doi:10.11959/j.issn.1000-436x.2017139

共享所有权证明：协作云数据安全去重新方法

熊金波[1,2]，李素萍[1]，张媛媛[1]，李璇[1]，叶阿勇[2]，姚志强[1,3]

（1. 福建师范大学软件学院，福建 福州 350117；2.福建省网络安全与密码技术重点实验室，福建 福州 350007；
3. 福建省公共服务大数据挖掘与应用工程技术研究中心，福建 福州 350117）

摘 要：针对共享文件的安全访问与去重问题，提出共享所有权证明（PoSW）的新概念，对其进行形式化定义，构造 PoSW 方案并对其进行扩展。在 PoSW 方案中，充分利用共享文件散布、收敛加密、秘密共享等技术对共享文件进行变换处理，实现对共享文件的分享授权；提出一种新的挑战—响应协议，实现共享所有权证明，为共享文件的安全去重提供基础；引入多云服务提供商对 PoSW 进行扩展，通过复制策略和安全文件散布策略提高不同类型共享文件的可用性和可靠性。安全性分析和性能分析表明所提 PoSW 方案是安全和高效的。

关键词：共享所有权证明；安全去重；收敛加密；安全文件散布；云存储

中图分类号：TP309.2 **文献标识码**：A

PoSW: novel secure deduplication scheme for collaborative cloud applications

XIONG Jin-bo[1,2], LI Su-ping[1], ZHANG Yuan-yuan[1], LI Xuan[1], YE A-yong[2], YAO Zhi-qiang[1,3]

(1.Faculty of Software, Fujian Normal University, Fuzhou 350117, China;
2.Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou 350007, China;
3.Fujian Engineering Research Center of Public Service Big Data Mining and Application, Fuzhou 350117, China)

Abstract: In order to solve the problems of secure access and deduplication to the shared file in the cloud environment, a novel notion called proof of shared ownership (PoSW) was formalized, and a formal definition of the PoSW was given. Furthermore, a PoSW scheme and an enhanced version of that were proposed. In the PoSW scheme, secure shared file dispersal, convergent encryption and secret sharing algorithm were employed to transform the shared file realize the sharing and authorization for the shared file, and then a novel challenge−response protocol was proposed to achieve the proof of shared ownership and the foundation for the secure deduplication of the shared file was provided. An enhanced PoSW scheme was designed to improve the availability and reliability for different kinds of the shared files by introducing the multi-cloud server providers and using the strategies of both data duplication and secret file dispersal. Security analysis and performance evaluation show the security and efficiency of the proposed scheme.

Key words: proof of shared ownership, secure deduplication, convergent encryption, secure file dispersal, cloud storage

1 引言

随着云计算、大数据技术的快速发展，越来越多的企业与个人利用云计算平台进行协同工作，以实现资源共享、无缝沟通，提高服务质量[1,2]。如工程设计领域的易图云是一个中国工程设计云服务

收稿日期：2016-11-10；**修回日期**：2017-04-27
通信作者：姚志强，yzq@fjnu.edu.cn
基金项目：国家自然科学基金资助项目（No.61402109, No.61370078,No.61502102, No.61502103）；福建省自然科学基金资助项目（No.2015J05120, No.2017J05099）；福建省网络安全与密码技术重点实验室开放课题基金资助项目（No.15008）；福建省高校杰出青年科研人才培育计划基金资助项目（No.2015）

Foundation Items: The National Natural Science Foundation of China (No.61402109, No.61370078,No.61502102, No.61502103), The Natural Science Foundation of Fujian Province (No.2015J05120,No.2017J05099), Fujian Provincial Key Laboratory of Network Security and Cryptology Research Fund (No.15008), Distinguished Young Scientific Research Talents Plan in Universities of Fujian Province (No.2015)

平台，为每个项目的建设方、工程总监和监理师等提供无缝协作服务；办公领域的 eteams、阿里云办公平台等均为一站式办公协作云平台，可实现随时随地与成员共同推进目标成果、与团队实时分享等协作服务；在教育学习领域有云教学平台、AWS 教育云等；在医疗健康领域有百度的健康云智能医疗平台、医疗健康云等。由此可见，协作云应用已经在人们生活的各个领域得到广泛应用并蓬勃发展。协作云应用平台采用多用户协同参与的工作模式，其协作过程中产生的共享文件由多参与者共同所有。与普通文件由单用户拥有、可单方面设定访问权限不同，共享文件的访问、决策控制等都要求参与者共同决定。在共享文件的产生、使用、访问、更新和存储等过程中，都将产生多副本文件，增加云平台的存储开销。

为了提高存储效率和实现盈利，云服务提供商需采用有效的数据缩减方法来存储用户数据。最直观的方式是数据压缩技术[3]，但不同的压缩算法将得到不同的压缩文件，一份文件需存储多个副本，浪费大量存储空间。重复数据删除（数据去重）作为一种高效的数据缩减方法，被各大云服务提供商广泛使用，如 DropBox、Google Drive 和阿里云 OSS 等。Harnik 等[4]根据去重数据粒度的不同将数据去重分为文件级去重与块级去重。根据数据去重的实施主体可以分为面向目标的数据去重、面向源的数据去重和跨用户的数据去重[5]。文献[6]显示跨用户的数据去重可以节约超过 50%的存储空间，甚至可以达到 90%~95%的去重概率[7]。从技术实现角度，云数据安全去重问题主要基于收敛加密（CE, convergent encryption）[8~10]、消息锁加密（MLE, message-lock encryption）[11~13]和密钥更新算法[14,15]。然而，这些算法的安全性都依赖原始文件的散列值，即用户只需拥有文件的散列值就可以试图向云服务器获取原始文件，而散列值本身容易遭受离线蛮力攻击。

为了解决上述攻击，Halevi 等[16]提出了一种所有权证明（PoW, proof of ownership）方案，用户对需上传的文件进行纠删码编码，服务器使用 Merkle 树对文件进行存储与管理，该方案可以帮助云服务器确认用户是否真实拥有请求的文件，实验结果表明该方案具有较大的计算开销。为了进一步提高 PoW 的有效性，Pietro 等[17]提出了一种 s-PoW 方案，通过广泛的基准测试和与现有方案的对比分析可知该方案具有较小的开销，能够安全高效地实现所有权证明。Blasco 等[18]提出了一种 bf-PoW 方案，运用布隆过滤器进一步提高服务器端的效率和灵活性。González-Manzano 等[19]提出了一种基于收敛加密的 ce-PoW 方案，该方案无可信第三方、没有复杂密钥管理开销，云服务器存储一个由采用 CE 加密每个文件块的密文、挑战、应答和客户身份列表组成的四元组，在随机预言机模型下该方案可证明是安全的，仿真实验结果表明，与其他 PoW 方案相比，该方案具有较小开销，能有效实现 PoW。

上述方案均能实现对数据的安全去重与所有权证明。然而，上述方案的研究对象是只针对单用户完全拥有所有权和管控权的普通文件，无法适应协作云应用中多用户共同拥有所有权的共享文件。因此，Soriente 等[20]提出一种针对共享文件的共享访问控制策略，但并未考虑对共享文件的所有权证明与去重问题。本文结合所有权证明、秘密共享和收敛加密等技术，提出一种面向协作云服务的共享所有权证明（PoSW, proof of shared ownership）新方案，实现协作云应用中对共享文件的安全访问、所有权证明与安全去重。本文主要贡献如下。

1) 形式化定义 PoSW，在此基础上提出一种 PoSW 方案。首先，对共享文件块进行收敛加密，将密文块的标识符秘密共享成 n 份并由 n 个共享所有者共同存储，对共享文件的访问必须得到其他 $t-1$（$t<n$）个共享所有者的共同授权；然后，构造一种验证共享所有权的挑战—响应新协议，从而实现协作云应用中对共享文件的所有权校验与安全去重。

2) 引入多个云服务提供商对 PoSW 方案进行扩展，依据密文块引用率的不同，通过复制策略和安全文件散布策略对共享文件进行合理的分散、冗余存储，以提高共享文件的可用性与可靠性，防止单点失效问题。

3) 安全性分析表明所提 PoSW 方案是安全的，不需要复杂的密钥管理；性能分析和仿真实验结果表明 PoSW 方案是高效的，且能大幅节约云服务器的存储空间。

2　基础知识

2.1　安全文件散布

安全文件散布由 Soriente 等[20]提出，首先，将文件编码成 n 个文件块，然后，从中随机获取 t 个

文件块，便可以解码恢复成原始文件。安全文件散布的安全性体现在任意用户获得少于 t 个文件块将无法恢复出原始文件。它主要包含以下 2 个算法：$\{bl_1,bl_2,\cdots,bl_n\}\leftarrow SF.Encode(t,n,f,k,\lambda)$，该算法将一份文件 f 编码成 n 个文件块，k 为编码过程中的加密密钥，λ 为安全参数；$f'\leftarrow SF.Combine(k,Bl,\lambda)$，该恢复算法将 t 个文件块 $Bl\subseteq\{bl_1,bl_2,\cdots,bl_n\}$ 解码恢复成 f'，如果 $|Bl|\geqslant t$，则 $f'=f$。

在 PoSW 方案中，运用算法 *SF.Encode* 与 *SF.Combine* 编码和恢复共享文件块标识符 *token*。

2.2 秘密共享

秘密共享[21,22]将一份秘密文件拆分成 n 份共享数据，只有得到大于或等于 t 份共享数据才能恢复出原始文件，主要由 2 个算法组成：$\{s_1,s_2,\cdots,s_n\}\leftarrow SF.Share(s,t,n)$，该算法使用纠删码将一份秘密数据拆分成同等长度的 n 份共享数据；$s'\leftarrow SF.Recover(\{s_1,s_2,\cdots,s_d\})$，该算法将 d 份共享数据恢复成原始秘密文件，当 $d\geqslant t$ 时，有 $s'=s$。

在 PoSW 方案中，运用算法 *SF.Share* 与 *SF.Recover* 秘密共享和恢复共享文件的加密密钥 k。

2.3 收敛加密

Douceur 等[8]提出了一种收敛加密算法，其加密密钥由原始数据经过散列运算获得。假设用户需要上传一份文件 f，首先，计算密钥（即对文件 f 进行散列运算）

$$k=H(f) \tag{1}$$

从而，相同的文件得到相同的加密密钥。服务器根据客户端上传的 $H(f)$ 检查是否已经存储该文件，若没有，则告知客户端上传文件，若已存储，则客户端对文件进行加密

$$C_f=E_{H(f)}(f) \tag{2}$$

然后，将 C_f 与 $H(f)$ 上传给服务器存储；若服务器已存储该文件，则客户端无需再次上传文件 f，从而大幅节省服务器存储空间。当客户端需要下载该文件时，只需上传 $H(f)$ 给服务器，服务器根据 $H(f)$ 将对应的密文 C_f 传给客户端，客户端进行解密

$$f=D_{H(f)}(C_f) \tag{3}$$

在 PoSW 方案中，为了实现共享文件的安全去重，使用 CE 算法加密共享文件块。

3 共享所有权证明

本节首先介绍 PoSW 方案中常用的符号与相应的描述，如表 1 所示。然后介绍系统模型和威胁模型，给出 PoSW 的形式化定义，在此基础上，给出 PoSW 方案的具体构造。最后对 PoSW 方案分策略进行扩展。

表 1 符号及其描述

符号	符号描述
f	协作云应用产生的共享文件
$f[i]$	共享文件 f 的第 i 个文件块
z	共享文件 f 的长度
b	文件块长度
k_i	第 i 个文件块的加密密钥，$k_i=H_2(f[i])$
$\Theta[i]$	共享文件的第 i 个密文块
$id[i]$	第 i 个所有者的 *id*
token	文件块标识符 $token[i]\leftarrow H_2(\Theta[i])$
n	*token* 的数量
h_f	$h_f\leftarrow H_1(token)$，共享文件的标识符
SO_i	共享文件的第 i 位所有者
H_1,H_2,H_3	抗碰撞散列函数
λ	安全参数

在表 1 中，抗碰撞散列函数 H_1：$\{0,1\}^*\rightarrow\{0,1\}^x$，其中，$x$ 为正整数；函数 H_2：$\{0,1\}^b\rightarrow\{0,1\}^l$，其中，$b$ 为文件块的长度，l 为 *token* 的长度；函数 H_3：$\{0,1\}^q\rightarrow\{0,1\}^m$，其中，$q$ 为所有者 *id* 长度，m 为散列 *id* 后的长度。

3.1 系统模型

本文构造一种 PoSW 方案并对其进行扩展，系统模型如图 1 所示。PoSW 方案包含多个共享所有者和一台云服务器。

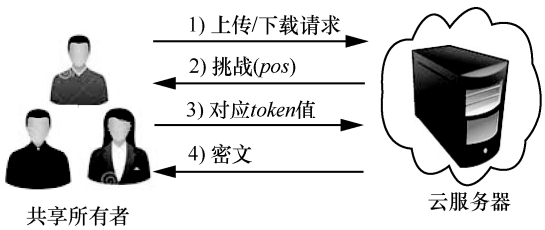


图 1 PoSW 系统模型

共享所有者包含一个共享文件创建者与其他多个共享所有者，共享文件创建者负责创建共享文件。若某个共享所有者想访问共享文件，必须得到至少 $t-1$ 个其他所有者的授权，并成功通过云服务器发送的挑战—响应协议的验证。云服务器的主要

任务为存储共享文件，建立与该文件相关联的挑战信息。当共享所有者请求下载文件时，实施 PoSW 协议，发出挑战并校验该所有者是否挑战成功，若成功，则返回相应信息给该所有者。

3.2　威胁模型

本文假设云服务器是“诚实且好奇”的服务器，即它会诚实地运行 PoSW 方案的协议，不会主动篡改共享文件所有者上传的数据，但它会尽可能地试图获取共享文件的隐私信息。本文还假设共享文件创建者是安全可信的，共享所有者是“诚实且好奇”的，即它会诚实地运行 PoSW 方案的协议，但它会与敌手共谋，提供自己所知的密钥分量和 *token* 分量信息。

基于以上假设，本文考虑敌手会对 PoSW 协议进行攻击，如在少于 t 个共享所有者的授权下访问共享文件，或与云服务器共谋获得敏感信息。同时，基于 s-PoW[17]和 ce-PoW[19]的信息理论，假设 64 MB 的文件块大小足够抵抗上述共谋攻击[17,19]。但是，PoSW 不允许敌手在交互过程中使用合法共享所有者作为交互协议预言机获得 PoSW 挑战的正确响应。

本文的主要设计目标是敌手成功通过 PoSW 校验的概率是关于安全参数 λ 的可忽略函数，且云服务器不知道任何关于共享文件的敏感信息。

3.3　形式化定义

PoSW 是云服务器与共享所有者之间的一种交互协议，交互的桥梁为 S_t，即共享文件的 *token* 值。在 PoSW 协议中，某个所有者（共享文件提供者 P）拥有 S_t，云服务器（共享文件验证者 V）根据 S_t 产生挑战消息 *pos*，然后根据该所有者发回的响应信息确定其是否通过 PoSW 验证。具体地，为了得到 S_t，共享所有者（分量请求者 E）需向其他所有者（分量发送者 S）请求共享文件的密钥与 *token* 分量。PoSW 可以抽象为随机输入 S_t 与安全参数 λ 的函数 $F_{\mathrm{S}}(\cdot)$、随机输入共享所有者 *id* 与安全参数 γ 的函数 $F_{\mathrm{V}}(\cdot)$、双方交互协议 $\mathcal{B}$(P↔V)与多方交互协议 $\mathcal{H}$(E↔S)。

1) 有效性。方案 $SP=(F_{\mathrm{S}},\mathcal{B})$是有效的，当且仅当满足：① F_{S} 与 $\mathcal{B}$ 是有效的；② 对于每个共享文件的所有者 $id\in\{0,1\}^{\gamma}$ 与安全参数 γ，$\mathcal{H}(\mathrm{E}(id,1^{\gamma})\leftrightarrow \mathrm{S}(F_{\mathrm{V}}(id,1^{\gamma})))\Rightarrow token$ S_t 成立；③ 对于每个共享文件的 $S_t\in\{0,1\}^{\lambda}$ 与安全参数 λ，$\mathcal{B}(P(S_t,1^{\lambda})\leftrightarrow \mathrm{V}(F_{\mathrm{S}}(S_t,1^{\lambda})))\Rightarrow Accept$(校验通过)成立。

2) 效率因素。PoSW 的效率因素主要体现在：① 参数 $pos=F_{\mathrm{S}}(S_t,1^{\lambda})$的长度；② 协议 $\mathcal{B}$ 与 $\mathcal{H}$ 的通信复杂度；③ 函数 F_{S} 与协议 $\mathcal{B}$、$\mathcal{H}$ 的计算复杂度。本文方案对安全参数的依赖是线性相关的，F_{S} 与 P 的计算复杂度与 S_t 的长度线性相关，$\mathcal{B}$ 的计算开销与共享所有者的个数线性相关，且其他函数在逻辑上均与 S_t 的长度线性相关。此外，由于协议为一次性交互协议，验证消息 *pos* 只需通过缓存读取一次，从而保证了协议的执行效率。

3.4　PoSW 方案构造

3.4.1　上传共享文件阶段

上传共享文件阶段如图 2 所示，该阶段包含以下 3 个步骤。

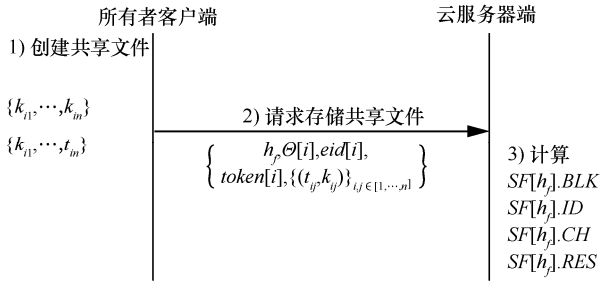


图 2　上传共享文件阶段

1) 创建共享文件。当共享所有者请求上传文件 f 时，云服务器根据其上传的 h_f 判断是否已经存储，若没有，则通知上传该共享文件。这时共享文件创建者创建文件，为了实现快速去重，本文将共享文件分为大小相等的文件块 $f[i]$进行操作，首先，采用 CE 算法加密文件块，即对每个文件块 $f[i]$进行散列运算得到收敛密钥 k_i。

$$k_i=H_2(f[i]) \tag{4}$$

接着，使用密钥 k_i 对文件块 $f[i]$进行加密得到密文块

$$\Theta[i]=E_{k_i}f[i] \tag{5}$$

并对 $\Theta[i]$ 使用 H_2 进行散列运算，得

$$token[i]=H_2\left(\Theta[i]\right) \tag{6}$$

然后，使用 H_1 对 *token* 进行散列运算得到文件标识符

$$h_f=H_1(token) \tag{7}$$

最后，通过安全文件散布算法和秘密共享算法将 *token* 和密钥进行分片处理，如式(8)和式(9)所示。

$$\{t_{i1},\cdots,t_{in}\}\leftarrow SF.Encode\left(t,n,token[i],k_i,\lambda\right) \tag{8}$$

$$\{k_{i1},\cdots,k_{in}\}\leftarrow SF.Share\left(k_i,t,n\right) \tag{9}$$

对于共享文件的其他所有者，文件创建者根据其 *id* 将 $\{(t_{i,j},k_{i,j})\}$ 上传至云服务器中该所有者对应 *id* 的信息库，因此，每个所有者都收到了文件的部分 *token*。除此之外，为了对共享所有者的 *id* 进行保护，需将每个所有者的 *id* 进行散列运算，得

$$eid[i]=H_3(id[i]) \tag{10}$$

2) 请求存储共享文件。共享文件创建者将 $\left\{h_f,\Theta[i],eid[i],token[i],\{(t_{ij},k_{ij})\}_{i,j\in[1,\cdots,n]}\right\}$ 上传至云服务器，请求云服务器存储该文件相关信息。

3) 存储共享文件。云服务器根据收到的信息建立与该文件关联的数组 *SF*，该数组 *SF* 包含一个四元组：*SF.BLK* 用来存储加密文件块 $\Theta[i]$；*SF.ID* 存储全部所有者 *id* 的散列值 *eid*[*i*]；*SF.CH* 存储预设的 1 000 个挑战 *CH*[*i*,*j*]，使用伪随机数生成器(PRNG)为每个挑战随机生成 *J* 个 *token* 的位置，且 *J*<*n*，函数 *PRNG* 以种子 *seed* 作为输入，*seed* 的生成方式参考 s-PoW[17]，挑战的具体计算为

$$CH[i,j]\leftarrow pos[j]\leftarrow PRNG(seed) \tag{11}$$

SF.RES 存储与挑战对应的响应信息 *RES*[*i*,*j*]。

$$RES[i,j]\leftarrow token[pos[j]] \tag{12}$$

当一个所有者请求文件时，云服务器将校验该所有者的 *eid* 是否存储在 *SF.ID* 中，如果存在，则正常执行该过程；否则，被认为是恶意用户，直接拒绝该请求。当 *SF.CH* 中的预设值都使用完之后，PoSW 将重新给 *SF.CH* 赋值。

3.4.2 共享所有权证明阶段

共享所有权证明阶段如图 3 所示，该阶段包含 8 个步骤。

1) 请求共享文件。共享所有者 SO_i 将 $\{h_f,eid_{SO_i}\}$ 上传给云服务器，请求下载共享文件。

2) 验证身份是否合法。云服务器检查 h_f 是否已经存储，如果 h_f 已经存在，则查找到相应的 *SF* 四元组，检查 eid_{SO_i} 是否存在于 *SF.ID* 中，若存在，则从 *SF.CH* 中选择一个挑战位置数组 *pos*。

3) 发送挑战。云服务器将挑战位置数组 *pos* 发送给 SO_i。

4) 计算 *token*。SO_i 收到 *pos* 后，向至少 *t*−1 个其他共享所有者获取 k_i 分量与 *token* 分量，其他所有者根据 SO_i 的 *id* 判断是否为共享文件所有者，若是，则将 k_i 分量与 *token* 分量发送至 SO_i，SO_i 收到相应分量后，加上自己拥有的第 *t* 个分量，并计算

$$k_i\leftarrow SF.Recover(\{k_{i1},\cdots,k_{it}\}) \tag{13}$$

$$token[pos]\leftarrow SF.Combine(k_i,\{t_{posj1},\cdots,t_{posjt}\},\lambda) \tag{14}$$

5) 上传与挑战位置数组 *pos* 相对应的响应数组 *res*。SO_i 上传 *res*=*token*[*pos*]数组给云服务器。

6) 验证。云服务器接收到所有者客户端发回的一个响应数组 *res* 后，验证该数组是否与原存储数组一致，如果一致，则 SO_i 成功通过了挑战，表明该所有者的确是该共享文件的所有者之一。

7) 发送结果。云服务器将密文块 $\Theta[i]$ 发送给 SO_i。

8) 解密文件块。SO_i 使用 k_i 解密文件块密文 $\Theta[i]$ 得到共享文件的明文块数据。

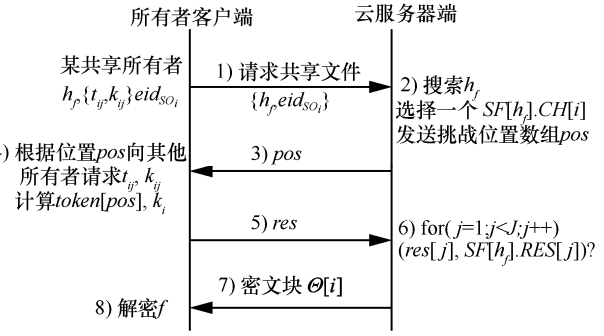


图 3 共享所有权证明阶段

在 PoSW 方案中，增加对共享文件所有者的身份验证，通过共享所有者之间分享 k_i 和 *token* 分量，实现对共享文件的安全访问与所有权证明。当云服务器接收到共享文件上传信息后，先根据文件标识符 h_f 依次检测对应的 *token* 值是否已经存储，若已存储，则说明该 *token* 对应的文件块已存储，无需再次上传；反之，则存储该文件块，最终实现对共享文件的块级去重。

3.5 PoSW 方案扩展

单一云服务提供商如果服务中断或遭黑客攻击[23]，则存储的数据可能面临巨大损失，此外，还将面临数据可用性低、可靠性差、安全性弱等诸多问题[24]。为了解决上述问题，本文对 PoSW 方案进行扩展，结合数据冗余[25]与散布技术，引入多云服务提供商，提高协作云应用中共享文件的可用性和可靠性。

将共享文件分散存储在多云服务提供商的主要优势是：当某个云服务提供商发生故障时，可以采用容错技术从其他云服务提供商安全重构出原共享文件，使数据更加安全可靠。在具体实施多云架构时，首先对上传到云端的共享文件执行基于 PoSW 方案的安全去重，在数据去重时，为每个存储的密文块设置引用率 r，当访问该密文块时，根据索引列表搜索该密文块，每访问一次，将 r 加 1。

本文针对该引用率采用如下 2 种策略对 PoSW 方案进行扩展：对引用率高的密文块采用基于复制的数据分布机制[26]，对引用率低的密文块采用基于安全文件散布[20]的数据分布机制。

3.5.1　基于复制的分布策略

共享所有者将共享文件上传至多云服务提供商，在整个访问和去重的过程中，各个密文块的引用率将发生改变，在变化过程中判断引用率是否超过阈值 R，若超过 R，则将高引用率的密文块复制成多份并分散存储到多云服务提供商，依据数据重要程度的不同，共享文件所有者可以自行设置不同的备份数量。当密文块所存储的云服务器中断或该密文块被破坏时，系统将依据备份数量从其他云服务提供商复制数据。

如图 4 所示，假设阈值 R 为 3，有 3 份共享文件 f_1、f_2、f_3 将上传至云服务器，A~F 分别表示不同的密文块，图 4 中密文块后的数字代表 3 份文件中相同密文块的数量。经过数据去重后，发现密文块 C 的引用率 r 达到阈值 R，假设备份数量为 3，则系统将复制 3 份密文块 C，分别存储到不同的云服务提供商（阿里云、百度云、有道云或易图云等），多云服务提供商可以根据需要将各个云服务提供商进行组合。

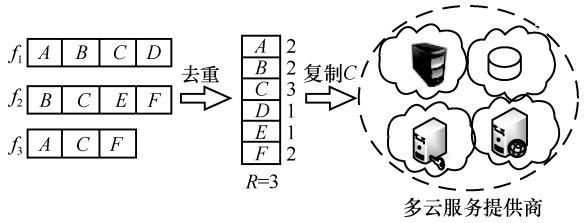


图 4　基于复制的数据块分布策略

3.5.2　基于安全文件散布的数据分布策略

共享所有者将共享文件上传至多云服务提供商，在访问和去重过程中，当数据块的引用率低于设定的阈值 R，则使用基于安全文件散布的分布策略进行存储。

本文采用 3.1 节安全文件散布[20]的数据分布策略将引用率低的密文块分散存储到多云服务提供商。安全文件散布的安全性体现在任意所有者获得低于 t 个数据分量将无法恢复出原始密文块。如图 5 所示，首先，由 3.4.1 节共享文件创建阶段的安全文件散布算法可知，对引用率低的某密文块 $\Theta[i]$ 进行编码后可以得到 n 个分量。

$$\{\Theta_{i1},\cdots,\Theta_{in}\} \leftarrow SF.Encode(t,n,\Theta[i],k_i,\lambda) \quad (15)$$

然后，伪随机数生成器(PRNG)随机产生 n 个索引，将这些分量根据索引分别存储到 n 个云服务提供商。当需要从多云架构中获取某密文块 $\Theta[i]$ 时，依据 3.4.2 节计算 *token* 步骤，根据存储索引从至少 t 个云服务提供商获取数据分量信息可重构原密文块。

$$\Theta[i] \leftarrow SF.Combine(k_i,\{\Theta_{i1},\cdots,\Theta_{in}\},\lambda) \quad (16)$$

其中，k_i 为密文块 $\Theta[i]$ 的加密密钥，λ 为安全参数。

上述多云存储策略针对块级的共享文件进行处理，同样适用于文件级处理，当共享文件的引用率高于某个阈值时，采用复制的策略，低于该阈值时采用安全文件散布的策略。

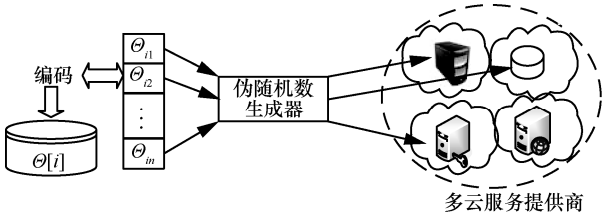


图 5　基于安全文件散布的数据块分布策略

4　安全性分析

PoSW 的安全性依赖 s-PoW[17]和 ce-PoW[19]建立的信息理论假设，对于共享文件 f，敌手的目标是成功通过云服务器发出的 PoSW 验证。根据威胁模型，敌手不允许在 PoSW 交互过程中使用合法共享所有者作为交互协议预言机获得 PoSW 挑战的正确响应。但为了通过验证，敌手在交互协议执行前可能跟某个共享所有者共谋。下面，将通过 2 个命题证明本文方案的安全性。

命题 1　敌手成功通过 PoSW 校验的概率 P(*token*)是关于安全参数 λ 的可忽略函数。

证明　为了建模 PoSW 交互过程并充分考虑敌

手可能知道共享文件 f 的部分信息，假设敌手知道该信息的概率为 p，即敌手知道一个数据字节在文件中任意位置的概率为 p；同时，假设敌手在不拥有特定数据字节的前提下，成功猜测其在共享文件中任意位置的概率为 g。对于一个长度为 b 的文件块 $f[i]$，相应地，用 p 来表示敌手知道整个文件块 $f[i]$中任意一个位置的字节的概率。

PoSW 的挑战阶段要求敌手可以获得随机选择的 J 个 *token* 值。每个 *token* 值被安全散布为 n 个分量，敌手至少需要获得 t 个分量才可重构为正确的 *token* 值。敌手收到一个随机的数组挑战时，定义 $token_i$ 为敌手得到正确的第 i 个文件块 $f[i]$事件，w_i 为敌手知道文件块 $f[i]$事件，其概率为 p。敌手知道文件块 $f[i]$后，即可对 $f[i]$进行散列运算得到 *token* 值，因敌手无法知道全部文件块信息，对于未知的 y 个独立比特，敌手猜测成功的概率为 g^y；或敌手可直接猜测长度为 l 的 *token* 值，每比特猜中的概率为 0.5，从而直接猜测 *token* 正确的概率为 0.5^l。又因为 l 的长度远小于文件块的长度，因此，直接猜测 *token* 的概率远小于得到文件块的概率，即 $g^y \ll 0.5^l$。故敌手得到一个 *token* 的概率可以描述为

$$\begin{aligned} &P\left(token_i\right) \\ &= P\left(token_i \cap \left(w_i \cup \overline{w_i}\right)\right) \\ &= P\left(token_i \mid w_i\right) P\left(w_i P\left(token_i \mid \overline{w_i}\right) P\left(\overline{w_i}\right)\right) \\ &= p + 0.5^l\left(1-p\right) \end{aligned} \tag{17}$$

对于 J 个独立的 *token* 事件，敌手成功的概率可以描述为

$$P(token)=P(token_i)^J=(p+0.5^l(1-p))^J \tag{18}$$

借鉴 s-PoW[17]和 ce-PoW[19]的分析，从式(18)可以看出，为了确保 $P\left(token\right) \leqslant 2^{\lambda}$ 成立，其中，λ 为安全参数，可以推断出 J 的下界为

$$J \geqslant \frac{\lambda \ln 2}{\left(1-p\right)\left(1-0.5^l\right)} \tag{19}$$

从式(19)可以看出，J 的大小不受文件长度的影响，只要 J 满足上述要求，则敌手成功通过 PoSW 校验的概率 $P(token)$是关于安全参数 λ 的可忽略函数，故满足 PoSW 方案的安全需求。

命题 2 若敌手知道小于某个值大小的文件信息，则通过所有权验证的概率为关于安全参数 λ 的可忽略函数，则可以抵抗敌手的共谋攻击。

证明 在 PoSW 中，为了抵抗敌手的共谋攻击，假设敌手需要知道至少 $z_{\min}$ 长度的比特值才能通过 PoSW。因为 z 为文件 f 的长度，b 为文件块的长度，则共有 $\left\lceil \frac{z}{b} \right\rceil$ 个 *token*，其长度 l 可以表示为

$$l \geqslant z_{\min}\left(\frac{b}{z}\right) \tag{20}$$

根据 ce-PoW[19]，文件的长度必须大于 $z_{\min}$，对于较小的共享文件，$z_{\min}$ 也相应减小。Halevi 经过充分的实验分析，其结果表明当文件大于 64 MB 时，若敌手知道小于 64 MB 的文件信息，则通过所有权验证的概率为关于安全参数 λ 的可忽略函数，从而将 $z_{\min}$ 值设为 64 MB，以抵抗敌手的共谋攻击[16]。在后续研究中，s-PoW[17]、bf-PoW[18]和 ce-PoW[19]等方案均将该值设为 64 MB，本文也采用该值以抵抗敌手的共谋攻击。

5 性能分析与评价

5.1 算法复杂度分析

对于算法复杂度分析，本文主要考虑 PoSW 方案中云服务器端与共享所有者客户端的计算开销与读写开销，并与现有相关方案进行对比分析，如表 2 所示。表 2 列出了文献[17]、文献[18]、文献[19]、PoSW 与 PoSW 扩展方案的算法复杂度。文献[17~19]的方案是最新关于验证单用户拥有文件所有权的 PoW 方案，与本文验证共享文件所有权的方案比较接近，因此，本文将与这 3 个经典方案进行对比分析。由于散列运算的开销相对较小，本文假设 3 个散列函数 H_1、H_2 和 H_3 具有相同的计算开销。

从表 2 中可以看出，在客户端计算开销方面，文献[17]与文献[18]方案仅需对整个文件进行散列运算得到文件标识符，开销较小，而 PoSW 及其扩展方案与文献[19]类似，需对文件先进行分块处理，再执行 2 次散列运算，此外，PoSW 还需对 *token* 与密钥进行安全文件散布与秘密共享操作，因此，PoSW 的客户端计算开销较大。PoSW 与其他方案一样，在客户端与云服务器初始化的读写开销主要为读取整个文件，因此，其大小取决于共享文件的长度。PoSW 在云服务器端初始化时仅对共享文件进行存储，而文献[17~19]方案需对文件或文件块进行散列运算，因此，PoSW 无云服务器初始化计算开销。PoSW 的常规计算开销为预先生成 Q 个挑战，生成挑战方式与文献[19]方案类似，因此，与文献[19]

表 2　　PoSW 与相关方案的开销对比分析

方案	客户端计算开销	客户端读写开销	云服务器初始化读写开销	云服务器初始化计算开销	云服务器常规计算开销	云服务器内存开销
文献[17]方案	$O(z)\cdot hash$	$O(z)$	$O(z)$	$O(z)\cdot hash$	$O(Q\cdot\lambda)\cdot PRF$	$O(Q\cdot\lambda)$
文献[18]方案	$O(z)\cdot hash$	$O(z)$	$O(z)$	$O(z)\cdot hash$	$O\left(\frac{l\cdot\lambda\left(\lg\frac{1}{p_f}\right)}{p_f}\right)\cdot hash$	$O\left(\frac{\lg\left(\frac{1}{p_f}\right)}{l}\right)$
文献[19]方案	$O(b)\cdot CE\cdot hash\cdot hash$	$O(z)$	$O(z)$	$O(b)\cdot hash\cdot hash$	$O(Q\,l\lambda)\cdot PRNG$	$O(Q\,l\lambda)$
PoSW	$O(b)\cdot CE\cdot hash\cdot hash\cdot SF$	$O(z)$	$O(z)$	$O(0)$	$O(Q\,l\lambda)\cdot PRNG$	$O(Q\,l\lambda)$
PoSW 扩展方案 1	$O(b)\cdot CE\cdot hash\cdot hash\cdot SF$	$O(z)$	$O(z)$	$O(0)$	$O(Q\,l\lambda)\cdot PRNG$	$O(Q\,l\lambda)$
PoSW 扩展方案 2	$O(b)\cdot CE\cdot hash\cdot hash\cdot SF$	$O(z)$	$O(z)$	$O(0)$	$O(Q\,l\lambda)\cdot PRNG\cdot SF$	$O(Q\,l\lambda)$

方案相同。云服务器常规计算开销取决于生成 Q 个挑战和与之对应的响应，即长度为 l 的 *token* 值。在云服务器内存开销方面，PoSW 与文献[19]需要存储 Q 个预先计算的挑战，每个挑战需要存储对应的 λ 个长度为 l 的 *token* 值，即相应的响应信息。

在表 2 中，z 表示文件长度，b 表示文件块的长度，Q 表示预先计算挑战的数量，λ 表示安全参数，p_f 表示 BF 的误判率，l 表示 *token* 的长度，R 表示阈值，*hash* 表示执行一次散列函数所需开销，*CE* 表示执行一次收敛加密所需开销，*PRF* 表示执行一次伪随机函数所需开销，*PRNG* 表示执行一次伪随机数生成器所需开销，*SF* 表示执行一次安全文件散布与秘密共享操作所需开销。

PoSW 扩展方案 1 与 PoSW 相比，只在存储共享文件方面对高引用率的密文块使用复制的分布策略，因此，表 2 中的开销均一致，此外，PoSW 扩展方案 1 还需增加多云服务提供商之间的通信开销 $O(z)$ 与存储开销 $O(z)$。PoSW 扩展方案 2 对低引用率的密文块使用安全文件散布的存储策略，因此，在云服务器常规计算开销中除了预先计算 Q 个挑战外，还需对密文块进行安全文件散布操作，并增加多云服务提供商之间的通信开销 $O(b)$，其他开销均与 PoSW 一致。

5.2　性能评价

本文采用 Linux 系统下 C++语言进行系统仿真实验，使用 OpenSSL 函数库，采用其中的 AES-256 与 SHA-256 对数据进行对称加密与散列运算。本文实验采用的测试计算机的配置如下：CPU 为 Intel Core i5-4590 3.30 GHz，RAM 为 DDR3 8 GB，磁盘为 WDC WD10EZEX-08M2NA0(1 TB/7 200 r/min)，操作系统为 Ubuntu 12.04.4 LTS。

本文实验将对 PoSW 方案的 2 个阶段进行时间开销计算，即上传共享文件与共享所有权证明，实验数据取 1 000 次实验的平均值。本文选择 8 种不同大小的共享文件进行实验，分别为 4 MB、8 MB、16 MB、32 MB、64 MB、128 MB、256 MB、512 MB，测试的时间单位为 ms，其实验结果如图 6 所示。

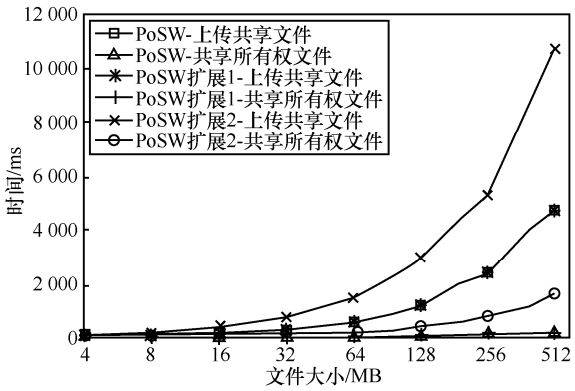


图 6　2 个阶段的计算开销

由图 6 可以看出，随着共享文件大小的增加，上传共享文件阶段的时间也随之增大，当共享文件增加到 512 MB 时，PoSW 方案上传时间仅为 4.7 s，在可接受范围内；PoSW 扩展方案 1 由于只在存储共享文件方面对高引用率的密文块使用复制的分布策略，因此，上传共享文件的时间开销与 PoSW 方案一致；PoSW 扩展方案 2 对低引用率的密文块使用安全文件散布的存储策略，在上传共享文件时需对密文块进行秘密共享编码处理，因此，所需存储时间比 PoSW 方案稍大，当共享文件为 512 MB 时，上传时间需 10.7 s。共享所有权证明阶段的时

间随着共享文件的增大而缓慢增加，PoSW 方案由于仅需完成 J 个 *token* 的合并与一次文件块的解密操作，相对上传阶段的时间有大幅减小，当共享文件增加到 512 MB 时，其证明阶段的时间仅需 0.18 s，具有较高的效率；PoSW 扩展方案 1 所需时间与 PoSW 一致；PoSW 扩展方案 2 需对密文块分量进行秘密共享解码处理，因此，所需时间比 PoSW 方案大，当共享文件为 512 MB 时，所需时间为 1.6 s。

本文实验还对共享所有者数量的变化进行效率测试，结果如图 7 所示，共享所有者数量 n 为 4~20 个，因为所有者数量不同，相同的文件分块后大小不同，为了防止文件块的大小影响实验数据准确性，本文实验采用同为 2 MB 的文件块。实验数据取 1 000 次实验的平均值，测试结果的时间单位为 ms。由图 7 可知，随着所有者数量的增大，上传共享文件阶段与共享所有权证明阶段的编码时间也相应增大，当所有者数量为 20 时，上传共享文件阶段的单个文件块编码时间约为 0.9 μs，同样具有较高的效率。PoSW 扩展方案对密文块使用的安全文件散布算法与上传共享文件阶段对数据块进行编码的算法相同，因此，密文块编码时间随着多云服务提供商个数的变化而变化的趋势与上传共享文件阶段数据块编码的时间随着共享所有者数量的变化趋势一致。

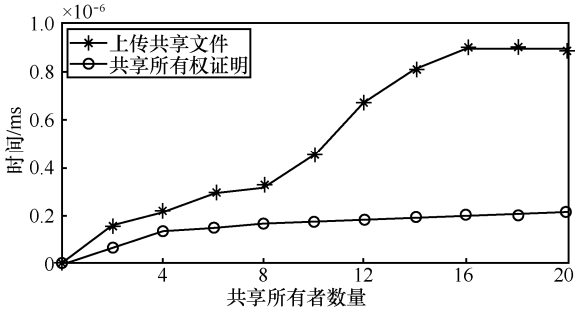


图 7　共享所有者数量与编码时间开销的关系

6　结束语

随着协作云应用的普及和日益发展，医疗会诊文件、联合项目申请等共享文件的安全访问与存储成为当前的研究热点。如何安全有效地存储这些共享文件，并对其实现安全访问与去重成为当前的研究热点。本文形式化定义共享所有权证明（PoSW）的概念，并构造一种 PoSW 方案。本文方案中，共享文件的所有者可以通过获得其他所有者授权的共享信息来计算相应的响应，以证明其确实是共享文件的所有者，在此基础上，实现对共享文件的安全去重。为了解决云服务器的单点失效问题，引入多云服务提供商，通过复制策略和安全文件散布策略对 PoSW 进行扩展，以提高共享文件的可用性和可靠性。安全性分析与性能分析表明所提方案是安全且有效的。

下一步工作将引入隐私计算相关理论[27]，将所提 PoSW 思想与隐私计算相结合，以博弈论安全模型为指导，探索云数据安全去重的隐私、可用、效率综合平衡机理。此外，还可以引入确定性删除与自毁相关理论[28, 29]，将安全去重与确定性删除相结合，构造全新的云安全存储系统。

参考文献：

[1] WU D, YANG B, WANG H, et al. Privacy-preserving multimedia big data aggregation in large-scale wireless sensor networks[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2016, 12(4): 1-19.

[2] XIONG J B, LI F H, MA J F, et al. A full lifecycle privacy protection scheme for sensitive data in cloud computing[J]. Peer-to-Peer Networking and Applications, 2015, 8(6): 1025-1037.

[3] MITTAL S, VETTERJ S. A survey of architectural approaches for data compression in cache and main memory systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(5): 1524-1536.

[4] HARNIK D, PINKASB, SHULMAN-PELEGA. Side channels in cloud services: deduplication in cloud storage[J]. IEEE Security & Privacy, 2010, 8(6): 40-47.

[5] 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.

[6] LIU J, ASOKAN N, PINKAS B. Secure deduplication of encrypted data without additional independent servers[C]//The 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015: 874-885.

[7] MEYER D, BOLOSKY W. A study of practical deduplication[J]. ACM Transactions on Storage (TOS), 2012, 7(4): 14-26.

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

[9] STANEK J, SORNIOTTI A, ANDROULAKI E, et al. A secure data deduplication scheme for cloud storage[C]//The Financial Cryptography and Data Security. 2014: 99-118.

[10] LI M, QIN C, LI J, et al. CDStore: toward reliable, secure, and cost-efficient cloud storage via convergent dispersal[J]. IEEE Internet Computing, 2016, 20(3): 45-53.

[11] BELLARE M, KEELVEEDHI S, RISRENPART T. Message-locked encryption and secure deduplication[C]//The Advances in Cryptology-EUROCRYPT 2013. 2013: 296-312.

[12] BELLARE M, KEELVEEDHI S, RISTENPART T. DupLESS: serv-

er-aided encryption for deduplicated storage[C]//The 22nd Usenix Conference on Security, Berkeley. 2013:179-194.

[13] 杨超, 纪倩, 熊思纯, 等. 新的云存储文件去重删除方法[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.

[14] LI J, QIN C, LEE P P C, et al. Rekeying for encrypted deduplication storage[C]//The 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. 2016: 618-629.

[15] QIN C, LI J, LEE P P C. The design and implementation of a rekeying-aware encrypted deduplication storage system[J]. ACM Transactions on Storage, 2017, 13(1): 9:1-9:30.

[16] 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.

[17] PIETRO R D, SORNIOTTI A. Boosting efficiency and security in proof of ownership for deduplication[C]//The 7th ACM Symposium on Information, Computer and Communications Security. 2012: 81-82.

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

[19] GONZÁLEZ-MANZANO L, ORFILA A. An efficient confidentiality-preserving proof of ownership for deduplication[J]. Journal of Network and Computer Applications, 2015, 50: 49-59.

[20] SORIENTE C, KARAME G, RITZDORF H, et al. Commune: shared ownership in an agnostic cloud[C]//The 20th ACM Symposium on Access Control Models and Technologies. 2015: 39-50.

[21] LI J, CHEN X, HUANG X, et al. Secure distributed deduplication systems with improved reliability[J]. IEEE Transactions on Computers, 2015, 64(12): 3569-3579.

[22] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613.

[23] 毛波, 叶阁焰, 蓝琰佳, 等. 一种基于重复数据删除技术的云中云存储系统[J]. 计算机研究与发展, 2015, 52(6): 1278-1287.
MAO B, YE G Y, LAN Y J, et al. A data deduplication-based primary storage system in cloud-of-clouds[J]. Journal of Computer Research and Development, 2015, 52(6):1278-1287.

[24] ABU-LIBDEH H, PRINCEHOUSE L, WEATHERSPOON H. RACS: a case for cloud storage diversity[C]//The 1st ACM Symposium on Cloud Computing. 2010: 229-240.

[25] 刘莎, 楚传仁. 基于文件等级的 Ceph 数据冗余存储策略的研究[J]. 信息网络安全, 2016(4): 50-54.
LIU S, CHU C R. Research on data placement strategy for Ceph based on file level[J]. Netinfo Security, 2016 (4): 50-54.

[26] BESSANI A, CORREIA M, QUARESMA B, et al. DepSky: dependable and secure storage in a cloud-of-clouds[J]. ACM Transactions on Storage (TOS), 2013, 9(4): 31-46.

[27] 李凤华, 李晖, 贾焰, 等. 隐私计算研究范畴及发展趋势[J]. 通信学报, 2016, 37(4): 1-11.
LI F H, LI H, JIA Y, et al. Privacy computing: concept, connotation and its research trend[J]. Journal on Communications, 2016, 37(4): 1-11.

[28] 熊金波, 李凤华, 王彦超, 等. 基于密码学的云数据确定性删除研究进展[J]. 通信学报, 2016, 37(8): 167-184.
XIONG J B, LI F H, WANG Y C, et al. Research progress on cloud data assured deletion based on cryptography[J]. Journal on Communications, 2016, 37(8): 167-184.

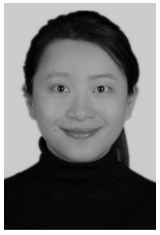
[29] XIONG J B, LIU X, YAO Z, et al. A secure data self-destructing scheme in cloud computing[J]. IEEE Transactions on Cloud Computing, 2014, 2(4):448-458.

作者简介：

熊金波（1981-），男，湖南益阳人，博士，福建师范大学副教授、硕士生导师，中国科学院信息工程研究所博士后，主要研究方向为云数据安全与隐私保护技术。

李素萍（1991-），女，福建三明人，福建师范大学硕士生，主要研究方向为云数据安全与隐私保护技术。

张媛媛（1992-），女，河南南阳人，福建师范大学硕士生，主要研究方向为云数据安全与隐私保护技术。



李璇（1984-），女，湖北黄石人，博士，福建师范大学副教授、硕士生导师，主要研究方向为信息安全。

叶阿勇（1977-），男，福建漳州人，博士，福建师范大学教授、硕士生导师，主要研究方向为隐私保护与网络安全等。

姚志强（1967-），男，福建莆田人，博士，福建师范大学教授、硕士生导师，主要研究方向为信息安全。