抗密钥泄露的支持密态数据去重的完整性审计方案

张襄松[1]，李晨[2]，刘振华[2]
（1. 西安工业大学理学院，陕西 西安 710021；2. 西安电子科技大学数学与统计学院，陕西 西安 710071）

摘 要：针对云存储环境中密钥泄露、数据重复和完整性检验的问题，提出了一种支持密钥更新和密文数据去重的完整性审计方案。所提方案利用布隆过滤器实现了密态数据的客户端去重，且每一次密钥更新能保证更新结果不能由其余时间周期的密钥猜测得到。该方案首次解决了在支持密态数据去重的审计方案中密钥更新困难的问题。安全性分析表明，所提方案在随机预言机模型下基于计算性 Diffie-Hellman 困难问题假设具有强抗密钥泄露、机密性、可检测性以及认证标签和证明值的不可伪造性。

关键词：云存储；完整性审计；抗密钥泄露；密态数据；客户端去重
中图分类号：TN309
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2019076

Key-exposure resilient integrity auditing scheme with encrypted data deduplication

ZHANG Xiangsong[1], LI Chen[2], LIU Zhenhua[2]
1. School of Science, Xi'an Technological University, Xi'an 710021, China
2. School of Mathematics and Statistics, Xidian University, Xi'an 710071, China

Abstract: For the problems of key-exposure, encrypted data duplication and integrity auditing in cloud data storage, a public auditing scheme was proposed to support key update and encrypted data deduplication. Utilizing Bloom filters, the proposed scheme could achieve client-side deduplication, and guaranteed that the key exposure in one time period didn't effect the users' private key in other time periods. The proposed scheme could solve the conflict between key-exposure resilient and encrypted data deduplication in public auditing scheme for the first time. Security analysis indicates that the proposed scheme is strong key-exposure resilient, confidentiality, detectability, and unforgeability of authentication tags and tokens under the computation Diffie-Hellman hardness assumption in the random oracle model.

Key words: cloud storage, integrity auditing, key-exposure resilient, encrypted data, client-side deduplication

1 引言

云平台具有强大的计算能力和存储能力，个人或企业可以将各种复杂的任务外包给云平台进行处理[1]，而不再需要花费高昂的代价购置和维护软硬件。例如大数据云存储服务，用户将数据上传给云存储服务提供商（CSP, cloud service provider）存储管理，这大大减轻了用户的存储负担，方便用户随时随地访问云端数据[2]。但是，由于用户失去对云端数据的绝对控制权，不可避免地带来了一系列数据安全问题[3]，如数据完整性、重复性、密钥泄露等。

收稿日期：2018−10−09；**修回日期**：2019−01−22
通信作者：刘振华，zh_liu@mail.xidian.edu.cn
基金项目：国家自然科学基金资助项目（No.61807026, No.61472470）；国家重点研发计划基金资助项目（No.2017YFB0802000）；陕西省教育厅专项科研计划基金资助项目（No.17JK0362）
Foundation Items: The National Natural Science Foundation of China (No.61807026, No.61472470), The National Key Research Development Program of China (No.2017YFB0802000), The Scientific Research Plan Project of Education Department of Shaanxi Province (No.17JK0362)

为了自身经济利益，CSP 可能故意删除用户的一部分数据。即使 CSP 能够诚实地存储用户的数据，也不可避免地存在因软硬件故障而导致的数据损坏。当上述问题发生时，CSP 可能瞒报，声称数据仍完整正确地存储在云服务器上，因此，需要定期地对云端数据完整性进行审计检查。如果需要将数据下载到本地再进行验证，这显然是低效的，甚至在数据量较大时是不可行的。为此，Ateniese 等[4]结合随机采样技术和随机化数字签名中的同态认证标签，提出了数据持有性证明（PDP, provable data possession）概念——在不进行数据取回的前提下, 云服务器向用户证明云端存储数据的完整性。进而，Juels 等[5]采用检查点和纠错码的方法，提出了数据可恢复性证明（PoR, proof of retrievability）概念，不仅可以进行完整性审计，而且还能实现损坏数据的恢复。

相同的数据文件可能被不同用户上传到云服务器进行存储，产生多个文件副本，这不仅占用了云存储服务商的存储空间，也浪费了用户的带宽资源，因此需要解决数据重复存储的问题。通过数据去重来消除冗余的数据，相同文件只保存一个副本，可以大大减少用户上传带宽及服务器存储空间。熊金波等[6]综述了数据去重的研究进展。

在云存储环境中同时实现数据的完整性验证和去重功能[7-10]是十分有意义的，但是机械地将去重功能和完整性验证组合起来不可避免地会导致上传者的身份混淆，以及由此产生的数据安全问题。这是因为后续用户上传一个服务器中已有的文件时，在生成的标签中含有用户的身份，这会导致标签与服务器中相应的文件之前的标签不一致，使云服务器存储了许多额外的数据，并且与完整性验证工作造成冲突。为此，Yuan 等[11]提出了一种支持数据去重的公开审计方案，将用户端的计算和传输开销减小至常数级，且支持批量审计，但是该方案的数据去重仅限于文件级，不够灵活。随后，Li 等[12]设计了支持数据审计和安全去重的 SecCloud 方案，利用 MapReduce 云代替第三方审计者（TPA, third party auditor）完成数据的完整性审计，但是将数据直接发送给 MapReduce 云破坏了用户身份的隐私性，且该方案不能降低用户的带宽消耗。为了解决带宽问题，Kiraz[13]提出了一种支持公开审计的安全客户端去重方案，该方案具有隐私保护性质，但是在上传数据过程中，用户完成所有权证明（PoW, proof of ownership）挑战的计算量较大，效率较低。此外，Kiraz 所提方案中的密钥服务器可能会根据用户发送的消息恢复出数据的部分加密密钥。

另外，由于用户安全意识薄弱或疏于管理的原因，在完整性审计过程中用户的私钥可能会丢失或遭到窃取。一旦恶意云得到了用户的私钥就能通过伪造认证标签来隐藏数据丢失，甚至为了节省存储空间删除用户不常访问的数据而不被察觉。在传统的完整性验证方案中，如果用户私钥遭到泄露，则需要用户下载其上传的所有数据，更换新的公私钥对后重新上传，显然这种方法效率极低，甚至是不可行的。因此，设计一种高效的抗密钥泄露的完整性审计方案也是十分必要的。Yu 等[14]基于二叉树结构构造了一个密钥更新算法，该算法保证了密钥泄露之前的时间周期内认证标签的安全性，即事前泄露。但从密钥泄露之后到发现泄露之前，该方案无法保证之后认证标签的安全性，这是由于在实际情况中密钥泄露通常只能在用户发现认证标签不是自己生成的时候被察觉，因此 Yu 所提方案的实用性较差。为了解决 Yu 所提方案中的事前泄露问题，Yu 等[15]又提出了强抗密钥泄露的云存储审计方案，使恶意的云不能通过已泄露时间周期的签名私钥获得其他时间周期用户的签名私钥，从而保护了除密钥泄露发生的时间周期外任意时间周期审计方案的安全性。

虽然现有数据完整性验证方案中已经实现了抗密钥泄露或数据去重功能，但目前尚没有同时支持这两项功能的数据完整性验证方案。如果只是单纯地再支持去重的审计方案中加入密钥更新，则可能会导致在不同时间周期由于用户私钥不同而无法成功去重。为解决这一问题，本文提出了一种具有密钥更新和密态数据去重的完整性验证方案。在本文方案中，采用收敛加密（convergent encryption）[12]方法保证文件的机密性；由第三方审计者协助用户进行密钥更新，能够保证用户在某一时间周期的私钥与其余时间周期的私钥相互独立，从而使得半可信的云服务器不能通过用户某一时间周期的私钥得到该用户其余时间周期私钥的任何信息。

2 预备知识

2.1 双线性映射

令 $\mathbb{G}_1$、$\mathbb{G}_2$ 表示阶为素数 p 的乘法循环群，g 表

示 $\mathbb{G}_1$ 的一个生成元，据此定义 $\hat{e}:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$ 为双线性映射[16]，该映射满足如下性质。

1) 双线性：对于任意的 $a,b\in\mathbb{Z}_p$，满足 $\hat{e}(g^a,g^b)=\hat{e}(g,g)^{ab}$。

2) 非退化性：存在 $u,v\in\mathbb{G}_1$，使 $\hat{e}(u,v)\neq 1$。

3) 可计算性：对于所有的 $u,v\in\mathbb{G}_1$，可高效地计算出 $\hat{e}(u,v)$。

2.2 CDH 困难问题假设

定义 1　对于任意的 $a,b\in\mathbb{Z}_p^*$，给出 (g,g^a,g^b)，则计算性 Diffie-Hellman（CDH）困难问题是计算 g^{ab}，挑战者 $\mathcal{C}$ 解决该困难问题的优势 ε 定义为

$$\Pr[g^{ab}\leftarrow\mathcal{A}(g,g^a,g^b)]\geqslant\varepsilon$$

如果没有多项式时间的算法能以至少 ε 的优势解决 CDH 困难问题，那么就称 CDH 困难问题假设成立。

2.3 布隆过滤器

布隆过滤器（Bloom filter）[17]作为一种概率数据结构，能够近似表示集合中的元素并验证某元素是否属于该集合。由于存储空间和插入/查询请求时间均为常数，布隆过滤器较现有的其他支持以上功能的数据结构具有存储时间与空间更为高效的优势。与此同时，为了取得时空高效，布隆过滤器不可避免地牺牲了一定的准确率，即存在误判。具体来说，误判的定义是若某一元素不在集合中，在利用布隆过滤器验证时也会以一定的概率将该元素判定为属于集合；但是，若该元素是集合中的元素，则不会发生误判。

布隆过滤器由 k 个随机的散列函数 $h_1(\cdot),h_2(\cdot),\cdots,h_k(\cdot)$ 和一个 m bit 的数组组成。初始化布隆过滤器时，需要将数组中的所有比特置为 0。若要向布隆过滤器中插入某一元素 x，则需计算该元素在布隆过滤器的数组中的 k 个位置 $a_1=h_1(x)\bmod m$，$a_2=h_2(x)\bmod m,\cdots,a_k=h_k(x)\bmod m$，并将数组中相应比特置为 1。为了确定某元素是否在集合中，需要计算它的 k 个散列值 $h_1',h_2',\cdots,h_k'$，然后验证数组中相应比特的值是否均为 1。在不考虑误判的情况下，该元素在集合中的充要条件是布隆过滤器数组中相应位置的值均为 1。此外，通过对布隆过滤器中误判的分析[18]，只要 k 选取适当的值，布隆过滤器的误判率可以很小。

2.4 系统描述

本文方案包含 4 个实体：云服务器、用户、第三方审计者（TPA）和密钥服务器（KS, key server），系统模型如图 1 所示。

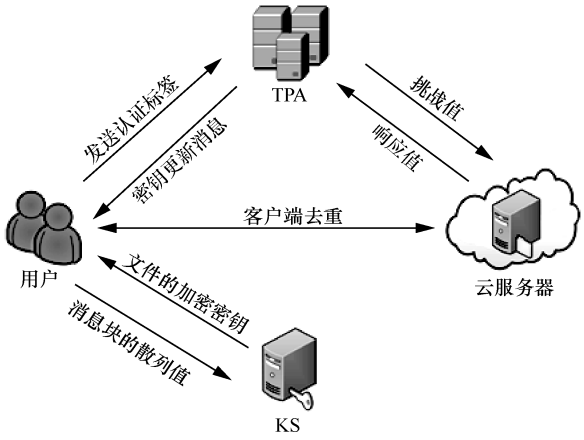


图 1　本文方案系统模型

在每个时间周期开始时，TPA 通过向用户发送密钥更新消息来协助用户更新文件的签名私钥，从而达到抗泄露的目的。TPA 在完整性审计时是诚实的，但在密钥更新时可能是不完全可信的。此外，在数据去重的过程中，用户需要配合云服务器验证是否需要将本地数据上传至云服务器，或是在云服务器的协助下进行 PoW 挑战而不需要上传数据本身。当用户需要对云服务器中存储的数据进行完整性审计时，首先需要将数据的认证标签发送给 TPA 验证其有效性，若能通过验证则 TPA 与云服务器执行挑战与响应协议。密钥服务器负责根据文件内容向用户分发私钥用于加密数据，用户需要从密钥服务器获取收敛密钥完成对文件进行加密。

2.5 一般方案

参考现有的支持密钥更新的公开审计方案[15]和密文数据去重的公开审计方案[12,18-19]，支持密钥更新和密文数据去重的公开审计方案通常包含以下 5 个算法。

密钥生成（KeyGen）算法：利用密钥生成算法，产生用户 u_i 的公私钥对 $(\mathrm{sk}_i,\mathrm{pk}_i)$、系统公开参数 pp、TPA 的私钥 $\mathrm{sk}_{\mathrm{TPA}}$、密钥服务器的收敛密钥种子 ks 及用于加密文件的加密密钥 ck。

更新消息生成（UMGen）算法：在每一时间周期开始时，由 TPA 输入系统公开参数 pp、当前时间周期 t 和 TPA 的私钥 $\mathrm{sk}_{\mathrm{TPA}}$，并输出更新消息 δ_t。

文件上传（FileUpload）算法：若用户 u_i 需要上传索引为 id_F 的文件 F，首先需要将文件 F 分为 n 块，计算文件 F 的散列值 $H(F)$ 并将其发送给云

服务器，判断云服务器中是否已经存储了文件 F，判断结果分 2 种。

情形 1 若云服务器中没有存储文件 F，则需要通知用户 u_i 上传数据。用户 u_i 收到通知后利用当前时间周期 t、系统公开参数 pp、用户自己的私钥 sk_i 和 TPA 发送的更新消息 δ_t 来计算时间周期 t 的签名私钥 sk_t，并利用密钥服务器生成的加密密钥对每个消息块 m_j 加密得到相应的密文 ct_j。随后用户 u_i 生成文件 F 的一组认证标签 Φ，将文件 F 的 n 个消息块 $m_j(j=1,\cdots,n)$ 分别存储到布隆过滤器 BF_j 中，并将文件 F 的所有密文集合 $(\mathrm{id}_F,\{\mathrm{ct}_j \mid j=1,\cdots,n\})$、索引值 id_j 和布隆过滤器 BF_j 一同发送给云服务器。

情形 2 若云服务器中已经存储了文件 F，则云服务器向用户 u_i 发出验证消息 K，用户 u_i 收到之后生成一组证明值，这时云服务器就可以通过判断这些证明值是否在布隆过滤器 BF_j 中来确认用户 u_i 是否确实拥有文件 F。

响应值生成（ResGen）算法：云服务器执行该算法。输入系统公开参数 pp、时间周期 t、挑战值 chal、文件 F 的密文集合 $(\mathrm{id}_F,\{\mathrm{ct}_j \mid j=1,\cdots,n\})$ 和文件 F 的一组认证标签 Φ，输出响应值 P。其中，云服务器输入的 (t,chal) 由 TPA 发布。

响应值验证（ResVerify）算法：TPA 执行该算法。输入系统公开参数 pp、时间周期 t、挑战值 chal 和响应值 P。若验证通过，返回“真”；否则，返回“假”。

2.6 安全模型

用户的隐私信息包含两部分：一部分是用于在时间周期 t 内生成消息认证标签的用户 u_i 的签名私钥 sk_t；另一部分是用于在时间周期 t 内生成签名私钥的用户 u_i 的私钥 sk_i。基于 Yu 等[15]的支持强抗密钥泄露的公开审计方案，本节定义了敌手 $\mathcal{A}$ 和挑战者 $\mathcal{C}$ 之间的安全性游戏。在游戏中，$\mathcal{A}$ 强大到甚至可以询问用户 u_i 在除一个时间周期之外的所有时间周期用户的私钥。具体安全模型如下。

1) 系统建立（setup）：首先令时间周期 $t=0$，$\mathcal{C}$ 执行 KeyGen 算法，生成系统公开参数 pp、TPA 的私钥 $\mathrm{sk}_{\mathrm{TPA}}$ 和用户 u_i 的私钥 sk_i。$\mathcal{C}$ 将 pp 发送给 $\mathcal{A}$，然后保留 sk_i。

2) 询问（query）：$\mathcal{C}$ 生成用户在时间周期 t 的签名私钥 sk_t。

① 认证标签询问：$\mathcal{A}$ 适应性地选择一系列消息 $m_1,\cdots,m_n$，并发送给 $\mathcal{C}$ 来询问时间周期 t 对应的认证标签。$\mathcal{C}$ 计算并返回该时间周期中消息 $m_i(i=1,\cdots,n)$ 的认证标签。

② 私钥询问：$\mathcal{A}$ 选择是否询问时间周期 t 内用户 u_i 的私钥，若是，$\mathcal{C}$ 将用户在时间周期 t 的私钥 sk_i 和签名密钥 sk_t 发送给 $\mathcal{A}$；否则，发送 $\perp$。

每个时间周期结束时，$\mathcal{A}$ 可选择再次询问或进行挑战。

3) 挑战（challenge）：$\mathcal{C}$ 选择一个 $\mathcal{A}$ 在私钥询问阶段中没有询问过的时间周期 t^*，并将挑战值 $\mathrm{chal}=\{\mathrm{id}_j,v_j\}_{\mathrm{id}_j\in\mathcal{I}}$ 发送给 $\mathcal{A}$，其中索引集合 $\mathcal{I}=\{\mathrm{id}_1,\mathrm{id}_2,\cdots,\mathrm{id}_s\}$，$1\leqslant s\leqslant n$，且集合中的元素均属于 $\{1,2,\cdots,n\}$。$\mathcal{C}$ 向 $\mathcal{A}$ 请求时间周期 t^* 下根据挑战值 chal 生成的响应值 P。

4) 伪造（forgery）：$\mathcal{A}$ 输出响应值 P，若验证通过，则 $\mathcal{A}$ 赢得游戏。

在上述安全模型中，$\mathcal{A}$ 在没有拥有挑战值 chal 中涉及的所有消息的情况下，若不能猜测出所有的消息的内容，则无法提供私钥没有泄露的时间周期的有效响应。在每个时间周期内 $\mathcal{A}$ 均可询问所有消息的认证标签。$\mathcal{A}$ 还可询问除挑战阶段的时间周期外所有时间周期的用户私钥。$\mathcal{A}$ 的目的是构造时间周期 t^* 的有效响应值 P。本文定义一个数据完整性审计方案的强抗密钥泄露，即存在一个提取器能够在 $\mathcal{A}$ 生成时间周期 t^* 的有效响应值 P 时提取出挑战的消息。

在实际情况下，许多用户对云服务器并不完全相信，为了保证用户数据的机密性，还需要在上传文件前对其进行加密处理。然而如果采用一般的加密方法，在去重过程中会导致不同用户利用不同密钥加密相同的文件，从而导致方案无法实现去重，因此需要对文件采用特殊的加密方法。文件的机密性要求阻止云服务器获取文件内容的攻击。特别地，要求方案能够抵抗字典攻击，也就是说，即使半可信的云服务器预先获得了包含所有可能文件及其密文的“字典”，仍旧无法恢复出目标文件。

在密钥生成算法中，TPA 在每一时间周期均生成更新消息来帮助用户更新私钥。进一步要求 TPA 不能根据自身私钥和生成的认证消息伪造任何认证标签，也就是认证标签的不可伪造性。

在文件上传算法过程中，考虑到有些恶意用户

可能会通过伪造消息的证明值来试图利用 PoW 挑战获取合法用户上传的文件，因此，本文方案的安全性还需要达到 PoW 挑战中证明值的不可伪造性。如果恶意用户需要通过 PoW 挑战来获得文件 F，假设该恶意用户已经拥有文件 F 的散列值和 F 的部分密文，那么恶意用户就可以伪造剩余部分的证明值从而达到通过 PoW 挑战的目的。本文方案不考虑恶意用户已经拥有绝大部分密文的情况，因为这种情况下恶意用户将不再需要伪造证明值。此外，本文方案利用收敛加密[7]的方法对文件 F 的 n 个消息块进行了加密，在确保文件加密密钥确定性的同时保证了文件的机密性，因此恶意用户即使已经获得了文件 F 的全部密文，也很难恢复出文件 F 对应的明文。

3　抗密钥泄露和支持去重的完整性审计方案

在实现抗密钥泄露时，本文方案中每一时间周期用户的签名私钥均为两部分的乘积，一部分是 TPA 根据自身私钥生成的更新消息，另一部分是由用户的私钥和当前时间周期计算而来的。任何时间周期的签名私钥都需要用户和 TPA 共同生成，这种方法能同时保证密钥更新的安全性和高效性，因此在没有 TPA 私钥的情况下，即使攻击者在某一时间周期入侵了用户，该攻击者也不能获得用户在其余时间周期的签名私钥。由于时间周期是认证标签的一部分，且无法分离出来，因此相同消息在不同时间周期的认证标签是不同的。在许多已有的云存储审计方案[2]中，利用数字签名 SSig（secure signature）来保证文件索引 id_F 的完整性。本文方案同样采用这种方式来保证文件索引 id_F 和时间周期 t 的完整性，且签名 SSig 对应的公私钥对 (ssk,spk) 由用户生成。

定义 3 个抗碰撞散列函数 H、H_1 和 H_2，其中，$H:\{0,1\}^*\to\mathbb{G}_1$，$H_1:\{0,1\}^m\to\{0,1\}^{\mathrm{len}}$，$H_2:\{0,1\}^*\times\mathbb{G}_1\to\mathbb{G}_1$，其中，$m$ 和 len 分别表示消息和证明值的长度。$\mathrm{Prf}:\{0,1\}^{\mathrm{len}}\times\{0,1\}^*\to\{0,1\}^{\kappa}$ 和 $\mathrm{Prf}_1:\{0,1\}^*\times\mathbb{G}_1\to\{0,1\}^*$ 是伪随机函数[18]，其中，κ 是正整数。方案的具体构造如下。

1) 密钥生成：随机选择 $x_i\in\mathbb{Z}_p^*$ 并计算 g^{x_i}，得到用户 u_i 的公私钥对 $(\mathrm{sk}_i,\mathrm{pk}_i)=(x_i,g^{x_i})$。TPA 随机选择私钥 $\mathrm{sk}_{\mathrm{TPA}}\in\mathbb{Z}_p^*$，并计算对应的公钥 $\mathrm{pk}_{\mathrm{TPA}}=g^{\mathrm{sk}_{\mathrm{TPA}}}$。系统公开参数为 $\mathrm{pp}=(g,\omega,\mathrm{pk}_{\mathrm{TPA}},\mathrm{pk}_i)$。密钥服务器随机选择一个收敛密钥种子 ks 用于协助用户生成文件加密密钥，并为每个用户分配一个私钥 ck 用于加密文件的加密密钥。

2) 更新消息生成：在时间周期 t 开始时，TPA 利用私钥 $\mathrm{sk}_{\mathrm{TPA}}$ 计算时间周期 t 的更新消息 $\delta_t=H(t)^{\mathrm{sk}_{\mathrm{TPA}}}$ 并发送给用户。用户接收到更新消息 δ_t 之后，利用 $\hat{e}(g,\delta_t)=\hat{e}(\mathrm{pk}_{\mathrm{TPA}},H(t))$ 验证消息的有效性。

3) 文件上传：若用户 u_i 在时间周期 t 内上传文件 F，首先需要将文件分为 n 个消息块 $m_1,m_2,\cdots,m_n$，其中，$m_j\in\mathbb{Z}_p^s,j=1,2,\cdots,n$，然后计算 $h_i=H(F)$ 并上传给云服务器，云服务器收到 h_i 之后判断 h_i 是否已存在。

情况 1　若云服务器中没有找到相同的 h_i，则说明云服务器中没有存储过文件 F，这时云服务器发送“No”给用户 u_i。用户 u_i 收到后利用更新消息 δ_t 计算 $\mathrm{sk}_t=H(t)^{\mathrm{sk}_i}\delta_t$，并与密钥服务器一起执行如下步骤对文件 F 进行加密并生成认证标签 Φ。

① 用户 u_i 计算每个消息块的散列值 $h_{m_j}=H(m_j)$，$j=1,\cdots,n$，并将所有计算结果 $(h_i,h_{m_1},h_{m_2},\cdots,h_{m_n})$ 发送给密钥服务器。

② 密钥服务器收到后对所有 $j=1,\cdots,n$，计算 $\mathrm{ks}_j=\mathrm{Prf}_1(\mathrm{ks},h_{m_j})$，其中，$\mathrm{ks}=(\mathrm{ks}_1,\cdots,\mathrm{ks}_n)$ 由密钥服务器保管。

③ 用户 u_i 加密每个消息块，计算 $\mathrm{ct}_j=\mathrm{Enc}(\mathrm{ks}_j,m_j)$，即得到文件 F 的密文 $(\mathrm{id}_F,\mathrm{ct}_1,\cdots,\mathrm{ct}_n)$，其中，$\mathrm{Enc}(\cdot)$ 是一个对称加密算法，id_F 为文件 F 的标识符。

④ 用户 u_i 用密钥服务器为其分配的私钥 ck 加密收敛密钥 ks_j，并存储在云服务器中。

⑤ 用户 u_i 随机选择 $r\in\mathbb{Z}_p^*$，计算 $R=g^r$，并由此计算每个消息块 m_j 的认证标签为

$$\sigma_j=H_2(t\|\mathrm{id}_j\|\mathrm{id}_F,R)^r\omega^{\mathrm{rct}_j}\mathrm{sk}_t$$

其中，id_j 和 id_F 分别为消息块 m_j 和文件 F 的标识符。

⑥ 用户 u_i 生成文件 F 的标签 $\mathrm{tag}_t=\mathrm{id}_F\|t\|\mathrm{SSig}_{\mathrm{ssk}}(\mathrm{id}_F\|t)$ 和认证标签集合 $\Phi=\{t,R,\sigma_1,\cdots,\sigma_n\}$。

用户 u_i 计算文件 F 中每个消息块 $m_j(j=1,\cdots,n)$ 相应的证明值 $T_j=H_1(m_j)$ 和伪随机值 $P_j=\mathrm{Prf}(T_j,\mathrm{id}_j)$，然后将每个伪随机值 P_j 插入布

隆过滤器 BF_F 中，并将布隆过滤器 BF_F 连同文件 F 的密文集合 $(\mathrm{id}_F,\mathrm{ct}_1,\cdots,\mathrm{ct}_n)$、文件标签 tag_t 和认证标签集合 Φ 一起上传给云服务器。云服务器计算 $H(F)$，验证认证标签的正确性以及 $h_i=H(F)$ 是否成立。若验证通过，云服务器将用户 u_i 上传的内容存储起来，并返回用户 u_i 文件 F 密文的链接和标签 tag_t；否则，云服务器返回出错消息。

情况 2 若云服务器中已经存储过文件 F，则需要用户 u_i 通过与云服务器的 PoW 挑战。首先，云服务器随机选择文件 F 中的 s 个密文消息块，并将这些消息块的索引集合 $K=\{k_1,\cdots,k_l\}(1\leqslant l\leqslant n)$ 发送给用户 u_i。用户 u_i 收到集合 K 后，计算集合中每个索引 $k_j(j=1,\cdots,l)$ 对应的证明值 $T_{k_j}=H_1(m_{k_j})$，然后将 $\{T_{k_j}\mid j=1,\cdots,l\}$ 返回给云服务器。对于所有云服务器选出的 l 个消息块，云服务器利用用户 u_i 返回的 $\{T_{k_j}\mid j=1,\cdots,l\}$ 计算其伪随机值 $P_{k_j}=\mathrm{Prf}(T_{k_j},\mathrm{id}_{k_j})$，并验证是否所有的 P_{k_j} 均属于布隆过滤器 BF_F。若属于，则向用户 u_i 发送文件 F 密文的链接和认证标签 tag_t；否则，返回出错消息。

4) 响应值生成：若用户 u_i 需要验证已上传文件 F 的完整性，则首先将文件 F 的认证标签 tag_t 发送给 TPA。TPA 收到后利用 spk 验证文件认证标签 tag_t 的有效性，若有效，则选择一个索引集合 $\mathcal{I}=\{\mathrm{id}_1,\mathrm{id}_2,\cdots,\mathrm{id}_s\}$，其中的每个元素均属于 $\{1,2,\cdots,n\}$。对于每个 $\mathrm{id}_j\in\mathcal{I}$，TPA 随机选择 v_j，并生成挑战值 $\mathrm{chal}=\{\mathrm{id}_j,v_j\}_{\mathrm{id}_j\in\mathcal{I}}$ 发送给云服务器。云服务器在收到挑战值 chal 后，计算

$$\sigma=\prod_{\mathrm{id}_j\in\mathcal{I}}\sigma_j^{v_j},\quad \mu=\sum_{\mathrm{id}_j\in\mathcal{I}}\mathrm{ct}_j v_j$$

并将响应值 $P=(t,R,\sigma,\mu)$ 返回给 TPA。

5) 响应值验证：TPA 收到响应值 P 后，验证

$$\hat{e}(g,\sigma)=\hat{e}(R,\prod_{\mathrm{id}_j\in\mathcal{I}}H_2(t\|\mathrm{id}_j\|\mathrm{id}_F,R)^{v_j}\omega^{\mu})\cdot \hat{e}(\mathrm{pk}_i\mathrm{pk}_{\mathrm{TPA}},H(t)^{\sum_{\mathrm{id}_j\in\mathcal{I}}v_j})$$

是否成立，若成立，返回“真”；否则，返回“假”。

4 安全性分析

4.1 方案的一致性

对于一个随机的挑战值 $\mathrm{chal}=\{\mathrm{id}_j,v_j\}_{\mathrm{id}_j\in\mathcal{I}}$ 和有效的证明值 $P=(t,R,\sigma,\mu)$，响应值验证算法总会返回“真”，因为有以下等式成立。

$$\hat{e}(g,\sigma)=$$
$$\hat{e}(g,\prod_{\mathrm{id}_j\in\mathcal{I}}\sigma_j^{v_j})=$$
$$\hat{e}(g,\prod_{\mathrm{id}_j\in\mathcal{I}}(H_2(t\|\mathrm{id}_j\|\mathrm{id}_F,R)^r\omega^{\mathrm{rct}_j}\mathrm{sk}_i)^{v_j})=$$
$$\hat{e}(g,\prod_{\mathrm{id}_j\in\mathcal{I}}(H_2(t\|\mathrm{id}_j\|\mathrm{id}_F,R)^r\omega^{\mathrm{rct}_j}H(t)^{\mathrm{sk}_i}\delta_i)^{v_j})=$$
$$\hat{e}(g,\prod_{\mathrm{id}_j\in\mathcal{I}}(H_2(t\|\mathrm{id}_j\|\mathrm{id}_F,R)^{v_j}\omega^{\sum_{\mathrm{id}_j\in\mathcal{I}}v_j\mathrm{ct}_j})^r H(t)^{(\mathrm{sk}_i+\mathrm{sk}_{\mathrm{TPA}})\sum_{\mathrm{id}_j\in\mathcal{I}}v_j})=$$
$$\hat{e}(g,(\prod_{\mathrm{id}_j\in\mathcal{I}}(H_2(t\|id_j\|\mathrm{id}_F,R)^{v_j}\omega^{\mu})^r)\hat{e}(g,H(t)^{(\mathrm{sk}_i+\mathrm{sk}_{\mathrm{TPA}})\sum_{\mathrm{id}_j\in\mathcal{I}}v_j})=$$
$$\hat{e}(R,\prod_{\mathrm{id}_j\in\mathcal{I}}H_2(t\|\mathrm{id}_j\|\mathrm{id}_F,R)^{v_j}\omega^{\mu})\hat{e}(\mathrm{pk}_i\mathrm{pk}_{\mathrm{TPA}},H(t)^{\sum_{\mathrm{id}_j\in\mathcal{I}}v_j})$$

若用户 u_i 需要取回存储在云服务器上的文件，那么在取回文件 F 的同时还需要下载事先已加密的收敛密钥 ks_j，并用私钥进行解密，从而得到相应的明文文件。

4.2 强抗密钥泄露

定理 1 若 $\mathbb{G}_1$ 上的 CDH 问题是困难的，则方案是强抗密钥泄露的。

证明 定义以下一系列安全性游戏，并分析敌手 $\mathcal{A}$ 在游戏中的行为差异。

游戏 0 游戏 0 是第 2.6 节中定义的安全性游戏。

游戏 1 游戏 1 基本与游戏 0 相同，不同之处在于，挑战者 $\mathcal{C}$ 维护一个所有经过他签名的认证标签列表。当 $\mathcal{A}$ 提交一个未通过 $\mathcal{C}$ 签名的有效标签时，$\mathcal{C}$ 中止游戏。

分析 若 $\mathcal{A}$ 在游戏 1 中使 $\mathcal{C}$ 中止游戏，则容易利用 $\mathcal{A}$ 构造一个攻击攻破签名方案 SSig。以下假设 id_F 和 t 在交互过程中均由 $\mathcal{C}$ 生成。

游戏 2 游戏 2 与游戏 1 基本相同。不同之处在于，$\mathcal{C}$ 维护一个对 $\mathcal{A}$ 认证标签询问的响应值列表。若 $\mathcal{A}$ 取得游戏胜利，而响应值 P 中的 R 与 $\mathcal{C}$ 维护的列表中不同，则 $\mathcal{C}$ 中止游戏。

分析 若 $\mathcal{A}$ 能够使 $\mathcal{C}$ 在游戏 2 中中止游戏，则可以构造一个模拟算法 $\mathcal{S}$ 以不可忽略的概率解决 CDH 困难问题。$\mathcal{S}$ 与游戏 1 中的 $\mathcal{C}$ 类似，但有以下不同：给予 $\mathcal{C}$ 一个 CDH 挑战 $(g,X=g^a,Y=g^b)$。$\mathcal{C}$ 通过 $\mathcal{A}$ 计算 g^{ab}，然后选择签名公私钥对 (ssk,spk) 生成 id_F 和 t 的签名。假设 $\mathcal{A}$ 进行了 q_k 次私钥询问和 q_s 次认证标签询问。

系统建立（setup）：$\mathcal{C}$ 随机选择 $\mathrm{sk}_i \in Z_p^*$，令 $\mathrm{pk}_{\mathrm{TPA}} = X$，$\mathrm{pk}_i = g^{\mathrm{sk}_i}$，再选择 $\alpha \in \mathbb{Z}_p^*$，并令 $\omega = g^{\alpha}$。然后 $\mathcal{C}$ 将公参 $\mathrm{pp} = (g, \omega, \mathrm{pk}_{\mathrm{TPA}}, \mathrm{pk}_i, \mathrm{spk})$ 发送给 $\mathcal{A}$。

询问（query）：将 H 和 H_2 看作 2 个由 $\mathcal{C}$ 控制和存储的随机预言机，$\mathcal{C}$ 需要对 $\mathcal{A}$ 发出的询问做出回答。

1) H 预言机询问：$\mathcal{C}$ 存储一个初始为空的 H 列表。当 $\mathcal{A}$ 向 $\mathcal{C}$ 发送时间周期 t 来询问 H 预言机时，$\mathcal{C}$ 查找 H 列表，检查是否有包含输入的时间周期 t 的元素 (t, c, λ, h)。若有，则向 $\mathcal{A}$ 发送 h；否则，$\mathcal{C}$ 掷一枚硬币 $c \in \{0,1\}$，满足结果为 0 的概率为 $\dfrac{p_k + p_s}{p_k + p_s + 1}$，结果为 1 的概率为 $\dfrac{1}{p_k + p_s + 1}$。若投掷结果为 0，$\mathcal{C}$ 随机选择 $\lambda \in \mathbb{Z}_p^*$，并计算 g^{λ}，然后将 $(t, 0, \lambda, h = g^{\lambda})$ 加入 H 列表；否则，$\mathcal{C}$ 随机选择 $\lambda \in \mathbb{Z}_p^*$ 并计算 Y^{λ}，然后将 $(t, 1, \lambda, h = Y^{\lambda})$ 加入 H 列表，最后，$\mathcal{C}$ 向 $\mathcal{A}$ 发送 h。

2) H_2 预言机询问：$\mathcal{C}$ 存储一个初始为空的 H_2 列表。当 $\mathcal{A}$ 向 $\mathcal{C}$ 发送 $(t \| \mathrm{id}_j \| \mathrm{id}_F, R)$ 进行询问时，$\mathcal{C}$ 查找 H_2 列表，检查是否有包含该输入的元素 $(t \| \mathrm{id}_j \| \mathrm{id}_F, R, \varphi, h_2)$。若有，则向 $\mathcal{A}$ 发送 h_2；否则，$\mathcal{C}$ 随机选择 $\varphi \in \mathbb{Z}_p^*$ 并将 $(t \| \mathrm{id}_j \| \mathrm{id}_F, R, \varphi, h_2 = g^{\varphi})$ 添加到 H_2 列表，最后，$\mathcal{C}$ 向 $\mathcal{A}$ 发送 h_2。

3) 私钥询问：当 $\mathcal{A}$ 在时间周期 t 内进行私钥询问时，$\mathcal{C}$ 从 H 列表中取出 $(t, c, \lambda, h = g^{\lambda})$，不失一般性，假设 $\mathcal{A}$ 向 H 预言机询问过时间周期 t。若 $c = 1$，则 $\mathcal{C}$ 中止（用 E_1 表示）；否则，$\mathcal{C}$ 计算 $\mathrm{sk}_t = X^{\lambda} h^{\mathrm{sk}_i}$，其中，$\mathrm{sk}_t = h^{\mathrm{sk}_{\mathrm{TPA}}} h^{\mathrm{sk}_i} = g^{\lambda \cdot \mathrm{sk}_{\mathrm{TPA}}} h^{\mathrm{sk}_i} = X^{\lambda} h^{\mathrm{sk}_i}$，最后，$\mathcal{C}$ 返回 $(\mathrm{sk}_i, \mathrm{sk}_t)$。

4) 认证标签询问：当 $\mathcal{A}$ 向 $\mathcal{C}$ 发送 (t, m_j, id_j) 来询问认证标签时，$\mathcal{C}$ 从 H 列表中取出 (t, c, λ, h)。若 $c = 1$，$\mathcal{C}$ 中止（用 E_2 表示）；否则，$\mathcal{C}$ 随机选择 $r \in \mathbb{Z}_p^*$ 和 $\sigma_j \in \mathbb{G}_1$，并计算 $R = g^r$。$\mathcal{C}$ 定义散列值 $H_2(t \| \mathrm{id}_j \| \mathrm{id}_F, R) = \dfrac{(\sigma_j X^{-\lambda} h^{-\mathrm{sk}_i})^{r^{-1}}}{\omega^{m_j}}$，这里有

$$H_2(t \| \mathrm{id}_j \| \mathrm{id}_F, R)^r \omega^{r m_j} \mathrm{sk}_t = \left(\frac{(\sigma_j X^{-\lambda} h^{-\mathrm{sk}_i})^{r^{-1}}}{\omega^{m_j}} \right)^r u^{r m_j} h^{\mathrm{sk}_{\mathrm{TPA}}} h^{\mathrm{sk}_i} = \sigma_j X^{-\lambda} (g^{\mathrm{sk}_{\mathrm{TPA}}})^{\lambda} = \sigma_j$$

最后，$\mathcal{C}$ 返回 (t, R, σ_j)。

挑战（challenge）：$\mathcal{C}$ 选择一个 $\mathcal{A}$ 没有进行过私钥询问的时间周期 t^*，然后向 $\mathcal{A}$ 发送一个挑战值 $\mathrm{chal} = \{\mathrm{id}_j, v_j\}_{\mathrm{id}_j \in \mathcal{I}}$ 和时间周期 t^*，其中 $\mathcal{I}$ 是消息块 $\{m_1, m_2, \cdots, m_n\}$ 索引的一个子集。此外，$\mathcal{C}$ 还要求 $\mathcal{A}$ 提供时间周期 t^* 文件 $F \to \{m_1, m_2, \cdots, m_n\}$ 在挑战值 $\mathrm{chal} = \{\mathrm{id}_j, v_j\}_{\mathrm{id}_j \in \mathcal{I}}$ 下的响应值 P。

伪造（forgery）：$\mathcal{A}$ 输出响应值 $P = (t^*, R, \sigma, \mu)$。$\mathcal{C}$ 从 H 列表中找出对应的 $(t^*, c^*, \lambda^*, h^*)$，若 $c^* = 0$，$\mathcal{C}$ 中止（用 E_3 表示）；否则，若 R 与认证标签集合中的 R 不同，则 $\mathcal{C}$ 从 H_2 列表中找到对应的 $(t^* \| \mathrm{id}_j \| \mathrm{id}_F,\ R, \varphi_j^*, h_2 = g^{\varphi_j^*})$，在这种情况下 $h^* = Y^{\lambda^*}$。如果 $\mathcal{A}$ 伪造的响应值是有效的，那么

$$\hat{e}(g, \sigma) = \hat{e}(R, \prod_{\mathrm{id}_j \in \mathcal{I}} H_2(t \| \mathrm{id}_j \| \mathrm{id}_F, R)^{v_j} \omega^{\mu}) \hat{e}(\mathrm{pk}_i \mathrm{pk}_{\mathrm{TPA}}, H(t)^{\sum_{\mathrm{id}_j \in \mathcal{I}} v_j}) = \hat{e}(R, g^{\sum_{\mathrm{id}_j \in \mathcal{I}} \varphi_j^* v_j} g^{\alpha\mu}) \hat{e}(g^{\mathrm{sk}_i} X, Y^{\lambda^* \sum_{\mathrm{id}_j \in \mathcal{I}} v_j}) = \hat{e}(g, R^{\alpha\mu + \sum_{\mathrm{id}_j \in \mathcal{I}} \varphi_j^* v_j}) \hat{e}(g, Y^{\lambda^* \mathrm{sk}_i \sum_{\mathrm{id}_j \in \mathcal{I}} v_j}) \hat{e}(X, Y^{\lambda^* \sum_{\mathrm{id}_j \in \mathcal{I}} v_j})$$

也就是说

$$\hat{e}(X, Y^{\lambda^* \sum_{\mathrm{id}_j \in \mathcal{I}} v_j}) = \hat{e}(g, \sigma) \hat{e}(g, R^{-(a\mu + \sum_{\mathrm{id}_j \in \mathcal{I}} \varphi_j^* v_j)}) \hat{e}(g, Y^{-\lambda^* \mathrm{sk}_i \sum_{\mathrm{id}_j \in \mathcal{I}} v_j})$$

若 $\sum_{\mathrm{id}_j \in \mathcal{I}} v_j = 0$，$\mathcal{C}$ 中止（用 E_4 表示）；否则，$\mathcal{C}$ 计算

$$g^{ab} = (\sigma R^{-(a\mu + \sum_{\mathrm{id}_j \in \mathcal{I}} \varphi_j^* v_j)} Y^{-\lambda^* \mathrm{sk}_i \sum_{\mathrm{id}_j \in \mathcal{I}} v_j} {}^{(\lambda^* \sum_{\mathrm{id}_j \in \mathcal{I}} v_j)^{-1}})$$

分析 $\mathcal{C}$ 在上述模拟算法中不中止的概率为

$$\Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3 \wedge \neg E_4] \geqslant \left(\frac{p_k + p_s}{p_k + p_s + 1} \right)^{p_k} \left(\frac{p_k + p_s}{p_k + p_s + 1} \right)^{p_s} \frac{1}{p_k + p_s + 1} \frac{p - 1}{p} = \left(\frac{p_k + p_s}{p_k + p_s + 1} \right)^{p_k + p_s} \frac{1}{p_k + p_s + 1} \frac{p - 1}{p} \geqslant \frac{p - 1}{e p (p_k + p_s + 1)}$$

其中，p_k 和 p_s 分别为 $\mathcal{A}$ 私钥询问和认证标签询问的数量。

如果 $\mathcal{A}$ 通过了挑战，但响应值 P 中的 R 与 $\mathcal{C}$ 从

H_2 列表中找到的 R 不相同，那么 $\mathcal{C}$ 能够以不可忽略的概率解决 CDH 困难问题。

游戏 3 游戏 3 与游戏 2 类似，不同之处在于 $\mathcal{C}$ 存储一个生成的 $\mathcal{A}$ 认证标签询问的返回结果列表。$\mathcal{C}$ 检查列表中的元素，若任何元素中有 $\mathcal{A}$ 虽通过挑战但 $\mathcal{A}$ 的认证标签 $\sigma \neq \prod_{\mathrm{id}_j \in \mathbb{I}} \sigma_j^{v_j}$，则 $\mathcal{C}$ 中止。

分析 假设使 $\mathcal{C}$ 中止的文件为 $F \to \{m_1, m_2, \cdots, m_n\}$，时间周期为 t，文件的认证标签集合为 $\Phi = \{t, R, \sigma_1, \cdots, \sigma_n\}$，挑战值为 $\{\mathrm{id}_j, v_j\}_{\mathrm{id}_j \in \mathcal{I}}$，$\mathcal{A}$ 的响应值为 $P' = (t, R, \sigma', \mu')$，$\mathcal{A}$ 诚实情况下生成的预期响应值 $P = (t, R, \sigma, \mu)$。因此，有效的响应值可由式(1)验证。

$$\hat{e}(g, \sigma) = \hat{e}(R, \prod_{\mathrm{id}_j \in \mathcal{I}} H_2(t \| \mathrm{id}_j \| \mathrm{id}_F, R)^{v_j} \omega^{\mu}) \cdot \hat{e}(\mathrm{pk}_i \mathrm{pk}_{\mathrm{TPA}}, H(t)^{\sum_{\mathrm{id}_j \in \mathcal{I}} v_j}) \tag{1}$$

根据 $\mathcal{C}$ 中止可以推断出 $\sigma' \neq \sigma$，且 σ' 能够通过如下等式的验证

$$\hat{e}(g, \sigma') = \hat{e}(R, \prod_{\mathrm{id}_j \in \mathcal{I}} H_2(t \| \mathrm{id}_j \| \mathrm{id}_F, R)^{v_j} \omega^{\mu}) \cdot \hat{e}(\mathrm{pk}_i \mathrm{pk}_{\mathrm{TPA}}, H(t)^{\sum_{\mathrm{id}_j \in \mathcal{I}} v_j}) \tag{2}$$

设 $\Delta\mu = \mu' - \mu$，显然有 $\Delta\mu \neq 0$，否则与假设矛盾。构造如下模拟算法 $\mathcal{S}$ 解决 CDH 困难问题。

向 $\mathcal{S}$ 输入 $(g, g^{a'}, v)$，最终输出 $v^{a'}$。$\mathcal{S}$ 代替游戏 2 中的 $\mathcal{C}$，但有以下不同：在系统建立阶段，$\mathcal{S}$ 选择 $\lambda, \eta \in \mathbb{Z}_p^*$，计算 $\omega = g^{\lambda} v^{\eta}$，并生成实际情况下所有时间周期的私钥。$H_2$ 可看作一个由 $\mathcal{S}$ 控制并存储的随机预言机，当 $\mathcal{A}$ 向 H_2 预言机发送 $(t \| \mathrm{id}_j \| \mathrm{id}_F, R)$ 时，$\mathcal{S}$ 验证 H_2 列表中是否已经存在相应的元素 $(t \| \mathrm{id}_j \| \mathrm{id}_F, R, h_2, \tau, \gamma)$。若有，$\mathcal{S}$ 返回 h_2 给 $\mathcal{A}$；否则，$\mathcal{S}$ 随机选择 $\tau \in \mathbb{Z}_p^*$，计算 $h = g^{\tau}$，将 $(t \| \mathrm{id}_j \| \mathrm{id}_F, R, h_2, \tau, *)$ 添加到 H_2 列表，并向 $\mathcal{A}$ 返回 h_2。

当 $\mathcal{A}$ 询问消息块 m_j 在时间周期 t 的认证标签时，$\mathcal{S}$ 随机选择 $\varsigma, r_j \in \mathbb{Z}_p^*$，计算 $R = (g^a)^{\varsigma}$ 和 $h_2 = \dfrac{g^{r_j}}{(g^{\lambda} v^{\eta})^{m_j}}$。$H_2(t \| \mathrm{id}_j \| \mathrm{id}_F, R)$ 与已有值碰撞的概率是可忽略的。$\mathcal{S}$ 计算认证标签

$$\sigma_j = H_2(t \| \mathrm{id}_j \| \mathrm{id}_F, R)^{a'\varsigma} \omega^{a'\varsigma m_j} \mathrm{sk}_t = \left(\frac{g^{r_j}}{(g^{\lambda} v^{\eta})^{m_j}}\right)^{a'\varsigma} (g^{\lambda} v^{\eta})^{a'\varsigma m_j} \mathrm{sk}_t = g^{r_j a'\varsigma} \mathrm{sk}_t$$

$\mathcal{S}$ 将 $(t \| \mathrm{id}_j \| \mathrm{id}_F, R, h_2, r_j, \varsigma)$ 添加到 H_2 列表。若 $\mathcal{A}$ 通过上传 $P' = (t, R, \sigma', \mu')$ 赢得游戏且 $\sigma' \neq \sigma$，则可以从 H_2 列表中提取 $(t \| id_j \| id_F, R, h_2, r_j, \varsigma)$。联立式(1)与式(2)，得到

$$\hat{e}\left(g, \frac{\sigma'}{\sigma}\right) = \hat{e}(R, \omega^{\Delta\mu}) = \hat{e}(R, (g^{\lambda} v^{\eta})^{\Delta\mu})$$

即

$$\hat{e}(g, \sigma' \sigma^{-1} R^{-\lambda\Delta\mu}) = \hat{e}((g^{a'})^{\varsigma}, v^{\eta\Delta\mu})$$

由此解决了 CDH 困难问题 $v^{a'} = (\sigma' \sigma^{-1} R^{-\lambda\Delta\mu})^{(\varsigma\eta\Delta\mu)^{-1}}$。

若 $\mathcal{A}$ 赢得游戏 2 和游戏 3 的概率有不可忽略的差别，则 $\mathcal{S}$ 能够解决 CDH 困难问题，所以在任何能够通过验证的响应值中 σ 必须是正确的。

游戏 4 游戏 4 与游戏 3 除以下区别外基本相同：$\mathcal{C}$ 存储并检查进行认证标签询问时给 $\mathcal{A}$ 的返回值。若存在 $\mathcal{A}$ 赢得游戏但 $\mathcal{A}$ 的聚合消息 $\mu' \neq \mu = \sum_{\mathrm{id}_j \in \mathcal{I}} m_j v_j$，则 $\mathcal{C}$ 中止游戏。

分析 假设 $\mathrm{chal} = \{\mathrm{id}_j, v_j\}_{\mathrm{id}_j \in \mathbb{I}}$ 是使游戏中止的询问，$P' = (t, R', \sigma', \mu')$ 是 $\mathcal{A}$ 上传的响应值，若 $\mathcal{A}$ 诚实情况下生成的预期响应值 $P = (t, R, \sigma, \mu)$。从游戏 2 和游戏 3 中可以看出 $R = R'$，$\sigma = \sigma'$，因此 P' 和 P 中不同的值为 μ' 和 μ。定义 $\Delta\mu = \mu' - \mu$，有 $\Delta\mu \neq 0$。若 $\mathcal{A}$ 能导致游戏 4 中止，则可以构造一个 $\mathcal{S}$ 解决离散对数问题。

$\mathcal{S}$ 随机选择 $g, v \in \mathbb{G}_1$，其目的是求得 x 满足 $v = g^x$。$\mathcal{S}$ 与游戏 3 中的挑战者 $\mathcal{C}$ 基本相同，不同之处在于在系统建立阶段，$\mathcal{S}$ 随机选择 $\lambda, \eta \in \mathbb{Z}_p^*$，令 $\omega = g^{\lambda} v^{\eta}$。$\mathcal{S}$ 与敌手 $\mathcal{A}$ 交互直到 $\mathcal{A}$ 成功返回不同于预期响应值 μ 的值 μ'。通过 μ' 和 μ 的验证式，可得

$$\hat{e}(R, \prod_{\mathrm{id}_j \in \mathcal{I}} H_2(t \| \mathrm{id}_j \| \mathrm{id}_F, R)^{v_j} \omega^{\mu}) \hat{e}(\mathrm{pk}_i \mathrm{pk}_{\mathrm{TPA}}, H(t)^{\sum_{\mathrm{id}_j \in \mathcal{I}} v_j}) = \hat{e}(g, \sigma) = \hat{e}(g, \sigma') = \hat{e}(R, \prod_{\mathrm{id}_j \in \mathcal{I}} H_2(t \| \mathrm{id}_j \| \mathrm{id}_F, R)^{v_j} \omega^{\mu'}) \hat{e}(\mathrm{pk}_i \mathrm{pk}_{\mathrm{TPA}}, H(t)^{\sum_{\mathrm{id}_j \in \mathcal{I}} v_j})$$

由此可得

$$\omega^{\mu} = \omega^{\mu'} \Rightarrow \omega^{\Delta\mu} = 1 \Rightarrow g^{\Delta\mu\lambda} v^{\Delta\mu\eta} = 1 \Rightarrow v = g^{-\frac{\lambda}{\eta}}$$

因此，若 $\mathcal{A}$ 在赢得游戏 3 和游戏 4 的概率之间存在一个不可忽略的差别，就可以构造一个模拟算法 $\mathcal{S}$ 解决离散对数问题。

根据本文分析，这些游戏之间仅存在可忽略的差别，且解决 CDH 问题的困难程度蕴含着解决离散对数问题的困难程度。若 $\mathbb{G}_1$ 上的 CDH 问题是困难的，且签名算法 SSig 是不可伪造的，则 $\mathcal{C}$ 除了 $\mathcal{A}$ 上传正确的响应值 P 的情况外将无法验证通过。

4.3　机密性

由于方案中引入了密钥服务器来帮助用户生成文件的收敛密钥，因此若不利用密钥服务器存储的收敛密钥种子，半可信的云服务器就不能以不可忽略的概率生成任何文件有效的收敛密钥。由于文件的散列值可以看作一个有效的消息认证码，因此云服务器在没有密钥服务器的帮助下就无法实施暴力攻击。此外，所有消息块在上传给云服务器前均已加密，且收敛密钥由密钥服务器生成，并由上传文件的用户公钥加密后存至云服务器的，也可以认为收敛密钥是由文件本身和密钥服务器的收敛密钥种子共同产生的，这就意味着收敛密钥并不仅仅源于文件本身。

假设半可信的云服务器不能与密钥服务器共谋，那么即使文件是可预测的，半可信的云服务器也不能利用暴力攻击推测出文件的内容，由此说明本文方案能够抵抗字典攻击。

4.4　可检测性

假设云服务器存储一个 n 块的文件，其中的 k 块均被损坏，在挑战阶段 TPA 的挑战值中包含 s 块，则损坏的块被发现的必要条件是当且仅当被挑战的块中至少一个是损坏的块。利用一个独立的随机变量 X 表示挑战阶段选择的块中损坏的块数量，用 P_X 表示在挑战阶段选择的块中至少有一个被损坏块的概率。因此，有 $P_X \geqslant 1-\left(\frac{n-k}{n}\right)^s$。也就是说，损坏的块被检测出的概率至少是 $1-\left(\frac{n-k}{n}\right)^s$，说明损坏的块越多，TPA 选出进行挑战的块越多，损坏的块被检测出的概率越大。

$$
\begin{aligned}
&P_X = P\{X \geqslant 1\} = \\
&1 - P\{X = 0\} = \\
&1 - \frac{n-k}{n}\frac{n-1-k}{n-1}\cdots\frac{n-s+1-k}{n-s+1}
\end{aligned}
$$

4.5　认证标签的不可伪造性

从文件上传算法中可以看出，时间周期 t 内计算任何认证标签均需要私钥 sk_t，而用户私钥的生成过程中私钥是由用户 u_i 的私钥 sk_i 和 TPA 的私钥 $\mathrm{sk}_{\mathrm{TPA}}$ 这两部分私钥构成的，具体来说，就是 $\mathrm{sk}_t = H(t)^{\mathrm{sk}_i}\delta_t = H(t)^{\mathrm{sk}_i}H(t)^{\mathrm{sk}_{\mathrm{TPA}}}$。由于方案中的 TPA 只拥有 $\mathrm{sk}_{\mathrm{TPA}}$ 且仅需要计算更新消息 δ_t，因此 TPA 不知道用户 u_i 的私钥 sk_i 也就不能计算出 sk_t。因此 TPA 不能通过自己的私钥 $\mathrm{sk}_{\mathrm{TPA}}$ 和更新消息 δ_t 伪造任何消息块的认证标签。

4.6　证明值的不可伪造性

在文件上传算法中，若某一用户需要向云服务器上传文件，在上传消息之前需要先发送文件的散列值来验证云服务器中是否已经存储该文件，若没有存储，则需要用户上传文件 F、文件标签 tag_t、认证标签集合 Φ 和消息的布隆过滤器 BF_F；若云服务器中已经存储过，则用户不再需要上传文件本身，而改为执行 PoW 挑战，目的在于避免恶意用户通过拥有文件的散列值而不是整个文件来恶意获取合法用户的数据。简单地说，本文方案利用布隆过滤器匹配消息的证明值来实现 PoW 挑战。

PoW 挑战要求用户生成云服务器随机选取的 k 个消息块的证明值，云服务器收到证明值后用伪随机函数处理并用相应的布隆过滤器验证是否属于选定的消息。假设恶意用户需要获得消息 m_j，若恶意用户拥有 m_j 的一些消息块，在进行 PoW 挑战时恶意用户需要生成所有云服务器选择的 k 个消息块的证明值。在 Blasco 等[18]的方案中已经得出，如果恶意用户仅仅拥有少数消息块，布隆过滤器的参数选取得当，且 k 足够大，那么恶意用户成功伪造消息证明值，从而通过 PoW 挑战的概率是可忽略的。

不能否认的是，若用户拥有绝大多数的消息块，则会以很大概率通过 PoW 挑战。在这种情况下，由于用户几乎已经拥有该消息，也就可以将这个用户看作一个合法的拥有者了。

5　性能分析与实验结果

本节从方案实现的功能和效率这 2 个方面进行分析。由于目前没有能同时支持抗密钥泄露和密文数据去重的完整性审计方案，本文方案在功能上更有优势。在方案的效率方面将从计算效率和通信效

率两部分进行评价。为便于表示，用 exp 、 mul 、 hash 和 pair 分别表示指数运算、乘法运算、散列运算和对运算。

在文件上传阶段，若云服务器中没有存储需要上传的文件，则用户需要计算签名私钥 sk_t 、文件标签 tag_t 、认证标签集合 $\boldsymbol{\Phi}$ 和布隆过滤器 BF_F ，计算开销为 $(n+1)\mathrm{hash}+2\mathrm{exp}+3\mathrm{mul}+\mathrm{sig}+\mathrm{prf}$ ，其中，sig 和 prf 分别为数字签名和伪随机函数；反之，若云服务器中已经存储过该文件，则用户需要生成 PoW 挑战的证明值，相应的计算开销为 $(l+1)\mathrm{hash}$ 。在挑战阶段，TPA 仅随机选择一些消息块的索引生成挑战值，因此计算开销很小。然而在响应阶段，云服务器生成响应值时的计算开销为 $s\exp+(s-1)\mathrm{mul}+s\ \mathrm{mul}$ ，其中 s 表示挑战值中选取消息块的个数。最后，在验证阶段，TPA 判断响应值是否有效，计算开销为 $(s+2)\exp+(s+2)\mathrm{mul}+3\mathrm{pair}$ 。表 1 分析了方案的总体计算开销。

表 1　支持密钥更新和密文数据去重的公开审计方案的计算开销

算法	云服务器中存储文件副本	云服务器中没有存储文件副本
密钥生成算法	2exp	2exp
更新消息生成算法	2hash+exp+2pair	2hash+exp+2pair
文件上传算法	$(l+1)\mathrm{hash}+l\ \mathrm{prf}$	$(n+4)\mathrm{hash}+(2n+2)\exp+(2n+1)\mathrm{mul}+\mathrm{sig}+\mathrm{prf}$
响应值生成算法	$s\exp+(s-1)\mathrm{mul}+s\ \mathrm{mul}$	$s\exp+(s-1)\mathrm{mul}+s\ \mathrm{mul}$
响应值验证算法	$(s+2)\exp+(s+2)\mathrm{mul}+3\mathrm{pair}$	$(s+2)\exp+(s+2)\mathrm{mul}+3\mathrm{pair}$

本文方案的通信开销主要包括文件上传、挑战阶段和响应阶段这 3 个部分。在文件上传的过程中，用户需要将 $\{F,t,\sigma_1,\cdots,\sigma_n,\mathrm{BF}_F\}$ 发送给云服务器，这部分的通信开销为 $|F|+(n+1)|q|$ ，其中，$|F|$ 和 $|q|$ 分别表示文件的大小和 $\mathbb{Z}_p$ 或 $\mathbb{G}_1$ 上一个元素的大小。挑战阶段挑战值 TPA 挑出的 s 个 (id_j,v_j) 组成，因此通信开销为 $s(|I|+|q|)$ ，其中，$|I|$ 为索引的长度。相应阶段响应值 (t,R,σ,μ) 的通信开销为 $3|q|$ 。

实验通过对方案的计算开销和通信开销进行比较，利用 PBC 库给出 Windows 7 环境下的效率分析结果，实验配置为 Intel Core I3-2120 CPU@3.30 GHz，4 GB RAM。假设 $\mathbb{G}_1$ 和 $\mathbb{Z}_p$ 上的元素大小均为 160 bit，消息块的大小为 2 KB，集合 $\mathcal{I}$ 中的元素大小为 20 bit，以下实验结果为 50 次实验的平均值。

图 2 显示了方案中用户、云服务器和 TPA 在云服务器中有副本和无副本这 2 种情况下的计算开销随文件块数和挑战块数取值的变化趋势。仿真实验选定文件的分块数量为 400，选取挑战块数分别为 50、100、150、200 和 250。图 2(a)表明当云服务器中无副本时，TPA 和云服务器的计算开销随着审计过程中挑战块数量的增加呈线性增长，而用户的计算开销只与文件分块的个数 n 有关，故用户的计算开销呈直线，且大于 TPA 和云服务器的计算开销。图 2(b)表明当云服务器中有副本时，云服务器承担了密文数据去重的大多数计算开销，随着审计过程中 TPA 挑战块数量的增加而呈线性增长，而用户和 TPA 的计算开销只与文件分块的个数 n 有关，故呈直线，且用户的计算开销小于 TPA 的计算开销。因此，在云服务器中无副本时，用户的计算开销较大；而有副本时，云服务器的计算开销较大。

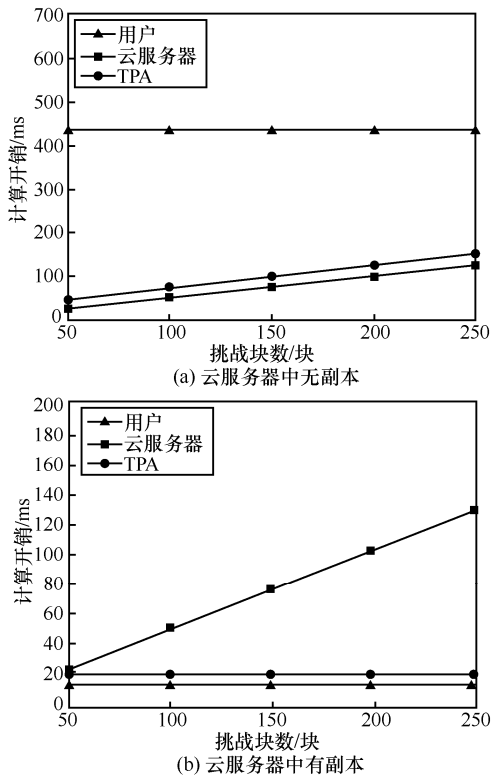


图 2　用户、云服务器和 TPA 计算开销的变化趋势比较

图 3 显示了方案中用户、云服务器和 TPA 三方在云服务器中有副本和无副本这 2 种情况下的通信开销随文件块数和挑战块数取值的变化趋

势。在图 3(a)中，选取文件分块数量分别为 250、300、350、400 和 450，结果表明，在云服务器中无副本时，用户的通信开销随着文件分块数量增加而线性增加，而 TPA 和云服务器的通信开销较小。在图 3(b)中，选定文件的分块数量 n 为 400，选取挑战块数分别为 50、100、150、200 和 250，仿真结果表明，在云服务器中有副本时，TPA 的通信开销随着审计过程中挑战的文件块数量的增加线性增长，而用户和云服务器的通信开销较小。因此，当云服务器中无副本时，用户的通信开销较大；而云服务器中有副本时，通信开销主要在审计过程中由 TPA 承担。

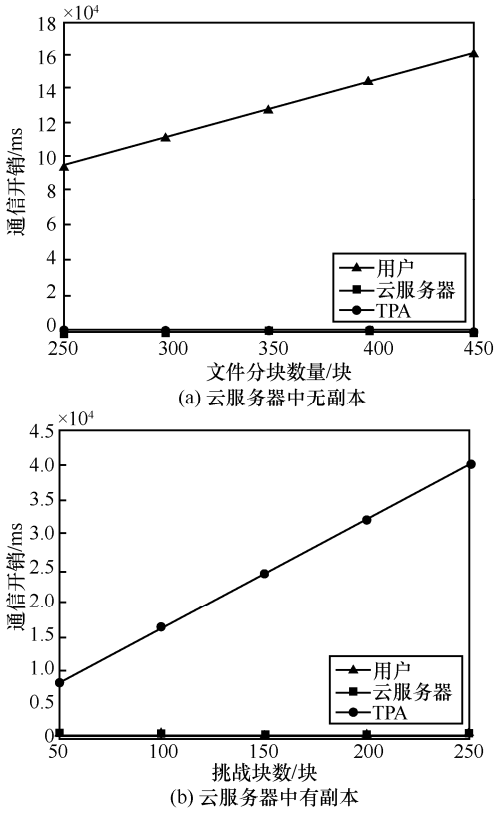


图 3　用户、云服务器和 TPA 通信开销的变化趋势比较

6　结束语

本文提出了同时支持密钥更新和密文数据去重的公开审计方案。新方案首次在支持数据去重的完整性审计方案中解决密钥泄露问题，且方案不仅考虑密钥泄露之前的安全性，还考虑到密钥泄露之后的用户数据机密性保护问题，使用户在某一时间周期内的私钥不会影响到其他时间周期，且敌手即使通过了 PoW 挑战也很难恢复出明文数据。安全性分析表明，新方案达到了强抗密钥泄露、机密性、可检测性以及认证标签和证明值的不可伪造性。下一步需要提高效率和进行全面形式化安全性证明。

参考文献：

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

[2] WANG Q, WANG C, LI J, et al. Enabling public verifiability and data dynamics for storage security in cloud computing[C]//The 14th European Symposium on Research in Computer Security. 2009: 355-370.

[3] 冯登国, 张敏, 张妍, 等. 云计算安全研究[J]. 软件学报, 2011, 22(1): 71-83.
FENG D G, ZHANG M, ZHANG Y, et al. Study on cloud computing security[J]. Journal of Software, 2011, 22(1): 71-83.

[4] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//The ACM Conference on Computer and Communications Security. ACM, 2007: 598-609.

[5] JUELS A, KALISKI B S. Pors: proofs of retrievability for large files[C]//The 14th ACM Conference on Computer and Communications Security. ACM, 2007: 584-597.

[6] 熊金波, 张媛媛, 李凤华, 等. 云环境中数据安全去重研究进展[J]. 通信学报, 2016, 37(11): 169-180.
XIONG J B, ZHANG Y Y, LI F H, et al. Research progress on secure data deduplication in cloud[J]. Journal on Communications, 2016, 37 (11): 169-180.

[7] 熊金波, 张媛媛, 田有亮, 等. 基于角色对称加密的云数据安全去重[J]. 通信学报, 2018, 39(5): 59-73.
XIONG J B, ZHANG Y Y, TIAN Y L, et al. Cloud data secure deduplication scheme via role-based symmetric encryption[J]. Journal on Communications, 2018, 39(5): 59-73.

[8] 郭晓勇, 付安民, 况博裕, 等. 基于收敛加密的云安全去重与完整性审计系统[J]. 通信学报, 2017, 38 (Z2): 156-163.
GUO X Y, FU A M, KUANG B Y, et al. Secure deduplication and integrity audit system based on convergent encryption for cloud storage[J]. Journal on Communications, 2017, 38(Z2): 156-163.

[9] 熊金波, 李素萍, 张媛媛, 等. 共享所有权证明：协作云数据安全去重新方法[J]. 通信学报, 2017, 38 (7): 18-27.
XIONG J B, LI S P, ZHANG Y Y, et al. PoSW: novel secure deduplication scheme for collaborative cloud applications[J]. Journal on Communications, 2017, 38(7): 18- 27.

[10] 杨超, 纪倩, 熊思纯, 等. 新的云存储文件去重复删除方法[J]. 通信学报, 2017, 38(3): 25-33.

YANG C, JI Q, XIONG S C, et al. New method for file deduplication in cloud storage[J]. Journal on Communications, 2017, 38(3): 25-33.

[11] YUAN J, YU S. Secure and constant cost public cloud storage auditing with deduplication[C]//The 1st IEEE Conference on Communications and Network Security. IEEE, 2013: 145-153.

[12] LI J, LI J, XIE D, et al. Secure auditing and deduplicating data in cloud[J]. IEEE Transactions on Computers, 2016, 65(8): 2386-2396.

[13] KIRAZ M S. Solving the secure storage dilemma: an efficient scheme for secure deduplication with privacy-preserving public auditing[EB]. IACR Cryptology ePrint Archive, 2016: 696.

[14] YU J, REN K, WANG C, et al. Enabling cloud storage auditing with key-exposure resistance[J]. IEEE Transactions on Information Forensics and Security, 2017, 10(6): 1167-1179.

[15] YU J, WANG H. Strong key-exposure resilient auditing for secure cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(8): 1931-1940.

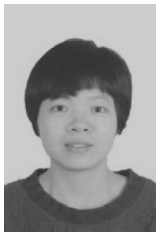
[16] CHATTERJEE S, SARKAR P. Identity-based encryption[M]. Boston: Springer Science & Business Media, 2011: 29-48.

[17] BLOOM B H. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM, 1970, 13(7): 422-426.

[18] BLASCO J, PIETRO R D, ORFILA A, et al. A tunable proof of ownership scheme for deduplication using Bloom filters[C]//The IEEE Conference on Communications and Network Security. IEEE, 2014: 481-489.

[19] WANG C, CHOW S S, WANG Q, et al. Privacy-preserving public auditing for secure cloud storage[J]. IEEE Transactions on Computers, 2013, 62(2):362-375.

[作者简介]



张襄松（1980− ），女，河南南阳人，博士，西安工业大学副教授，主要研究方向为信息科学中的安全与优化问题。

李晨（1992− ），女，河北石家庄人，西安电子科技大学硕士生，主要研究方向为密态数据去重与审计。

刘振华（1978− ），男，河南信阳人，博士，西安电子科技大学教授、硕士生导师，主要研究方向为密态数据的计算。