基于角色对称加密的云数据安全去重

熊金波[1,2]，张媛媛[2]，田有亮[1]，应作斌[3]，李琦[4]，马蓉[2]

（1. 贵州省公共大数据重点实验室（贵州大学），贵州 贵阳，550025；2. 福建师范大学数学与信息学院，福建 福州 350117；
3. 安徽大学计算机科学与技术学院，安徽 合肥 230601；4. 南京邮电大学计算机学院，江苏 南京 210023）

摘 要：云计算和大数据技术的飞速发展促使人们进入大数据时代，越来越多的企业和个人选择将数据外包至云服务提供商。数据量的爆炸式增长态势、占据大量存储空间以及庞大的管理开销给云存储带来巨大压力。同时，如何有效防止个人隐私泄露、实现授权访问、云数据安全去重以及密钥更新与权限撤销问题也给云服务提供商提出更大挑战。针对上述问题，提出一种角色对称加密算法，利用角色对称加密将用户角色与密钥相关联，构建角色密钥树，不同角色可根据访问控制策略访问对应权限的文件；同时，提出一种基于角色对称加密的云数据安全去重方案，有效保护个人隐私信息、实现分层结构下的云数据授权去重，并通过群组密钥协商解决角色与密钥映射关系中密钥更新与权限撤销等带来的安全问题。安全性分析表明所提角色对称加密算法和云数据安全去重方案是安全的，性能分析和实验结果表明所提安全去重方案是高效的。

关键词：角色对称加密；隐私保护；授权去重；重复数据删除；权限撤销
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.1000-436x.2018077

Cloud data secure deduplication scheme via role-based symmetric encryption

XIONG Jinbo[1,2], ZHANG Yuanyuan[2], TIAN Youliang[1], YING Zuobin[3], LI Qi[4], MA Rong[2]

1. Guizhou Provincial Key Laboratory of Public Big Data (Guizhou University), Guiyang 550025, China
2. College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China
3. College of Computer Science and Technology, Anhui University, Hefei 230601, China
4. School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Abstract: The rapid development of cloud computing and big data technology brings prople to enter the era of big data, more and more enterprises and individuals outsource their data to the cloud service providers. The explosive growth of data and data replicas as well as the increasing management overhead bring a big challenge to the cloud storage space. Meanwhile, some serious issues such as the privacy disclosure, authorized access, secure deduplication, rekeying and permission revocation should also be taken into account. In order to address these problems, a role-based symmetric encryption algorithm was proposed, which established a mapping relation between roles and role keys. Moreover, a secure deduplication scheme was proposed via role-based symmetric encryption to achieve both the privacy protection and the authorized deduplication under the hierarchical architecture in the cloud computing environment. Furthermore, in the proposed scheme, the group key agreement protocol was utilized to achieve rekeying and permission revocation. Finally, the security analysis shows that the proposed role-based symmetric encryption algorithm is provably secure under the standard model, and the deduplication scheme can meet the security requirements. The performance analysis and experimental results indicate that the proposed scheme is effective and efficient.

Key words: role-based symmetric encryption, privacy protection, authorized deduplication, data deduplication, permission revocation

收稿日期：2017-10-12；**修回日期**：2018-03-29

通信作者：田有亮，youliangtian@163.com

基金项目：国家自然科学基金资助项目（No.61772008, No.U1405255, No.61502248, No.61402109, No.61502489, No.61502103）；贵州省科技重大专项计划基金资助项目（No.20183001）；贵州省公共大数据重点实验室开放课题基金资助项目（No.2017BDKFJJ028）
Foundation Items: The National Natural Science Foundation of China (No.61772008, No.U1405255, No.61502248, No.61402109, No.61502489, No.61502103), The Science and Technology Major Support Program of Guizhou Province (No.20183001), Guizhou Provincial Key Laboratory of Public Big Data Research Fund (No.2017BDKFJJ028)

1 引言

随着云计算、大数据技术的不断发展，越来越多的企业和个人通过数据外包享受到云服务提供商经济高效的计算和存储服务。Excelcom公司发布的“互联网一分钟产生数据”信息显示，一分钟内Facebook共产生701 389个账号登录，1.5亿封电子邮件已发送，Google上产生240万个的搜索请求，Instagram平台上传243万多张照片。从信息单位的角度计算，2011年全世界每天发送的数据量达到40亿或更多，全球数据产生量达到1.8 ZB，IDC（international data corporation）的报告显示，2013年全球数据量已达到4.4 ZB，并且数据总量每年的增长速度也超过50%，预计到2020年，全球数据总量将超过44 ZB[1]。研究表明，超过一半的云存储空间被重复数据的副本占用，用于管理重复数据的预算开销是管理原数据本身的8倍。数据量的爆炸式增长态势、占据大量存储空间的重复数据以及庞大的管理开销给云存储空间带来巨大压力。因此，如何经济高效地存储和管理数据成为云服务提供商面临的严峻挑战。

为了提高存储效率、降低管理开销，重复数据删除技术（数据去重）被云服务提供商广泛采用。云服务器通过随机抽样、提取散列值等方法校验用户上传的数据是否已经存储，经验证，若用户新上传的数据与原存储数据相同则执行数据去重[1]。根据不同的分类标准，数据去重的分类结果也不尽相同。根据数据的处理单位，可以分为文件级数据去重和块级数据去重；根据数据去重的执行对象，可以分为基于目标的数据去重即服务器端数据去重、基于文件源的数据去重（即客户端数据去重）以及跨用户数据去重[2]。实验表明，跨用户数据去重将节省一半以上的存储空间，去重率达到90%~95%[3,4]。

然而，大数据时代下的企业与个人外包给云服务提供商的数据涉及大量隐私信息。因此，在保护用户隐私数据的同时实施数据安全去重是云服务提供商面临的新挑战，是否能够提供安全的数据去重服务也是满足用户外包数据需求的衡量标准之一。保护用户数据隐私的安全去重技术迅速成为云存储领域的研究热点，得到学术界和产业界的广泛关注。收敛加密（CE, convergent encryption）算法首先由Douceur等[5]提出，保证相同的数据经过散列运算及对称加密算法得到相同的密钥和密文。CE算法不仅满足对密文数据进行重复性检测的需求，而且有效地减少云存储空间的浪费，能够很好地适应云计算环境，如Puzio等[6]、Li等[7]和Stanek等[8]均结合CE算法实现云数据安全去重。为了达到语义安全，Bellare等[9]提出了一种消息锁加密（MLE, message-locked encryption）算法，Chen等[10]、Jiang等[11]、Li等[12]和Qin等[13]分别结合MLE算法实现数据安全去重。然而，传统的CE和MLE算法存在许多隐私泄露问题以及新的安全挑战。

1) 隐私泄露。云服务提供商在采用数据去重技术控制单个文件副本数量的同时，敌手可能利用云数据去重过程并通过相关攻击手段窃取用户的隐私信息，包括个体隐私和集体隐私。不仅如此，在云服务器执行去重检测的同时，用户的身份、位置信息以及用户之间重复数据的数量可能被泄露，这些隐私数据遭到泄露将严重阻碍云服务的健康发展，因此，在云数据去重过程中保护数据隐私尤为重要。

2) 未授权访问。基于传统内容加密算法的数据去重方案存在严重的安全漏洞，敌手仅通过上传文件的散列值即可通过离线蛮力攻击得到用户信息，难以保障云端用户隐私数据的安全性和授权访问。云环境中用户存在分层结构，不同层次的用户所拥有的权限也不同，对于云端存储的文件，只有拥有访问权限的用户才能访问。在执行云数据去重过程中，如何保证只有拥有特定权限的用户才能访问指定文件是亟待解决的一个关键问题。

3) 权限的更新与撤销。在实际应用中，执行数据去重的企业和个人的角色与需求灵活多变，这些用户的权限也是动态变化的。因此，需要合理的更新和撤销机制来及时处理用户的权限变更，保证用户的合法访问权限；同时，满足复杂多变的用户需求，更好地应用在实际生产生活中。在云数据授权访问过程中，用户权限的更新与撤销问题亟待解决。

综上所述，为了解决现有云数据安全去重方案存在的上述问题，本文提出一种基于角色对称加密的云数据安全去重方案。该方案设计一种角色对称加密算法将用户角色与密钥相关联，构建角色密钥树，满足不同角色根据访问控制策略访问对应权限文件的需求，实现角色分层结构下的云数据授权去重，并通过群组密钥协商解决角色与密钥映射关系中由密钥更新、权限撤销等带来的安全问题。本文

主要贡献如下。

1) 提出角色对称加密算法，建立分层角色密钥树映射用户角色关系，设计角色密钥生成函数和文件密钥生成函数获得角色密钥及文件密钥，使用户角色与其密钥相关联，为云数据安全去重提供理论基础。

2) 针对云环境下的隐私泄露、未授权访问等安全问题，提出一种基于角色对称加密的云数据安全去重方案，实现角色分层结构下不同角色用户对文件的授权访问与安全去重。

3) 针对角色与密钥映射关系中由于密钥更新、权限撤销等带来的安全问题，设计一种授权密钥更新机制，引入群组密钥协商协议，对角色密钥树进行处理，在保证授权访问和安全去重的基础上，实现授权密钥的更新和用户权限的撤销。

4) 安全分析表明，角色对称加密算法是可证明安全的，基于角色对称加密的云数据安全去重方案能够满足安全目标，性能分析和实验结果表明所提方案是高效的。

2　相关工作

国内外学者对云环境中数据安全去重问题进行了深入研究，并取得了一定的成果。Puzio 等[6]提出了一种基于 CE 算法的安全高效存储系统 ClouDedup，增加访问控制机制与语义安全加密算法实现数据块级去重。Stanek 等[8]将文件分为流行文件和非流行文件，分别对应不同的安全等级，针对这些不同安全等级的文件采用不同级别的加密算法。针对用户自定义文件安全等级过程中出现的安全隐患，Puzio 等[14]提出了一种 PerfectDedup 方案，使用完美散列函数获得数据块重要程度的索引，实现数据安全去重。Li 等[7]将 CE 算法与收敛扩散机制结合，提出了一种 CDStore 方案以实现数据安全去重，实验表明所提方案节省近 70%的存储开销。为了达到语义安全加密，Bellare 等[9]提出了一种 MLE 算法，并给出明确的安全性目标和严格的形式化定义来实现云数据安全去重；在此基础上，提出了一种跨用户的数据安全去重方案 iMLE[15]，采用交互消息锁加密算法实现关联文件的安全去重。Chen 等[10]提出了一种适用于大规模加密文件环境中的数据安全去重方案 BL-MLE，使用少量的元数据高效安全地实现文件级和块级数据去重。Jiang 等[11]提出了一种 R-MLE2 方案，采用随机化标识的方式实现跨用户的高效数据去重。为了解决密钥管理问题，Li 等[16]将密钥分布存储在多服务器中，Miao 等[17]提出了一种基于门限盲签名与可校验秘密共享机制的多密钥服务器数据去重方案，可以防止第三方的密钥服务器与云服务器合谋。针对密钥更新问题，Li 等[12]和 Qin 等[13]提出了一种通过更新密钥状态实现文件密钥更新的 REED 方案，结合 MLE 和 AONT-RS 秘密共享机制满足数据安全去重的要求。然而，上述结合 CE、MLE 等基于内容加密的数据去重方案易遭受离线蛮力攻击和侧信道攻击，存在隐私泄露和未授权访问等安全问题。

通过上述攻击，敌手仅依据上传文件散列值就可以猜测得到文件信息，为此，Halevi 等[18]提出了一种所有权证明（PoW, proof of ownership）的概念，服务器和客户端分别根据原文件生成 Merkle Hash Tree（MHT），由服务器验证客户端所返回的给定挑战对应的回答是否正确，进而得出所有权证明结果。为了减少客户端的计算开销，文献[19,20]提出了一种 s-PoW 方案，根据特定算法获得文件随机位置的比特值作为挑战，要求待验证的客户端返回对应结果，进而实现文件的所有权证明，扩展方案 s-PoW1 和 s-PoW2 有效提高了算法的执行效率。为了提高服务器端的计算和查询效率，Blasco 等[21]提出了一种基于布隆过滤器（BF, bloom filter）的 BF-PoW 方案，服务器端建立三元数组分别存储文件、挑战和应答，要求客户端上传一定数量的验证信息证明文件的所有权，实验表明，服务器加入布隆过滤器能够大幅减少计算开销。González-Manzano 等[22]提出了一种基于 CE 算法的所有权证明方案 ce-PoW，该方案无可信第三方参与，不涉及复杂密钥管理，服务器采用四元数据结构映射密文块、挑战、应答和身份标识，通过与客户端进行挑战—应答交互机制实现文件所有权证明。针对敌手或未授权用户利用侧信道访问文件信息的问题，Li 等[23]提出了一种混合云环境下的授权去重方案，文件密钥的生成与用户权限相关，实现了云环境下的授权去重及重复数据检测。González-Manzano 等[24]综合考虑授权去重问题和所有权证明方案，提出了一种 ase-PoW 方案，文件密钥的生成与用户的属性相关，使用轻量级的访问控制策略实现分层环境下的授权去重和文件所有权证明，然而，该方案未考虑密钥的更新和撤销，客户端和服务器端的计算开销较大。表 1 总结上述典型数据去重

表 1 现有云数据去重方案的比较

方案	主要技术	授权去重	第三方服务器	数据去重级别	数据去重执行对象	密钥更新
文献[6]方案	CE+访问控制策略	—	密钥服务器	块级去重	跨用户	—
文献[7]方案	CE+AONT-RS	—	—	块级去重	客户端	是
文献[10]方案	BL-MLE+PoW	—	—	块级去重+文件级去重	客户端	—
文献[17]方案	门限盲签名+可校验秘密共享	—	多密钥服务器	文件级去重	客户端	—
文献[12,13]方案	MLE+AONT-RS	—	—	块级去重	客户端	是
文献[23]方案	身份认证协议+授权检测	是	私有云服务器	文件级去重	跨用户	—
文献[24]方案	属性加密+随机抽样	是	属性认证中心	块级去重	跨用户	—
本文方案	角色对称加密+群组密钥协商	是	角色认证中心	文件级去重	跨用户	是

方案的相关特性并将其与本文所提方案进行对比。

综上所述，现有解决云环境下数据安全去重问题的方案较少考虑到数据去重过程中的隐私泄露与未授权访问等问题，缺乏对分层结构下授权用户密钥更新和权限撤销等方面的研究。同时，在实现云数据安全去重与权限更新的基础上，如何有效减少计算开销、提高 I/O 读写效率以及降低通信开销等问题也亟待研究。

3 系统模型、威胁模型和实现目标

为了方便描述本文所提安全去重方案，表 2 列出常用的符号及对应的描述。

表 2 符号及其描述

名称	描述
T_i	角色密钥树
$Root_i$	根节点
MK_i	主密钥
G_i	角色群组节点
h_i	角色密钥树层数
K_{Gi}	角色群组节点密钥
f	文件
r_k	角色密钥
f_k	文件密钥
H_1,H_2,H_3,H_4	抗碰撞散列函数，$\{0,1\}^* \to \{0,1\}^\varepsilon$，其中，$\varepsilon$ 为正整数
Θ	使用文件密钥加密的密文
h_f	文件索引值
id	用户身份标识
eid	加密的用户身份标识
Ω	更新因子，群组协商协议得出
Φ	更新因子，角色认证中心指定

3.1 系统模型

基于角色对称加密的云数据安全去重方案的系统模型如图 1 所示，主要包含 3 个实体：用户、角色认证中心和云服务器。

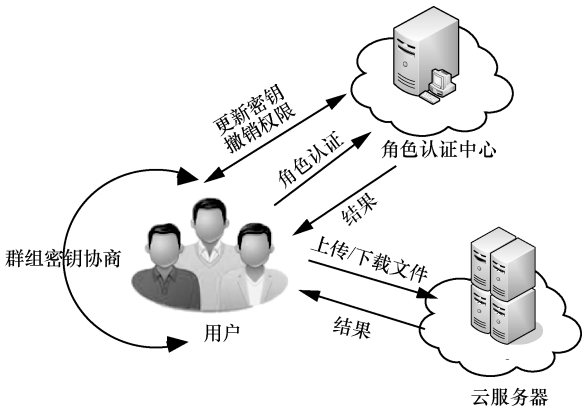


图 1 系统模型

用户：在云环境的分层结构中，不同角色的用户拥有对应的角色密钥，再结合访问控制策略得到文件密钥，用户使用该文件密钥对称加密文件，向云服务器发送上传或下载文件的请求。用户通过群组密钥协商协议更新角色密钥，并与角色认证中心交互，实现用户权限的撤销。

角色认证中心：负责认证用户角色和撤销用户权限，通过用户身份认证用户角色，返回角色密钥给用户。用户执行群组密钥协商协议与角色认证中心交互更新角色密钥树，撤销用户权限。

云服务器：负责安全存储文件和执行授权去重，将用户上传的文件信息保存在存储服务器中，当用户再次请求上传相同文件时执行授权去重，并返回对应的结果给用户。

3.2 威胁模型

本文所提角色对称加密算法[25]的角色密钥和文件密钥均需采用散列函数，假设所使用的散列函数均能够抵抗弱碰撞攻击和强碰撞攻击，角色对称

加密算法的加密部分采用标准对称加密算法，如AES-256。

本文考虑 2 种层面的攻击：敌手对角色对称加密算法的攻击和对云数据安全去重方案的攻击。

结合敌手的攻击强度，本文考虑以下 3 种类型的敌手对算法发起攻击。

1) 敌手 $\mathcal{A}_0$ 的攻击能力可以描述如下：能够向挑战者发起询问，获取有向无环图 T 中所有公共信息 $Pub=\{ID_i,H_1\}$，攻陷算法的方式是成功猜测节点 G_i 的角色密钥 K'_{G_i}，成功的概率为 $P_r[K'_{G_i}=K_{G_i}]$。

2) 敌手 $\mathcal{A}_1$ 的攻击能力可以描述如下：能够向挑战者发起询问，获取有向无环图 T 中所有公共信息 $Pub=\{ID_i,H_1\}$、部分节点 G_i 的角色密钥，攻陷算法的方式是成功恢复节点 G_u 的角色密钥 K'_{G_u}，成功的概率为 $P_r[K'_{G_u}=K_{G_u}]$。

3) 敌手 $\mathcal{A}_2$ 的攻击能力可以描述如下：能够向挑战者发起询问，获取有向无环图 T 中所有公共信息 $Pub=\{ID_i,H_1\}$、部分节点 G_i 的角色密钥，并且可以选定任一节点 G_v 质询挑战者获得对应的角色密钥，攻陷算法的方式是成功区分挑战者返回的角色密钥是否为该节点的真正角色密钥，成功的概率为 $P_r[K'_{G_v}=K_{G_v}]$。

根据上述 3 种类型敌手攻击能力的定义，敌手 $\mathcal{A}_0$ 可获得的信息包含于敌手 $\mathcal{A}_1$ 和敌手 $\mathcal{A}_2$ 掌握的挑战信息中。因此，敌手的攻击能力具有 $\mathcal{A}_1$ 角色密钥恢复和 $\mathcal{A}_2$ 角色密钥的不可区分。如果敌手 $\mathcal{A}_1$ 攻陷算法的概率 $\epsilon_1=P_r[K'_{G_u}=K_{G_u}]$，敌手 $\mathcal{A}_2$ 攻陷算法的概率 $\epsilon_2=P_r[K'_{G_v}=K_{G_v}]$ 是可忽略的，即敌手不能以不可忽略的概率攻陷该算法，则所提算法是安全的。

对云数据安全去重方案的攻击中，敌手试图非授权访问、下载云端存储的文件，存在以下类型的攻击。

1) 内容猜测攻击或文件伪造攻击。敌手拦截合法用户向云服务器上传的数据，或云服务器向合法用户反馈的数据，并试图猜测所拦截数据的内容或伪造所拦截数据。

2) 共谋攻击。在基于角色对称加密算法的安全去重过程中，合法用户可以与敌手共谋，泄露部分文件内容给敌手，根据文献[17]和文献[24]，当泄露不超过 64 MB 内容时足够抵抗这种共谋攻击。

3.3　系统实现目标

本文系统实现的目标主要考虑安全目标和性能目标 2 个方面。

安全目标主要包含算法安全性、抵抗内容猜测攻击或文件伪造攻击、抵抗共谋攻击、细粒度访问控制等方面。

1) 算法安全性：所提角色对称加密算法是可证明安全的。

2) 抵抗内容猜测攻击或文件伪造攻击：在安全去重过程中，一个拥有文件 f 部分内容的敌手，能够以可忽略的优势成功猜测或伪造目标文件。

3) 抵抗共谋攻击：在安全去重过程中，一个拥有文件 f 部分内容的敌手，必须和 f 的合法用户交换至少 $S_{\min}$ 的信息才能成功通过安全去重协议。根据 Halevi 等[18]的方案，$S_{\min}$ 设置为 64 MB。

4) 细粒度访问控制：本文所提方案，除了保障文件安全之外，还需要提供对用户和文件的细粒度访问控制支持，且不需要云服务器和用户承担额外的任务。

性能目标主要包含最小化传输带宽、云服务器内存消耗和用户端存储空间。

1) 通信带宽有效性：在执行安全去重过程中，用户端和云服务器端交换的文件字节数应该尽可能小，以保证低通信开销。

2) 服务器内存有效性：在执行安全去重过程中，云服务器端内存中加载的信息应比较小，与上传的文件大小无关，以保证低内存开销。

3) 用户端存储有效性：用户端存储的密钥、密文数据尽可能少；此外，存储密钥的数量和长度都应该与文件大小无关，以保证低存储开销。

4　方案设计

本节首先提出一种新型的角色对称加密算法，然后详细描述基于所提算法的云数据安全去重方案。

4.1　角色对称加密算法

角色认证中心建立分层角色密钥树以映射用户角色和密钥之间的关系，角色密钥树的每个节点表示不同的角色群组，拥有唯一的身份标识，相同角色的用户属于同一个角色群组，拥有相同的角色标识[4]。不同的文件由特定角色群组的用户管理，根据不同的访问控制策略，每个文件可由多个角色群组的用户共同管理，但是有且仅有一个主角色群组[25]。

为了方便描述密钥管理和文件管理过程，图 2 给出一种角色密钥树的实例。定义 2 棵角色密钥树 T_1 和 T_2，根节点分别为 $Root_1$ 和 $Root_2$，并拥有各自的主密钥 MK_1 和 MK_2。T_1 包含 2 个角色群组 G_1 和 G_2，群组 G_1 由子群组 G_3 和 G_4 组成，而群组 G_2 只有一个子群组 G_5。T_2 包含一个群组 G_6，并由一个子群组 G_7 组成。文件 f_1 属于群组 G_3，由 G_3 中的所有用户管理，文件 f_1 的主群组为 G_3。文件 f_2 由 G_5 中的用户和 G_4 中的用户共同管理，文件 f_2 的主群组为 G_5。文件 f_3 由 G_7 中的用户和 G_5 中的用户共同管理，文件 f_3 的主群组为 G_5。

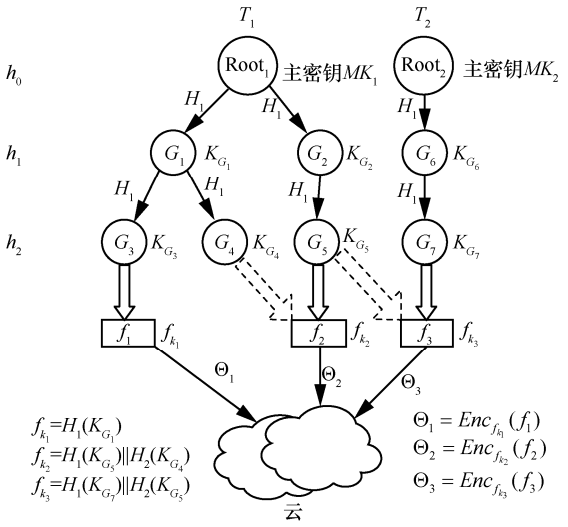


图 2 角色对称加密算法实例

所提角色对称加密算法主要分为 3 个阶段：角色密钥生成、文件密钥生成和对称加密。

1) 角色密钥生成阶段。上述角色密钥树 T_1 可以和一个有向无环图形成映射关系，形式化定义为 $T_1 =< G, E >$，其中，$G = \{Root_1, G_1, G_2, \cdots, G_n\}$，表示图 T_1 中的节点集合，每个节点 G_1 表示一类角色，也表示一类安全级别；$E = \{E_1, E_2, \cdots, E_m\}$，表示图 T_1 中有向边的集合，每条有向边 E_i 表示 2 个安全级别的角色之间具有从属关系。

算法初始化（Setup）时，给定有向无环图 $T_1 =< G, E >$ 和安全参数 λ、ρ，取图 T_1 中的每个节点 $G_i \in G$ 分配唯一的角色标识符 $id_i = G_i \in \{0,1\}^{\lambda}$，随机选取相应的密钥材料 $MK_1 \in \{0,1\}^{\rho}$，角色之间具有的从属关系用散列函数 H_1 描述。将角色标识符和角色从属关系作为公开参数 $Pub = \{ID_i, H_1\}$，MK_1 作为根节点主密钥。

各层角色密钥推导过程（derivation）如下。由根节点的主密钥及各个角色群组的标识计算得到各级群组节点的角色密钥。G_1 的角色密钥 $K_{G_1} = H_1(H_1(MK_1) \| G_1)$，$G_3$ 的角色密钥 $K_{G_2} = H_1(H_1(MK_1) \| G_3)$。类似地，上级节点的散列值串联该节点的群组标识再进行散列运算的结果作为各个角色群组节点的角色密钥。

2) 文件密钥生成阶段。文件密钥由主群组角色密钥的散列值串联访问控制策略中其他拥有管理权的群组角色密钥的散列值计算得到。文件 f_1 属于群组 G_3，该角色群组中的所有用户拥有文件 f_1 的管理权，文件 f_1 的文件密钥为角色密钥 K_{G_3} 的散列值，$f_{k_1} = H_1(K_{G_3})$。文件 f_2 由群组 G_5 和群组 G_4 中的用户共同管理，群组 G_5 为主群组，则文件 f_2 的文件密钥由主群组角色密钥和其他拥有管理权的群组的角色密钥计算得到，$f_{k_2} = H_1(K_{G_5}) \| H_2(K_{G_4})$。类似地，文件 f_3 的文件密钥与主群组 G_7 和其他群组 G_5 相关，$f_{k_3} = H_1(K_{G_7}) \| H_2(K_{G_5})$。

3) 对称加密阶段。由文件密钥对称加密原文件得到密文。使用文件密钥 f_{k_1} 对称加密文件 f_1 得到密文，$\Theta_1 = Enc_{f_{k_1}}(f_1)$，类似地，得到文件 f_2 和文件 f_3 的密文，$\Theta_2 = Enc_{f_{k_2}}(f_2)$，$\Theta_3 = Enc_{f_{k_3}}(f_3)$。然后将使用文件密钥对称加密的密文 Θ_1、Θ_2、Θ_3 上传至云服务器。

4.2 云数据安全去重方案构造

基于角色对称加密的云数据安全去重方案包括 4 个阶段：文件加密阶段、文件上传阶段、文件存储阶段和文件去重阶段。

1) 文件加密阶段

用户向角色认证中心发送认证请求，角色认证中心认证用户身份、搜索角色密钥树，根据从根节点到用户所属群组节点的角色节点路径以及根节点的主密钥 MK_α，执行角色密钥生成函数，得到角色密钥 r_k 并发送给用户。其中，节点路径可以表示为 $< Root_1, G_1, G_2, \cdots, G_i >, i \in [1, m]$。

角色密钥 r_k 计算过程可以表示为 $r_{k_i} = H_1\left(\cdots H_1\left(H_1\left(H_1\left(MK_\alpha\right) \| G_1\right) \| G_2\right) \cdots \| G_i\right), i \in [1, m]$。

角色密钥与节点路径的节点标识及主密钥相关，从主密钥散列值开始，执行上级节点的散列值串联用户所属群组标识再进行散列运算的递归操作，获得角色密钥。

用户根据角色密钥 r_k 以及对应的访问控制策略

执行文件密钥生成函数，获得文件密钥 f_k。

$$f_k = H_x\left(r_{k_1}\right) \| H_x\left(r_{k_2}\right) \cdots \| H_x\left(r_{k_i}\right),$$
$$i \in [1,n], x = 1 \text{或} x = 2$$

文件密钥 f_k 与角色密钥及访问控制策略相关，而散列函数的选择取决于拥有文件管理权的角色群组是否为主群组，如果文件属于单个角色群组，则 x=1，即对该角色群组的角色密钥进行 H_1 操作，如果文件属于多个角色群组，则对主群组的角色密钥执行 H_1 操作，对其他拥有管理权的角色群组的角色密钥执行 H_2 操作，最后串联上述结果，作为文件密钥。

用户使用文件密钥对称加密原文件得到密文

$$\Theta = Enc_{f_k}(f)$$

文件加密阶段的交互过程如图 3 所示。

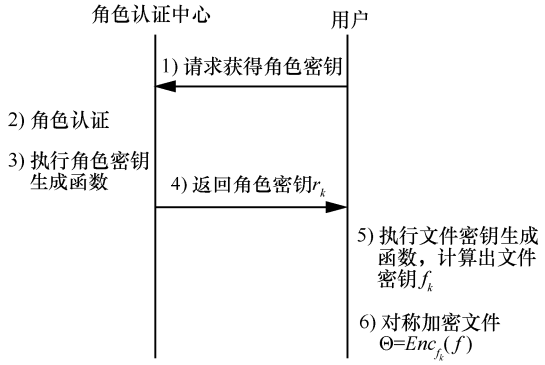


图 3　文件加密阶段

文件加密阶段的具体描述如算法 1 所示。

算法 1　客户端和角色认证中心——文件加密

输入　角色群组节点列表，主密钥 MK_α，文件 f

输出　使用文件密钥对称加密的密文 Θ

$r_{k_i} = H_1\left(\cdots H_1\left(H_1\left(H_1\left(MK_\alpha\right) \| G_1\right) \| G_2\right) \cdots \| G_i\right)$, $i \in [1,m]$;

$f_k = H_x\left(r_{k_1}\right) \| H_x\left(r_{k_2}\right) \cdots \| H_x\left(r_{k_i}\right)$, $i \in [1,n]$, $x = 1$或 $x = 2$;

$\Theta = Enc_{f_k}(f)$

2) 文件上传阶段

用户执行角色对称加密算法得到密文 Θ，对密文执行散列操作得到文件索引值，$h_f = H_3(\Theta)$，然后对用户的身份标识进行散列操作得到加密的身份标识，$eid = H_4(id)$，最后，用户整合上述结果，首次向云服务器发送{Θ,h_f,eid}，请求存储文件。文件上传阶段具体描述如算法 2 所示。

算法 2　客户端——文件上传

输入　使用文件密钥对称加密的密文 Θ，用户身份标识 id

输出　文件索引值 h_f，加密的身份标识 eid

$h_f = H_3(\Theta)$;

$eid = H_4(id)$;

return Θ,h_f,eid

发送{Θ,h_f,eid}给云服务器

3) 文件存储阶段

接收到用户存储文件的请求后，云服务器首先计算密文的散列值，$h'_f = H_3(\Theta)$，验证计算结果 h'_f 是否与用户上传的文件索引值 h_f 一致，用来抵抗文件伪造攻击。如果通过验证，云服务器根据接收到的信息创建一个二元的映射结构数组 $\mathfrak{F}$，包括 $\mathfrak{F}\cdot ENC$ 和 $\mathfrak{F}\cdot EID$，分别存储加密文件 Θ 和加密的用户身份标识 eid，并使用密文的散列值 h_f 作为检索数据结构的索引，如果结果不一致，则返回失败。文件上传阶段和文件存储阶段的交互过程如图 4 所示。

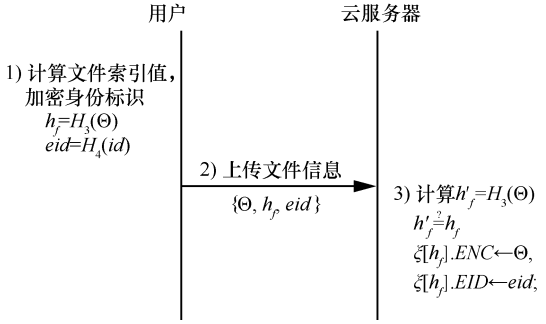


图 4　文件上传阶段和文件存储阶段

文件存储阶段的具体描述如算法 3 所示。

算法 3　云服务器——文件存储

输入　文件索引值 h_f，使用文件密钥对称加密的密文 Θ，加密的身份标识 eid

输出　二元的映射结构数组 $\mathfrak{F}$

$h'_f = H_3(\Theta)$;

if $\neg\left(h'_f == h_f\right)$

return ⊥;

end if

$\mathfrak{F}\left[h_f\right].ENC \leftarrow \Theta$;

$\mathfrak{F}\left[h_f\right].EID \leftarrow eid$;

return $\mathfrak{F}\left[h_f\right]$

4) 文件去重阶段

当用户向云服务器发送上传文件的请求时，执行文件去重过程。首先，云服务器要求用户上传文件索引值和加密的用户身份标识{h_f,*eid*}，然后，云服务器根据文件索引值h_f检索存储服务器中是否存储对应的结构数组，如果不存在，则要求用户上传使用角色密钥加密的密文 Θ，存储到云存储服务器中的二元映射结构数组中；如果存在，则表明云服务器中已存储该文件，不需要用户再次上传文件，实现云数据安全去重，并返回给用户存储文件的地址，然后，验证用户上传的加密身份标识是否属于$\mathfrak{F}\left[h_f\right].EID$，若属于该数组，则表明该用户使用同一身份上传过相同文件，若不属于该数组，则表明用户使用该身份首次上传文件，但云服务器中已存储同一角色群组中的其他用户上传的文件，将用户的加密身份标识添加到数组$\mathfrak{F}\left[h_f\right].EID$中。文件去重阶段的交互过程如图 5 所示。

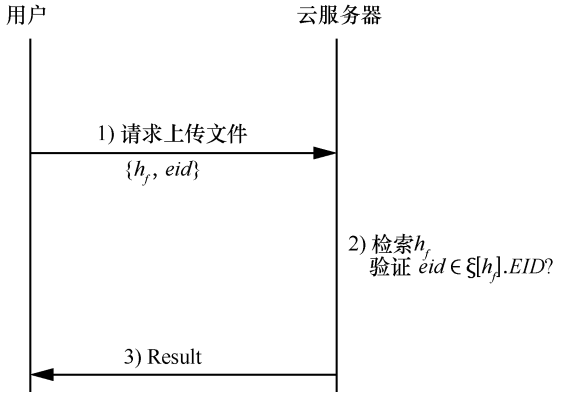


图 5 文件去重阶段

文件去重阶段的具体描述如算法 4 所示。

算法 4 云服务器——文件去重

输入 文件索引值 h_f，加密的身份标识 *eid*

输出 结果

if h_f 存储 then

执行数据去重操作

renturn 发送文件地址

if $eid \notin \mathfrak{F}\left[h_f\right].EID$ then

$\mathfrak{F}\left[h_f\right].EID \leftarrow eid$

end if

else

return 上传加密文件

end if

当用户向云服务器发送下载文件的请求时，发送文件索引值和加密的用户身份标识{h_f,*eid*}，云服务器验证用户的身份，根据文件索引值进行检索，返回密文给用户。最后，用户使用文件密钥解密文件得到原文件。

4.3 密钥更新机制

为了解决密钥更新问题，通过群组密钥协商协议得到更新因子 Ω，并与角色认证中心交互，提出一种密钥更新机制，高效实现云环境下分层结构的角色密钥更新和用户权限撤销。

用户权限撤销通过对用户的角色密钥进行更新实现，而更新角色密钥主要是对角色树进行操作的，用户权限撤销可分为以下 3 种情况。

1) 撤销中间节点或叶子节点的部分用户的权限

用户通过密钥协商协议协商出更新因子 Ω，设参与密钥协商的用户个数为 n，分别用 $U_0, U_1,\cdots, U_{n-1}$ 表示。固定轮数群组密钥协商协议基于可认证的群组密钥协商协议[26]和椭圆曲线密码系统，通过 2 轮协商得出群组密钥即更新因子 Ω。协商的过程可以描述如下。

①第一轮协商。参与协商的 n 个用户分别随机产生 $r_i \in {}_R Z_a^*, i \in [0, n-1]$，由预先在椭圆曲线上选取的公共点 P 计算出 $Z_i=r_iP$，每个用户 U_i, $i \in [0, n-1]$发送 Z_i给用户 $U_{(i-1) \bmod n}$和 $U_{(i+1) \bmod n}$。

②第二轮协商。用户 U_i, $i \in [0, n-1]$根据接收到的 $Z_{(i-1) \bmod n} = r_{(i-1) \bmod n} P$ 和 $Z_{(i+1) \bmod n} = r_{(i+1) \bmod n} P$ 计算 $\Upsilon_i = \left(Z_{(i+1) \bmod n} - Z_{(i-1) \bmod n}\right) r_i$，并广播 Υ_i 至所有参与协商的用户。用户 U_i计算出协商密钥，即更新因子 Ω，$\Omega_i = n r_i Z_{(i+1) \bmod n} + (n-1)\Upsilon_i + (n-2)\Upsilon_{(i+1) \bmod n} + \cdots + \Upsilon_{(i-2) \bmod n} = \Omega_{(i-1) \bmod n} = \Omega$。

用户将更新因子 Ω 发送给角色认证中心，角色认证中心将该群组节点标识与更新因子进行异或操作，作为更新后的群组节点标识，被撤销权限的用户不参与协商过程，无法获得更新因子，也无法得到更新后的节点标识，从而不能执行角色对称加密算法获得密文，实现用户权限的撤销。撤销中间节点或叶子节点的部分用户权限过程如图 6 所示。

2) 撤销中间节点全部用户的权限

角色认证中心指定更新因子 Φ，与该群组节点标识进行异或操作，作为更新后的群组节点标识，该角色节点的全部用户均无法获得更新因子，也无法得到更新后的节点标识，从而不能执行角色对

称加密算法获得密文，实现中间角色节点全部用户权限的撤销。撤销中间节点全部用户权限的过程如图 7 所示。

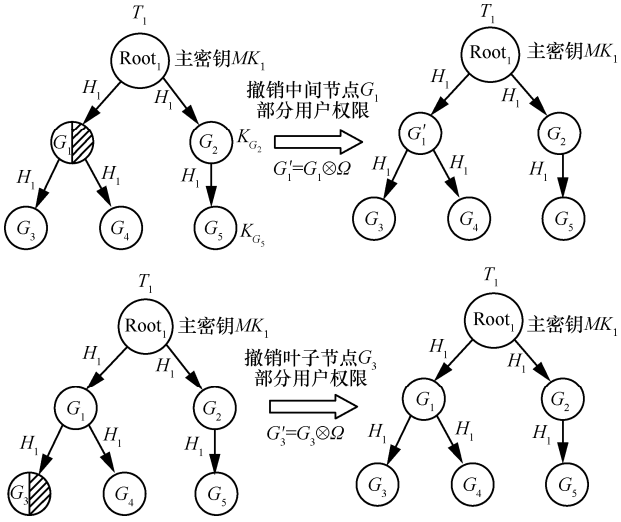


图 6　撤销中间节点或叶子节点的部分用户权限过程

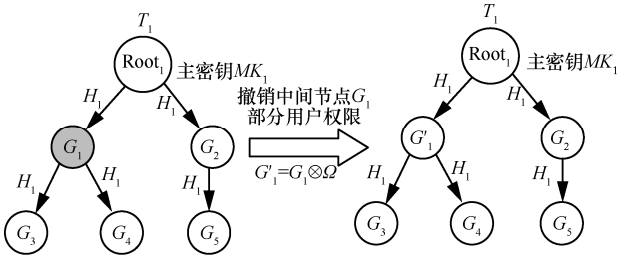


图 7　撤销中间节点全部用户权限的过程

3) 撤销叶子节点全部用户的权限

角色认证中心直接删除该叶子节点，即删除了该角色群组节点的节点标识，从而实现叶子角色节点全部用户权限的撤销。撤销叶子节点全部用户权限的过程如图 8 所示。

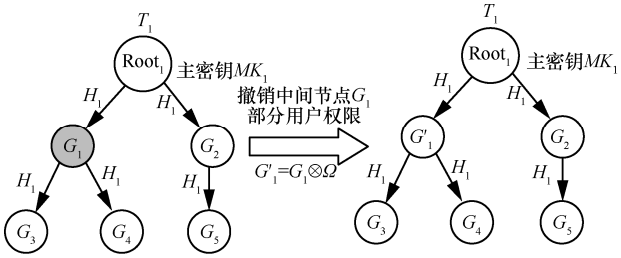


图 8　撤销叶子节点全部用户权限的过程

角色认证中心由更新后的角色树获得新的节点路径，执行角色密钥生成函数，得到角色密钥 r'_k 并发送给对应角色的所有用户。为了实现对已存储密文的更新，角色认证中心在更新的角色群组中选择参与密钥协商的用户，随机发送该角色群组所管理的文件，参与密钥协商的用户个数为 n, 角色群组 G_x 中管理的索引列表{$token_1$, $token_2$,…, $token_q$}，$token$ 为云服务器存储密文的索引，文件个数为 q，随机发送过程可以描述为以下 2 种情况。

1) 当 $n>q$ 时，即参与协商的用户个数大于该角色群组中管理的文件个数。

①角色认证中心从 n 个用户中随机选择 q 个用户依次发送文件索引{$token_1$, $token_2$, …, $token_q$}；②用户执行文件密钥生成函数得到更新后的文件密钥 f'_k，并向云服务器发送更新文件请求，云服务器根据对应的文件索引返回密文，用户使用文件密钥 f'_k 得到更新后的密文上传给云服务器。

2) 当 $n\leqslant q$ 时，即参与协商的用户个数小于或等于该角色群组中管理的文件个数。

①角色认证中心计算 $a=\left[\frac{q}{n}\right], b=q \bmod n$，将文件索引{$token_1$, $toke_2$,…, $token_n$}和{$token_{n+1}$, $toke_{n+2}$,…, $token_{n+n}$}随机发送给 n 个用户，重复 a 轮，最后随机从 n 个用户中选取 b 个，将{$token_{q-b}$, $token_{q-b+1}$,…, $token_{q-1}$, $token_q$}依次发送给 b 个随机用户；②用户执行情况 1)的步骤②，完成更新密文操作。

5　安全性分析

本文的安全性分析主要包含算法安全性证明和系统安全性分析。

5.1　算法安全性证明

本文借鉴文献[27,28]的构造思想，利用标准模型，对所提角色对称加密算法进行形式化安全证明。通过将所提算法规约到随机函数的安全性和加密方案的选择明文攻击的安全性上来证明所提算法是安全的，即如果存在敌手 $\mathcal{A}$ 攻陷所提算法的概率等价于存在敌手 $\mathcal{A}'$ 攻陷所提算法采用的一个多项式时间的计算复杂难题上，则可以证明所提算法是安全的[29]。

命题 1　令 RSE 表示本文所提角色对称加密算法，设 H_1 是一个安全、抗碰撞的散列函数，$H_1:\{0,1\}^*\to\{0,1\}^\varepsilon$。设 F_R 是一个伪随机函数集合，$F_R:\{0,1\}^X\times\{0,1\}^Y\to\{0,1\}^R$。对于任意的有向无环图 $T=<G,E>$，如果存在一个敌手 $\mathcal{A}_1$ 以 ϵ 的概率攻陷 RSE，则存在敌手 $\mathcal{A}'_1$ 以 ϵ' 的概率攻陷随机函数 F_R。

证明 定义 $Game_0$, $Game_1$, …, $Game_w$ 为敌手发起的一系列游戏，敌手发起的真正游戏为 $Game_0$，每个游戏 $Game_i$ 对应有向无环图 $T=<G,E>$ 中节点展开角色密钥恢复的操作，通过 *Game* 之间获得的密钥不可区分性来证明敌手进行 $Game_0$ 攻陷算法 RSE 的概率是可忽略的。

1) $Game_0$

初始化阶段。挑战者 C 调用 RSE 的初始化函数，即输入有向无环图 T_1，主密钥 MK_1，安全参数 λ、ρ，其中，$T_1=<G,E>, G_i\in\{0,1\}^{\lambda}=ID_i$, $MK_1\in\{0.1\}^{\rho}$。将输出结果中的公开信息 $Pub=\{ID_i,H_1\}$ 交给 $\mathcal{A}_1'$。

询问阶段。$\mathcal{A}_1'$ 向 C 发起询问，询问任意节点 G_i 对应的角色密钥材料，即 K_{G_i} 的结果。

挑战阶段。挑战者 C 由角色密钥生成算法计算得到对应的角色密钥材料 K_{G_i}，具体可以描述为 $K_{G_i}=H_1\left(\cdots H_1\left(H_1\left(H_1\left(MK_1\right)\|G_1\right)\|G_2\right)\cdots\|G_i\right)$, $G_i\in\{0,1\}^{\lambda}=ID_i, MK_1\in\{0,1\}^{\rho}$，并将上述角色密钥材料 K_{G_i} 的结果返回给 $\mathcal{A}_1'$。

猜测阶段。敌手 $\mathcal{A}_1'$ 指定一个节点 G_0，且 G_0 与质询阶段的 G_o 不构成从属关系，$\mathcal{A}_1'$ 通过猜测得到最接近 G_0 的真实角色密钥 K_{G_0} 的密钥 K'_{G_0}，$\mathcal{A}_1'$ 赢得 $Game_0$ 的优势定义为 $A\mathrm{dv}_{\mathcal{A}_{1,0}'}=\epsilon_0=P_r\left[K'_{G_0}=K_{G_0}\right]$。

2) $Game_1$

$Game_1$ 的初始化阶段、询问阶段和挑战阶段均与上述 $Game_0$ 过程相同，区别在于猜测阶段获得角色密钥 K_{G_1} 的算法由伪随机函数集合来替代，即 $K_{G_1}\approx F_R\left(MK_1,ID_1\right)$。利用 $Game_0$ 和 $Game_1$ 之间的可区分性，能构造一个多项式时间算法，以不可忽略的概率优势攻陷安全随机函数，有

$$\left|\epsilon_1-\epsilon_0\right|<negl\left(F_R\right) \tag{1}$$

成立。

同理，下面描述 $Game_i$ (i=0,1,2,⋯,w)。

3) $Game_i$

$Game_i$ 的初始化阶段、询问阶段和挑战阶段均与上述 $Game_{i-1}$ 过程相同，区别在于猜测阶段获得角色密钥 K_{G_1} 的算法由另一伪随机函数集合来替代，即 $K_{G_1}\approx F_R'\left(MK_1,ID_1\right)$。利用 $Game_i$ 和 $Game_{i-1}$ 之间的可区分性，能构造一个多项式时间算法，以不可忽略的概率优势攻陷安全随机函数，有

$$\left|\varepsilon_i-\varepsilon_{i-1}\right|<negl\left(F_R\right) \tag{2}$$

成立。

敌手 $\mathcal{A}_1'$ 在整个游戏 $Game_{i-w}$ 过程中，无法通过询问获得真实角色密钥 K_{G_1}，因此，敌手 $\mathcal{A}_1'$ 能够猜测出正确角色密钥 K_{G_1} 赢得游戏的优势定义为

$$A\mathrm{dv}_{\mathcal{A}_{1,w}'}=\epsilon_w=P_r\left[K'_{G_w}=K_{G_w}\right]=\frac{1}{2^{\lambda\rho}} \tag{3}$$

合并式(1)~式(3)，即 $\epsilon_0<w\cdot negl\left(F_R\right)+\frac{1}{2^{\lambda\rho}}$，证毕。

命题 2 令 RSE 表示本文所提角色对称加密算法，设 H_1 是一个安全、抗碰撞的散列函数，$H_1:\{0,1\}^{*}\to\{0,1\}^{\varepsilon}$。设 F_R 是一个伪随机函数集合，$F_R:\{0,1\}^{X}\times\{0,1\}^{Y}\to\{0,1\}^{R}$。对于任意的有向无环图 $T=<G,E>$，如果存在一个敌手 $\mathcal{A}_2$ 以 ϵ 的概率攻陷 RSE，则存在敌手 $\mathcal{A}_2'$，$\mathcal{A}_2'$ 以 ϵ' 的概率攻陷随机函数 F_R。

证明 定义 $Game_0$, $Game_1$, …, $Game_w$ 为敌手发起的一系列游戏，敌手发起的真正游戏为 $Game_0$，每个游戏 $Game_i$ 对应有向无环图 $T=<G,E>$ 中节点展开角色密钥获取的操作，敌手的优势为可以区分挑战者返回的结果是真正的角色密钥还是与密钥等长的随机值，通过 *Game* 之间获得的密钥不可区分性来证明敌手进行 $Game_0$ 攻陷算法 RSE 的概率是可忽略的。

1) $Game_0$

初始化阶段。挑战者 C 调用 RSE 的初始化函数，即输入有向无环图 T_1，主密钥 MK_1，安全参数 λ、ρ，其中，$T_1=<G,E>, G_i\in\{0,1\}^{\lambda}=ID_i, MK_1\in\{0.1\}^{\rho}$。将输出结果中的公开信息 $Pub=\{ID_i,H_1\}$ 交给 $\mathcal{A}_2'$。

询问阶段。$\mathcal{A}_2'$ 向 C 发起询问，询问任意节点 G_i 对应的角色密钥材料，即 K_{G_1} 的结果。

挑战阶段。①挑战者 C 由角色密钥生成算法计算得到对应的角色密钥材料 K_{G_1}，即 $K_{G_i}=H_1\left(\cdots H_1\left(H_1\left(H_1\left(MK_1\right)\|G_1\right)\|G_2\right)\cdots\|G_i\right)$, $G_i\in\{0,1\}^{\lambda}=ID_i, MK_1\in\{0,1\}^{\rho}$，并将上述角色密钥材料 K_{G_i} 的结果返回给 $\mathcal{A}_2'$。② $\mathcal{A}_2'$ 选定任一节点 G_0，且 G_0 与询问阶段的 G_i 不构成从属关系，向挑战者 C 发起询问。挑战者 C 随机选择

$C'_r \in \{0,1\}$，若 $C'_r = 1$，则返回真实角色密钥 $K_{G_0} = H_1\left(\cdots H_1\left(H_1\left(H_1\left(MK_1\right)\|G_1\right)\|G_2\right)\cdots\|G_0\right)$，$G_0 \in \{0,1\}^{\lambda} = ID_0, MK_1 \in \{0,1\}^{\rho}$，若 $C'_r = 0$，则返回与密钥等长的随机值 K'_{G_0}。

猜测阶段。敌手 $\mathcal{A}'_2$ 被赋予挑战者 C 随机选择 $C'_r \in \{0,1\}$ 返回的对应值作为猜测结果，则 $\mathcal{A}'_2$ 赢得 $Game_0$ 的优势定义为 $Adv_{\mathcal{A}'_{2,0}} = \epsilon_0 = P_r\left[K'_{G_0} = K_{G_0}\right] = -\frac{1}{2}$。

2) $Game_1$

$Game_1$ 的过程与上述 $Game_0$ 过程相同，区别在于推导获得角色密钥 K_{G_1} 的算法由伪随机函数集合来替代，即 $K_{G_1} \approx F_R\left(MK_\alpha, ID_1\right)$。利用 $Game_0$ 和 $Game_1$ 之间的可区分性，能构造一个多项式时间算法，以不可忽略的概率优势攻陷安全随机函数，有

$$\left|\epsilon_1 - \epsilon_0\right| < negl\left(F_R\right) \tag{4}$$

成立。

同理，下面描述 $Game_i(i=1,2,\cdots,w)$。

3) $Game_1$

$Game_i$ 的过程与 $Game_{i-1}$ 过程相同，区别在于推导获得角色密钥 K_{G_1} 的算法由另一伪随机函数集合来替代，即 $K_{G_1} \approx F'_R\left(MK_\alpha, ID_1\right)$。利用 $Game_i$ 和 $Game_{i-1}$ 之间的可区分性，能构造一个多项式时间算法，以不可忽略的概率优势攻陷安全随机函数，有

$$\left|\epsilon_i - \epsilon_{i-1}\right| < negl\left(F_R\right) \tag{5}$$

成立。

敌手 $\mathcal{A}'_2$ 在整个游戏 $Game_w$ 过程中，无法通过区分获得的角色密钥是真实角色密钥还是与密钥等长的随机值，因此，敌手 $\mathcal{A}'_2$ 无法通过推导得到真实的角色密钥 K_{G_1}，则能够成功猜测挑战者返回结果为真实角色密钥，赢得游戏的优势定义为

$$A\mathrm{dv}_{\mathcal{A}'_{2,w}} = \epsilon_w = P_r\left[K'_{G_w} = K_{G_w}\right] = \frac{1}{2} \tag{6}$$

合并式(4)~式(6)，即 $\epsilon_0 < w \cdot negl\left(F_R\right) + \frac{1}{2}$，证毕。

5.2　系统安全性分析

本文所提云数据安全去重方案需要抵抗内容猜测攻击、文件伪造攻击和共谋攻击，与传统的基于内容加密的数据去重方案相比，本文所提方案基于角色对称加密算法建立密钥和用户角色之间的映射关系，确保密钥和文件内容无关，使得敌手无法通过内容猜测攻击获取隐私信息，即使敌手得到密文信息，也无法解密出原文件[30]。在假设合法用户与敌手交换至少 $S_{\min}$ 的信息成功通过安全去重协议的安全目标下，根据 Halevi 等[18]设置 $S_{\min}$ 为 64 MB 来实现抵抗共谋攻击。在文件存储阶段，服务器通过验证用户上传的密文 Θ 和文件索引值 h_f 是否一致来抵抗文件伪造攻击，使拥有部分文件信息的敌手以可忽略的优势成功访问目标文件。在实现细粒度访问控制的安全目标方面，基于角色的对称加密算法通过访问控制策略和用户角色权限关联文件密钥来实现不同权限的用户访问特定的文件，计算角色密钥的过程中使用递归散列的操作以及计算文件密钥的过程中使用串联操作保证了不同的访问控制策略的应用，如果用户权限被撤销，则不能访问该文件，实现了细粒度访问控制的目标。

本文所提方案的通信带宽与设置的安全参数相关，满足用户和服务器交换较小的文件字节数实现授权去重的系统目标，服务器的内存开销也与安全参数相关，但与上传的文件大小无关，保证了服务器的低内存开销。客户端存储的角色密钥由角色认证中心计算，用户根据访问控制策略和角色密钥得到文件密钥来加密原文件，因此，客户端存储的密钥长度与文件大小无关，实现用户端存储有效性的系统目标。

表 3 将本文所提云数据安全去重方案与相关方案在算法安全性、系统安全性、细粒度访问控制和性能目标等方面进行归纳与总结。

6　性能分析与评价

6.1　算法复杂度分析

本文主要从客户端、服务器端和第三方服务器的计算开销分析算法的复杂度。客户端的计算开销主要是使用散列和串联操作得出文件密钥、使用文件密钥对称加密文件得到密文、进而得到文件索引值，复杂度与密钥长度、访问控制策略以及文件大小相关。服务器端的计算开销主要是计算和匹配文件索引值、生成和检索二元映射结构，复杂度与具体的匹配算法、检索算法相关。第三方服务器的计算开销主要是角色认证中心初始化角色密钥树、搜索角色密钥树及获取角色密钥，复杂度与角色密钥树的层次，密钥长度及具体的搜索算法相关。表 4 将本文方案与实现数据安全去重和所有权证明的

表 3 相关方案安全性和性能目标的比较

方案	安全性证明	抵抗攻击类型	细粒度访问控制	性能目标		
				通信带宽有效性	服务器内存有效性	用户存储有效性
文献[6]方案	理论分析	目录攻击、侧信道攻击	是	—	—	是
文献[7]方案	理论分析	侧信道攻击	—	是	是	是
文献[10]方案	标准模型下可证明安全	文件分发攻击	—	是	是	是
文献[17]方案	理论分析	蛮力攻击、共谋攻击	—	—	—	—
文献[12,13]方案	理论分析	伪造攻击	—	—	—	—
文献[23]方案	理论分析	侧信道攻击	是	—	—	—
文献[24]方案	理论分析	内容猜测攻击、侧信道攻击、共谋攻击	是	是	是	是
本文方案	标准模型下可证明安全	内容猜测攻击、文件伪造攻击、共谋攻击	是	是	是	是

方案在计算复杂度和带宽方面进行对比分析。

从表 4 可以看出，在客户端计算开销方面，ce-PoW 和 ase-PoW 均对文件块进行加密操作，虽然比其他方案直接对文件操作更加高效，但是密钥的计算也需要较大的开销；在服务器端计算开销方面，ce-PoW 和 ase-PoW 均对文件块进行处理，与其他方案相比计算开销较小；在第三方服务器计算复杂度方面，AuthorizedDedup 方案、ase-PoW 和本文提出的方案均引入第三方服务器，AuthorizedDedup 方案的第三方服务器计算密钥与文件相关，而 ase-PoW 和本文方案与文件无关，因此，效率更高。

6.2 性能评价

本文采用 Linux 系统下 Java 语言进行系统仿真实验，选取公开的 OpenSSL 函数库中 AES-256 与 SHA-256 算法实现对称加密和散列运算。实验环境配置如下，CPU：Intel Core i5-4590 3.30 GHz；RAM：8 GB；磁盘：WDC WD10EZEX-08M2NA0（1TB/7200 r/min）；操作系统：Ubuntu 12.04.4 LTS。

实验测试文件加密阶段和文件上传阶段的运行时间，实验运行 1 000 次，取平均值作为实验结果，选择 9 个从 2 MB 到 512 MB 不同大小的文件：2 MB、4 MB、8 MB、16 MB、32 MB、64 MB、128 MB、256 MB 和 512MB。

文件加密阶段的运行时间主要分为生成角色密钥、生成文件密钥及对称加密。实验测试了生成不同层级的群组角色密钥运行时间，在不同访问控制策略下生成文件密钥运行时间和使用文件密钥对称加密文件的运行时间，如图 9~图 11 所示。

表 4 相关方案计算复杂度和带宽消耗的比较

方案	客户端计算复杂度	服务器端计算复杂度	第三方服务器计算复杂度	带宽
文献[24]方案	$O(F)\cdot Sym\cdot hash$	$O(F)\cdot hash$	$O(F)\cdot Sym\cdot hash$	$O(\lambda)$
文献[14]方案	$O(F)\cdot hash$	$O(F)\cdot hash$	—	$O(\lambda\cdot\log\lambda)$
文献[15,16]方案	$O(F)\cdot hash$	$O(F)\cdot hash$	—	$O(\lambda)$
文献[17]方案	$O(F)\cdot hash$	$O(F)\cdot hash$	—	$O\left(\frac{l\cdot\lambda}{p_f}\right)$
文献[18]方案	$O(b)\cdot CE\cdot hash\cdot hash$	$O(b)\cdot hash\cdot hash$	—	$O(l\cdot\lambda)$
文献[19]方案	$O(b)\cdot CE\cdot n_{re}\cdot hash$	$O(b)\cdot hash\cdot hash$	Sym	$O(l\cdot\lambda)$
本文方案	$O(F)\cdot Sym\cdot hash$	$O(F)\cdot hash$	Sym	$O(\lambda)$

注：F 表示文件长度，b 表示文件块长度，λ 为安全参数，p_f 表示布隆过滤器的误判率，l 表示 *token* 的长度，*hash* 表示散列函数运算，*Sym* 表示系统生成密钥的运算，n_{re} 表示预设挑战的数量。

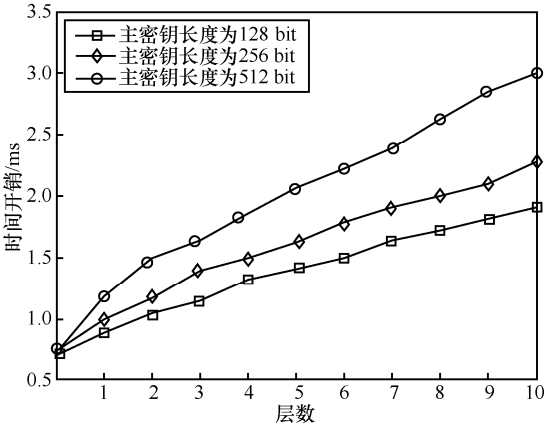


图 9　生成不同层级群组角色密钥的运行时间

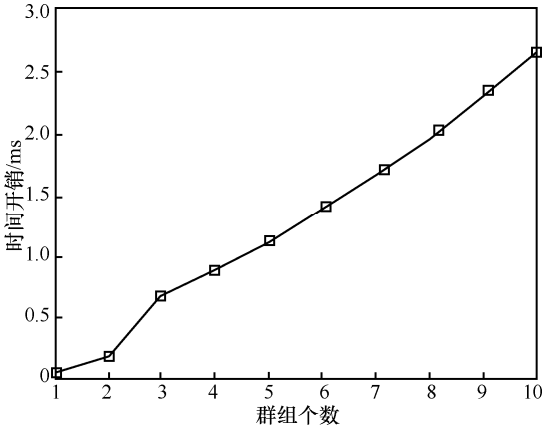


图 10　不同群组个数下生成文件密钥的运行时间

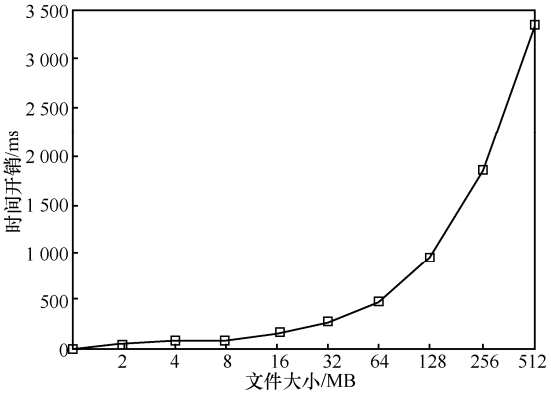


图 11　对称加密文件的运行时间

从图 9 可以看出角色密钥的计算复杂度与角色群组节点的层次和密钥长度相关，随着层次个数的增加，运行时间也不断增大，当设置主密钥长度为 512 bit，角色群组在角色密钥树的第 10 层时，计算角色密钥的时间开销仅需 3 ms，具有很高的计算效率。文件密钥的计算复杂度主要与访问控制策略相关，即文件所属的角色群组个数相关，由图 10 不同群组个数下生成文件密钥的运行时间可以看出，随着群组个数不断增加，得到文件密钥的运行时间也不断增大，当角色群组个数为 10 个时，计算文件密钥的时间开销约为 2.7 ms，具有较高的效率。文件加密阶段主要的时间开销在使用文件密钥对称加密文件上，如图 11 所示，随着文件大小的增加，对称加密的时间开销增大，曲线的增长趋势符合方案计算复杂度的分析。

文件上传阶段的运行时间主要是计算文件索引值，上传文件信息和运行时间的关系如图 12 所示。文件大小不断增加，上传文件所需时间开销有明显增长，曲线的增长趋势符合方案计算复杂度的分析。

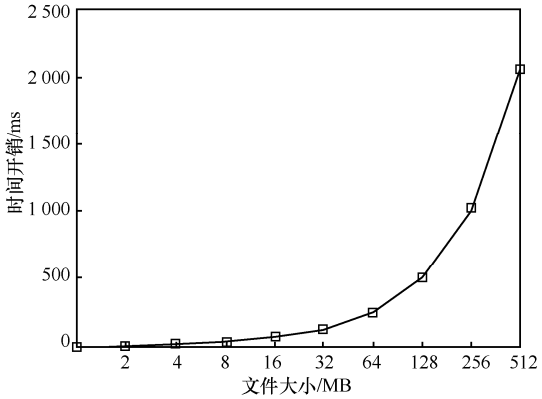


图 12　文件上传阶段运行时间

7　结束语

随着云计算和大数据技术的普及和发展，数据量的爆炸式增长和庞大的管理开销给有限的存储空间带来巨大压力，如何有效地存储管理这些文件，在保护个人隐私的同时实现安全访问和授权去重成为当前的研究热点。本文综合考虑隐私泄露、未授权访问、安全数据去重和密钥更新等问题，提出了一种全新的角色对称加密算法和基于该算法的云数据安全去重方案，通过构建角色密钥树，建立角色和密钥之间的映射关系，并利用角色对称加密关联用户角色与密钥，以满足不同角色根据访问控制策略访问对应权限文件的需求，有效保护个人隐私信息，实现云环境中分层结构下的云数据授权去重，并通过群组密钥协商解决角色与密钥映射关系中由密钥更新、权限撤销等带来的安全问题。安全性证明和性能分析表明所提方案是安全且高效的。

参考文献：

[1]　XIA W, JIANG H, FENG D, et al. A comprehensive study of the past,

present, and future of data deduplication[J]. Proceedings of the IEEE, 2016, 104(9):1681-1710.

[2] 熊金波, 张媛媛, 李凤华,等.云环境中数据安全去重研究进展.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.

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

[4] XIONG J, ZHANG Y, LI X, et al. RSE-PoW: a role symmetric encryption PoW scheme with authorized deduplication for multimedia data[J]. Mobile Networks and Applications, 2017:1-14.

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

[6] PUZIO P, MOLVA R, ONEN M, et al. ClouDedup: secure deduplication with encrypted data for cloud storage[C]//5th International Conference on Cloud Computing Technology and Science (CloudCom). 2013: 363-370.

[7] 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.

[8] STANEK J, SORNIOTTI A, ANDROULAKI E, et al. A secure data deduplication scheme for cloud storage[C]// International Conference on Financial Cryptography and Data Security, Springer Berlin Heidelberg, 2014, 8437: 99-118.

[9] BELLARE M, KEELVEEDHI S, RISTENPART T. Message-locked encryption and secure deduplication[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer Berlin Heidelberg, 2013, 7881: 296-312.

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

[11] JIANG T, CHEN X, WU Q, et al. Secure and efficient cloud data deduplication with randomized tag[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(3): 532-543.

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

[13] 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 (TOS), 2017, 13(1): 9.

[14] PUZIO P, MOLVA R, ÖNEN M, et al. PerfectDedup: secure data deduplication[C]//International Workshop on Data Privacy Management. Springer International Publishing, 2015: 150-166.

[15] BELLARE M, KEELVEEDHI S. Interactive message-locked encryption and secure deduplication[C]// IACR International Workshop on Public Key Cryptography. Springer Berlin Heidelberg, 2013, 7881: 296-312.

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

[17] MIAO M, WANG J, LI H, et al. Secure multi-server-aided data deduplication in cloud computing[J]. Pervasive and Mobile Computing, 2015, 24: 129-137.

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

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

[20] DI PIETRO R, SORNIOTTI A. Proof of ownership for deduplication systems: a secure, scalable, and efficient solution[J]. Computer Communications, 2016, 82: 71-82.

[21] BLASCO J, ROBERTO D P, ALEJANDRO O, et al. A tunable proof of ownership scheme for deduplication using bloom filters[C]// IEEE Conference on Communications and Network Security (CNS). 2014: 481-489.

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

[23] LI J, LI Y K, CHEN X, et al. A hybrid cloud approach for secure authorized deduplication[J]. IEEE Transactions on Parallel and Distributed Systems, 2015, 26(5): 1206-1216.

[24] GONZÁLEZ-MANZANO L, FUENTES J M D, CHOO K K R. ase-POW: a proof of ownership mechanism for cloud deduplication in hierarchical environments[C]// 12th EAI International Conference on Security and Privacy in Communication Networks.2016: 412-428.

[25] ZHANG Y, XIONG J, REN J, et al. A novel role symmetric encryption algorithm for authorized deduplication in cloud[C]//10th EAI International Conference on Mobile Multimedia Communications (EAI MOBIMEDIA). 2017: 104-110.

[26] 王宏远, 祝烈煌, 李龙一佳. 云存储中支持数据去重的群组数据持有性证明[J]. 软件学报, 2016, 27(6):1417-1431.
WANG H Y, ZHU L H, LI L Y J. Group provable data possession with deduplication in cloud storage[J]. Journal of Software, 2016, 27(6): 1417-1431.

[27] SANTIS A D, FERRARA A L, MASUCCI B. Efficient provably-secure hierarchical key assignment schemes[J]. Theoretical Computer Science, 2011, 412(41): 5684-5699.

[28] ATALLAH M, BLANTON M, FAZIO N, et al. Dynamic and efficient key management for access hierarchies[J]. ACM Transactions on Information and System Security (TISSEC), 2009, 12(3):1-43.

[29] 马骏, 郭渊博, 马建峰,等.物联网感知层一种分层访问控制方案[J]. 计算机研究与发展, 2013, 50(6): 1267-1275.
MA J, GUO Y B, MA J F, et al. A hierarchical access control scheme for perceptual layer of IoT[J]. Journal of Computer Research and Development, 2013, 50(6):1267-1275.

[30] 宋建业,何暖,朱一明,等.基于阿里云平台的密文数据安全去重系统的设计与实现[J].信息网络安全,2017(3):39-45.
SONG J Y, HE N, ZHU Y M, et al. Design and implementation of secure deduplication system for ciphertext data based on Aliyun[J]. Netinfo Security, 2017(3):39-45.

[作者简介]

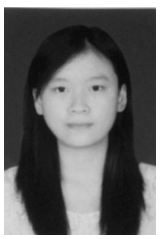
熊金波（1981-），男，湖南益阳人，博士，福建师范大学副教授、硕士生导师，主要研究方向为云数据安全、移动数据安全等。

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、移动数据安全等。

田有亮（1982-），男，贵州六盘水人，博士，贵州大学教授、博士生导师，主要研究方向为算法博弈论、密码学与安全协议、大数据安全与隐私保护、区块链与电子货币等。

应作斌（1982-），男，安徽芜湖人，博士，安徽大学讲师，主要研究方向为密码学与信息安全、基于位置的隐私保护等。

李琦（1989-），男，江苏淮安人，博士，南京邮电大学讲师，主要研究方向为基于属性的密码学与访问控制技术。



马蓉（1992-），女，甘肃兰州人，福建师范大学硕士生，主要研究方向为云数据安全、移动数据安全等。