一种混合云环境下基于 Merkle 哈希树的数据安全去重方案

张桂鹏 陈平华

(广东工业大学计算机学院 广州 510006)

摘 要 重复数据删除技术是云存储系统中一种高效的数据压缩和存储优化技术,能够通过检测和消除冗余数据来减少存储空间、降低传输带宽消耗。针对现有的云存储系统中数据安全去重方案所采用的收敛加密算法容易遭受暴力攻击和密文计算时间开销过大等问题,提出了一种混合云环境下基于 Merkle 哈希树的数据安全去重方案 MTHDedup。该方案通过引入权限等级函数和去重系数来计算去重标签,高效地实现了支持访问控制的数据安全去重系统;同时通过执行额外的加密算法,在文件级和数据块级的数据去重过程中构造 Merkle 哈希树来生成加密密钥,保证了生成的密文变得不可预测。安全性分析表明,该方案能够有效地抵制内部和外部攻击者发起的暴力攻击,从而提高数据的安全性。仿真实验结果表明,MTHDedup 方案能有效地降低密文生成的计算开销,减少密钥的存储空间,而且随着权限集数目的增加,性能优势将更加明显。

关键词 混合云存储,数据去重,Merkle 哈希树,访问控制,暴力攻击

中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.11.029

Secure Data Deduplication Scheme Based on Merkle Hash Tree in Hybrid Cloud Storage Environments

ZHANG Gui-peng CHEN Ping-hua

(School of Computers, Guangdong University of Technology, Guangzhou 510006, China)

Abstract Deduplication is an efficient data compression and storage optimization technology in cloud storage systems. It can reduce storage space and transmission bandwidth consumption by detecting and eliminating redundant data. The convergence encryption adopted by existing cloud storage systems is vulnerable to brute-force attacks and the time cost of ciphertext generation is excessive. In this paper, an efficient deduplication scheme based on Merkle hash tree in hybrid cloud environment was proposed. The tag used to detect duplicated data is calculated by introducing privilege level function and label coefficients which can realize a secure deduplication system with different privilege levels. At the same time, an additional encryption algorithm is implemented, and cryptographic keys are generated by a Merkle hash tree. These keys are used to encrypt the plaintext at a file-level and block-level deduplication which ensures that the ciphertext becomes unpredictable. The security analysis shows that this scheme can effectively resist the brute-force attacks from internal and external attackers, and improve the confidentiality of data. The simulation results show that the proposed MTHDedup scheme can effectively reduce the computation overhead of ciphertext generation and the storage space of cryptographic keys. With the increase of the number of privilege sets, the performance advantage of MTHDedup scheme is more obvious.

Keywords Hybrid cloud storage, Data deduplication, Merkle hash tree, Access control, Brute-force attacks

1 引言

当今云计算技术为整个互联网的用户或企业提供了巨大的计算能力和存储空间,越来越多的数据被外包给具有充足存储空间和计算资源的云存储服务提供商。根据 IDC 的分析报告,2020 年全球的数据量预计将达到 44 ZB[1]。随着用户存储数据的爆炸式增长,许多云存储服务提供商都会使用重复数据删除技术来对数据进行压缩,通过检测和消除冗余数据来减小传输带宽和云端存储空间。研究表明,有效的数据去重能节省超过 90%的存储空间[2],然而重复数据删除技术与传统的加密算法无法进行兼容。由于用户或企业在外包数据前,需要采用加密算法对数据进行加密,以防止攻击者或云服务提供商获取到原始数据[5-7],此时使用传统的加密算法(如 AES 和 DES)会随机化加密后所产生的密文,以至于对同

到稿日期:2018-08-08 返修日期:2018-09-15 本文受国家自然科学基金资助项目(61572144),广东省科技计划项目(2013B091300009,2014B070706007,2017B030307002)资助。

张桂鹏(1993—),男,硕士生,主要研究方向为云计算、大数据、区块链,E-mail:zhguipeng@outlook.com;陈平华(1967—),男,教授,主要研究方向为云计算、大数据、推荐系统,E-mail:phchen@gdut.edu.cn(通信作者)。

样的数据每进行一次加密都会产生不同的密文，使得云服务器难以确定原始数据是否相同，从而无法进行重复检测删除。

针对传统加密算法无法对密文进行重复性检测的问题，Douceur 等[8] 提出了收敛加密算法（Convergent Encryption，CE），其中加密数据的密钥由原始数据进行哈希运算获得，确保相同的数据能够得到相同的密文，这种密钥生成算法能够对密文进行重复检测。随后的一些研究方案[9-13] 都将 CE 算法与不同机制相结合来实现数据去重，然而由于 CE 算法缺乏严格的密码学原语定义和形式化描述的安全模型，这些方案都面临着暴力攻击、数据泄露、复制伪造攻击等一些安全问题。为了提供安全目标理论依据，Bellare 等[14] 提出了一种新的密码学原语——消息锁加密（Message - Locked Encryption，MLE），并将收敛加密归为 MLE 的一个特例，但是 MLE 只适用于保护具有比较高的不可预测性的数据。此外，Bellare 等[15] 提出了 DupLess 方案，在 MLE 的基础上通过引入第三方密钥服务器来产生去重标签，将可预测的数据转化为不可预测，从而保护数据的安全。

为了验证用户确实拥有数据，防止攻击者利用去重机制，通过文件的指纹信息来获取到完整文件，Halevi 等[16] 提出了所有权证明（Proof of Ownership，POW）的概念，以便用户在没有上传原始文件的情况下，可以向云服务器证明他拥有原始文件，但该方案仍然存在着暴力攻击的风险。Ng 等[5] 为加密的文件扩展了 POW 的概念，但是没有解决如何减少密钥管理开销的问题。Blasco 等[17] 提出了基于布隆过滤器的 POW 去重方案，服务器将文件分成大小相同的文件块，并计算出对应的标签信息，同时使用布隆过滤器存储处理后的标签信息，客户端需要上传一定数量的标签信息才能证明其拥有该文件。该方案能够大幅度减少计算开销，但未涉及到数据的安全问题。Yang 等[18] 提出了基于零知识证明的客户端重复数据删除方案（ZK-DE），用户能够通过原始的文件证明其文件所有权，而不会在与服务器交互的过程中泄漏任何文件信息给服务器，并且也提出了基于代理重加密的密钥分发方案，保证了数据和密钥的机密性，但同时会使得客户端承担过大的计算量。为了降低客户端的计算开销，Liu 等[19] 提出了一种基于 MLE 算法的完整性审计方案，该方案无需额外的代理服务器，且适用于文件级和数据块级重复数据删除系统，但仍然存在着 MLE 算法所面临的缺陷。Li 等[20] 提出了一种支持访问控制的混合云系统架构，通过执行额外的计算来增强数据的安全性，使得数据文件只能被具有特定权限的用户访问，但去重标签和密文的计算开销较大，且不支持数据块级的数据去重，亟需进一步进行方案的研究和改进。

为了降低去重标签和密文的计算开销，高效实现支持访问控制的数据去重系统。本文提出一种混合云环境下基于 Merkle 哈希树的数据安全去重方案 MTHDedup。本文所提方案的主要贡献如下：

1）本方案在现有的混合云系统模型中引入了密钥管理服务器和权限管理服务器，通过权限等级函数 T 和去重系数 δ 来计算出去重标签，保证了重复数据能够被检测，降低了去重标签的计算开销。

2）通过构造 Merkle 哈希树来生成加密密钥，保证生成的密文变得不可预测，有效地解决了内部和外部攻击者发起的暴力攻击，在一定程度上也降低了密文生成的计算开销。

3）有效地支持文件级和数据块级的重复数据删除，提高数据的去重率，高效地实现了支持访问控制的数据安全去重系统。

本文第 2 节介绍预备知识；第 3 节描述系统模型、威胁模型和安全目标；第 4 节提出了一种支持访问控制的数据安全去重系统；第 5 节对所提方案进行安全性分析；第 6 节介绍测试平台的实验仿真结果；最后对本方案的研究进行总结，并指出未来的研究方向。

2 预备知识

2.1 符号说明

文中相关符号的说明如表 1 所列。

表 1 符号说明

Table 1 Symbolic description

符号	描述说明
F	原始文件
P	权限的总集合
P_U	用户 U 的权限集合
P_F	文件 F 的权限集合
$ROOT$	Merkle 哈希树的根节点
Encrypt(K,D)	加密算法，以原始数据的密钥 K 和明文 D 作为输入，输出对应的密文 C
Decrypt(K,C)	解密算法，以原始数据的密钥 K 和密文 C 作为输入，输出明文 D

2.2 Merkle 哈希树

Merkle 哈希树是一种哈希二叉树，主要由一个根节点、一组中间节点和一组叶子节点构成，用于数据的完整性验证和用户的所有权证明。Merkle 哈希树的叶子节点由数据信息构成，其余非叶子节点由其孩子节点值串接后通过哈希运算得到；依次从下向上逐层运算，最终将得到唯一的根节点，此时 Merkle 哈希树构建完成。

2.3 身份认证协议

身份认证协议能够验证当前持有文件的用户身份是否正确，即是否为真实拥有文件的用户或者伪装的攻击者。一个身份认证协议的过程包括：证明阶段和验证阶段。协议运行时，证明者（即用户）根据个人相关的身份信息 ID（如用户名、密码等）生成身份证明数组，将证明发送给验证者，验证者验证证明者发送的证明数据，并输出接受或拒绝，从而判定该证明是否通过验证。已有的文献中包含了许多有效的认证协议，包括基于证书的认证协议等[3-4]。

2.4 所有权证明算法（POW）

所有权证明算法是一种由证明者（即客户端）和验证者（即服务器）共同参与的交互式算法，该算法用来证明客户端具有存储在云服务器中数据的所有权，以防止攻击者通过数据指纹信息从服务器中获取完整的文件。算法运行时，作为验证者的服务器根据文件生成相应的挑战发送给客户端；为了证明文件的所有权，客户端根据拥有的文件生成正确的应答并返回验证，以此来证明客户端确实拥有该文件。

3　问题阐述

3.1　系统模型

本节将介绍方案中存在的系统模型和安全威胁模型的定义，如图 1 所示，系统模型包含 3 种实体模型：用户实体、云存储服务提供商实体（又称公有云实体）和私有云实体。

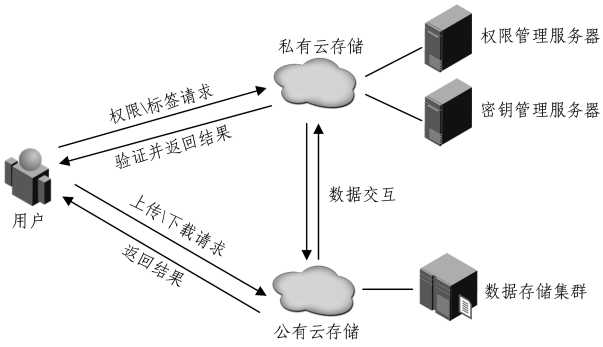

图 1　混合云环境下的数据安全去重系统模型

Fig. 1　Secure data deduplication system model in hybrid cloud environment

用户实体（User）：为了降低文件传输中的带宽并节省存储空间，用户需要上传数据文件，并将数据文件存储到云存储服务器中，数据文件只需要上传一份。在支持访问控制的数据去重系统中，用户上传文件后需要设定文件的访问权限，不符合文件访问权限的用户禁止下载该文件。

云存储服务提供商实体（S-CSP）：云存储服务提供商接收不同用户上传的数据文件和相关标签，并合理地存储在其服务器上。当用户上传相同的数据文件时，云存储服务提供商进行安全去重来降低存储成本。本方案假定 S-CSP 具有充足的存储容量和计算能力。

私有云实体（PrivateCloud）：私有云作为混合云主要的组成成分之一，提供了对数据安全性和服务质量的有效保障，并且相比于云存储，其具有较高的可靠性和安全性。在本方案中，私有云为诚实可信的第三方，包含了权限管理服务器和密钥管理服务器，权限管理服务器存储着用户的身份和权限的信息，并能对用户进行身份和权限的验证，密钥管理服务器能生成和管理权限密钥和加密密钥。

3.2　威胁模型和安全目标

本方案的威胁模型主要考虑了两种类型的攻击者：外部攻击者和内部攻击者。外部攻击者试图冒充合法用户来与云存储服务器进行交互，企图获得有关数据文件的信息，存在与其他实体串通合谋获取数据的风险；内部攻击者拥有部分有效的权限，对存储在云服务器上的部分数据信息有一定的了解，一个内部攻击者可被看作拥有部分有效权限的外部攻击者，他们企图通过拥有的部分权限来获取私有云的信任，并试图从私有云复制其权限范围外的去重标签，从而窃取公有云存储上的数据文件。

本文方案假定私有云为诚实可信的第三方，并提出了以下的安全性要求：去重标签的安全性、加密密钥的安全性和数据的机密性。

1）去重标签的安全性：拥有有效权限的用户可以在私有云的验证下，根据自身的权限，通过与私有云进行交互计算来获取文件的去重标签，从而执行数据重复检测操作。基于这种假设，非授权的用户或者没有被认证的用户将无法获取到任何的去重标签，进而无法下载数据文件，同时缺少私有云的验证授权，拥有部分权限的用户也将无法获取到有效的去重标签，这样可有效地阻止非授权的外部攻击者进行暴力破解。

2）加密密钥的安全性：在数据去重过程中，要求用户包括 S-CSP 在未进行身份认证前无法直接接触到加密密钥，使得恶意用户即使获取到其他用户的密钥，也无法从密钥中获取到关于文件或权限的信息。在本方案中，密钥是由可信的私有云进行计算生成的，同时密钥的生成需要存储在私有云上的权限集参与，未拥有权限集的外部或内部攻击者将无法获取到关于密钥的任何信息，当用户某一权限失效时，密钥会立即失效，此时用户需要重新向私有云进行申请才能获得新的密钥。此外，只要文件的密文上传到 S-CSP，即使 S-CSP 与恶意用户进行共谋，也无法获得其他用户的加密密钥或者伪造密钥来解密密文。

3）数据的机密性：意味着未被授权的用户或者 S-CSP 无法解密密文，无法直接获取到原始文件，用户只有拥有有效身份和权限，并且在私有云的授权下，才能接触到原始文件。在本方案中，通过构造 Merkle 哈希树来生成加密密钥，这样使得密文变得不可预测，而并非使用传统的收敛加密来生成密文，有效地抵制了暴力攻击。

4　方案实现

4.1　基本思路

为了实现混合云环境下对数据的安全去重，本方案改进了现有的混合云存储去重模型，引入权限等级函数和去重系数来计算去重标签，以提高去重标签的安全性。在进行文件级或数据级的数据去重过程中，加密数据的密钥通过构造 Merkle 哈希树来生成，保证生成的密文变得不可预测，有效地抵制了暴力攻击。

4.2　去重标签的生成方案要素设计

在公共层次关系中，权限之间存在着角色等级差异的关系，如在某项目管理中，项目经理的角色对于项目文件的管理权限会比工程师角色的权限级别高。Li 等[20] 假定等式 $\phi'_{F,p}=TagGen(F,k_p)$ 表示标签 $\phi'_{F,p}$ 只能由文件 F 和权限密钥 k_p 共同计算得到，这样的标签生成函数可以容易地实现为 $H(F,k_p)$，其中 H 表示散列函数。同时，在方案中结合角色等级差异给出了去重标签的生成方式：$\{\phi_{F,p\tau}=H(H(F),k_{p\tau})\}$，其中 $p\tau$ 满足等式 $R(p,p\tau)=1$，权限 $p\in P_U$，R 为二元关系式，表示 p 具有比 $p\tau$ 更高的权限等级。这种标签生成方式有效地实现了对数据授权访问的去重检测，然而当用户所拥有的权限集数目增多一倍时，去重标签的计算开销将增大数倍，倍数由权限所属的角色等级来决定。

为了降低去重标签的计算开销，我们构造等级函数 T 来量化等级差异，引入权限有效时间 t_p 和去重系数 δ 来计算去重标签，因此我们给出了以下几个定义。

定义 1　设定权限集 P 的等级函数为 T，权限 p 的等级值表示为 $T(p)(T(p)\in Z)$，等级函数 T 的取值范围为 $[0,\varepsilon]$，

ε 值的大小由私有云根据所对应的公共层次关系来设定。当 $T(p)=0$ 时，表示权限 p 为最低等级；当 $T(p)=\varepsilon$ 时，表示权限 p 为最高等级，权限 p 的等级越高，等级值 $T(p)$ 就越大。

定义 2 设定一个三元关系式 $Y=\{(t_p,p,p')\}$，其中 p，p' 为权限，t_p 为权限 p 的有效时间，$\delta(\delta\in Z)$ 为去重系数（当 δ 越大时，需要计算的有效去重标签数目就越少，整个去重系统的安全性就越差，反之则亦），t_p 和 Y 的值满足以下关系式：

$$t_p=\begin{cases}1, & \text{权限 } p \text{ 有效}\\ 0, & \text{权限 } p \text{ 已失效}\end{cases} \tag{1}$$

$$Y=\begin{cases}1, & \text{当 } t_p=1, T(p')\leqslant T(p)/\delta \text{ 时}\\ 0, & \text{其他}\end{cases} \tag{2}$$

4.3 系统设置

假定用户的数量为 N，总权限集为 P，用户的身份标识为 $uuid$（$uuid$ 可由加密的用户名、密码或用户申请的权限等组成），在用户注册登录成功时，私有云将生成该用户的身份标识 $uuid$，并将其发送给该用户进行存储。本方案中的所有用户的权限和文件的设定权限都来源于权限集 P。

4.4 基于文件级的安全去重方案

用户向私有云发送文件上传请求，私有云为了确保数据的安全去重，需要先与用户进行身份认证，以验证用户的身份是否正确。首先，用户将身份标识 $uuid$ 发送给私有云；其次，私有云与用户执行身份认证协议并输出验证结果，若验证不通过，则私有云将停止当前的数据操作，拒绝用户与 S-CSP 进行数据交互，若验证通过，则继续执行下列操作。

4.4.1 文件上传

假设用户所拥有的权限集为 $P_U(P_U\in P)$，用户设定上传文件 F 的访问限制权限集为 $P_F(P_F\in P)$，用户选取哈希函数 H_0，将 $H_0(F)$ 发送给私有云，私有云根据 P_U 计算出相应的权限密钥 $\{k_p\}$，选取哈希函数 H_1 来计算出用户的去重标签集 $\{\phi_{F,p'}=H_1(H_0(F),k_{p'})\}$，其中 p' 满足三元关系式 $Y\{(t,p,p')\}=1$，$p\in P_U$，将标签集 $\{\phi_{F,p'}\}$ 发送给用户，用户将标签集 $\{\phi_{F,p'}\}$ 上传到 S-CSP，$\{\phi_{F,p'}\}$ 用于对文件 F 的重复检测，S-CSP 返回检测结果。

若检测结果为文件 F 重复，则进行以下操作：S-CSP 与用户之间运行 POW 算法，通过所有权证明的检测来判定用户是否确实拥有该文件。若用户没有通过所有权证明，则 S-CSP 将停止当前的数据操作，拒绝用户访问数据；若用户通过所有权证明，S-CSP 将授权允许用户拥有文件 F，用户不需要再上传文件 F，私有云计算出去重标签集的权限差集 $\{\phi_{F,p'}=H_1(H_0(F),k_{p'})\}$，其中 p' 满足三元关系式 $Y\{(t,p,p')\}=1$，$p\in P_F-P_U$，将标签集 $\{\phi_{F,p'}\}$ 发送给 S-CSP，对原先存储的 $\{\phi_{F,p'}\}$ 进行更新，以便其他用户进行去重检测。

若检测结果为文件 F 不重复，则进行以下操作：用户将文件 F 的访问限制权限集 P_F 发送给私有云，私有云根据 P_F 选取出权限密钥 k_{pF}，计算出权限标签集 $\{\phi_{F,pF}=H_1(H_0(F),k_{pF})\}$，并将标签集 $\{\phi_{F,pF}\}$ 和 $\{\phi_{F,p'}\}$ 发送给 S-CSP；此外，私有云还需要计算加密密钥 k_F，k_F 用于对文件 F 进行加密，私有云使用用户的权限密钥来构建该用户的 Merkle 哈希树，从而得到文件密钥 k_F。具体的构造流程如下：

1）私有云将用户权限密钥 $\{k_p\}$ 的哈希值作为 Merkle 哈希树的叶子节点，即权限密钥为 $k_{p1},k_{p2},\cdots,k_{pj}$，其叶子节点为 $H(k_{p1}),H(k_{p2}),\cdots,H(k_{pj})$。

2）对每两个叶子节点进行串接，并计算其哈希值作为父节点，父节点与其兄弟节点继续串接，并哈希运算得到其父节点，依次从下向上逐层运算。当某一层为奇数个节点时，将出现最后一个节点没有兄弟节点与其进行串接的情况，此时其父节点由该节点直接哈希获得，最终将得到唯一的根节点 $ROOT$，并将此根节点 $ROOT$ 作为文件加密密钥 k_F。如图 2 所示，当用户的权限集数目为 4 时，私有云计算分配的权限密钥分别为 $k_{p1},k_{p2},k_{p3},k_{p4}$，其权限哈希值（该 Merkle 哈希树的叶子节点）分别为代号 P_0,P_1,P_2,P_3，最终通过构造 Merkle 哈希树得到其密钥 k_F 为 P_{0123}。

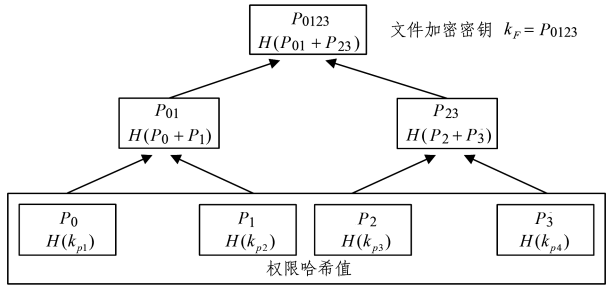

图 2 当权限集的数目 $n=4$ 时文件加密密钥的生成过程

Fig. 2 Generation process of file-level encryption key when $n=4$

每当用户发起上传文件的请求时，私有云直接从密钥管理服务器中获取该用户的密钥 k_F 并将其发送给用户，以降低密钥的计算开销。当用户的权限集需要更新时，k_F 也会直接失效，私有云重新构造 Merkle 哈希树来更新文件密钥 k_F。

私有云将密钥 k_F 发送给用户，用户使用加密密钥 k_F 对文件进行加密，得到密文 C_F，即 $C_F=\text{Encrypt}(k_F,F)$，将密文 C_F 上传到 S-CSP，S-CSP 存储该密文后返回指向文件的指针。

4.4.2 文件下载

用户需要下载文件 F 时，发送下载请求给私有云，私有云验证用户所具有的权限是否符合文件访问权限 P_F，如果验证通过，私有云发送信号给 S-CSP，S-CSP 接收到信号后发送密文 C_F 给用户，用户使用密钥 k_F 对 C_F 进行解密，即 $F=\text{Decrypt}(k_F,C_F)$，最终用户获取文件 F。如果验证不通过，S-CSP 将拒绝用户下载文件。

4.5 基于块级的安全去重方案

在对文件进行块级去重前，用户需要先执行 4.4 节所描述的文件级数据去重方案，如果检测结果为文件不重复，则将文件切分成数据块，进而执行粒度更细小的块级去重方案。与 4.4 节所描述的文件级去重方案不同，在基于块级的去重方案中，Merkle 哈希树的叶子节点为数据块的哈希值，而并非用户的权限密钥集 $\{k_p\}$ 的哈希值，所得到的根节点 $ROOT$ 为验证密钥 k_C，密钥 k_C 可用于对数据块的完整性进行校验，最终加密密钥 k_B 需要由验证密钥 k_C 与文件密钥 k_F 进行异或运算得到，即 $k_B=k_C\oplus k_F$，下面给出详细的阶段。

4.5.1 文件上传

用户将文件 F 分成 n 块数据块 $\{B_i\}$，其中 $1\leqslant i\leqslant n$，并将

$H(B_i)$发送给私有云，私有云将计算出块标签集$\{\phi_{B_i,p'}=H_1(H_0(B_i),k_{p'})\}$，其中 p'满足三元关系式 $Y\{(t,p,p')\}=1$，$p\in P_U$，私有云将块标签集发送给用户，用户上传到 S-CSP 进行重复检测，S-CSP 将检测结果返回给用户。

若数据块存在重复，用户只上传不重复的数据块到 S-CSP，S-CSP 返回指向数据块的指针，私有云计算出权限差集$\{\phi_{B_i,p'}=H_1(H_0(B_i),k_{p'})\}$，其中 p'满足三元关系式 $Y\{(t,p,p')\}=1$，$p\in P_F-P_U$，将标签集发送给 S-CSP 进行权限集更新。

若数据块不重复，私有云将根据 P_F 计算出权限标签集$\{\phi_{B_i,pF}=H_1(H_0(B_i),k_{pF})\}$，将标签集$\{\phi_{B_i,p'}\}$和$\{\phi_{B_i,pF}\}$发送给 S-CSP，并通过构建 Merkle 哈希树来得到数据块密钥 k_B。具体的构造流程与 4.4 节所描述的构造流程类似：

1）将数据块$\{B_i\}$的哈希值作为 Merkle 哈希树的叶子节点，即叶子节点为 $H(B_1),H(B_2),\cdots,H(B_n)$。

2）对每两个叶子节点进行串接，计算其哈希值，将该值作为父节点，父节点与其兄弟节点继续串接，并通过哈希运算得到其父节点，依次从下向上逐层运算，最终将得到根节点 *ROOT*，*ROOT* 将作为数据块的验证密钥 k_C，密钥 k_C 能验证数据块的完整性，数据块加密密钥为 $k_B=k_C\oplus k_F$。如图 3 所示，当数据块的数目为 4 时，即 B_1,B_2,B_3,B_4，其数据块哈希值（该 Merkle 哈希树的叶子节点）分别为代号 L_0,L_1,L_2,L_3，最终通过构造 Merkle 哈希树得到其验证密钥 k_C 为 L_{0123}，数据块加密密钥为 $k_B=k_C\oplus k_F$。

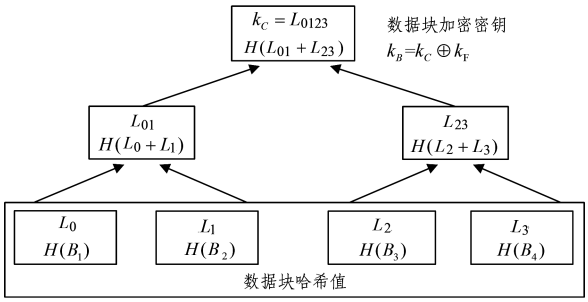

图 3　当数据块的数目 $n=4$ 时数据块加密密钥的生成过程

Fig. 3　Generation process of block-level encryption key when $n=4$

私有云将密钥 k_B 和 k_C 发送给用户，用户使用密钥 k_B 对数据块$\{B_i\}$进行加密能得到密文 $C_{B_i}=\mathrm{Encrypt}(k_B,B_i)$，用户只需将$\{C_{B_i}\}$上传到 S-CSP，S-CSP 返回指向数据块的指针。

4.5.2　文件下载

当需要下载文件 F 时，用户同样发送下载请求给私有云，私有云将验证用户所拥有的权限是否符合访问权限 P_F。当验证通过时，私有云才授权用户获取密文$\{C_{B_i}\}$，用户获取密文$\{C_{B_i}\}$后使用密钥 k_B 解密出全部的数据块$\{B_i\}$，即 $B_i=\mathrm{Decrypt}(k_B,C_{B_i})$，从而恢复文件 F。

4.5.3　数据块的完整性校验

当用户解密获取到全部数据块$\{B_i\}$时，用户需要验证从 S-CSP 下载的数据块是否完整，用户只需通过下载到的数据块$\{B_i\}$来构造 Merkle 哈希树（与 4.5.1 节的方法一样），也将得到根节点 *ROOT*，从而获取到密钥 k_C'，此时只需将 k_C'与本地存储的 k_C 进行比较，只有当 k_C 与 k_C'相同时，才能说明存储在 S-CSP 的数据块没有被丢失或篡改。

5　安全性分析

本方案通过引入权限等级函数和去重系数来进行安全去重，解决了传统的收敛加密算法容易遭受暴力攻击、密文计算开销过大等问题。在假定私有云为可信第三方的前提下，本节将从去重标签的安全性、加密密钥的安全性和数据的机密性 3 个方面进行安全性分析。基于 3 个方面的安全性分析，我们能够证明本文提出的数据去重方案是安全的。

5.1　去重标签的安全性

在本方案中，去重标签由数据内容和用户的权限计算得到，即$\{\phi_{F,p'}=H_1(H_0(F),k_{p'})\}$。缺少有效的权限 p_U 的外部攻击者，将无法通过私有云的验证，从而获取不到权限密钥和去重标签。内部攻击者可以看作拥有部分有效权限的外部攻击者，因此相比于外部攻击者，内部攻击者具有更多的威胁性。假设内部攻击者拥有的权限集为 p_U'，伪造了去重标签 $\phi_{F,p'}$并上传到 S-CSP，如果去重标签 $\phi_{F,p'}$是有效的，则必然满足等式$\{\phi_{F,p'}=H_1(H_0(F),k_{p'})\}$，其中 p'也需要满足三元关系式 $Y\{(t,p,p')\}=1$。计算有效的去重标签需要其他用户的权限参与，由 δ 来决定，δ 越大，需要其他用户的权限数目就越少。当 $\delta=1,t_p=1$ 时，Y 等同于二元式 R；当 δ 接近无限大，即 $T(p')=0$ 时，在 $t_p=1$ 的前提下，标签的计算开销最小，但系统的安全性也最低，攻击者只需获取到最低等级的权限，即可计算出所有的去重标签。此外，由于其余用户的权限集 p'存储在私有云上，伪造的去重标签 $\phi_{F,p'}$中的权限参数 p'只会满足 $Y\{(t,p,p')\}\neq1$，因此内部攻击者将无法得到有效的标签集与 S-CSP 进行交互。

5.2　加密密钥的安全性

在基于文件级和数据块级的去重方案中，通过构造 Merkle 哈希树得到了加密密钥 k_F 和 k_B，加密密钥只存储在对应的用户和私有云上的密钥管理服务器中。由于加密密钥是由多层哈希运算计算得到的，即使攻击者与恶意用户进行串谋获取到了加密密钥 k_F，他也无法通过加密密钥来获得关于文件内容的任何信息。

5.3　数据的机密性

在去重过程中，用户使用加密密钥 k_F 来加密数据内容，并把密文 C 上传到 S-CSP 进行存储，而非使用传统的收敛加密来生成密文 C，这样使得生成密文变得不可预测，有效地抵制了暴力攻击。在私有云服务器为可信第三方的假设下，保存在 S-CSP 中的只有密文 C，S-CSP 即使与未授权的用户进行共谋，也无法通过伪造密钥 k_F 来解密密文。

6　性能分析与实验评估

本节将对 Li 等[20]提出的方案与本文提出的 MTHDedup 方案进行仿真实验对比。仿真实验环境为：英特尔酷睿四核 CPU、i5-7300HQ、2.50 GHz 主频、8 GB 内存的 PC 机，使用 Java 编程语言进行性能仿真测试。我们使用的哈希算法为具有强碰撞性的 SHA-256 哈希算法。

实验 1　测试不同权限集数目对生成最终密文的时间开销的影响。实验分别选取权限集数目为 100，200，400，800 和 1000 进行测试，测试文件的大小恒定为 20MB，实验结果如图 4 所示。可以看出，随着使用权限集数目的增多，两方案中密

文计算所需要的时间开销也会增加。文中提出的 MTHDedup 方案与 Li 的方案相比，计算密文所需要的时间开销会更少，并且在权限集数目增多的情况下，两个方案中所需要的时间开销差异会更加明显。因为在 MTHDedup 方案中，加密密钥由 Merkle 哈希树得到(其中 n 为权限集数目)，因此当 n 逐渐增大时，计算密文的时间开销与 Li 的方案相比，也仅仅呈现出缓慢增长的趋势。

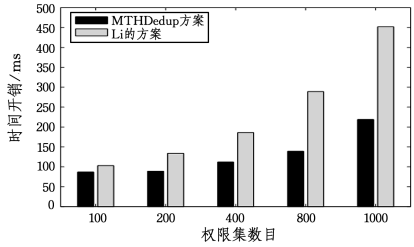

图 4 不同权限集数目下生成密文的时间开销

Fig. 4 Time overhead of ciphertext generation on different number of privilege sets

实验 2 测试不同文件大小下生成去重标签的时间开销。实验分别选取文件大小为 10，20，30，40，50，60，80 和 100(单位为 MB)进行仿真测试，实验结果如图 5 所示。随着文件大小的增加，计算去重标签的时间开销也会增加，文件大小和生成去重标签的时间开销大致趋于一定的线性关系。

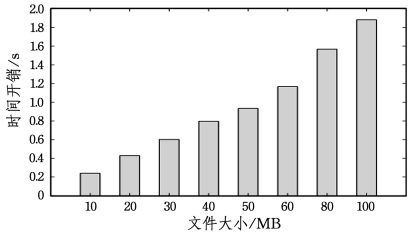

图 5 不同文件大小下去重标签生成的时间开销

Fig. 5 Time overhead of tag generation with different file sizes

实验 3 测试不同数据块数目下生成 Merkle 哈希树的时间开销。实验选取数据块数目分别为 1000，2000，4000，8000 和 10000，使用两组数据块大小分别为 4 kB 和 8 kB 进行仿真测试，实验结果如图 6 所示。当数据块数目越多时，生成 Merkle 哈希树所需要的时间开销也会增加，并且数据块越大，需要的时间开销也将越多，当数目越多时，数据块大小对所需时间开销的影响也将更加明显。

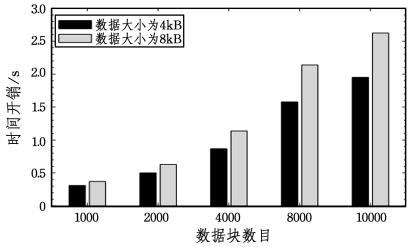

图 6 不同数据块数目下生成 Merkle 哈希树的时间开销

Fig. 6 Time overhead generated by Merkle hash trees with different number of blocks

结束语 本文提出了一种混合云环境下基于 Merkle 哈希树的数据安全去重方案 MTHDedup，以有效地支持文件级和数据块级的加密数据去重检测，提高数据的去重率；通过 Merkle 哈希树来构造加密密钥，降低了密文生成的计算开销。安全性分析表明，MTHDedup 方案能够有效地抵制外部攻击者和内部攻击者发起的暴力攻击，提高数据的安全性；仿真实验结果表明，该方案能有效地降低生成密文的计算开销。未来的研究工作主要集中在基于多用户参与协同的情况，验证云存储系统中数据的完整性，保护各参与方的数据不被泄露。

参 考 文 献

[1] GANTZ J, REINSEL D. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the fareast[OL]. http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf.

[2] CLEMENTS A T, AHMAD I, VILAYANNUR M, et al. Decentralized deduplication in SAN cluster file systems[C]// Conference on Usenix Technical. 2009: 8-8.

[3] BELLARE M, NAMPREMPRE C, NEVEN G. Security Proofs for Identity-Based Identification and Signature Schemes[J]. Journal of Cryptology, 2009, 22(1): 1-61.

[4] BELLARE M, PALACIO A. GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks[M]// Advances in Cryptology — CRYP10 2002. Berlin: Springer, 2002: 149-162.

[5] NG W K, WEN Y, ZHU H. Private data deduplication protocols in cloud storage[C]// Acm Symposium on Applied Computing. ACM, 2012: 441-446.

[6] STORER M W, GREENAN K, LONG D D E, et al. Secure data deduplication[C]// ACM International Workshop on Storage Security and Survivability. ACM, 2008: 1-10.

[7] BARACALDO N, ANDROULAKI E, GLIDER J, et al. Reconciling End-to-End Confidentiality and Data Reduction In Cloud Storage[J]. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2017, 6(3): 206-210.

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

[9] STANEK J, SORNIOTTI A, ANDROULAKI E, et al. A secure data deduplication scheme for cloud storage[OL]. http://www.ifca.ai/fc14/papers/fc14_submission_5.pdf.

[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] LIU Z S, HE Z. Deduplication with encrypted data based on Merkle hash tree in Cloud Storage[J]. Computer Engineering and Applications, 2018, 54(5): 85-90. (in Chinese)
刘竹松，何喆. 基于 Merkle 哈希树的云存储加密数据去重复研究[J]. 计算机工程与应用，2018，54(5)：85-90.

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

(下转第 203 页)

续的学术研究工作具有积极的指导意义。

本文提出的基于混合优先级的 one-test-at-a-time 策略弥补了现有研究的一些缺陷,但是否存在普遍适用和效率更高的用例生成策略还需要进一步的探索和研究。

参 考 文 献

[1] WANG Z Y,XU B W,NIE C H,et al. Survey of combinatorial test generation [J]. Journal of Frontiers of Computer Science and Technology,2008,2(6):571-588. (in Chinese)
王子元,徐宝文,聂长海,等. 组合测试用例生成技术[J]. 计算机科学与探索,2008,2(6):571-588.

[2] JIA J T. Research of Automatic Testcase Generation Functions Based onParticle Swarm Optimization Algorithm [J]. Computer Technology and Development,2010,20(9):24-27. (in Chinese)
贾冀婷. 基于粒子群算法的测试用例自动生成方法研究[J]. 计算机技术与发展,2010,20(9):24-27.

[3] WANG Z Y,NIE C H,XU B W,et al. Optimal Test Suite Generation Methods for Neighbor Factors Combinatorial Testing [J]. Chinese Journal of Computers,2007,30(2):200-211. (in Chinese)
王子元,聂长海,徐宝文,等. 相邻因素组合测试用例集的最优生成方法[J]. 计算机学报,2007,30(2):200-211.

[4] WANG Z Y,QIAN J,CHEN L,et al. Generating Variable Strength Combinatorial Test Suite with one-test-at-a-time Strategy [J]. Chinese Journal of Computers,2012,35(12):2541-2552. (in Chinese)
王子元,钱巨,陈林,等. 基于 One-test-at-a-time 策略的可变力度组合测试用例生成方法[J]. 计算机学报,2012,35(12):2541-2552.

[5] BAO X A,YANG Y J,ZHANG N,et al. Test Case Generation Method Based on Adaptive Particle Swarm Optimization [J]. Computer Science,2017,44(6):177-181. (in Chinese)
包晓安,杨亚娟,张娜,等. 基于自适应粒子群优化的组合测试用例生成方法[J]. 计算机科学,2017,44(6):177-181.

[6] BERGH F V D,ENGELBRECHT A P. Cooperative learning in neural networks using particle swarm optimizers[J]. South African Computer Journal,2000,26:84-90.

[7] RATNAWEERA A,HALGAMUGE S K,WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Transactions on Evolutionary Computation,2004,8(2):240-255.

[8] HU W,LI Z S. A Simpler and More Effective Particle Swarm Optimization Algorithm [J]. Journal of Software,2007,18 (4):861-868. (in Chinese)
胡旺,李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报,2007,18(4):861-868.

[9] COHEN D M,DALAL S R,FREDMAN M L,et al. The AETG System:An Approach to Testing Based on Combinatorial Design [J]. IEEE Transactions on Software Engineering,1997,23(7):437-444.

[10] CHEN X,GU Q,WANG Z Y,et al. Framework of Particle Swarm Optimization Based Pairwise Testing [J]. Journal of Software,2011,22(12):2879-2893. (in Chinese)
陈翔,顾庆,王子元,等. 一种基于粒子群优化的成对组合测试算法框架[J]. 软件学报,2011,22(12):2879-2893.

[11] WILLIAMS A W. Determination of Test Configurations for Pair-Wise Interaction Coverage[C]// International Conference on Testing Communicating Systems:TOOLS and Techniques. DBLP,2000:59-74.

[12] SHI Y,EBERHART R C. Fuzzy adaptive particle swarm optimization[C]// Proceedings of the 2001 Congress on Evolutionary Computation. IEEE Xplore,1997:101-106.

[13] EBERHART R C,SHI Y. Tracking and optimizing dynamic systems with particle swarms[C]// Proceedings of the 2001 Congress on Evolutionary Computation,2001. IEEE,2002:94-100.

[14] YOU B,CHEN G,GUO W. A Discrete PSO-Based Fault-Tolerant Topology Control Scheme in Wireless Sensor Networks[C]// Advances in Computation and Intelligence—International Symposium(Isica 2010). Wuhan,China,DBLP,2010:1-12.

[15] GONG M G,JIAO L C,YANG D D,et al. Research on Evolutionary Multi-Objective Optimization Algorithms [J]. Journal of Software,2009,20(2):271-289. (in Chinese)
公茂果,焦李成,杨咚咚,等. 进化多目标优化算法研究[J]. 软件学报,2009,20(2):271-289.

[16] ZHANG N,YAO L,BAO X A,et al. Multi-Objective Optimization Based On-Line Adjustment Strategy of Test Case Prioritization[J]. Journal of Software,2015,26(10):2451-2464. (in Chinese)
张娜,姚澜,包晓安,等. 多目标优化的测试用例优先级在线调整策略[J]. 软件学报,2015,26(10):2451-2464.

(上接第 192 页)

[13] YIN Q Q. Secure deduplication approach based on Bloom Filter in hybrid cloud storage environments[J]. Computer Engineering and Applications,2018,54(10):73-80. (in Chinese)
尹勤勤. 基于 Bloom Filter 的混合云存储安全去重方案[J]. 计算机工程与应用,2018,54(10):73-80.

[14] BELLARE M,KEELVEEDHI S,RISTENPART T. Message-Lcked Ecryption and Secure Deduplication[M]// Advances in Cryptology — EUROCRYPT 2013. Berlin:Springer,2013:296-312.

[15] BELLARE M,KEELVEEDHI S,RISTENPART T. DupLESS:server-aided encryption for deduplicated storage[C]// Usenix Conference on Security. USENIX Association,2013:179-194.

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

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

[18] YANG C,ZHANG M,JIANG Q,et al. Zero knowledge based client side deduplication for encrypted files of secure cloud storage in smart cities [J]. Pervasive & Mobile Computing,2017,41:243-258.

[19] LIU X,SUN W,LOU W,et al. One-tag checker:Message-locked integrity auditing on encrypted cloud deduplication storage[C]// IEEE Conference on Computer Communications. IEEE,2017.

[20] LI J,LI Y,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.