doi:10.11959/j.issn.1000-436x.2017269

基于收敛加密的云安全去重与完整性审计系统

郭晓勇[1,2]，付安民[1,2]，况博裕[1]，丁纬佳[1]
（1. 南京理工大学计算机科学与工程学院，江苏 南京 210094；
2. 贵州大学贵州省公共大数据重点实验室，贵州 贵阳 550025）

摘 要：云存储应用以其便利性、可扩展性等优势迅速成为个人用户和企业存储的不二选择，但安全去重与完整性审计是云存储面临的关键问题。首先提出了基于盲签名的收敛密钥封装与解封算法，在安全存储收敛密钥的同时可以实现收敛密钥去重，提高了云存储空间利用率。另一方面，提出了基于收敛密钥的 BLS 签名算法，并利用可信第三方（TTP）存储审计公钥和代理审计，实现了对审计签名和审计公钥的去重，减轻了客户端存储和计算负担。在此基础上，进一步设计与实现了一个基于收敛加密的云安全去重和完整性审计系统。该系统能为云存储提供数据隐私保护、重复认证、审计认证等安全服务，且进一步降低了客户端、云端的存储和计算开销。
关键词：去重；收敛密钥；密钥管理；代理审计
中图分类号：TP393 **文献标识码**：A

Secure deduplication and integrity audit system based on convergent encryption for cloud storage

GUO Xiao-yong[1,2], FU An-min[1,2], KUANG Bo-yu[1], DING Wei-jia[1]
（1. School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China;
2. Guizhou Provincial Key Laboratory of Public Big Data, GuiZhou University, Guiyang 550025, China)

Abstract: Cloud storage applications quickly become the best choice of the personal user and enterprise storage with its convenience, scalability and other advantages, secure deduplication and integrity auditing are key issues for cloud storage. At first, convergent key encapsulation/decoupling algorithm based on blind signature was set up, which could securely store key and enable it to deduplicate. Besides, a BLS signature algorithm based on convergence key was provided and use TTP to store public key and proxy audit which enables signature and pubic key deduplication and reduces client storage and computing overhead. Finally, cloud-based secure deduplicaion and integrity audit system was designed and implemented. It offered user with data privacy protection, deduplication authentication, audit authentication services and lowered client and cloud computation overhead.
Key words: deduplication, convergent key, key management, proxy audit

1 引言

随着云计算技术的飞速发展，大量企业和个人选择将数据外包给云服务提供商。据 IDC 分析，云存储空间预计在 2020 年达到 40 万亿 GB。与此同时，EMC 的研究显示大约有 75%的云存储空间被重复数据所占有[1]。此外，确保云端数据的正确性与完整性也是用户迫切的诉求。因此安全去重和完整性审计是云存储数据安全研究的两大热点。

为解决远程数据去重面临的单一散列值代表整个文件的攻击，Halevi 等[2]提出了拥有权证明（PoW, proof of ownership）协议，基于证明者和验证者两者间交互认证的 PoW 协议可以确保证明者拥有指定的文件。杨等[3]提出了 MRN-CDE 方案来保证 PoW 证据的新鲜性，但是未考虑文件密钥的存储问题。Li 等[4]就基于收敛密钥数据去重[5,6]引起的密钥存储问题提出了 2 个方案，首先利用用户随机选取的主密钥以对称加密方式加密块级收敛密钥，并由云服务器存储密钥密文，从而避免存储大量的块收敛密钥；Li 进一步利用 Ramp 密钥共享和

收稿日期：2017-09-27

门限的思想，将收敛密钥分解后存储到 n 个密钥服务器中，只有当获得大于或等于 k 个密钥服务器的数据后，才能恢复出该收敛密钥。Chen 等[7]基于信息锁（message-locked-encryption）[8]提出了块级去重方案 BL-MLE，方案中认证标签封装了块密钥，解决了块密钥的存储问题。但是，其对于不同文件中相同数据块的重复检测效率低，而且也没有考虑文件级密钥同样存在的存储问题。

为能够高效地实现远程数据完整性验证，Ateniese 等[9]提出了可证明数据持有（PDP, provable data possession）模型，通过“挑战—应答”协议来实现数据的完整性验证。为了减少用户验证数据完整性所付出的开销，一个第三方审计者（TPA, third party auditor）被提出，然而，TPA 是不完全可信的，它可能在数据审计过程中窥探用户的隐私[10~14]。Yuan 等[15]针对用户间签名私钥的不同会导致重复文件的签名在云端多次存储的问题，提出在文件去重后，将此文件的所有签名标签聚合，只为该文件存储一份签名标签，减少重复签名标签带来的开销，但是此方法在签名标签重复情况下并没有减少客户端的计算开销以及通信开销。Li 等[16]提出引入第三方集群，通过第三方集群所生成的唯一私钥来为重复文件生成同样的签名标签，从而使云端只存储一份签名标签。但引入的第三方并没有考虑数据隐私问题。Liu 等[17]利用 MLE 和代理重签名的思想，实现了用户间审计标签的去重，同时降低了由此带来的公钥存储开销。但是审计过程须有用户参与，而且为支持用户间的相同文件的审计，云端存储了与用户量线性相关的代理重签名密钥，增加了用户计算开销和云存储开销。

本文设计与实现了一个基于收敛加密的云安全去重和完整性审计系统，具体贡献如下。

1) 利用盲签名的思想构建了安全的收敛密钥封装/解封算法，引入可信第三方 TTP 来执行封装环节，通过统一的 TTP 盲化封装，相同的文件会得到相同的收敛密钥密文，因此对收敛密钥密文实现了去重，既保证了收敛密钥安全，也节省了收敛密钥密文的冗余存储空间。

2) 利用基于收敛密钥的 BLS 签名[18]方案为文件生成 PDP、PoW 认证证据，对于相同明文数据，云端只要存一份认证证据即可提供两种安全服务。收敛密钥作签名私钥也使用户间的相同明文的认证证据实现了去重。此外，通过可信第三方 TTP 存储签名公钥并代理审计，避免了用户间存储冗余的签名公钥，也节省审计开销。

3) 设计与实现了一个基于收敛密钥云安全去重和审计系统，并对各环节会引起的数据冗余开销给予优化，确保了云可以高效、安全地为用户提供存储服务。

2　算法设计

2.1　基于盲签名的密钥封装/解封算法

基于盲签名的密钥封装/解封算法可安全高效地解决云端密文去重中引入的收敛加密所引起的密钥管理问题。

1) 密钥封装算法

密钥封装算法利用了盲签名[19]隐藏消息内容的特性。通过可信第三方对用户生成的收敛密钥进行盲签名，保证密钥安全的同时，对密钥实现了统一封装。算法核心内容包含：系统参数初始化、收敛密钥生成、随机数生成、盲化密钥、签名、去盲化。算法详情如下。

算法 1　密钥封装算法

输入　F 文件明文，U 用户标识

输出　CK 已封装的密钥

```
begin
1)   (e, G1,G2, g1, y1, y2, H(·))=system_ini()
2)   execute client_procedure
3)        k=generate_key(F)
4)        r=random()
5)        a=blind(k, r, g1)
6)        send(a, TTP)
7)        execute TTP_procedure
8)          while(TTP_check_request())
9)             b=sign(a, x)
10)            return(b, U)
11)         end while
12)       CK=remove_bilnd(b, r)
end
```

Step1　系统执行初始化操作系统随机选择一个素数 q 并创建 q 阶椭圆曲线方程 G_1，G_2，并且映射关系 $e{:}G_1\times G_1\to G_2$，g_1 是 G_1 的生成元，并构造散列函数 $H(\cdot)\to\{0,1\}^*\in G_1$，系统的公开参数为 $\{e,g_1,H(.),G_1,G_2\}$。TTP 随机一个选择整数 $x\in Z_q$ 作为私钥，并计算，$y_2=g_1^{x}$，$y_2=g_1^{x^{-1}}$，TTP 公开公钥参数 y_1，以及参数 y_2，并保持 x 私有。

Step2~Step6 执行客户端程序。首先，用户根据文件明文 F 计算得出收敛密钥 $k=H(F)$；随机选择 r 作为盲化因子，$r\in Z_q$；然后用户根据盲化因子 r、系统参数 g_1、收敛密钥 k 得出盲化后的收敛密钥 $a=kg_1^r$；用户将 a 发送至 TTP 端。

Step7~Step11 执行 TTP 端程序。TTP 利用私钥 x 对 a 进行签名得到 $b=a^x$。并将 b 返回给用户 U。

Step12 执行客户端程序。用户对 b 进行去盲化处理，得到封装后的收敛密钥 CK，其构成为 $CK=by_1^{-r}$，并通过构造的双线性映射关系 $e(CK,g_1)\stackrel{?}{=}e(k,y_1)$ 验证 CK 的签名合法性，若验证成功，则 CK 为密钥密文，用户可以安全地将 CK 外包存储至云。

2) 密钥解封算法

密钥解封算法是密钥封装算法的逆运算，该算法核心内容包括：随机数生成、盲化、签名、去盲化。算法详情如下。

算法 2 密钥解封算法

输入 CK 已封装的密钥

输出 K 收敛密钥

begin

1) $(e, G_1, G_2, g_1, y_1, y_2, H(\cdot))=get\_syspara()$
2) execute client_procedure
3) $r=random()$
4) $a=blind(CK,r,g_2)$
5) $send(a, TTP)$
6) execute TTP_procedure
7) while ($TTP\_check\_request()$)
8) $b=sign(a,x^{-1})$
9) return(b,U)
10) end while
11) $k=remove\_bilnd(b,-r)$

end

Step1 用户首先获取系统的公共参数，其中包含参数$\{e, G_1, G_2, g_1, y_1, y_2, H(\cdot)\}$，以便实现解封功能。

Step2~Step5 执行客户端程序。用户随机选择 r 作为盲化因子，$r\in Z_q$；然后用户依公共参数 g_1，对密钥密文 CK 盲化，得到 $a=CK\cdot g_1^r$；然后用户将 a 上传至 TTP 端。

Step6~ Step10 执行 TTP 端程序。当 TTP 检测到请求，接收 a，并利用私钥 x，计算得出签名后的密钥 $b=a^{x^{-1}}$，并将 b 返回给用户。

Step11 执行客户端程序。用户收到 TTP 的返回值 b 后，则进行去盲化处理得到 $k=by_2^{-r}$，通过双线性映射关系 $e(CK,g_1)\stackrel{?}{=}e(k,y_1)$ 验证签名的合法性，若验证成功，则 k 即为收敛密钥。

2.2 基于收敛密钥的 BLS 签名算法

基于收敛密钥的 BLS 签名算法，利用收敛密钥的特性，可使相同明文数据产生同样的 PDP 认证证据，达到了签名去重的目的。同时通过利用 TTP 来存储文件签名公钥并代理审计，进一步节省了客户端的存储和计算开销。本算法的核心内容：系统初始化、公私钥生成、密文生成、签名生成等。算法详情如下。

算法 3 基于收敛密钥的 BLS 算法

输入 F 文件明文

输出 $\sigma_i(1\leqslant i\leqslant n)$ 文件签名

begin

1) $(e,G_1,G_2,g_2,H(\cdot),h(\cdot))=system\_ini()$
2) $(ssk, spk)=generate\_signkeypair(F, syspara)$
3) $k=generate\_enckey(F)$
4) $(B_1, B_2,\cdots, B_n)=split\_file(F)$
5) $C_i=Enc(B_i, k)\ (1\leqslant i\leqslant n)$ //对称加密
6) $\sigma_i=sign(ssk, C_i)\ (1\leqslant i\leqslant n)$

end

Step1~Step2 系统执行初始化操作。系统随机选择素数 q 并创建 q 阶椭圆曲线方程 G_1，G_2。并且映射关系 $e:G_1\times G_1\to G_2$，g_2 是 G_1 的生成元，并构造两个散列函数：$H(\cdot)\to\{0,1\}^*\in G_1$，$h(\cdot)\to\{0,1\}^*\in Z_q$，同时系统对公开参数进行发布$\{e, G_1, G_2, g_2, h(\cdot), H(\cdot.)\}$；用户利用映射函数 $h(\cdot)$和文件明文 F 生成签名私钥 $ssk_F=h(F)$，进而得出签名公钥 $spk_F=g_2^{sskF}$。

Step3~Step5 客户端为生成签名做准备。用户根据文件明文 F 计算得出收敛密钥 $k=H(F)$；然后用户切割文件 F 为相同大小的文件块 $B_i(1\leqslant i\leqslant n)$，并利用密钥 k 加密 B_i 为密文块 $C_i(1\leqslant i\leqslant n)$。

Step6 执行签名程序。用户利用系统参数$\{g_2,H(\cdot)\}$、密文块 $C_i(1\leqslant i\leqslant n)$、签名私钥 ssk_F 生成签名集 $\sigma_i=\left[H(i)g_2^{C_i}\right]^{ssk_F}(1\leqslant i\leqslant n)$。

3 系统设计与实现

系统采用 B/S 架构开发，系统包含 3 个实体。客户端、云服务器、可信第三方。三方相互协作使

系统能够提供远程数据安全去重、远程数据审计的安全服务。本系统利用 JPBC 中的库函数来进行算法实现。

3.1　系统框架

客户端为用户提供数据操作、数据加密的基本服务，并且额外提供本系统特色的远程安全去重、远程审计、密钥管理、身份判定等安全服务。TTP 一方面可以封装密钥，另一方面执行代理审计，并记录审计日志。云服务器主要针对可信第三方和客户端对应的模块进行管控，并提供认证服务、安全存储服务。系统设计框架如图 1 所示。

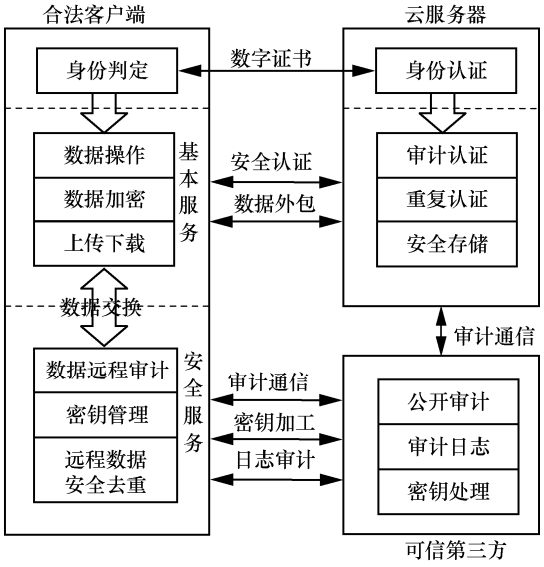


图 1　系统总体结构

客户端主要功能模块有：数据操作模块、数据加密模块，再辅之以身份判定模块、数据远程审计模块、远程数据安全去重模块、密钥管理模块来为用户提供更加安全可靠的数据存储服务。其中，数据操作模块主要负责切分数据、计算文件密文的唯一标签；数据加解密模块是利用密钥加密/解密数据；数据密钥管理模块是基于盲签名的密钥封装算法的实现部分，主要负责密钥生成、密钥封装、密钥密文解封；数据远程审计模块主要是初始化审计证据、公钥外包存储 TTP、请求审计；远程数据安全去重模块主要是远程数据重复检测、远程数据重复认证协议。

可信第三方主要功能模块有：密钥处理、公开审计、审计日志管理。密钥处理模块封装用户端产生的收敛密钥，以便安全存放至云端；公开审计主要是代理用户执行审计过程中数据的挑战集计算及证据验证；审计日志管理模块主要负责记录已审计的数据信息，可延伸至审计提醒、历史查询等功能。

云服务器端主要功能模块有：身份认证、审计认证模块、重复认证模块、安全存储模块。安全存储模块主要是为用户提供安全的存储区域存储数据密文、密钥密文、PDP 和 PoW 认证证据；审计认证模块主要是 TTP 代理用户向云端以挑战—响应的方式求证文件的完整性；重复认证模块则是用户与云端之间以挑战—响应方式来进行文件拥有权认证。

3.2　系统初始化

系统随机选择一个素数 q 并创建 q 阶的椭圆曲线方程 G_1，G_2，并产生一个可接受的线性配对 $e{:}G_1 \times G_1 \to G_2$，$g_1$、$g_2$ 是 G_1 的 2 个不同的生成元，令散列函数 $H(\cdot) \to \{0,1\}^* \in G_1$，散列函数 $h(\cdot) \to \{0,1\}^* \in Z_q$，即系统部分公开参数为 $\{e,G_1,G_2,g_2,h(.),H(.)\}$。TTP 随机选择 $x \in Z_q$，令私有参数 $tsk=x$ 并保持私有，计算，$y_2 = g_1^{x}, y_2 = g_1^{x^{-1}}$，令 $tpk=\{y_1, y_2\}$并添加至系统参数中。

3.3　文件上传

文件上传涉及 3 个实体多个模块的协同工作，具体包含客户端身份判定、密钥操作模块、证据操作、数据操作等模块，TTP 端的密钥封装模块，以及云服务器端所对应管控模块、存储模块。文件上传的整体流程如图 2 所示。

用户首先为文件 F 计算收敛密钥 $k=H(F)$，然后加密得到文件 F 的密文 C_F，即 $C_F=Enc(k,F)$。进而得出文件唯一标识 $Tag=Sha1(C_F)$，上传 Tag 至云服务器进行去重检测，云端管控模块对应部分检索是否存在标识符 Tag。如果重复存在，则云端重复认证模块与用户执行 PoW(详见文件去重部分)。如果认证成功，用户不需上传文件密文、PDP 和 PoW 证据、收敛密钥密文至云端，并且也不需上传文件公钥至 TTP，云端只需将文件的所有权授予该用户。否则，首先执行系统初始化操作；其次执行第 2.1 节提出的收敛密钥封装算法，算法执行完成后得到收敛密钥密文 CK_F；同时，利用 2.2 节提出的基于收敛密钥的 BLS 签名算法初始化安全认证所需的证据，证据包含文件的 PoW 重复证据、PDP 审计证据，算法执行完毕后获得文件 F 的认证证据 $\sigma_i\ (1 \leqslant i \leqslant n)$，$spk_F$；同时数据操作模块可计算得

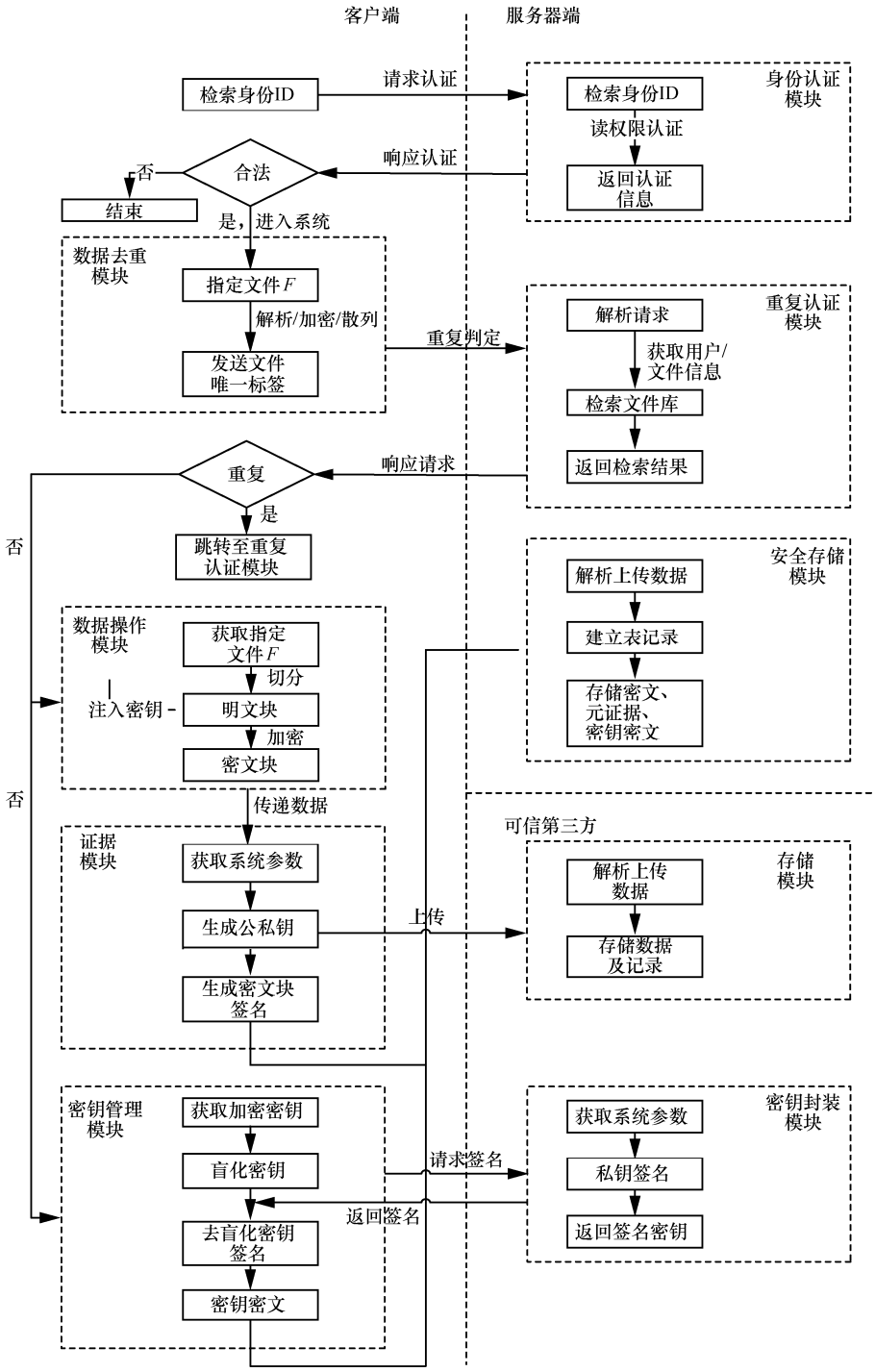


图 2 文件上传的整体流程

出文件 F 的密文 C_F 及密文块 C_i ($1 \leqslant i \leqslant n$)。至此，客户端上传准备就绪，用户可上传 C_F，C_i ($1 \leqslant i \leqslant n$)，$\sigma_i$ ($1 \leqslant i \leqslant n$)，$CK_F$ 至云，同时上传文件公钥 spk_F 至 TTP。

3.4 文件审计

文件审计流程与 PDP 协议一致，本系统中用户是审计的发起者，TTP 是审计的实际执行者。审计认证证据的构造方式是基于收敛密钥的 BLS 算法。文件审计详情如下。

1) 用户向系统请求审计指定的文件 F。

2) TTP 接收到用户请求后，解析请求信息，获取到用户身份和文件标识，然后根据文件 F 的总块数 n，从 1~n 中随机生成 c 个数，组成数据块索引集合 $I=\{s_1,s_2,\cdots,s_c\}$，且对于 $\forall s_i,s_j \in I\,(i \neq j)$，$s_i$ 和 s_j 是相

互独立的。对于$\forall i\in I$，随机选择 $v_i \in Z_q$，组成挑战信息$chal=(i,v_i)_{i\in I}$，并将挑战集 *chal*、用户标识 *U*、文件标识 *Tag* 一起发送给云端。

3) 云端收到来自 TTP 端的审计请求后，解析得到用户标识 *U* 和文件标识 *Tag* 后，检索并得到文件审计证据 σ_i、密文块 C_i，云端根据 *chal*、σ_i、C_i 计算得出对应的聚合证据 $\sigma=\prod_{(i,v_i)\in I}\sigma_i^{v_i}$，以及聚合密文 $\mu=\sum_{(i,v_i)\in I}C_i v_i$。然后云端将{$\mu$，$\sigma$}、用户标识 *U*、文件标识 *Tag* 返回至 TTP。

4) TTP 收到来自云端反馈的内容后，解析并获取到用户标识 *U*、文件标识 *Tag*，然后依文件标识 *Tag* 检索到公钥信息 spk_F。然后 TTP 验证等式 $e(\sigma,g_2)\stackrel{?}{=}e(\prod_{(i,v_i)\in I}H(i)^{v_i}g_2{}^{\mu},spk_{_F})$ 是否成立，若左右相等，则表示用户存储在云端文件是完整的。否则，表示该文件受损。然后 TTP 将审计结果记录至日志表中，并返回审计结果至客户端。等式的正确性可通过式(1)得出。

$$\begin{aligned}e(\sigma,g_2)&=e(\prod_{(i,v_i)\in I}\sigma_i^{v_i},g_2)\\&=e(\prod_{(i,v_i)\in I}[H(i)g_2{}^{C_i}]^{ssk_F\cdot v_i},g_2)\\&=e(\prod_{(i,v_i)\in I}H(i)^{v_i}\prod_{(i,v_i)\in I}g_2{}^{C_i\cdot v_i},g_2{}^{ssk_F})\\&=e(\prod_{(i,v_i)\in I}H(i)^{v_i}g_2{}^{\sum_{(i,v_i)\in I}C_i\cdot v_i},g_2{}^{x})\\&=e(\prod_{(i,v_i)\in I}H(i)^{v_i}g_2{}^{\mu},spk_{_F})\end{aligned}\tag{1}$$

3.5 文件去重

系统中重复认证证据的构造方式也是基于收敛密钥的 BLS 算法，具体步骤如下。

1) 云服务器解析用户的请求后，发现有重复文件 *F* 的上传请求。然后云服务器获取文件 *F* 的记录，根据文件 *F* 的总块数 *n*，从 1~*n* 中随机生成 *c* 个数，组成 $I=\{s_1,s_2,\cdots,s_c\}$，且对于$\forall i\in I$，随机选择 $v_i\in Z_q$，组成挑战信息$chal=(i,v_i)_{i\in I}$，并将 *chal*、文件标识 *Tag* 一起发至指定用户。

2) 用户收到来自云服务器的去重挑战，解析挑战集 *chal*，获取到文件的块索引集 *I*。用户按系统设置大小切分文件 *F*，得到 $B_i(1\leqslant i\leqslant n)$，并利用收敛密钥 *k* 加密文件块 $B_i\ (i\in I)$，得到 $C_i\ (i\in I)$，然后用户利用密钥模块生成的签名私钥 ssk_F、系统参数 g_2，对密文块 $C_i\ (i\in I)$进行签名得到文件块签名 $\sigma_i'=\left[H(i)g_2{}^{C_i}\right]^{ssk_F}(i\in I)$，进一步聚合得到 $\sigma'=\prod_{(i,v_i)\in I}\sigma_i'^{v_i}$，然后用户将 σ'、用户标识 *U*、文件 *F* 标识一起发至云端等待重复认证。

3) 云端收到响应后，解析得到聚合签名 σ'，服务器利用元证据 σ_i、*chal* 计算得到 σ，$\sigma=\prod_{(i,v_i)\in I}\sigma_i^{v_i}$，并验证$\sigma\stackrel{?}{=}\sigma'$，如果两者相同，则说明重复认证成功，云服务器将此用户 ID 添加此文件的所有权列表中。否则，重复认证失败，用户上传文件。云服务器将重复认证结果反馈给用户。

4) 用户收到反馈信息，若显示认证成功，则用户无需上传文件数据；否则用户执行文件上传。

4 实验与结果分析

本文实验平台包括 1 台 PC 机和 2 台服务器。其中 PC 机部署客户端程序，一台服务器部署可信第三方服务器，另一台部署云服务器。服务器为阿里云提供的 ECS 实例，同时利用其提供的对象存储服务 OSS 以及关系型数据库服务 RDS。ECS 实例配置详情：4 核 CPU、8 GB 内存、40 GB 云盘、CentOS 6.5 操作系统；本地 PC 机的配置：Intel(R) Core i5-4590 CPU @ 3.30 GHz 3.30 GHz 处理器、8 GB 内存、Windos 10/64 位操作系统，编程环境为：Eclipse oxygen IDE、Apache Tomcat8.x、MySQL5.6。

4.1 系统对比分析

本系统与其他方案对比结果如表 1 所示。

表 1　本系统与其他方案对比

方案	数据隐私	密钥管理	密钥去重	标签去重	拥有权证明	公开审计
文献[4]方案	✓	✓	✓			
文献[16]方案			✓	✓	✓	✓
文献[17]方案	✓			✓	✓	✓
本文	✓	✓	✓	✓	✓	✓

表 1 分别针对数据隐私、密钥管理、密钥去重、标签去重、拥有权证明、公开审计指标与文献[4,16,17]中的方案作对比，其中，标签去重指去重认证或者审计认证中所用到的 BLS 签名去重。从表 1 中可看出本系统所提供的安全服务相较与其他方案而言，安全保障覆盖范围更为广泛，而且本系统对云服务器面临的密文重复、收敛密钥重复、签名重复等问题给出了针对性的方案。在保证云提供安全服务的同时，减少冗余数据的存储，提高存储空间的利用率。

4.2　实验验证分析

实验验证主要针对公开审计、安全去重的有效性进行验证，各模块依系统设计而实现，通过多次调试，各模块功能正常，达到性能稳定要求，如图 3 和图 4 所示。

图 3　云服务器重复认证

图 4　TTP 服务器代理审计

5　结束语

本文针对当前云存储面临的隐私保护以及信任危机问题，设计与实现了一个基于收敛加密云安全去重和完整性审计系统。本系统对于用户和云提供者而言，都可以从中获益。对于用户而言，系统提供数据隐私保护、安全去重、完整性审计服务的同时，降低了重复收敛密钥、重复认证证据的计算开销。对于云存储提供者而言，在为用户提供安全服务的同时，节省了重复的认证证据以及重复收敛密钥密文的存储空间。

参考文献：

[1] GANTZ B J, REINSEL D. Big data, bigger digital shadows, and biggest growth in the far east executive summary: a universe of oppor tunities and challenges[C]//Idc.2007: 1-16.

[2] HALEVI S, HARNIK D, PINKAS B, et al. Proofs of ownership in remote storage systems[C]//The 18th ACM conference on Computer and communications security. 2011: 491-500.

[3] 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. New method for file deduplication in cloud storage[J]. Journal on Communications, 2017, 37(3):25-33.

[4] LI J, CHEN X F, LI M Q, et al. Secure deduplication with efficient and reliable convergent key management[J]. IEEE Trans on Parallel and Distributed Systems, 2014, 25(6); 1615-1625.

[5] DOUCEUR J R, ADYA A, BOLOSKY W J, et al. Reclaiming space from duplicate files in a serverless distributed file system[C]//22nd International Conference on Distributed Computing Systems, 2002: 617-624.

[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(1 1) :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, 31(1 1):169-180.

[7] CHEN R, MU Y, YANG G, et al. BL-MLE: block-level message-locked encryption for secure large file deduplication[J].IEEE Trans actions on Information Forensics and Security, 2015, 10(12): 2643-2652.

[8] BELLARE M, KEELVEEDHI S, RISTENPART T. Message-locked encryption and secure deduplication[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2013: 296-312.

[9] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//The 14th ACM conference on Computer and communications security. 2007: 598-609.

[10] 黄龙霞, 张功萱, 付安民.基于层次树的动态群组隐私保护公开审计方案[J].计算机研究与发展, 2016, 53(10): 2334-2342.
HUANG L X, ZHANG G X, FU A M. Privacy-preserving public auditing for dynamic group based on hierarchical tree[J]. Journal of Computer Research and Development, 2016, 53(10): 2334-2342.

[11] LI Y, FU A, YU Y, et al. IPOR: an efficient IDA-based proof of retrievability scheme for cloud storage systems[C]//2017 IEEE International Conference on Communications (ICC). 2017: 1-6.

[12] FU A, YU S, ZHANG Y, et al. NPP: a new privacy-aware public auditing scheme for cloud data sharing with group users[J]. IEEE Transactions on Big Data, 2017.

[13] 付安民, 秦宁元, 宋建业, 等.云端多管理者群组共享数据中具有隐私保护的公开审计方案[J] .计算机研究与发展, 2015, 52(10): 2353-2362.
FU A M, QIN N Y, SONG J Y. Privacy-preserving public auditing for multiple managers shared data in the cloud [J][J]. Journal of Computer Research and Development, 2015, 52(10): 2353-2362.

[14] HUANG L, ZHANG G, FU A. Certificateless public verification scheme with privacy-preserving and message recovery for dynamic group[C]//The Australasian Computer Science Week Multiconference.

2017: 76.

[15] YUAN J, YU S. Secure and constant cost public cloud storage auditingwith deduplication[C]//2013 IEEE Conference on Communications and Network Security (CNS). 2013: 145-153.

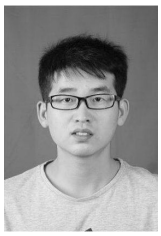
[16] LI J, LI J, XIE D, et al. Secure auditing and deduplicating data in cloud[J].IEEE Transactions on Computers, 2016, 65(8): 2386-2396.

[17] LIU X, SUN W, LOU W, et al. One-tag checker: message-locked integrity auditing on encrypted cloud deduplication storage[J]. IEEE International Conference on Computer Communications, 2017.

[18] SHACHAM H, WATERS B. Compact proofs of retrievability[C]// Asiacrypt. 2008: 90-107.

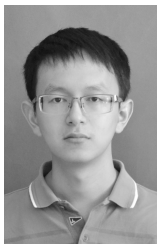
[19] BOLDYREVA A. Threshold signatures, multi signatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme[C]// International Workshop on Public Key Crypto graphy. 2003: 31-46.

作者简介：



郭晓勇（1993-），男，山西忻州人，南京理工大学硕士生，主要研究方向为云存储安全。

付安民（1981-），男，湖北通城人，博士，南京理工大学副教授、博士生导师，主要研究方向为无线网络安全、云计算、大数据安全等。



况博裕（1994-），男，四川绵阳人，南京理工大学硕士生，主要研究方向为物联网安全。

丁纬佳（1995-），女，浙江杭州人，南京理工大学本科生，主要研究方向为云存储安全。