方案的安全性，其中分别涉及到一个挑战者$\mathcal{C}$和对手$\mathcal{A}_I$，$\mathcal{A}_{II}$和$\mathcal{A}_{III}$。对三个安全游戏的描述如下：

游戏 I：由$\mathcal{C}$和$\mathcal{A}_I$来完成。

初始化：$\mathcal{C}$运行 Setup 算法，获得主密钥和公开参数。$\mathcal{C}$自己保留主密钥，把公开参数发送给$\mathcal{A}_I$。

询问：$\mathcal{A}_I$可以在多项式时间内对$\mathcal{C}$做不同的询问。$\mathcal{C}$按照如下方式对$\mathcal{A}_I$的询问进行回应：

哈希询问：$\mathcal{A}_I$适应性的向$\mathcal{C}$做哈希询问。$\mathcal{C}$向$\mathcal{A}_I$回应哈希值。

部分私钥询问。$\mathcal{A}_I$自适应的选择不同的ID并提交给$\mathcal{C}$来询问ID的部分私钥。$\mathcal{C}$执行 PartialKeyGen 算法获得ID的私钥并发给$\mathcal{A}_I$。

秘密值询问。$\mathcal{A}_I$自适应地选择不同的ID并提交给$\mathcal{C}$来询问ID的秘密值。$\mathcal{C}$运行 SecretValueGen 算法生成秘密值并发送给$\mathcal{A}_I$。

公钥查询：$\mathcal{A}_I$自适应地选择不同的ID并提交给$\mathcal{C}$来询问ID的公钥。$\mathcal{C}$运行 PublicKeyGen 算法计算公钥并发送给$\mathcal{A}_I$。

公钥替换：$\mathcal{A}_I$重复地选择一个值来替换任意ID的公钥。

标签询问：$\mathcal{A}_I$自适应地选择元组(ID, m)并提交给$\mathcal{C}$，以询问块m由ID生成的标签。

通过 TagGen 算法，$\mathcal{C}$生成m的标签并发送给$\mathcal{A}_I$。

伪造：最后，$\mathcal{A}_I$输出数据块m'由身份ID'和公钥$PK_{ID'}$生成的标签T'。

如果满足以下条件，$\mathcal{A}_I$将获胜。

1）伪造的标签T'是在身份ID'和公钥$PK_{ID'}$下对数据块m'的有效标签。

2）$\mathcal{A}_I$没有询问过ID'的私钥。

3）$\mathcal{A}_I$没有同时询问ID'的部分私钥并替换了ID'的公钥。

4）$\mathcal{A}_I$没有在身份ID'和公钥$PK_{ID'}$下对m'做过标签询问。

游戏 II：由$\mathcal{C}$和$\mathcal{A}_{II}$来完成。

初始化：$\mathcal{C}$运行 Setup 算法，获得主密钥和公开参数。$\mathcal{C}$把主密钥和公开参数都发送给$\mathcal{A}_{II}$。

询问：$\mathcal{A}_{II}$可以在多项式时间内对$\mathcal{C}$做不同的询问。$\mathcal{C}$按照如下方式对$\mathcal{A}_{II}$的询问进行回应：

哈希询问。$\mathcal{A}_{II}$适应性的向$\mathcal{C}$做哈希询问。$\mathcal{C}$向$\mathcal{A}_{II}$回应哈希值。

秘密值询问。$\mathcal{A}_{II}$自适应地选择不同的ID并提交给$\mathcal{C}$来询问ID的秘密值。$\mathcal{C}$运行 SecretValueGen 算法生成秘密值并发送给$\mathcal{A}_{II}$。

公钥询问。$\mathcal{A}_{II}$自适应地选择不同的ID并提交给$\mathcal{C}$来询问ID的公钥。$\mathcal{C}$运行 PublicKeyGen 算法计算公钥并发送给$\mathcal{A}_{II}$。

标签询问。$\mathcal{A}_{II}$自适应地选择元组(ID, m)并提交给$\mathcal{C}$，以询问块 m由ID生成的标签。

通过 TagGen 算法，$\mathcal{C}$生成m的标签并发送给$\mathcal{A}_{II}$。

伪造。最后，$\mathcal{A}_{II}$输出数据块m'由身份ID'标签T'。

如果满足以下条件，$\mathcal{A}_{II}$将获胜。

1）伪造的标签T'是在身份ID'下对数据块m'的有效标签。

2）$\mathcal{A}_{II}$没有询问过ID'的秘密值。

3）$\mathcal{A}_{II}$没有在身份ID'下对m'做过标签询问。

定义 6：如果对于任意的多项式时间敌手$\mathcal{A}$（$\mathcal{A}_I$或$\mathcal{A}_{II}$），$\mathcal{A}$赢得游戏 I 和游戏 II 的概率是可忽略的，数据块的单个标签是存在性不可伪造的。

游戏 III： 挑战者$\mathcal{C}$ 和敌手$\mathcal{A}_{III}$ i 在游戏中进行交互。在这里，$\mathcal{A}_{III}$ 被认为是不可信的 CSP。如果数据损坏，$\mathcal{A}_{III}$ 会试图欺骗验证者数据是保持完整的。从定义 6 中，我们知道任何敌手在没有正确的私钥的情况下都不能伪造单个块的标签。因此，在这个游戏中，我们只关注 $\mathcal{A}_{III}$ 是否可以在不正确的数据上伪造完整性证据来通过验证。受[26]的启发，游戏 3 的过程定义如下。

初始化：$\mathcal{C}$为所有用户生成公共参数、主密钥和私钥。$\mathcal{C}$将所有私钥和主密钥私有，但将公共参数发送给$\mathcal{A}_{III}$。

标签询问：$\mathcal{A}_{III}$自适应地选择元组(ID, m)并发送给$\mathcal{C}$，询问由 ID 生成的m的标签。$\mathcal{C}$通过 TagGen 算法生成m的标签并发送给$\mathcal{A}_{III}$。

挑战。$\mathcal{C}$ 生成一个随机的挑战消息$chal$，把$chal$发送给$\mathcal{A}_{III}$，并要求$\mathcal{A}_{III}$给出$chal$对应的数据拥有证据P。
伪造。对于挑战消息$chal$，AIII 生成证据P并把它交给$\mathcal{C}$。如果P可以通过完整性验证，并且P中的块信息是错误的，$\mathcal{A}_{III}$获胜。

定义 7。如果任意的多项式时间敌手$\mathcal{A}_{III}$只能以可忽略的概率赢得关于一个数据块的游戏 III，那么在没有正确数据的情况下，伪造完整性证据的概率是可忽略的。

3 我们的方案

3.1 方案构造

我们假设在一个群组中存在z个用户，让ID_i表示用户u_i的唯一标识（$1 \le i \le z$）。在不丧失一般性的情况下，我们将u_1设置为群组管理员，他将设置系统并为其他用户生成部分密钥。我们假设文件F被分成n个块，表示为$F = (m_1, m_2, ..., m_n)$，每个块都是Z_q中的一个元素。方案详细实施如下：

Setup 算法：令$\mathbb{G}_1$和$\mathbb{G}_2$是素数q阶的乘法群。$e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$是一个双线性映射，$g$是$\mathbb{G}_1$的生成元。$u_1$选择选择两个安全哈希函数$H_1: \{0,1\}^* \to \mathbb{G}_1^*$ 和$H_2: \{0,1\}^* \to \mathbb{G}_1^*$，一个伪随机排列(PRP) $\pi: Z_q^* \times \{1, ..., n\} \to \{1, ..., n\}$ 和一个伪随机函数(PRF) $\phi: Z_q^* \times Z_q^* \to Z_q^*$。$u_1$随机选择$s \in Z_q^*$作为主私钥，并计算$P_0 = g^s$。$u_1$自己保留主私钥并公开系统参数$params = (q, g, \mathbb{G}_1, \mathbb{G}_2, e, P_0, H_1, H_2, \phi, \pi)$。

PartitialKeyGen 算法：当收到群组用户u_i的身份ID_i，管理员u_1计算$D_i = H_1(ID_i)^s$，返回D_i给u_i作为它的部分私钥。注意D_i在群组中是唯一的。

SecretValueGen 算法：u_i随机选择$x_i \in Z_q^*$并设置$S_i = x_i$作为秘密值并保密。

PublicKeyGen 算法：u_i 使用秘密值 S_i 计算公钥 $PK_i = g^{S_i} = g^{x_i}$。

TagGen 算法：群组中的每个用户都可以访问文件块，并使用其私钥为它们生成标签。假设用户$u_i (1 \le i \le z)$想要为块$m_j (1 \le j \le n)$生成标签，计算标签的方程是$T_j = D_i m_j \cdot H_2(\omega_j)^{S_i}$，其中$\omega_j = F_{id} \| n \| j$，$F_{id}$表示唯一的文件标识。群组管理员维护一个公共日志文件，该文件存储$\omega_j (1 \le j \le n)$的信息和标签生成者的身份。在添加或修改数据块后，用户应该添加或更新日志文件中相应的$\omega_j (1 \le j \le n)$和的标签生成者的身份。用户上传这些块和标签到 CSP。CSP 可以通过下列等式检查标签的验证情况：

$$e(T_j, g) = e(H_2(\omega_j), PK_i) \cdot e(H_1(ID_i)^{m_j}, P_0) \qquad (1)$$

Challenge 算法： 验证者随机选择挑战块的数量$c \in [1, n]$和两个随机值(k_1, k_2)，其中$k_1, k_2 \in Z_q^*$分别是 PRP 和 PRF 的种子。验证者发送$chal = (c, k_1, k_2)$给 CSP 作为挑战信息。

ProofGen 算法：当 CSP 收到挑战消息$chal = (c, k_1, k_2)$，CSP 计算集合$C = (v_i, a_i)$，其中$v_i = \pi(k_1, i)$, $a_i = \phi(k_2, i)$，对于 $1 \le i \le c$。根据每个被挑战块的标签生成者，CSP 把集合C分割成d个子集$C = \{C_1, ..., C_d\} (d \le z)$，其中子集$C_j$是由用户$u_{l_j} \left(1 \le j \le d, 1 \le l_j \le z\right)$生成的标签的集合。令$c_j$表示$C_j$中元素的数量。我们可以得到$c = \sum_{j=1}^{d} c_j, C = C_1 \cup C_2 \cdots \cup C_d$，且对于$j \ne j'$，$C_j \cap C_{j'} = \emptyset$。对于每一个子集$C_j$，CSP 计算出$\left(\overline{T_j}, \overline{F_j}\right)$，通过$\overline{T_j} = \prod_{v_i \in C_j} T_{v_i}{}^{a_i}$和$\overline{F_j} = \sum_{v_i \in C_j} a_i m_{v_i}$。CSP 返回$P = (\bar{T}, \bar{F})$作为最终的证据，这里的$\bar{T} = \{\overline{T_1}, ..., \overline{T_d}\}, \bar{F} = \{\overline{F_1}, ..., \overline{F_d}\}$。

Verify。当收到完整性证据P，验证者计算$v_i = \pi(k_1, i), a_i = \phi(k_2, i)$，并按照 CSP 那样根据标签生成者对挑战集合进行划分。通过在日志文件中搜索，验证者获得了所有被挑战文件块的ω_{v_i}。最后，验证者检查等式（2）是否成立：

$$e\left(\prod_{j=1}^{d} \bar{T}_j, g\right) = \prod_{j=1}^{d} \left(e\left(\prod_{v_i \in C_j} H_2(\omega_{v_i})^{a_i}, PK_{l_j}\right)\right) \cdot e\left(\prod_{j=1}^{d} H_1\left(ID_{l_j}\right)^{\bar{F}_j}, P_0\right) \quad (2)$$

如果等式（2）成立，验证者输出 1，否则输出 0。如果 CSP 和验证者诚实的运行该协议，我们可以通过以下等式来检查协议的正确性：

$$e\left(\prod_{j=1}^{d} \bar{T}_j, g\right) = \prod_{j=1}^{d} e\left(\bar{T}_j, g\right) = \prod_{j=1}^{d} e\left(\prod_{v_i \in C_j} T_{v_i}^{a_i}, g\right)$$

$$= \prod_{j=1}^{d} e\left(\prod_{v_i \in C_j} \left(H_1\left(ID_{l_j}\right)^{sa_i m_{v_j}} \cdot \left(H_2(\omega_{v_i})\right)^{x_{l_j} a_i}\right), g\right)$$

$$= \prod_{j=1}^{d} \left(e\left(\prod_{v_i \in C_j} H_1\left(ID_{l_j}\right)^{a_i m_{v_i}}, g^s\right) \cdot e\left(\prod_{v_i \in C_j} H_2(\omega_{v_i})^{a_i}, g^{x_{l_j}}\right)\right)$$

$$= \prod_{j=1}^{d} \left(e\left(H_1\left(ID_{l_j}\right)^{\bar{F}_j}, P_0\right) \cdot e\left(\prod_{v_i \in C_i} H_2(\omega_{v_i})^{a_i}, PK_{l_j}\right)\right)$$

$$= \prod_{j=1}^{d} e\left(\prod_{v_i \in C_j} H_2(\omega_{v_i})^{a_i}, PK_{l_j}\right) \cdot e\left(\prod_{j=1}^{d} H_1\left(ID_{l_j}\right)^{F_j}, P_0\right)$$

3.2 支持用户撤销

如果任何用户$u_i (2 \le i \le z)$ l 离开该群组，则应声明u_i的密钥和公钥无效。因此，应该从u_i生成的标签中删除u_i的密钥。否则，将无法验证这些标记的有效性，也无法检查文件的完整性。重新生成被撤销用户的标签的传统方法是从 CSP 下载块，重新生成标签，然后再次上传新的标签。这不可避免地增加了用户的通信成本和计算成本。因此，为了减轻这些潜在的开销，我们的方案设计了将标签更新工作外包给 CSP，并降低了通信成本。我们使用 ReTagGen 算法来更新由被撤销的用户生成的标签。ReTagGen 的构造如下所示，其中u_i是被撤销的用户，$u_j (1 \le j \le z, j \ne i)$是组中的另一个有效用户。值得注意的是，$u_1$是组的创建者和管理者，我们认为$u_1$不应该被撤销。

ReTagGen: 该算法包含u_i、u_j和 CSP 之间的三个交互。我们假设u_i、u_j和 CSP 之间不存在勾结，并且在交互过程中使用了安全通道。此外，在撤销程序中，要求u_i、u_j和 CSP 同时在线。

1 CSP P 随机选择一个值 $\rho \in Z_q^*$并将ρ 通过安全信道发送给 u_j 。

2 u_j 计算并发送 $\left(W_1 = (D_j)^{\frac{1}{S_j}}, W_2 = \rho \cdot S_j\right)$ 给 u_i。

3 u_i 计算并发送 $\left(R_1=\frac{W_1 S_i}{D_i}, R_2=\frac{W_2}{S_i}\right)$ 给 CSP。

4 当收到 (R_1, R_2)，CSP 计算$R_3=\frac{R_2}{\rho}=\frac{S_j}{S_i}$。使用等式（1）来检查由$u_i$生成的所有标签-块对$\left[T_{i'}, m_{i'}\right]\left(1 \leq i' \leq n\right)$ 。然后 CSP 将块$m_{i'}$的标签$T_{i'}$转换为：

$$T_{i'}^{'}=\left(R_1^{m_{i'}} \cdot T_{i'}\right)^{R_3}=\left(\left(\frac{D_j^{S_j}}{D_i}\right)^{m_{i'}} \cdot D_i^{m_{i'}} \cdot H_2\left(\omega_{i'}\right)^{S_i}\right)^{\frac{S_j}{S_i}}=\left(D_j^{\frac{S_i m_i S_j}{}} \cdot H_2\left(\omega_{i'}\right)^{S_i}\right)^{\frac{S_j}{S_i}}$$

$=D_j^{m_{i'}} \cdot H_2\left(\omega_{i'}\right)^{S_j}$ ，其中 $T_{i'}^{'}$是由u_j生成的$m_{i'}$的有效标签。

4 安全证明

通过以下三个定理证明了该方案是安全的。

4.1 安全证明

定理 1。如果一个概率多项式时间敌手 $\mathcal{A}_I$ 在时间 t 内，最多分别以 $q_{H_1}, q_{k1}, q_{k2}, q_{k3}, q_{kr}, q_{H_2}, q_T$次发起$H_1$-哈希询问、部分私钥询问、秘密值询问、公钥询问、$H_2$-哈希询问和标签询问后，以$\varepsilon$的优势赢得 2.5 节中定义的游戏 I，那么就有一个模拟者$\mathcal{B}$ 可以在时间$t' \le t + O\left(q_{H_1} + q_{k1} + q_{k2} + q_{k3} + q_{kr} + q_{H_2} + q_T\right)$内，以$\varepsilon' \ge \varepsilon/\left((q_{k1} + q_T) \cdot 2e\right)$ 的概率攻破 CDH 问题。

证明。给定一个 CDH 问题的实例 $\left(g, \mathbb{G}_1, g^a, g^b\right)$. 如果敌手$\mathcal{A}_I$ 以不可忽略的优势赢得游戏 I，模拟者 $\mathcal{B}$就能通过$\mathcal{A}_I$的能力以不可忽略的概率计算出g^{ab}的值。$\mathcal{B}$模拟了与$\mathcal{A}_I$ 的每个交互步骤，如下所示。

初始化。 $\mathcal{B}$产生公开参数，并设置 $P_0 = g^a$ ，其中主密钥是隐匿的未知值 a 。

H_1-哈希询问。 $\mathcal{A}_I$ 自适应地对任意身份ID^* 做H_1-哈希询问。$\mathcal{B}$ 维护一个H_1-哈希询问的列表 $L_1 = \{(ID, h_1, Q, T)\}$ 。 如果 ID^* 存在于 L_1，$\mathcal{B}$ 检索元组 (ID^*, h_1^*, Q^*, T^*) 并返回 Q^* 给$\mathcal{A}_I$。否则， $\mathcal{B}$ 随机选择一个值 $h_1^* \in Z_q^*$ 并抛出一个硬币 $T \in \{0,1\}$。假设 $T^* = 0$ 的概率是 γ，则$T^* = 1$ 的概率是 $1 - \gamma$.。如果$T = 0$，$\mathcal{B}$ 计算$Q^* = g^{h_1^*}$。如果$T = 1$，$\mathcal{B}$ 设置$Q^* = \left(g^b\right)^{h_1^*}$。$\mathcal{B}$ 回复Q^* 给$\mathcal{A}_I$ 并将运足 (ID^*, h_1^*, Q^*, T^*) 添加到L_1中。

部分私钥询问。 $\mathcal{A}_I$ 自适应地执行对任意身份ID^*的部分私钥询问。$\mathcal{B}$ 首先检查$(ID^*, h_1{}^*, Q^*, T^*)$是否存在$L_1$列表中。如果不存在，$\mathcal{B}$对身份$ID^*$做$H_1$-哈希询问。注意，如果$T^* = 1$，$\mathcal{B}$中止。$\mathcal{B}$还会维护一个部分私钥询问的列表$L_2 = \{(ID, D_{ID}, PK_{ID}, S_{ID})\}$。

1 如果 ID^* 存在于 L_2，$\mathcal{B}$ 检查是否对应的值 $D_{ID^*} = \perp$。如果 $D_{ID^*} = \perp$ ，$\mathcal{B}$ 从L_1中搜索元组(ID^*, h_1^*, Q^*, T^*)，对于$T^* = 0$，计算 $D_{ID^*} = (Q^*)^a = g^{ah_1^*}$ 并更新元组中的 D_{ID^*} 。如果$T^* = 1$，$\mathcal{B}$ 终止。对于 $D_{ID^*} \neq \perp$ ，$\mathcal{B}$ 直接得到 D_{ID^*} 。然后 $\mathcal{B}$ 返回D_{ID^*} 给$\mathcal{A}_I$。

2 如果 ID^* 不存在与 L_2，$\mathcal{B}$ 从L_1中搜索元组(ID^*, h_1^*, Q^*, T^*) ，如果$T^* = 0$，计算 $D_{ID^*} = (Q^*)^a = g^{ah_1^*}$ 。如果$T^* = 1$，$\mathcal{B}$ 终止。最后，$\mathcal{B}$ 返回D_{ID^*} 给$\mathcal{A}_I$，并添加心得元组 $(ID^*, D_{ID^*}, \perp, \perp)$ 到L_2中。

秘密值询问。 $\mathcal{A}_I$ 自适应地对任意身份 ID^*做秘密值询问。$\mathcal{B}$检查是否元组(ID^*, h_1^*, Q^*, T^*)已经在 L_1中存在。如不存在， $\mathcal{B}$ 对ID^*做 H_1-哈希询问。$\mathcal{B}$检查 ID^* 是否在 L_2中存在。

1 如果ID^*存在于L_2，$\mathcal{B}$ 检查是否对应的值 $S_{ID^*} = \perp$.。如果 $S_{ID^*} = \perp$ ，$\mathcal{B}$ 随机选择一个值 $x \in Z_q^*$ 并设置 $S_{ID^*} = x$, $PK_{ID^*} = g^x$。然后$\mathcal{B}$ 使用 S_{ID^*} and PK_{ID^*}更新元组。如果$S_{ID^*} \neq \perp$ ，$\mathcal{B}$ 在元组中检索它并返回给 $\mathcal{A}_I$。

2 如果ID^* 不存在于L_2，$\mathcal{B}$ 随机选择一个值 $x \in Z_q^*$ 并设置 $S_{ID^*} = x$, $PK_{ID^*} = g^x$。最

后，$\mathcal{B}$ 添加元组 $(ID^*, \perp, g^x, x)$ 到L_2中并返回 S_{ID^*} 给$\mathcal{A}_I$。

公钥询问。 $\mathcal{A}_I$ 自适应地对任意身份 ID^*做公钥询问。

1 如果元组 $(ID^*, D_{ID^*}, PK_{ID^*}, S_{ID^*})$ 在 L_2 中存在，$\mathcal{B}$ 检查是否 $PK_{ID^*} = \perp$ 。 如果 $PK_{ID^*} = \perp$，$\mathcal{B}$ 随机选择一个值 $x \in Z_q^*$ 并设置 $S_{ID^*} = x, PK_{ID^*} = g^x$ 。$\mathcal{B}$ 返回PK_{ID^*} 并更新元组中的S_{ID^*}, PK_{ID^*} 。如果 $PK_{ID^*} \neq \perp$，$\mathcal{B}$ 直接把它返回给 $\mathcal{A}_I$ 。

2 如果 L_2 中不包含元组 $(ID^*, D_{ID^*}, PK_{ID^*}, S_{ID^*})$,$\mathcal{B}$ 随机选择 $x \in Z_q^*$ 并设置 $S_{ID^*} = x, PK_{ID^*} = g^x$.$\mathcal{B}$ 把元组 $(ID^*, \perp, PK_{ID^*}, S_{ID^*})$ 添加到L_2 并发送 PK_{ID^*} 给 $\mathcal{A}_I$.

公钥替换。$\mathcal{A}_I$ 自适应地用 $\left(ID^*, PK'_{ID^*}\right)$进行公钥替换。

1 如果元组 $(ID^*, D_{ID^*}, PK_{ID^*}, S_{ID^*})$ 存在于 L_2 ，$\mathcal{B}$ 将该元组更新为 $\left(ID^*, D_{ID^*}, PK'_{ID^*}, \perp\right)$。

2 如果 L_2 中不包括该元组 $(ID^*, D_{ID^*}, PK_{ID^*}, S_{ID^*})$，$\mathcal{B}$ 添加一个新的元组 $(ID^*, \perp, PK'_{ID^*}, \perp)$ 到L_2中。

H_2-哈希询问。$\mathcal{A}_I$ 自适应地执行H_2-哈希询问， 对于 $\omega^* \in Z_q^*$。$\mathcal{B}$ 同样为H_2-哈希询问维持一个列表 $L_3 = \{(\omega, h_2)\}$ 。如果 L_3 中包含了 ω^*，$\mathcal{B}$ 检索 h_2^* 并返回 $g^{h_2^*}$ 给$\mathcal{A}_I$。否则,$\mathcal{B}$ 选择一个随机值 $h_2^* \in Z_q^*$ 并发送 $g^{h_2^*}$ 给$\mathcal{A}_I$。然后 $\mathcal{B}$ 将 (ω^*, h_2^*) 插入到 L_3。

标签询问。$\mathcal{A}_I$ 自适应地执行标签询问，使用(ω^*, m^*, ID^*)。$\mathcal{B}$ 首先检查是否 $T^* = 0$ 存在列表 L_1 中，对于ID^*。 如果 $T^* = 1$，$\mathcal{B}$ 终止。否则，$\mathcal{B}$ 从L_3中获得 $H_2(\omega^*)$ 、D_{ID^*} 并从L_2 中获得 S_{ID^*} 。然后$\mathcal{B}$ 通过 TagGen 算法计算(ω^*, m^*, ID^*) 的标签并返回给 $\mathcal{A}_I$。

伪造。最后，$\mathcal{A}_I$ 输出一个元组 $O = \left(T', \omega', m', ID', PK_{ID'}\right)$ 其中 T' 是由身份ID'使用公钥$PK_{ID'}$对块m' 伪造的标签。

分析。如果$\mathcal{A}_I$ 赢得游戏 I，$\mathcal{B}$ 可以根据等式（1）得到 $e(T', g) = e\left(H_1\left(ID'\right)^{m'}, P_0\right) \cdot e\left(H_2\left(\omega'\right), PK_{ID'}\right)$ 。$\mathcal{B}$ 在L_1中检索元组 $\left(ID', h_1', Q', T'\right)$。如果$T^* = 0$ ，$\mathcal{B}$ 终止并输出“失败”。否则， $\mathcal{B}$ 在L_1 中检索$H_1\left(ID'\right) = g^{bh_1'}$ 以及在L_3中检索 $H_2\left(\omega'\right) = g^{h_2'}$ 。 根据上面提到的验证等式，$\mathcal{B}$ 可以得到 $e(T', g) = e\left(g^{bh_1'm'}, g^a\right).\ e\left(g^{h_2'}, PK_{ID'}\right)$。 因此我们可以得到 $g^{ab} = \left(\frac{T'}{\left(PK_{ID'}\right)^{h_2'}}\right)^{1/h_1'm'}$ 。 现在，我们评估 $\mathcal{B}$ 输出正确的结果的概率。显然，如果 $\mathcal{B}$ 在上述的过程中不终止， $\mathcal{B}$ 和 $\mathcal{A}_I$进行了完美的交互。 另外我们可以知道 H_1-哈希询问、秘密值询问、公钥询问、公钥替换、 H_2-哈希询问都可以被无意外地完美执行。 $\mathcal{B}$ 的终止只发生在部分私钥询问和标签询问的过程中。因此$\mathcal{B}$ 与$\mathcal{A}_I$完美模拟交互过程而不终止的概率高于 $(1-\gamma)^{q_{k1}+q_T}$。因此，$\mathcal{B}$ 以$\varepsilon' \geq \varepsilon \cdot \gamma' \ (1-\gamma)^{q_{k1}+q_T} \geq \varepsilon/\left((q_{k1}+q_T) \cdot 2e\right)$的概率输出$g^{ab}$ 正确的值。相应的时间代价为 $t' \leq t + O\left(q_{H_1} + q_{k1} + q_{k2} + q_{k3} + q_{kr} + q_{H_2} + q_T\right)$ 。□

定理 2： 如果一个概率多项式时间敌手 $\mathcal{A}_{II}$ 在时间 t 内最多分别以$q_{H_1}, q_{k1}, q_{k2}, q_{H_2}, q_T$次发起$H_1$-哈希询问、秘密值询问、公钥询问、$H_2$-哈希询问和标签询问后，以$\varepsilon$ 的概率赢得节 2.5 中定义的游戏 II，那么就有一个模拟者$\mathcal{B}$ 可以在时间 $t' \leq t + O(q_{H_1} + q_{k1} + q_{k2} + q_{H_2} + q_T)$内，以 $\varepsilon' \geq \varepsilon/((q_{k1} + q_T) \cdot 2e)$ 的概率攻破 CDH 问题。

证明：给定一个 CDH 实例 $(g, \mathbb{G}_1, g^a, g^b)$。如果敌手$\mathcal{A}_{II}$ 以不可忽略的优势赢得游戏 II，模拟者 $\mathcal{B}$ 可以通过$\mathcal{A}_{II}$的能力以不可忽略的概率计算出g^{ab} 的值。at non-negligible probability by the capability of $\mathcal{A}_{II}$.$\mathcal{B}$ 模拟了与$\mathcal{A}_{II}$的每个交互步骤如下。

初始化：$\mathcal{B}$ 随机选择一个值 $s \in Z_q^*$ 作为主密钥并产生公开参数。 $\mathcal{B}$ 返回主密钥和公开参数给 $\mathcal{A}_{II}$。

H_1-哈希询问： $\mathcal{A}_{II}$ 自适应地对任意身份ID^*发起 H_1-哈希询问。$\mathcal{B}$ 为H_1-哈希询问维护一个列表 $L_1 = \{(ID, h_1)\}$ 。如果 ID^* 存在于L_1，$\mathcal{B}$ 检索元组 (ID^*, h_1^*) 并返回 $g^{h_1^*}$ 给$\mathcal{A}_{II}$。 否则，$\mathcal{B}$选择一个随机值 $h_1^* \in Z_q^*$ 并计算 $Q^* = g^{h_1^*}$。$\mathcal{B}$ 回复 Q^* 给$\mathcal{A}_{II}$ 并添加 (ID^*, h_1^*) 到L_1 中。

秘密值询问：因为 $\mathcal{A}_{II}$ 知道主密钥， $\mathcal{A}_{II}$ 不需要进行部分私钥询问。 $\mathcal{A}_{II}$ 自适应地对任意身份 ID^*做秘密值询问。$\mathcal{B}$ 维护一个秘密值询问的列表 $L_2 = \{(ID, PK_{ID}, S_{ID}, T)\}$ 。$\mathcal{B}$首先检查 ID^* 是否在 L_2中存在。

1 如果ID^* 不存在于L_2，$\mathcal{B}$ 随机选择一个值 $x^* \in Z_q^*$ 并抛掷一个硬币 $T^* \in \{0,1\}$。假设 $T^* = 0$ 的概率是 γ，则$T^* = 1$ 的概率是 $1 - \gamma$。如果 $T^* = 1$，$\mathcal{B}$ 计算$PK_{ID^*} = (g^a)^{x^*}$ 并添加元组 $(ID^*, PK_{ID^*}, x^*, T^*)$ 到L_2中。然后 $\mathcal{B}$ 输出“失败” 并终止。 如果 $T^* = 0$，$\mathcal{B}$ 计算$PK_{ID^*} = g^{x^*}$ 并添加 $(ID^*, PK_{ID^*}, x^*, T^*)$ 到L_2中。然后$\mathcal{B}$ 返回x^*给$\mathcal{A}_{II}$。

2 如果ID^* 存在于L_2，$\mathcal{B}$ 检查对应的值 T^*。如果 $T^* = 1$，$\mathcal{B}$ 输出“失败” 并终止。否则， $\mathcal{B}$ 检索 S_{ID^*} 并把它返回给 $\mathcal{A}_{II}$（当$T^* = 0$ 是S_{ID^*}是一定在L_2 中存在的）。

公钥询问： $\mathcal{A}_{II}$ 自适应地对任意身份 ID^*做公钥询问。

1 如果元组 $(ID^*, PK_{ID^*}, S_{ID^*}, T^*)$ 在 L_2 中存在，$\mathcal{B}$ 直接返回PK_{ID^*} 给$\mathcal{A}_{II}$ 。

2 如果L_2 中不包含元组 $(ID^*, PK_{ID^*}, S_{ID^*}, T^*)$，$\mathcal{B}$ 随机选择一个值 $x^* \in Z_q^*$ 并抛掷一个硬币 $T^* \in \{0,1\}$。假设 $T^* = 0$ 的概率是 γ，则$T^* = 1$ 的概率是 $1 - \gamma$。如果 $T^* = 1$，$\mathcal{B}$ 计算$PK_{ID^*} = (g^a)^{x^*}$。 如果$T^* = 0$，$\mathcal{B}$ 设置$S_{ID^*} = x^*$ 并计算$PK_{ID^*} = g^{x^*}$。$\mathcal{B}$ 添加一个新的元组 $(ID^*, PK_{ID^*}, x^*, T^*)$ 到L_2 中并返回 PK_{ID^*} 给$\mathcal{A}_{II}$。

H_2-哈希询问：$\mathcal{A}_{II}$ 自适应地对$\omega^* \in Z_q^*$发起 H_2-哈希询问。$\mathcal{B}$ 为H_2-哈希询问维护一个列表 $L_3 = \{(\omega, h_2)\}$ 。如果 L_3中包含了ω^*，$\mathcal{B}$检索元组h_2^* 并返回 $(g^b)^{h_2^*}$ 给$\mathcal{A}_{II}$。 否则，$\mathcal{B}$选择一个随机值 $h_2^* \in Z_q^*$ 并返回 $(g^b)^{h_2^*}$ 给$\mathcal{A}_{II}$。然后$\mathcal{B}$并添加 (ω^*, h_2^*) 到L_3中。

标签询问。$\mathcal{A}_{II}$ 自适应地执行标签询问，使用(ω^*, m^*, ID^*)。$\mathcal{B}$ 首先检查是否 $T^* = 0$ 存在列表 L_2 中，对于ID^*。 如果 $T^* = 1$，$\mathcal{B}$ 终止。否则，$\mathcal{B}$ 从L_3中获得 $H_2(\omega^*)$ 、D_{ID^*} 并从L_2中获得 S_{ID^*} 。然后$\mathcal{B}$ 通过 TagGen 算法计算(ω^*, m^*, ID^*) 的标签并返回给 $\mathcal{A}_I$。

伪造：最后，$\mathcal{A}_{II}$输出一个元组 $O = (T', \omega', m', ID')$ 其中 T' 是由身份ID'对块m' 伪造的标签。

分析： 如果$\mathcal{A}_{II}$赢得游戏 II，$\mathcal{B}$ 可以根据等式（1）得到 $e(T', g) = e\left(H_1(ID')^{m'}, P_0\right) \cdot e\left(H_2(\omega'), PK_{ID'}\right)$ 。然后$\mathcal{B}$ 在L_2中检索元组(ID', $PK_{ID'}, x', T'$) 。如果$T^* = 0$ ，$\mathcal{B}$ 终止并输出“失败”。否则， $\mathcal{B}$ 在L_1 中得到$H_1(ID') = g^{bh_1'}$ ，从L_2中得到 $PK_{ID'} = g^{ax'}$ 以及从L_3中得到$H_2(\omega') = g^{bh_2'}$。根据上面提到的等式，$\mathcal{B}$ 可以得到 $e(T', g) = e\left(g^{h_1'm'}, g^s\right) \cdot e\left(g^{bh_2'}, g^{ax'}\right)$。 因此我们可以得到 $g^{ab} = (T')^{1/(x'h_2'h_1's m')}$。 现在，我们评估 $\mathcal{B}$ 输出正确的结果的概率。显然，如果 $\mathcal{B}$ 在上述的过程中不终止， $\mathcal{B}$ 和 $\mathcal{A}_{II}$进行了完美的交互。另外我们可以知道 H_1-哈希询问、公钥询问、H_2-哈希询问都可以被无意外地完美执行。$\mathcal{B}$ 的终止只发生在秘密值询问和标签询问的过程中。因此$\mathcal{B}$ 与$\mathcal{A}_{II}$完美模拟交互过程而不终止的概率高于 $(1-\gamma)^{q_{k1}+q_T}$。因此，$\mathcal{B}$ 以 $\varepsilon' \geq \varepsilon \cdot \gamma \cdot (1-\gamma)^{q_{k1}+q_T} \geq \varepsilon/\left((q_{k1}+q_T) \cdot 2e\right)$的概率输出 g^{ab} 正确的值。相应的时间代价为 $t' \leq t + O\left(q_{H_1} + q_{k1} + q_{k2} + q_{H_2} + q_T\right)$。□

定理 3：如果离散对数假设成立， 敌手 $\mathcal{A}_{III}$ 只能以可忽略的概率赢得游戏 III 。

证明： 令挑战信息为 $chal = (c, k_1, k_2)$。如果 $\mathcal{A}_{III}$ 输出了完整性证据 $P' = (\overline{T'}, \overline{F'})$ 并以不可忽略的概率赢得游戏 III ，我们可以得到以下的验证等式：

$$e\left(\prod_{i=1}^{d'} \overline{T_i'}, g\right) = \prod_{i=1}^{d'} \left(e\left(\prod_{l \in C_i} H_2(\omega_l)^{a_l}, PK_i\right)\right) \cdot e\left(\prod_{i=1}^{d'} H_1(ID_i)^{\overline{F_i'}}, P_0\right)$$

其中 d' 表示挑战中的群组子集的数量。假设挑战信息 chal $= (c, k_1, k_2)$ 的真正证据是 $P = (\bar{T}, \bar{F})$，我们仍可以得到验证等式：$e\left(\prod_{i=1}^{d'} \overline{T_i}, g\right) = \prod_{i=1}^{d'} \left(e\left(\prod_{l \in C_i} H_2(\omega_l)^{a_l}, PK_i\right)\right) \cdot e\left(\prod_{i=1}^{d'} H_1(ID_i)^{\overline{F_i}}, P_0\right)$。因为 $\mathcal{A}_{III}$ 赢得了游戏 III，一定存在 $\bar{T} = \overline{T'}$ 而$\bar{F} \neq \overline{F'}$。根据上面的两个等式，有 $\prod_{i=1}^{d'} H_1(ID_i)^{\overline{F_i'}} = \prod_{i=1}^{d'} H_1(ID_i)^{\overline{F_i}}$。定义$\Delta\overline{F_i} = \overline{F_i'} - \overline{F_i}$ 对于每一个 $1 \leq i \leq d'$，我们知道 $\prod_{i=1}^{d'} \left(H_1(ID_i)\right)^{\overline{\Delta F_i}} = 1$。基于这个结论，离散对数问题可以被如下解决。给定两个元素 $h, u \in \mathbb{G}_1$ 其中 $u = h^x$，我们将计算出 $x \in Z_q^*$。令 $H_1(ID_i) = \chi_i = h^{\alpha_i} u^{\beta_i}$，其中$\alpha_i$和$\beta_i$是从 Z_q^*中随机选择的。 我们可以得到下面的等式

$$\prod_{i=1}^{d'} \chi_i^{\overline{\Delta F_i}} = \prod_{i=1}^{d'} (h^{\alpha_i} u^{\beta_i})^{\overline{\Delta F_i}} = h^{\sum_{i=1}^{z} \alpha_i \overline{\Delta F_i}} u^{\sum_{i=1}^{z} \beta_i \overline{\Delta F_i}} = 1，\ u = h^{-\frac{\sum_{i=1}^{z} \alpha_i \overline{\Delta F_i}}{\sum_{i=1}^{z} \beta_i \overline{\Delta F_i}}}$$

然后我们可以得到$x = h^{\frac{\sum_{i=1}^{z}\alpha_i\overline{\Delta F_i}}{\sum_{i=1}^{z}\beta_i\overline{\Delta F_i}}}$。因为$\bar{F} \neq \bar{F}'$，只有有一个$\overline{\Delta F_i}\left(1 \leq i \leq d'\right)$ 不为 0。 $\beta_i(1 \leq i \leq d')$ 是一个 Z_q^*中的随机值， 所以 $\sum_{i=1}^{z}\beta_i\overline{\Delta F_i} = 0$ 的概率只有 $1/q$。因此，我们可以以不可忽略的概率 $1 - 1/q$输出x 正确的值。

4.2 错误检测概率

假设 CSP 上n个块的p被篡改。CSP 随机选择$p(c_1 \leq p)$ 中的c_1个块来生成证据。为了检测数据损坏，必须满足 $c_1 \geq 1$。因此，错误检测的概率就等于 $c_1 \geq 1$ 的概率。 令P_a 表示错误检测的概率，我们有 $P_a = P\{c_1 \geq 1\} = 1 - P\{c_1 < 1\} = 1 - P\{c_1 = 0\} = 1 - \frac{n-p}{n} \cdot \frac{n-p-1}{n-1} \ldots\ldots \frac{n-p-c+1}{n-c+1}$。它表示: $1 - \left(1 - \frac{p}{n}\right)^c \leq P_a \leq 1 - \left(1 - \frac{p}{n-c+1}\right)^c$。从上面的式子中我们可以看到，更多的挑战块会导致更高的错误检测概率。通过对文献 [5]的分析，对于有 1%篡改块的文件，300 个挑战块将使 $P_a \geq 95\%$ 而 460 个挑战块会使 $P_a \geq 99\%$。因此，我们的方案可以实现较高的错误检测概率。

参考文献

[1] Dropbox for Business. [Online]. Available: https://wwW dropbox.com/business, Accessed on: Sep. 16, 2016.

[2] TortoiseSVN. [Online]. Available: https://tortoisesvn.net/, Accessed on: Sep. 16, 2016 .

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Gener. Comp. Syst., vol. 25, no. 6, pp. 599-616, 2009

[4] Y. Deswarte, J. J. Quisquater, and A. Saïdane, "Remote integrity checking," in Proc. 6 th Working Conf. Integr. Internal Control Inf. Syst., 2003, pp. 1-11.

[5] G. Ateniese, et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 598609.

[6] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int'l Conf. Security Privacy Commun. Netw., 2008, pp. 1-10.

[7] F. Sebé, J. Domingo-Ferrer, A. Martinez-balleste, Y. Deswarte, and J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. Knowl. Data Eng., vol. 20, no. 8, pp. 1034-1038, Aug. 2008.

[8] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession,'" in Proc. 16th ACM Conf. Comput. Commun. Security, 2009, pp. 213-222.

[9] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847-859, May, 2011.

[10] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy preserving public auditing for secure cloud storage," IEEE Trans. Comput., vol. 62, no. 2, pp. 362-375, Feb. 2013. [11] L. Chen, S. Zhou, X. Huang and L. Xu, "Data dynamics for remote data possession checking in cloud storage," Comput. Elect. Eng., vol. 39, no. 7, pp. 2413-2424, 2013

[12] Y. Yu, Y. Zhang, J. Ni, M. H. Au, L. Chen, and H. Liu, "Remote data possession checking with enhanced security for cloud," Future Generation Comput. Syst., no. 52, pp. 77-85, 2015

[13] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717-1726, Sep. 2013.

[14] H. Yan, J. Li, J. Han, and Y. Zhang, "A novel efficient remote data possession checking protocol in cloud storage," IEEE Trans. Inf. Foren. Sec., vol. 12, no. 1, pp. 78-88, Jan. 2017.

[15] Y. Feng, Y. Mu, G. Yang, and J.K Liu, "A new public remote integrity checking scheme with user privacy," in Proc 20th Australasian Conf. Inf. Security Privacy, 2015, pp. 377-394.

[16] Y. Zhu, H. Hu, G. J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231-2244, Dec. 2012.

[17] H. Wang, "Identity-based distributed provable data possession in Multicloud storage," IEEE Trans. Service Comput., vol. 8, no. 2, pp. 328-340, Mar./Apr. 2015 .

[18] H. Wang, D. He, J. Yu, and Z. Wang, "Incentive and unconditionally anonymous identity-based public provable data possession," IEEE Trans. Services Comput., vol. PP, no. 99, pp. 1-1. doi: 10.1109/ TSC.2016.2633260

[19] H. Wang, D. He, and S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud,"' IEEE Trans. Inf. Foren. Sec., vol. 11, no .6, pp. 1165-1176, Jun. 2016

[20] B. Wang, B. Li, H. Li, and F. Li, "Certificateless public auditing for data integrity in the cloud," in Proc. IEEE Conf. Commun. Network Security, 2013, pp. 136-144.

[21] H. Wang and J. Li, "Private certificate-based remote data integrity checking in public clouds," in Proc. 21th Int. Comput. Combinatorics, 2015, pp. 575-586.

[22] B. Wang, B. Li, and H. Li, "Knox: Privacy-preserving auditing for shared data with large groups in the cloud," in Proc. 10th Int. Conf. Applied Cryptography Netw. Security, 2012, pp. 507-525.

[23] B. Wang, H. Li, and M. Li, "Privacy-preserving public auditing for shared cloud data supporting group dynamics," in Proc. IEEE Int. Conf. Commun., 2013, pp. 1946-1950.

[24] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure multi-owner data sharing for dynamic groups in the cloud," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1182-1191, Jun. 2013

[25] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 43-56, Jan.-Mar. 2014.

[26] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," IEEE Trans. Service Comput., vol. 8, no. 1, pp. 92-106, Jan./Feb. 2015.

[27] Y. Yu, Y. Mu, J. Ni, J. Deng, and K. Huang, "Identity privacy-preserving public auditing with dynamic group for secure mobile cloud storage," in Proc. 8th Int. Conf. Netw. Syst. Security, 2014, pp. 28-40.

[28] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," IEEE Trans. Inf. Forensics Security, vol. 10, no. 8, pp. 1717-1726, Aug. 2015.

[29] A. Juels and B. S. Kaliski Jr., "PORs: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 584 – 597.

[30] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. 14th Int. Conf.

Theory Appl. Cryptology Inf. Security, 2008, pp. 90 – 107

[31] K. D. Bowers, A. Juels, and A. Oprea, "Hail: A high-availability and integrity layer for cloud storage," in Proc. 16th ACM Conf. Comput. Commun. Security, 2009, pp. 187-198

[32] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in Proc. 6th Theory Cryptography Conf., 2009, pp. 109-127.

[33] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in Proc. 9th Int. Conf. Theory Appl. Cryptology and Inf. Security, 2003, pp. 452-473.

[34] D. Boneh, H. Shacham, and B. Lynn, "Short signatures from the weil pairing,"]. Cryptology, vol. 17, no. 4, pp. 297-319, Sep. 2004.

[35] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and finegrained attribute-based data storage in cloud computing," IEEE Trans. Service Comput., vol. 10, no. 5, pp. 785-796, Sep./Oct. 2017. [36] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," IEEE Syst. J., 2017, doi: 10.1109/JSYST.2017.2667679.

[37] H. Wang, D. He, and J. Han, "VOD-ADAC: Anonymous distributed fine-grained access control protocol with verifiable outsourced decryption in public cloud," IEEE Trans. Services Comput., vol. PP, no. 99, pp. 1-1. doi: 10.1109/TSC.2017.2687459

[38] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage," IEEE Trans. Service Comput., vol. 10, no. 5, pp. 715-725, Sep.-Oct. 2017.

[39] J. Li, Y. Shi, and Y. Zhang, "Searchable ciphertext-policy attributebased encryption with revocation in cloud Storage,"Int. J. Com mun. Syst., vol. 30, no. 1, 2017, Art. no. e2942.

[40] J. Li, H. Wang, Y. Zhang, and J. Shen, "Ciphertext-policy attributebased encryption with hidden access policy and testing," KSII Trans. Internet Inf. Syst., vol. 10, no. 7, pp. 3339-3352, 2016.

[41] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy preserving personal health record using multi-authority attribute-based encryption with revocation," Int. J. Inf. Security, vol. 14, no. 6, pp. 487-497, 2015.

[42] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with efficient verifiable outsourced decryption," IEEE Trans. Inf. Forensics Security, vol. 10, no. 10, pp. 2119-2130, Oct. 2015.

[43] J. Li, F. Sha, Y. Zhang, X. Huang, and J. Shen, "Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length," Security Commun. Netw., vol. 2017, 2017, Art. no. 3596205 , doi: 10.1155/2017/3596205.

[44] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," IEEE Trans. Services Comput., 2017, doi: 10.1109/TSC.2017.2710190.

[45] The Pairing-based Cryptography Library (PBC). [Online]. Available: https://crpto.stanford.edu/pbc/download.html, Accessed on: Sep. 16, 2016.

[46] The GNU Multiple Precision Arithmetic Library (GMP). [Online] Available: http:/ / gmplib.org/, Accessed on: Sep. 16, 2016.