doi:10.11959/j.issn.1000-436x.2015267

云存储中数据完整性的聚合盲审计方法

何凯[1,2]，黄传河[1,2]，王小毛[1,2]，王晶[1,2]，史姣丽[1,2]
（1. 武汉大学 计算机学院，湖北 武汉 430072；2. 地球空间信息技术协同创新中心，湖北 武汉 430072）

摘 要：针对云存储中数据完整性问题，提出了一种聚合盲审计方法。利用双线性对映射的性质，在云服务器端将数据证据和标签证据加密后再合并，实现审计者在不知数据内容的情况下进行盲审计。在此基础上，设计高效的索引机制支持数据更新，使数据更新操作不会导致大量额外的计算和通信开销，实现了动态审计。同时，针对多个审计请求，设计将不同的证据聚合的方法，以支持对多所有者多云服务器多文件的批量审计，使批量审计的通信开销与审计请求的数量无关。理论分析和实验结果表明，该方法是可证明安全的，与现有的其他审计方案相比，所提的单审计和批量审计的效率分别提高了 21.5%和 31.8%。

关键词：云存储；数据完整性；盲审计；动态审计；批量审计
中图分类号：TP309 **文献标识码**：A

Aggregated privacy-preserving auditing for cloud data integrity

HE Kai[1,2], HUANG Chuan-he [1,2], WANG Xiao-mao[1,2], WANG Jing [1,2], SHI Jiao-li[1,2]
(1. Computer School, Wuhan University, Hubei 430072, China;
2. Collaborative Innovation Center of Geospatial Technology, Hubei 430072, China)

Abstract: To solve the problem of data integrity in cloud storage, an aggregated privacy-preserving auditing scheme was proposed. To preserve data privacy against the auditor, data proof and tag proof were encrypted and combined by using the bilinearity property of the bilinear pairing on the cloud server. Furthermore, an efficient index mechanism was designed to support dynamic auditing, which could ensure that data update operations did not lead to high additional computation or communication cost. Meanwhile, an aggregation method for different proofs was designed to handle multiple auditing requests. Thus the proposed scheme could also support batch auditing for multiple owners and multiple clouds and multiple files. The communication cost of batch auditing was independent of the number of auditing requests. The theoretical analysis and experimental results show that the proposed scheme is provably secure. Compared with existing auditing scheme, the efficacy of the proposed individual auditing and batch auditing improves 21.5% and 31.8% respectively.

Key words: cloud storage; data integrity; privacy-preserving auditing; dynamic auditing; batch auditing

1 引言

云存储是云计算的一种重要服务，允许数据所有者将其数据托管在云服务器中，并通过云服务器向用户提供数据访问[1,2]。通过这种数据的外置服务，可以给数据所有者带来诸多方便：1）减少存储管理的压力；2）减少存储硬件和软件以及数据维护的费用；3）可以实现任意地点、任意时间的数据访问。云存储在给数据所有者带来优势和便利的同时，也带来新的安全问题。数据所有者将数据存储在云中后，就失去了对数据的物理控制，这将导致数据的安全性高度依赖于云服务提供商。事实上云服务提供商是不能被完全信任的。首先，尽管云服务提供商能提供安全性更高的存储设备，但海量数据存储在云中使数据更容易遭受主动攻击，同时由于一些不可抗拒的客观原因造成数据的丢失[3,4]。其次，云

收稿日期：2014-11-05；**修回日期**：2015-08-10
基金项目：国家自然科学基金资助项目（61373040, 61173137）；湖北省自然科学基金资助项目（2010CDA004）
Foundation Items: The National Natural Science Foundation of China (61373040, 61173137)；The Natural Science Foundation of Hubei Province (2010CDA004)

服务提供商为了自身的利益，通过各种手段对数据所有者的数据采取不可信的行为。如为了金钱等某些利益或目的，通过丢弃没有或很少被访问的数据来节省存储空间，或者隐瞒数据损坏事件来维护其声誉。由此可以看出，云存储虽然能带来诸多优势和便利，但是并不能保证数据所有者数据的完整性。如果这个问题得不到有效的解决，将严重阻碍云存储的发展。

为了随时知晓数据是否被破坏或被丢失，数据所有者需要对存储在云服务器上的数据进行完整性检查。检查数据的完整性而采用下载全部原始数据的方法是不切实际的，因为这样增加网络中 I/O 和传输成本。数据完整性检查需要周期性地执行，这将会给数据所有者造成很大的负担，而数据所有者也不愿意去承担这些成本，因为数据所有者选择云存储服务的目的就是为了节约成本。因此，云存储中数据完整性检查工作对数据所有者来说是非常困难的，并且代价是相当高的。同时检查结果并不能使云服务提供商信服，因为服务提供商可能会怀疑数据所有者是故意采用欺骗手段诬告服务提供商而获取非法赔偿或损害服务商的声誉。此外，这种完整性检查工作完全由服务提供商承担，检查结果同样不能使数据所有者信服。当数据完整性遭到质疑时，无论是数据所有者还是云存储提供商都无法提供一个可信、公平的结果。在这种情形下，第三方审计是比较合适的选择。第三方审计是由可信第三方提供的，为数据所有者和云服务提供商提供可信、公平的审计结果。

第三方审计方案的设计面临诸多挑战：1) 支持盲审计，即数据需要对第三方审计者保密，在审计过程中审计者不能获知数据内容；2) 支持动态审计，即审计方案支持数据动态更新操作，以及对更新后的数据进行完整性检查；3) 支持批量审计，即审计方案支持多个审计任务进行合并处理，以提高审计效率。近年来，研究者们提出了一些第三方审计方案。表 1 从盲审计、动态审计、批量审计以及性能等方面对这些方案进行了比较。可以看出，现有的方案或多或少地存在一些问题：1) 有些方案不支持盲审计；2) 有些方案不支持数据更新操作，或者更新开销过高；3) 有些方案不支持批量审计，或者假设所有者只有一份文件；4) 计算开销和通信开销过高；5) 存在安全漏洞[5,6]。

针对上述方案存在的问题和不足，提出了一种云存储中数据完整性的聚合盲审计方法。本文的贡献主要有 3 方面。

1) 设计了公开存储审计框架并提出了盲审计方案。利用双线性对的性质，在云服务器端对数据证据和标签证据加密后再合并，审计者直接使用合并的结果来验证数据的完整性，从而达到保护数据隐私的目的，同时还可以减少审计者的计算开销。

2) 提出了基于索引机制的动态审计方法。设计新的索引机制，使数据更新的开销是 $O(1)$，同时保证数据更新操作不会带来新的安全问题。

3) 提出了支持多所有者多云服务器多文件的批量审计方法。在云服务器端将不同的证据聚合，不仅能减少计算开销，还使通信开销独立于审计任务的数量。

2 相关工作

ATENIESE 等首次提出的“可证明数据拥有”PDP 方案支持公开审计[7]，将 RSA 密码与同态可验证标签结合起来。所有者先将数据分成块并加密，然后为每个数据块计算标签，并与加密后的数据一起保存在服务器上。审计者向服务器发出对数据块子集的质询而不需要检索整个文件。同时，JUELS 等提出的“可检索的证明”(POR)方案[8]，在远程存

表 1 **云存储中数据完整性审计方案的比较**

方案	盲审计	动态审计		批量审计			计算开销		通信开销	
		类型	开销	多所有者	多云服务器	多文件	审计者	服务器	审计者	服务器
PDP[7]	N	N	N	N	N	N	$O(Kc)$	$O(Kc)$	$O(Kc)$	$O(K)$
POR[8, 9]	N	N	N	N	N	N	$O(Kc+Ks)$	$O(Kc+Ks)$	$O(Kc)$	$O(Ks)$
ZHU[13, 22]	Y	索引表	$O(n)$	N	Y	N	$O(Kc+Ks)$	$O(Kcs)$	$O(Kc)$	$O(Ks)$
WANG[11]	Y	散列树	$O(\lg n)$	Y	N	N	$O(Kcs)$	$O(Kcs)$	$O(Kc)$	$O(Ks)$
YANG[15]	Y	索引表	$O(n)$	Y	Y	N	$O(Kc)$	$O(Kcs)$	$O(Kc)$	$O(S)$

c 是质询块的数量，s 是块内元素的数量，K 是审计任务的数量，S 是云服务器的数量。计算开销和通信开销是考虑批量审计的开销，对于不支持批量审计的方案相当于执行 K 次单审计。

储服务器中利用抽样和纠错码来确保数据文件的“拥有性”和“可检索性”。该方案只支持有限次数的验证，不支持公开验证。SHACHAM 等[9]改进了 POR，使用同态认证支持公开验证，并且次数是无限的。PDP 和 POR 方案都是直接将数据块进行线性组合发送给审计者，从而可能将数据内容暴露给审计者。

1) 盲审计。WANG 等提出了利用随机屏蔽方法实现盲审计协议[10~12]，ZHU 等利用随机数来保护审计过程的数据隐私[13]。利用随机屏蔽方法的做法是在计算数据证据时加入随机数而不是直接将数据块进行线性组合发送给审计者，同时也需要将加密后的随机数发送给审计者。HE 等提出利用同态性对数据证据和标签证据同时加密的来保护数据隐私[14]。利用随机屏蔽技术来保护数据隐私会增加审计的计算开销。YANG 等提出了利用双线性对的性质对数据证据加密的方法来保护数据的隐私[15]。该方法不需要对数据证据进行随机屏蔽，但是存在安全漏洞。

2) 动态审计。ATENIESE 等设计了一种部分动态可证明（DPDP）的数据拥有协议[16]，其基本思想是在开始阶段预先计算一些元数据，其中每个元数据对应一个更新，缺点是更新和验证的次数有限且是预先固定的，不支持数据插入操作。ERWAY 等提出了改进的方案[17]，其方法是利用基于排名的认证词典，其中数据和标签组织成 skip-list 或 RSA 树结构以支持更新操作。WANG 等提出了基于纠错码的可靠云存储服务，支持数据更新操作[18]。同时，他们还提出利用默克尔散列树的方法支持数据的更新[19]。基于树结构的更新方法在数据更新时，部分树节点的值需要重新计算，效率很低。为了提高更新效率，研究者们提出了使用索引表支持动态审计[15,20~24]。基于索引表的更新方法也存在问题，当插入或删除数据块时，在插入或删除位置之后的索引项都需要更新。当索引更新方法设计不合理时，更新效率同样也是不高的。

3) 批量审计。WANG 等提出利用同态标签的思想将数据标签聚合实现批量审计[10,11]。WANG[25]和 YUAN 等[26]分别提出了针对共享数据的批量审计方法。HE[14]和 YANG 等[15]分别提出了支持多所有者多云服务器的批量审计方法。这些批量审计方法都是假设数据所有者只有一份文件，同时，只考虑从计算开销上提高审计效率，而没考虑减少通信开销。

3　模型和问题描述

3.1　系统模型

考虑如图 1 所示的云存储审计系统，包括数据所有者（DO）、云服务器（CS）和第三方审计者（TPA）。

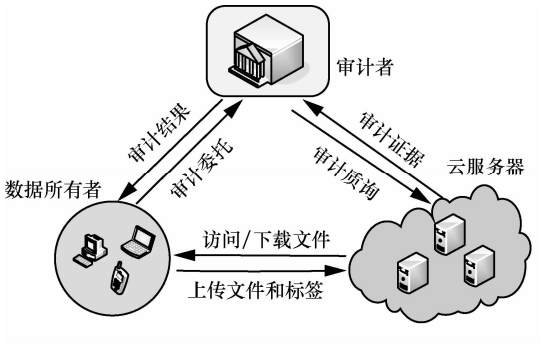


图 1　第三方审计系统模型

数据所有者：在云服务器中存储着拥有大量的数据，可以是单独的个体或机构。云服务器由服务提供商管理，拥有大量的存储空间和强大的计算资源，提供数据存储服务和数据访问服务。第三方审计者：能为数据所有者和云服务器双方提供审计服务。审计第三方可由可信部门监督和管理，提供可信、公正公平的审计结果。

在系统模型中，数据所有者依赖云服务器存储和管理数据，而云服务器和数据所有者不在同一个可信域，因此云服务器是不能被完全信任的。为了知道数据是否被破坏，数据所有者需要委托审计者检查云服务器中数据的完整性。尽管审计者是由可信部门监督和管理，表面上它们正常执行合约，不会与云服务器勾结，但不排除它们有可能为了某些目的而设法获取数据内容。因此，必须保证审计者在审计过程中不能从证据中获取到数据隐私，否则，第三方审计方案将会给所有者的数据带来新的安全威胁。

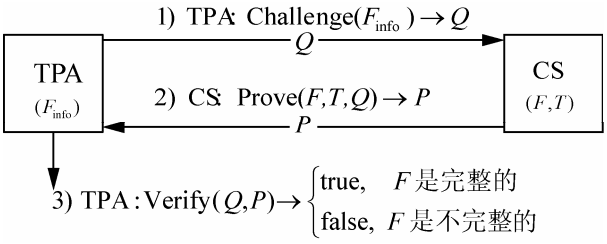
3.2　威胁模型

本文中所考虑威胁数据完整性的因素主要有：1) 硬件故障、自然灾害、黑客攻击和管理人员不正确的操作等因素外造成数据的丢失和破坏。2）云服务器可能为了自身的利益删除服务器中长时间不用的数据，以此减轻负担和费用，也有可能发现数据已被外部攻击损坏，却对数据所有者隐瞒实情，以此来维护自己的名誉。为此，在审计过程云服务器可能发起如下的攻击：a）替换攻击，云服

务器用正确的数据块和标签对去替换被质询的数据块和标签对，而被质询的数据块已经被破坏或丢失；b）伪造攻击，云服务器伪造证据欺骗审计者或伪造数据标签。特别是数据更新时所有者的标签密钥会被重复使用，使云服务器可能会伪造数据标签；c）重放攻击，云服务器用已通过验证的证据或者旧版本的数据块和标签欺骗审计者。本文中假设数据所有者、云服务器以及审计者他们两两之间的通信是安全可靠的，其安全信道可以通过密钥协商获得共享密钥建立，也可以基于公私钥加解密的方式建立。

3.3 问题描述

为了检查云服务器中数据的完整性，数据所有者为数据块生成数据标签。数据文件 F 和标签集合 T 都存储在云中。根据前面的模型，本文提出了一种交互式的审计方案，将要解决的问题可以表示为：



首先，审计者随机生成抽样数据块的质询集 Q，并发送给云服务器。然后，云服务器根据质询集 Q 及存储的数据文件和数据标签生成证据 P，并发送给审计者。最后，审计者验证证据的正确性。提出的审计方案必须满足下列要求：1）支持审计者在不知数据内容的前提下进行盲审计；2）支持数据更新操作的动态审计；3）支持将多个审计任务合并处理的批量审计。

4 基于双线性映射加密的盲审计方案

在审计过程中，审计者可能通过某些途径获取到数据隐私。对于明文数据，审计者可能会从接收到的数据证据中获取原始的数据块。对于加密数据，审计者可能会通过其他途径获取密钥从而能够解密数据。这样审计方案将会给所有者的数据带来新的安全威胁。本文利用双线性对的性质，在云服务器端对数据证据和标签证据加密，进一步将加密后的数据证据和标签证据合并，审计者直接使用合并后的结果来验证数据的完整性。该方法的好处有：1）利用双线性对的性质，将证据加密合并可以保护数据隐私而不增加额外的计算开销；2）将部分计算从审计者转移到云服务器可以提高审计效率，因为云服务器具有更强的计算能力；3）减少证据的通信开销。

4.1 盲审计方案的定义和框架

定义 1 （盲审计方案）本文的盲审计方案由 5 个算法组成。

1) KeyGen(λ) → (sk, pk)。密钥生成算法的输入是安全参数 λ，输出是计算数据标签的私钥和公钥对 (sk, pk)。

2) TagGen(F, sk) → T。标签生成算法的输入是数据文件 F 和私钥 sk，输出是数据标签集合 T。

3) Challenge(F_{info}) → Q。质询算法的输入是文件信息 F_{info} (包括文件标识，数据块索引等信息)，输出是质询信息 Q。

4) Prove(F,T,Q) → P。证据生成算法的输入是文件 F、标签 T 和质询信息 Q，输出是证据 P。

5) Verify(F_{info}, P, Q) → (true/false)。验证算法的输入是文件信息 F_{info}、证据 P 和质询信息 Q，输出是 true/false。

如图 2 所示，本文的盲审计过程包括初始化和抽样审计 2 个阶段。在初始化阶段，数据所有者生成密钥和数据标签，并将数据文件和数据标签上传到云服务器。在审计阶段，审计者周期性地抽样检查云服务器中数据的完整性。

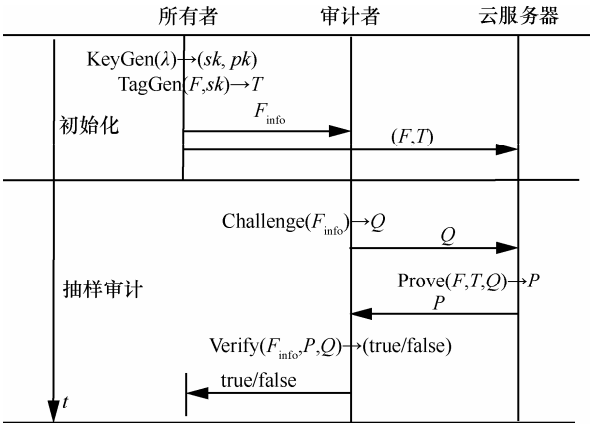


图 2 盲审计框架

阶段 1 初始化。首先，数据所有者执行密钥生成算法 KeyGen 生成私钥和公钥对（sk, pk）。然后执行标签生成算法 TagGen 为数据生成数据标签集合 T。最后数据所有者将数据 F 和数据标签 T 上传到云服务器，并将文件信息 F_{info} 发送给审计者，同时，数

据所有者将文件 F 和标签 T 从本地磁盘中删除。

阶段 2　抽样审计。收到审计委托后，审计者按如下过程周期性地检查数据的完整性。

1) 审计者执行质询算法 Challenge 生成质询集 Q，并发送给云服务器。

2) 收到质询集 Q 后，云服务器执行证据生成算法 Prove 计算相应的证据 P，并发送给审计者。

3) 收到证据 P 后，审计者执行验证算法 Verify 验证证据的正确性。并将审计结果发送给数据所有者。如果结果是 true，表示存储在云服务器上的数据是完整的。否则，表示数据已经被破坏或丢失。

为了提高审计效率，本文采取随机抽样审计。假设数据块中每个元素被破坏的概率是 ρ，抽样数据块的数量为 c。当云服务器中数据被破坏时，本文的审计方法能检测到数据被破坏的概率是 $\Pr(c,s)=1-(1-\rho)^{cs}$。文献[7]已证明，当抽样数据块数量 $c=460$ 时（无论文件数据块的总数是多少），检测概率高达 99%。

4.2　算法描述

假设文件 F 由 n 块数据组成 $F=(m_1,m_2,\cdots,m_n)$，其中 m_i 由 s 个元素组成 $m_i=(m_{i1},m_{i2},\cdots,m_{is})$。假设 G_1、G_2 和 G_T 是素阶为 p 的乘法循环群，g_1 和 g_2 分别是 G_1 和 G_2 的生成元，h 是散列函数：$\{0,1\}^*\to G_1$。$x_1,x_2,\cdots,x_s$ 是 Z_p 的随机数，并计算 $u_j=g_1^{x_j}\in G_1$（$j\in[1,s]$）。$e:G_1\times G_2\to G_T$ 是双线性映射对，具有如下性质：1）存在高效的算法计算 e；2）$e(g_1,g_2)\neq 1$；3）对于任意的 $u\in G_1,v\in G_2$ 和 $a,b\in Z_p$ 有 $e(u^a,v^b)=e(u,v)^{ab}$；4）对于任意的 $u_1\in G_1$，$u_2\in G_1$，$v\in G_2$ 有 $e(u_1u_2,v)=e(u_1,v)e(u_2,v)$。

下面详细描述本文盲审计方案的 5 个算法。

$\mathrm{KeyGen}(\lambda)\to(sk,pk)$。算法 1 是密钥生成算法。

算法 1　KeyGen

输入：安全参数 λ

输出：生成计算标签的私钥和公钥对 (sk,pk)

1) 随机选取一个值 $sk\in Z_p$ 作为私钥；

2) 计算相应的公钥 $pk=g_2^{sk}$；

3) return (sk,pk)。

$\mathrm{TagGen}(F,sk)\to T$。算法 2 是标签生成算法。

算法 2　TagGen

输入：数据文件 F 和私钥 sk

输出：数据标签集合 $T=\{t_i\}_{i\in[1,n]}$

1) for　i=1 to n do；

2) 为数据块 m_i 生成对应的数据标签 t_i：$t_i=(h(w_i)\prod_{j=1}^{s}\mu_j^{m_{ij}})^{sk}$；//其中 $w_i=fid\,\|\,i$，fid 是文件标识，i 是块号，||是连接操作；

3) end for；

4) return $T=\{t_i\}_{i\in[1,n]}$。

$\mathrm{Challenge}(F_{\mathrm{info}})\to Q$。算法 3 是质询算法。

算法 3　Challenge

输入：文件信息 F_{info}(包括文件标识、数据块索引等)

输出：质询信息 $Q=(\{i,v_i\}_{i\in I},R_1,R_2,R_3)$

1) 随机选取部分数据块组成质询集 $I(I\in subset([1,n]))$；

2) 为每个被质询的数据块生成相应的随机值 v_i；

3) 随机选择 2 个值 $r_1,r_2\in Z_p$，并计算 $R_j=(g_1^{x_j})^{r_1}$，$R_1=pk^{r_2}$，$R_2=g_2^{r_1r_2}$；

4) return $Q=(\{i,v_i\}_{i\in I},\{R_1\}_{j\in[1,s]},R_1,R_2)$。

$\mathrm{Prove}(F,T,Q)\to P$。算法 4 是证据生成算法。

算法 4　Prove

输入：文件 F、数据标签 T 和质询信息 Q

输出：质询信息 P

1) 计算数据证据(DP)：$DP=e(\prod_{j=1}^{s}R_j^{\sum_{i\in I}v_im_{ij}},R_1)$；

2) 计算标签证据(TP)：$TP=e(\prod_{i\in I}(t_i)^{v_i},R_2)$；

3) 合并数据证据和标签证据：$P=\dfrac{TP}{DP}$；

4) return P。

$\mathrm{Verify}(F_{\mathrm{info}},P,Q)\to\left(\dfrac{\mathrm{true}}{\mathrm{false}}\right)$。算法 5 是验证算法。

算法 5　Verify

输入：文件信息 F_{info}、证据 P 和质询信息 Q

输出：true/false

1) 按式(1)验证证据的正确性

$$P\overset{?}{=}e(\prod_{i\in I}h(w_i)^{v_ir_1r_2},pk)\tag{1}$$

2) if 式(1) 成立；

3)　return true；

4) else；

5)　return false；

5　基于索引机制的动态审计

存储在云服务器上的数据可能会被数据所有

者时不时地更新。数据更新操作可能导致的问题如下。1）当数据块被插入或删除时，会影响插入或删除位置之后的所有数据块，受影响数据块的数据标签可能需要重新计算，这将导致很大的计算开销和通信开销。2）服务器可能会实施重放攻击和伪造攻击带来安全漏洞。一方面，服务器可能并没有正确更新数据所有者的数据，而用之前存储的版本通过审计来欺骗数据所有者；另一方面，云服务器可能会伪造数据标签欺骗审计者，因为同一数据块的新版本重复使用相同的索引和标签私钥。如果云服务器能伪造标签，那么它就可以使用任意的数据和伪造的标签通过审计。

5.1 索引表

本文利用索引表来实现数据更新操作。索引表记录数据块的摘要信息，每条记录是一个四元组 $<i,B_i,V_i,T_i>$，其中 i 是数据块 m_i 的索引，B_i 是 m_i 的原始块编号，V_i 是 m_i 的版本号，T_i 是用于生成数据标签的时间戳。索引表是由所有者创建和初始化的，但由审计者存储和维护。数据更新时，审计者也需要同步更新索引表。为了使数据更新操作不带来新的安全问题，在标签中加入数据块的全部摘要信息。在标签生成算法 TagGen 中使用 $w_i = fid \| i \| B_i \| V_i \| T_i$ 而不是单独的 i 来计算数据 m_i 的标签 t_i，这样可以防止云服务器发起伪造攻击。同时，利用版本号 V_i 和时间戳 T_i，可以防止云服务器发起重放攻击。

表 2 记录针对不同的操作索引表的更新情况。

表 2 描述了文件 F 的初始化；修改数据块 m_2 后，对应的 V_2 和 T_2 也同步更新了；在数据块 m_2 前插入新的数据块 m_{n+1}，（m_1 和 m_2 之间），索引 i 即为 $\frac{1+2}{2}$，后面的索引项不用改变；删除数据块 m_1，对应的索引项直接删除，其他索引项不变。

5.2 数据更新

本文考虑基于块的数据更新，对于每一个更新请求可以表示为 $UpReq =< OP,i,B_i^*,V_i^*,T_i^*,m_i^*,t_i^* >$，其中，$OP$ 是数据操作类型包括修改（M）、插入（I）、删除（D），i 表示操作的位置，B_i^* 是新的块编号，V_i^* 是新的版本号，T_i^* 是新的时间戳，m_i^* 是新的数据块，t_i^* 是 m_i^* 对应的标签。

数据更新过程包括更新执行和更新验证 2 个阶段。在更新执行阶段，数据所有者生成数据更新请求 $UpReq$ 并发送给云服务器；云服务器收到更新请求后，执行相应的更新操作并将索引表更新信息发送给审计者；审计者接收到信息后更新索引表。在更新验证阶段，审计者对更新的数据进行审计并将结果发送给数据所有者。更新验证的目的是为了检查更新有没有被云服务器正确执行。尽管审计者会周期性地审计云中数据的完整性，但当检查到数据不完整时，并不能区分数据是被破坏还是没有被更新。

阶段 1 更新执行。更新执行阶段分为 3 步。

1) 数据所有者根据操作的类型生成相应的更新请求，并将请求发送给云服务器。按操作类型分 3 种情况。a）当操作类型是修改时，数据所有者生成新的版本号 V_i^* 和新的时间戳 T_i^*，修改操作不会改变索引 i 和块编号 B_i，并执行标签生成算法 TagGen 为修改后的数据块 m_i^* 生成新的标签 t_i^*。针对修改操作，更新请求可以表示为 $UpReq =< M,i,B_i,V_i^*,T_i^*,m_i^*,t_i^* >$。b）当操作类型是插入时，数据所有者为要插入的数据块 m_i 生成原始的版本号 V_i 和新的时间戳 T_i，索引号和块编号分别是 i 和 B_i，表示在第 i 块数据之前插入 m_i，并执行标签生成算法 TagGen 为新的数据块 m_i 生成新的标签 t_i。针对插入操作，更新请求可以表示为 $UpReq =< I,i,B_i,V_i,T_i,m_i,t_i >$。c）当操作类型是删除时，只需要索引号 i，表示删除第 i 个位置的数据块。针对删除操作，更新请求可以表示为 $UpReq =< D,i,B_i,0,0,0,0 >$。

表 2 索引表的更新情况

初始化				修改 m_2				在 m_2 前插入 m_{n+1}				删除 m_1			
i	B_i	V_i	T_i	i	B_i	V_i	T_i	i	B_i	V_i	T_i	i	B_i	V_i	T_i
1	1	1	T_1	1	1	1	T_1	1	1	1	T_1	$\frac{3}{2}$	$n+1$	1	T_{n+1}
2	2	1	T_2	2	2	2	T_2^*	$\frac{3}{2}$	$n+1$	1	T_{n+1}	2	2	2	T_2^*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	2	2	2	T_2^*	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	n	1	T_n	n	n	1	T_n	n	n	1	T_n	n	n	1	T_n

2) 收到数据所有者的更新请求*UpReq*后，云服务器根据操作类型执行相应的更新操作。按操作类型分 3 种情况。a）当操作类型是修改时，云服务器用新的数据块m_i^*和标签t_i^*替换旧的m_i和t_i。b）当操作类型是插入时，云服务器直接在第i个数据块之前插入m_i和相应的t_i。c）当操作类型是删除时，云服务器直接删除第i个位置的数据块及相应的标签。执行完更新后，云服务器将索引表更新信息发送给审计者。

3) 收到云服务器的索引表更新信息，审计者根据操作类型更新相应索引表项。按操作类型分 3 种情况。a）当操作类型是修改时，云服务器用新的版本号V_i^*和时间戳T_i^*替换旧的V_i和T_i。b）当操作类型是插入时，云服务器直接在第i个索引项之前插入新的索引项。c）当操作类型是删除时，云服务器直接删除第i个位置的索引项。

阶段 2　更新验证。针对被更新的数据块，审计者按照前面抽样审计的过程验证更新操作是否被云服务器正确执行。如果结果是 true，表示更新操作已被云服务器正确执行。否则，表示更新没有被正确地执行。

6　聚合批量审计

随着云存储的发展，越来越多的数据所有者会选择将数据存储在云服务器上，这样审计者可能会接收大量来自不同数据所有者的审计委托。由于审计者是周期性地执行审计任务，如果一个一个地执行这些大量的审计任务，将会造成巨大的通信开销和计算开销。在这种情况下，如果将众多审计请求合并进行批量审计，则会大大提高审计效率。另外，不同文件可能会被存储在不同的云服务器上，这样审计者可能会收到来自不同云服务器的证据。同时，同一个所有者可能有针对不同文件的多个审计请求。因此，批量审计方法必须支持多所有者多云服务器多文件的情形。

本文的方法是审计者在生成批质询集时，同一个云服务器上的不同审计任务使用同一个质询集，使批量审计的质询集的大小只与云服务器的数量有关，而与审计任务的数量无关。同样，云服务器生成证据时，将不同的证据聚合，而不是直接将每个证据发送给审计者，这样证据的大小也只与云服务器的数量有关。将不同的证据在云服务器聚合还可以进一步减少审计者的计算开销。

假设审计者收到了K个审计委托，可能来自不同数据所有者针对不同文件的审计，也可能同一所有者有多个审计委托。另外，不同数据文件可能存储在不同云服务器上，同一个云服务器上可能有多个被质询的文件。批量审计的过程分为 2 个阶段。

阶段 1　初始化。为了方便描述，每个审计任务对应的数据文件和标签可以分别表示为$F_k=\{m_{k,i}\}_{k\in K,i\in[1,n]}$和$T_k=\{t_{k,i}\}_{k\in K,i\in[1,n]}$，其中，$t_{k,i}=(h(w_{k,i})\prod_{j=1}^{s}\mu_j^{m_{k,ij}})^{sk_k}$。文件$F_k$对应的生成数据标签的私钥和公钥对是$(sk_k,pk_k)$。

阶段 2　批量审计。批量审计阶段由 3 个算法组成：批量质询算法 BChallenge、批量证据生成算法 BProve 和批量验证算法 BVerify。算法 6 详细描述了批量审计过程。

算法 6　批量审计

输入：K个审计任务

输出：true/false

// BChallenge　at　审计者

1) 审计者执行质询生成算法 Challenge 为K个审计任务随机选取数据块组成质询子集$I(I\in subset([1,n]))$和生成相应的随机值v_i，并随机选择$r_1,r_2\in Z_p$和计算$R_j=g_1^{r_1}$，$R_k=pk_k^{r_2}$，$R_2=g_2^{r_1r_2}$；

2) for　l=1 to S do

3)　审计者为云服务器CS_l生成相应质询集$Q_l=(\{i,v_i\}_{i\in I},\{R_j\}_{j\in[1,s]},\{R_k\}_{[k\in K_l]},R_2,K_l)$，$K_l$表示云服务器$CS_l$有$K_l$个文件被质询（$K_1+K_2+\cdots+K_S=K$）；

4)　审计者将质询集Q_l发送给对应的CS_l；

5) end for

// BProve　at　云服务器

6) for　l=1 to S do

7)　CS_l执行证据生成算法 Prove 按如下过程生成相应的证据：

$$DP_l=\prod_{k\in K_l}e(\prod_{j=1}^{s}R_j^{\sum_{i\in I}v_im_{k,ij}},R_k);$$
$$TP_l=e(\prod_{k\in K_l}\prod_{i\in I}(t_{k,i})^{v_i},R_2);$$
$$P_l=TP_l/DP_l;$$

8)　CS_l将证据P_l发送给审计者；

9) end for

// BVerify　at　审计者

10)审计者通过式(2)验证S个证据的正确性：

$$\prod_{l=1}^{S} P_l \stackrel{?}{=} \prod_{k=1}^{K} e(\prod_{i\in I} h(w_i)^{v_i r_1 r_2}, pk_k) \tag{2}$$

11) if 式(2)成立

12) return true;

13) else

14) return false

当式(2)成立时，表示 K 个证据都是正确的，即在所有数据块和数据标签都正确存储在云服务器上。如果 K 个被质询的文件中有任一数据块或数据标签被破坏或被删除，式(2)将不成立。在审计过程中，双线性映射对 e 将花费大部分的计算开销，而批量审计的一个明显特点就是可以减少双线性映射对 e 的运算次数。根据式(2)可知，批量审计 K 个任务只需要 $S+2K$ 次双线性映射对 e 运算，而逐一审计 K 个任务则需要 $3K$ 次双线性映射对 e 运算。本文批量审计的另一个特点就是可以减少通信开销。让同一个云服务器上的不同审计任务使用相同的质询集，同时在云端将同一个云服务器上证据聚合成一个证据，这样使通信开销独立于审计任务数，其大小只与云服务器的数量 S 有关。

7 安全性分析

7.1 盲审计方案的安全性分析

定理 1 在本文的盲审计方案中，云服务器能通过审计当且仅当所有被质询的数据块和标签都正确地存储在云服务器上。

证明 如果当且仅当被质询的数据块和标签都正确存储时式(1)成立，那么就说明本文提出的审计方案是正确的。根据双线性对的性质，式(1)的正确性证明如下

$$\begin{aligned}
&DPe(\prod_{i\in I} h(w_i)^{v_i r_1 r_2}, pk)\\
=&e(\prod_{j=1}^{s} R_j^{\sum_{i\in I} v_i m_{ij}}, R_1) e(\prod_{i\in I} h(w_i)^{v_i r_1 r_2}, pk)\\
=&e(\prod_{j=1}^{s} g_1^{x_j \cdot sk \cdot \sum_{i\in I} v_i m_{ij}}, g_2^{r_1 r_2}) e(\prod_{i\in I} h(w_i)^{sk v_i}, g_2^{r_1 r_2})\\
=&e(\prod_{i\in I} (h(w_i) \prod_{j=1}^{s} u_j^{m_{ij}})^{sk v_i}, g_2^{r_1 r_2})\\
=&e(\prod_{i\in I} (t_i)^{v_i}, R_3)\\
=&TP
\end{aligned}$$

即 $P = e(\prod_{i\in I} h(w_i)^{v_i r_1 r_2}, pk)$ 。

定理 2 如果云服务器所提供的证据能够通过审计者地验证，其证据必须是服务器通过正确计算得到的，而不是伪造的。

证明 如果存在攻击者以概率 ε 成功地伪造一个证据 $P' = (DP', TP')$ 使其通过验证，则必存在算法能以与概率 ε 相关的概率破解离散对数问题(DL)。假设质询集是 Q，云服务器生成的正确的证据是 $P = \dfrac{TP}{DP}$，而服务器伪造一个不正确的证据 $P' = \dfrac{TP'}{DP'}$，其中，$P' \neq P, DP' \neq DP, TP' \neq TP$。如果证据 P' 仍然能通过审计，根据验证式(1)可以得到

$$DP' e(\prod_{i\in I} h(w_i)^{v_i r_1 r_2}, pk) = TP' \tag{3}$$

由于 $P = \dfrac{TP}{DP}$ 是正确的证据，即有

$$DPe(\prod_{i\in I} h(w_i)^{v_i r_1 r_2}, pk) = TP \tag{4}$$

根据双线性对的性质，将等式(3)和等式(4)相除即可得到

$$e\left(\frac{t'}{t}, R_2\right) = e(\prod_{j=1}^{s} R_j^{\Delta m_j}, R_1) \tag{5}$$

其中，$t' = \prod_{i\in I} (t_i')^{v_i}$，$t = \prod_{i\in I} (t_i)^{v_i}$，$\Delta m_j = \sum_{i\in I} v_i m_{ij}' - \sum_{i\in I} v_i m_{ij}$。

进一步可得

$$r_1 \frac{t'}{t} = \prod_{j=1}^{s} R_j^{\Delta m_j} \tag{6}$$

即可求得

$$r_1 = t \prod_{j=1}^{s} \frac{R_j^{\Delta m_j}}{t'} \tag{7}$$

也就是在知道 $u_1, R_j = u_j^{r_1}$ 的条件下，攻击者可以获取 r_1，即攻击者就可以以概率 ε 成功破解 DL 问题，这样就违背了 DL 问题假设。

定理 3 审计过程中，审计者不能从证据 P 中获取数据隐私，即该方案支持盲审计。

证明 在证据计算过程中，利用双线性对的性质对数据证据加密和标签证据加密，并将数据证据和标签证据合并，再将合并的结果发送给审计者。审计者在不能也不需要解密的情况下直接使用合并的结果验证证据的正确性。审计者不能从合并的结果中恢复数据证据和标签证据，因此无法获取数据内容。再者，加密过程中数据块与随机数 R_1 之间是对数运算的($\prod_{j=1}^{s} R_j^{\sum_{i\in I} v_i m_{ij}}$)，如果能从中获取数据块

m_{ij}，就相当于能破解 DL 问题。可以看出本文采用双重机制保证审计者不能从证据 P 获取到数据隐私内容。因此，本文的审计方案支持盲审计。

定理 4　本文的审计方案可以抵抗替换攻击。

证明　假设云服务器可能用正确的数据块和标签对 (m_k,t_k) 去替换被质询的数据块和标签对 (m_l,t_l)，数据证据就应该表示为

$$DP' = e(\prod_{j=1}^{s} R_j^{\sum_{i\in I,i\neq l} v_i m_{ij}}, R_1) \tag{8}$$

同样，标签证据应该是

$$TP' = e(t_k^{v_l} \prod_{i\in I,i\neq l} (t_i)^{v_i}, R_3) \tag{9}$$

根据验证式（1）即有

$$DP' \cdot e(\prod_{i\in I} h(w_i)^{v_i r_1 r_2}, pk) = e\left(\left(\frac{h(w_l)}{h(w_k)}\right)^{v_l \cdot sk}, R_3\right) TP' \tag{10}$$

由于散列函数 h 是防碰撞的，就有 $h(w_l)\neq h(w_k)$，因此等式(10)是不能成立的。也就是说，云服务器用正确的数据块和标签对 (m_k,t_k) 去替换被质询的数据块和标签对 (m_l,t_l) 是不能通过审计的。因此，本文的审计方法可以抵抗替换攻击。

7.2　动态审计的安全性分析

定理 5　本文的动态审计可以抵抗标签伪造攻击。

证明　假设云服务器可以在不知道标签生成私钥的前提下，可以伪造标签。对于 2 个不同的数据块 m_i 和 m_i'，对应的标签分别是 $t_i = (h(w_i)\prod_{j=1}^{s} u_j^{m_{i,j}})^{sk}$ 和 $t_i = (h(w_i)\prod_{j=1}^{s} u_j^{m_{ij}'})^{sk}$。为了方便，用 $g_1^{m_i}$ 和 $g_1^{m_j}$ 分别表示 $\prod_{j=1}^{s} u_j^{m_{i,j}}$ 和 $\prod_{j=1}^{s} u_j^{m_{ij}'}$。即有

$$\frac{t_i}{t_i'} = g_i^{(m_i - m_i')sk} \tag{11}$$

即可求得

$$g_1^{sk} = \left(\frac{t_i}{t_i'}\right)^{\frac{1}{(m_i - m_i')}} \tag{12}$$

对于任一数据块和标签对 (m_k,t_k)，根据标签计算方法云服务器可以得到

$$h(w_k)^{sk} = \frac{t_k}{(g_1^{sk})^{m_k}} \tag{13}$$

因此对于修改后的数据块 m_k^*，云服务器可以伪造其对应的标签为

$$t_k^* = t_k \left(\frac{t_i}{t_i'}\right)^{\frac{m_k^* - m_k}{m_i - m_i'}} \tag{14}$$

等式(14)成立的前提是 $h(w_k) = h(w_k^*)$，也就是 $w_k = w_k^*$。在数据动态更新过程中，w_k 是等于 $fid \| k \| B_k \| V_k \| T_k$。数据修改时，数据所有者会生成新的版本号 V_k^* 和时间戳 T_k^*，而 $V_k^* \neq V_k$，$T_k^* \neq T_k$，所以 $w_k^* \neq w_k$。因此，云服务器是不能伪造数据标签的。

定理 6　本文的动态审计可以抵抗重放攻击。

证明　在审计过程中，每次质询集抽样的数据块不一样，R_1, R_2, R_3 也是随机选择的，因此针对 2 次不同的质询，云服务器计算的证据是不同的，云服务器不能使用之前正确的证据欺骗审计者。同时，针对数据更新操作，用于计算标签的版本号 V 和时间戳 T 是不同的，计算的散列值是不相同的。因此，数据更新也不会使云服务器发起重放攻击的。

7.3　批量审计的安全性分析

定理 7　批量审计具有正确性、不可伪造性、盲审计、抵抗替换攻击、抵抗伪造攻击和抵抗重放攻击等性质。

证明　批量审计是在盲审计和动态审计的基础上设计的。因此，批量审计仍然具有盲审计和动态审计相关的性质。证明过程和前面的类似，这里就不再详细证明。

8　性能分析及实验

8.1　性能分析

从表 1 针对现有数据完整性审计方案的对比结果可以看出，无论是从支持的功能全面性上还是从审计效率上考虑，本文方案与 WANG 的方案[11]和 YANG 的方案[15]比较接近。因此，接下来，将从存储开销、通信开销以及计算开销等方面进行详细分析比较。表 3 给出这 3 种方案的详细对比结果。本文盲审计方案是考虑单用户单任务的情况，为了方便描述，将其称为单审计。

1) 存储开销。存储开销主要是由存储云服务器的数据块和数据标签引起的。本文的方案和 YANG 的方案都需要在审计者上存储数据文件的索引信息。相对于数据和数据标签的大小，索引信息的存储开销是可以忽略不计的。例如，当安全参数是 160 bit，数据块内的元素总数是 $s = 50$ 时，对于 10 MB 的文件，

块数就是 1 000，索引表的大小也只有 500 B。因此，本文将不考虑索引表的存储开销。本文方案和 YANG 的方案在云服务器中只存储数据块和数据标签，所以存储开销都是 $|F|+\frac{|F|}{s}$。而在 WANG 的方案中，为了支持动态审计，云服务器中需要存储默克尔散列树(MHT)，其大小是 $2|F|$，所以存储开销都是 $3|F|+\frac{|F|}{s}$。因此，本文方案和 YANG 的方案减少了存储开销。

2) 计算开销。审计方法由 KeyGen 、TagGen 、Challenge 、 Prove 和 Verify 等 5 个算法组成的。由于 KeyGen 、 TagGen 、 Challenge 等 3 个算法的计算开销几乎是一样的，因此只比较 Prove 和 Verify 的计算开销。相比群运算，整数运算是可以忽略的，因此只需考虑群运算。E 表示群中一次指数运算，M 表示群中一次乘法运算，P 表示一次双线性对 e 运算。在群 G 内运算中，相比指数和乘法运算，双线性对 e 运算的计算代价更高，双线性对 e 运算的次数将直接影响审计方案的效率。从表 3 中可以看出，3 种方案的计算开销都与被质询的块数 c 成线性关系，被质询的块数 c 越大，计算开销越大。进一步可以看出，WANG 的方案和 YANG 的方案中双线性对 e 运算的次数还与块内元素的数量 s 有关。 WANG 的方案和 YANG 的方案都需要花费 $s+2$ 次的双线性对 e 运算。这是因为 WANG 的方案利用随机 $\{R_j=e(u,v)^{r_j}\}_{1\leqslant j\leqslant s}$ 保护数据隐私，YANG 的方案在计算数据证据 $DP=\prod_{j=1}^{s}e(u_j,R)^{MP_j}$ 需要将 s 个值聚合。在本方案中，只需要 3 次的双线性对 e 运算。因此，本文方案比其他 2 种方案花费更少的计算开销。相比单审计，批量审计可以大大减少双线性对 e 运算的次数。用单审计方案逐一审计 K 个任务，WANG 的方案和 YANG 的方案需要 $K(s+2)$ 次的双线性对 e 运算，而本文方案只需要 $3K$ 次的双线性对 e 运算。在批量审计方案中，WANG 的方案和 YANG 的方案需要 $Ks+K+1$ 次的双线性对运算 e，而本方案需要 $2K+S$ 次的双线性对 e 运算。

3) 通信开销。通信开销主要由初始化和审计过程中引起的。由于初始化引起的通信开销等同于存储开销，而且初始化只需要执行一次。审计过程需要周期性执行，通信开销将直接影响方案的性能。因此，只比较审计过程中的通信开销。审计过程中的通信开销主要有质询算法 Challenge 和证据生成算法 Prove 所引起的。在质询阶段，WANG 的方案中的质询集 $Q=\{i,v_i\}_{i\in I}$，其大小是 $c(|i|+|v_i|)$。YANG 的方案中的质询集 $Q=\{i,v_i,R\}_{i\in I}$，其大小是 $c(|i|+|v_i|)+|p|$。本方案的质询集 $Q=\{i,v_i,\{R_j\}_{j\in[1,s]},R_1,R_2\}_{i\in I}$，其大小是 $c(|i|+|v_i|)|+(s+2)|p|$。在证据生成阶段，WANG 的方案的证据 $P=\{\mu_i,\sigma,R_i\}_{i\in[1,s]}$，其大小是 $(2s+1)|p|$。YANG 的方案的证据是 $P=\{DP,TP\}$，其大小是 $2|p|$。本文方案的质询集 $P=\frac{TP}{DP}$，其大小是 $|p|$。从表 3 中，可以看出 3 种方案的通信开销都与被质询的块数 c 成线性关系，而 WANG 的方案还与块内元素的数量 s 有关。因此，本文方案和 YANG 的方案的通信开销比 WANG 的方案要少。

在批量审计中，WANG 的方案的通信开销与审计任务数 K 成正比。在 YANG 的方案中，质询的通信开销与审计任务数成正比，而证据的通信开销是

表 3　　性能开销详细对比

方案	存储开销	计算开销				通信开销			
		单审计		批量审计		单审计		批量审计	
		Prove	Verify	Prove	Verify	Challenge	Prove	Challenge	Prove
WANG	$3\|F\|+\frac{\|F\|}{s}$	$(c+s)E+cM+sP$	$(c+s+1)E+(2s+c)M+2M+2P$	$K(c+s)E+K(c+1)M+KsP$	$K(c+s+1)E+K(2s+c)M+2M+(K+1)P$	$c(\|i\|+\|v_i\|)$	$(2s+1)\|p\|$	$c(\|i\|+\|v_i\|)$	$K(2s+1)\|p\|$
YANG	$\|F\|+\frac{\|F\|}{s}$	$(c+s)E+cM+sP$	$(c+1)E+(c+1)M+2P$	$K(c+s)E+KcM+KsP$	$(Sc+1)E+S(c+2)M+(K+1)P$	$c(\|i\|+\|v_i\|)+s\|p\|$	$2\|p\|$	$cK(\|i\|+\|v_i\|)+Ks\|p\|$	$2S\|p\|$
本文方案	$\|F\|+\frac{\|F\|}{s}$	$(c+s)E+(c+s)M+2P$	$cE+(s+c)M+2M+P$	$K(c+s)E+K(c+s)M+(K+S)P$	$KcE+KsM+(2S+Kc)M+KP$	$c(\|i\|+\|v_i\|)+(s+2)\|p\|$	$\|p\|$	$cS(\|i\|+\|v_i\|)+(K+S+s)\|p\|$	$S\|p\|$

注：$|F|$ 是数据文件的大小，$|i|$ 是数据块索引的大小，$|v_i|$ 是随机值 v_i 的大小，c 是质询块的数量，s 是块内元素的数量，$|p|$ 是整数 Z_p 或群 G 中元素的大小，K 是审计任务的数量，S 是云服务器的数量。

与云服务器数 S 成正比的。在本文方案中，质询的通信开销和证据的通信开销都是与审计任务数 S 成有关的。实际中，云服务器的个数 S 是远远小于审计任务数 K 的。因此，本文的批量审计方案可以减少通信开销。

4) 参数分析。从前面的分析对比可以看出，块内元素的数量 s 将直接影响存储开销、通信开销和计算开销。从表 3 中可以看出：3 种方案的存储开销都与块内元素总数 s 成反比，WANG 方案的通信开销与块内元素总数 s 成正比，WANG 方案和 YANG 方案的双线性对 e 运算的次数与块内元素总数 s 成正比，同时 3 种方案中指数运算 E 和乘法运算 M 都与块内元素的数量 s 成线性关系的。

由于安全方面的要求，块内元素的数量 s 是不能超过安全系数 λ 的。对于大小固定的文件 F，数据块的数量 $n=\dfrac{\text{sizeof}(F)}{s\lg\lambda}$。在标签生成算法 TagGen 中，计算一个数据标签的开销是 $s(E+M)+E$，因此对于大小固定的文件 F，其标签生成的总计算开销 $T_{\text{tag}}=\dfrac{\text{sizeof}(F)}{\lg\lambda}(E+M+\dfrac{1}{s}E)$，而标签的总存储开销是 $\dfrac{|F|}{s}$。可以看出，随着 s 的增大，存储开销会减小，生成标签的总开销也会减小，而审计的计算开销会增大。从生成标签的总开销和审计开销的数值分析中可以看出，s 的取值对本文方案的影响不是决定性的，既不影响通信开销，也不影响双线性对 e 运算的次数。但是，s 的取值将直接影响其他 2 种方案的性能。

8.2　实验及分析

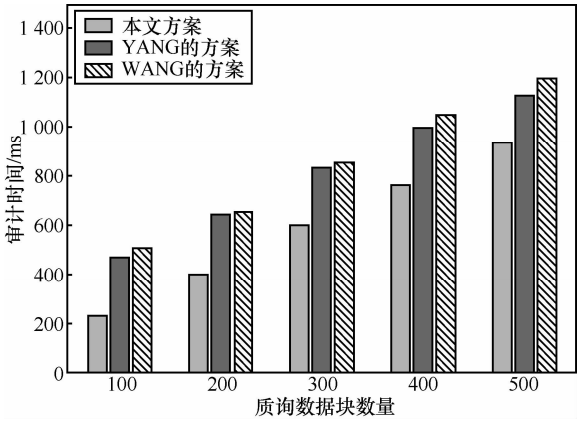
下面将设计实验对本文方案的性能进行分析。使用两台计算机搭建系统原型。一台模拟存储服务，一台模拟审计服务。计算机的性能是 Intel Core 2 2.67 GHz， 4 GB 内存，操作系统是 openSUSE 12.1。本文中所有的算法都是用 C 语言实现的。编码的实现是基于 Pairing-Based Cryptography（PBC）库的，使用的椭圆曲线是 MNT d159 曲线，其基础域大小和嵌入度分别是 159 bit 和 6。实验选择的安全参数 λ 是 80 bit，意味着 $|v_i|=80$ 和 $|p|=160$。文件 F 的数据块总数 $|n|=100\,000$，数据块大小是 4 KB，数据块内元素的数量 $s=50$。云服务器的数量 $S=5$。所有的实验结果都是取 50 次实验的平均值。

1) 单审计的效率对比。图 3 展示了验证不同大小质询集 c 所用的审计时间和通信开销。审计时间是指云服务器生成证据的时间与审计者验证证据的时间的总和。通信开销是指质询集的大小和证据的大小的总和。

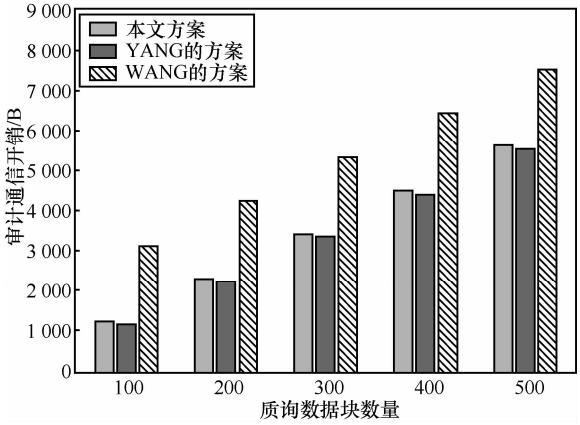
从图 3(a)易知，随着质询数据块数量的增加，3 种方案的审计时间都会增加，但本文方案比其他 2 种方案所用的时间要少。当质询数据块的数据 $c=500$ 时，本文方案所用的时间是 940 ms，而 YANG 的方案和 WANG 的方案所用的时间分别是 1 113 ms 和 1 198 ms，效率分别提高了 15.5%和 21.5%。

从图 3(b)可以看出，3 种方案的通信开销与质询数据块的数量成线性关系。本文方案和 YANG 的方案比 WANG 的方案所需的通信要少。当质询数据块的数据 $c=500$ 时，本文方案和 YANG 的方案所用的通信开销分别是 5 580 B 和 5 560 B，而 WANG 的方案所需的通信开销是 7 520 B。

2) 批量审计的效率对比。图 4 展示了批量验证不同数量审计任务所用的审计时间和通信开销。在批量审计测试中，质询数据块的数量 c 为 500。



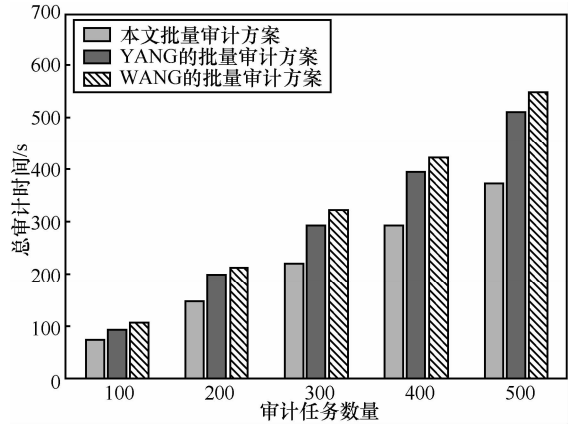
(a) 审计时间对比



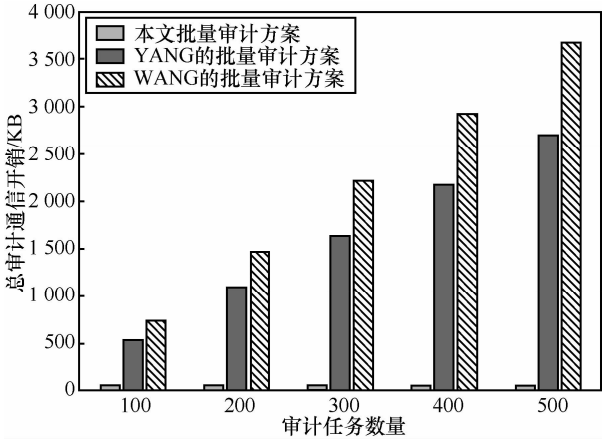
(b) 通信开销对比

图 3　单审计的效率对比

从图 4(a)可以看出，随着审计任务数量的增加，3 种批量审计方案所用的时间也会增加，但本文批量审计方案比其他 2 种批量审计方案所用的时间要少。当审计任务数量 K 为 500 时，本文批量审计方案所用的时间是 376 s，而 YANG 的批量审计方案和 WANG 的批量审计方案所用的时间分别是 507 s 和 552 s，效率分别提高 25.8% 和 31.8%。



(a) 批量审计时间对比



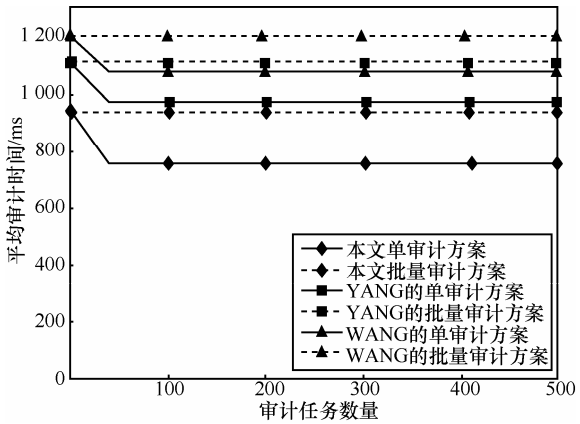
(b) 批量通信开销对比

图 4　批量审计的效率对比

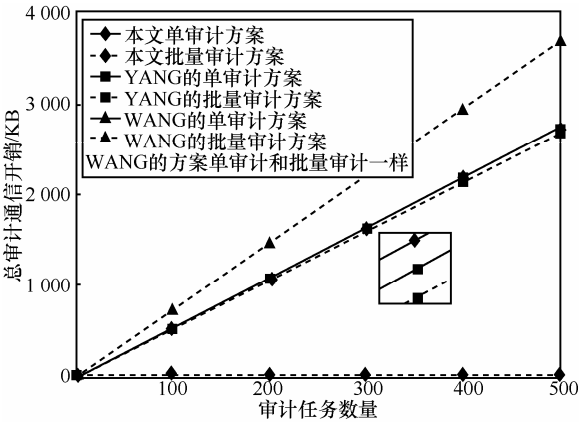
从图 4(b)可以看出，本文批量审计方案所用的通信开销是与审计任务数量无关，而 YANG 的批量审计方案和 WANG 的批量审计方案所需的通信开销与审计任务的数量成正比关系。本文批量审计方案所需的通信开销固定为 27.25 KB，比 YANG 的批量审计方案和 WANG 的批量审计方案所需的通信开销要少。当审计任务数量 K 为 500 时，Yang 的批量审计方案和 WANG 的批量审计方案所需的通信开销分别高达 2 695.4 KB 和 3 671.9 KB。

3) 单审计与批量审计的效率对比。图 5 给出验证不同数量的审计任务时，单审计和批量审计的效率对比关系。在单审计和批量审计测试中，质询数据块的数量 c 都是 500。

从图 5(a)可以看出，随着审计任务数量的增加，批量审计的平均审计时间是趋于稳定的。WANG 的批量审计方案的平均审计时间和单审计时间分别是 1 080 ms 和 1 198 ms，WANG 的批量审计方案提高了 9.8%的审计效率。Yang 的批量审计方案的平均审计时间和单审计时间分别是 970 ms 和 1 113 ms，YANG 的批量审计方案提高了 12.8%的审计效率。本文的批量审计方案的平均审计时间和单审计时间分别是 760 ms 和 940 ms，本文的批量审计方案提高了 19.1%的审计效率。因此，本文的批量审计方案更有效地提高了审计效率。



(a) 审计时间对比



(b) 通信开销对比

图 5　单审计与批量审计的效率对比

从图 5(b)可以看出，随着审计任务数量的增加，3 种单审计方案所需的总的通信开销也会增加，

YANG 的批量审计方案和 WANG 的批量审计方案所需的通信开销也是线性增加的，而本文的批量审计方案所需的通信开销是不变的。其中，WANG 的单审计的通信开销和批量审计的通信开销是一样的，所以图中 WANG 的单审计和批量审计的线条重合。YANG 的单审计通信开销比批量审计的通信开销是差不多的，所以图中 YANG 单审计和批量审计的线条几乎也是重合的。当审计任务数量 K 为 500 时，WANG 的单审计方案的总通信开销和批量审计的通信开销都是 3 671.9 KB ，YANG 的单审计方案的总通信开销和批量审计的通信开销分别是 2 714.8 KB 和 2 695.4 KB，本文的单审计方案的总通信开销和批量审计的通信开销分别是 2 724.6 KB 和 27.25 KB。因此本文的批量审计将大大减少通信开销。

9 结束语

本文主要研究了云存储中数据完整性问题，提出了一种聚合盲审计方法。首先，设计了第三方存储审计框架，数据所有者委托审计者检查云服务器中数据的完整性。利用双线性对的性质，对数据证据和标签证据加密后再合并，将合并后的结果直接发送给审计者，审计者在不需要解密也不能解密的情况下，直接使用合并后的结果验证数据的完整性，这样审计者无法获取数据内容，实现了盲审计。接着，提出了利用索引表支持高效安全的数据更新操作。在数据标签中加入版本号和时间戳等信息，防止服务器实施重放攻击和伪造攻击，保证数据更新操作不会带来新的安全问题。最后，提出了支持多云多所有者多文件的批量审计方法，将多个任务合并处理减少审计时间，将证据进行聚合进一步大大减少了审计通信开销。详细的安全性分析表明，本文的方案是可证明安全的。实验结果和分析表明，与现有的方案相比，本文方案有效提高了审计效率。特别是，批量审计的通信开销与任务数无关。

参考文献：

[1] ARMBRUST M, FOX A, GRIFFITH A, *et al*. A view of cloud computing[J]. Commun ACM, 2010, 53(4): 50-58.

[2] 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.

[3] BAIRAVASUNDARAM L N, GOODSON G R, PASUPATHY S, *et al*. An analysis of latent sector errors in disk drives[A]. Proc of ACM SIGMETRICS Int'l Conf Measurement and Modeling of Computer Systems[C]. 2007. 289-300.

[4] SCHROEDER B, GIBSON G A, disk failures in the real world:what does an MTTF of 1 000 000 hours mean to you[A]. Proc of USENIX Conf File and Storage Technologies[C]. 2007.1-16.

[5] WANG H, ZHANG Y. On the knowledge soundness of a cooperative provable data possession scheme in multicloud storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(1): 264-267.

[6] NI J, YU Y, Mu Y, *et al*. On the security of an efficient dynamic auditing protocol in cloud storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(10):2760-2761.

[7] ATENIESE G, BURNS R, CURTMOLA R, *et al*. Provable data possession at untrusted stores[A]. Proc of the 14th ACM Conference on Computer and Communications Security[C]. 2007. 598-609.

[8] JUELS, KALISKI B S. PORs: Proofs of retrievability for large files[A]. Proc of ACM CCS[C]. 2007.584-597.

[9] SHACHAM H, WATERS B. Compact proofs of retrievability[A]. Proc of the 14th International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology, ASIACRYPT'08[C]. Berlin, Heidelberg, 2008. 90-107.

[10] WANG C, WANG Q, REN K, *et al*. Privacy-preserving public auditing for data storage security in cloud computing[A]. Proc of IEEE INFOCOM[C]. 2010. 525-533.

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

[12] WANG C, WANG Q, REN K, *et al*. Toward secure and dependable storage services in cloud computing[J]. IEEE Transactions on Services Computing, 2012, 5(2):220-232.

[13] ZHU Y, HU H, AHN G J, *et al*. Cooperative provable data possession for integrity verification in multi-cloud storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(12):1-14.

[14] HE K, HUANG C H, WANG J H, *et al*. An efficient public batch auditing protocol for data security in multi-cloud storage[A]. Proc of China Grid[C].2013.51-56.

[15] YANG K, JIA X. An efficient and secure dynamic auditing protocol for data storage in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(9):1717-1726.

[16] ATENIESE G, PIETRO R D, MANCINI L V, *et al*. Scalable and efficient provable data possession[A]. Proc of the 4th International Conference on Security and Privacy in Communication Networks[C]. 2008. 1-10.

[17] ERWAY C, KUPCCU A, PAPAMANTHOU C, *et al*. Dynamic provable data possession[A]. Proc of the 16th ACM Conference on Computer and Communications Security[C]. 2009. 213-222.

[18] WANG Q, WANG C, LI J, *et al*. Enabling public verifiability and data dynamics for storage security in cloud computing[A]. Proc of ESORICS[C]. 2009. 355-370.

[19] WANG Q, REN K, Yu S, *et al*. Dependable and secure sensor data storage with dynamic integrity assurance[J]. ACM Transactions on

Senor Networks (ToSN), 2011, 8(1):1-24.

[20] WANG C, WANG Q, REN K, *et al*. Enabling public verifiability and data dynamics for storage security in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(5): 847-859.

[21] ZHU Y, WANG H, HU Z, *et al*. Dynamic audit services for integrity verification of outsourced storage in clouds[A]. Proc of ACM Symposium on Applied Computing[C]. 2011. 1550-1557.

[22] ZHU Y, HU Z, AHN G J, *et al*. Dynamic audit services for outsourced storages in clouds[J]. IEEE Transactions on Services Computing, 2013, 6(2):227-238.

[23] BARSOUM A, HASAN A. Enabling dynamic data and indirect mutual trust for cloud computing storage systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(12): 2375-2385.

[24] LIU C, CHEN J, YANG L, *et al*. Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(9): 2234-2244.

[25] WANG B, LI B, LI H. Panda: Public auditing for shared data with efficient user revocation in the cloud[J]. IEEE Transactions on Services Computing , 2014.

[26] YUAN J, YU S. Efficient public integrity checking for cloud data sharing with multi-user modification[A]. Proc of IEEE INFOCOM[C]. 2014. 2121-2129.

作者简介：

何凯（1987-），男，湖北黄冈人，武汉大学博士生，主要研究方向为云存储安全等。

黄传河[通信作者]（1963-），男，湖北随州人，武汉大学教授、博士生导师，主要研究方向为云计算、移动互联网、移动 ad hoc 网络、无线传感器网络、无线 mesh 网络、物联网、网络安全、分布并行处理。E-mail:huangch@whu.edu.cn。

王小毛（1984-），男，湖北天门人，武汉大学博士生，主要研究方向为移动 ad hoc 网络、计算机图形学等。

王晶（1986-），女，广西桂林人，武汉大学博士生，主要研究方向为云计算与物联网中的安全与隐私保护。

史姣丽（1979-），女，山西运城人，武汉大学博士生，主要研究方向为访问控制等。