doi:10.11959/j.issn.1000-436x.2017057

新的云存储文件去重复删除方法

杨超[1]，纪倩[1]，熊思纯[1]，刘茂珍[1]，马建峰[1]，姜奇[1]，白琳[2]
(1. 西安电子科技大学网络与信息安全学院，陕西 西安 710071；
2. 西安邮电大学计算机学院，陕西 西安 710121)

摘 要：去重被广泛地应用于云存储服务中以节省带宽和存储资源，然而，客户端去重复化删除中仍存在安全缺陷，使外部攻击者可访问用户私有数据。基于密文的跨用户的去重复化删除方案 Xu-CDE 被首次提出，该方案支持在外部攻击者和诚实且好奇的服务器同时存在的场景下保护数据隐私，具有良好的理论意义和代表性。然而该方案中的用户所有权认证凭据缺乏实时性保护，以致不能抵抗重放攻击。针对该缺陷，提出改进方案云存储中基于 MLE 与随机数改进的客户端密文去重（MRN-CDE），引入随机因子以保障认证凭据的实时性，并利用 MLE-K_P 算法通过原始文件提取密钥代替用文件本身作为加密密钥，在提高安全性的同时大大降低了运算量。经过安全性分析与测试，结果表明，所提出的改进方案 MRN-CDE 在 Xu-CDE 方案的基础上，不仅增强所有权认证的安全性，而且在时间效率上也有所提高，对于云端大文件的文件去重效果尤其佳，具有一定的应用价值。

关键词：加密数据；去重；所有权认证；实时性；云存储
中图分类号：TP391 **文献标识码**：A

New method for file deduplication in cloud storage

YANG Chao[1], JI Qian[1], XIONG Si-chun[1], LIU Mao-zhen[1], MA Jian-feng[1], JIANG Qi[1], BAI Lin[2]
(1. School of Cyber Engineering, Xidian University, Xi'an 710071, China;
2. School of Computer, Xi'an University of Posts & Telecommunications, Xi'an 710121, China)

Abstract: Deduplication is widely used in cloud storage service to save bandwidth and storage resources, however, the security of client deduplication still flaws in an external attack to access a user's private data. Xu-CDE, a deduplication solution of encrypting data for multi-client was first proposed, which could protect the privacy of data from the external attackers and honest but curious server, with favorable theoretical meaning and representativeness. However, in Xu-CDE, the user ownership authentication credentials were lack of instantaneity protection, which could not resist replay attack. As an improvement to the flaw, the protocol MRN-CDE (MLE based and random number modified client-side deduplication of encrypted data in cloud storage) was proposed, adding random number in order to ensure the instantaneity of the authentication credentials, and using the algorithm of MLE-K_P to extract key from original file to replace the file itself as an encryption key. As a consequence, the new protocol improved security while significantly reduced the amount of computation. After the safety analysis and the actual tests, results show that based on Xu-CDE, the proposed protocol MRN-CDE has stronger security of ownership, and improves time efficiency. Specially, the new protocol works better on large files in cloud with a certain value.

Key words: encrypted data, deduplication, proof of ownership, instantaneity, cloud storage

收稿日期：2016-10-31；**修回日期**：2016-12-29
通信作者：杨超，chaoyang@xidian.edu.cn
基金项目：国家自然科学基金资助项目（No.61672415, No.61671360, No.61672413）；陕西省教育厅科研基金资助项目（No.14JK1665）
Foundation Items: The National Natural Science Foundation of China (No.61672415, No.61671360, No.61672413), The Natural Science Project by Shaanxi Province Office of Education (No.14JK1665)

1 引言

在信息化时代，海量数据的产生使数据存储成为问题，企业及个人用户纷纷使用云存储服务来存储数据，当用户需要上传体量大的数据时，会造成巨大延迟，并且在全球范围内，有大量的数据冗余度。针对当前问题，研究人员提出了重复数据删除技术(data deduplication)——在数据块级别或文件级别降低数据冗余度来提升存储资源和网络带宽利用率。利用该技术，针对同一份文件，无论多少用户想要上传，若文件已存在于服务器，所有的文件拥有者链接到该文件，客户无需再上传。

然而，云存储重复数据删除技术的安全问题是用户最大的质疑和担心[1]。就目前云环境下的安全去重[2]来看，Harnik 等[3]第一次提出客户端重复数据删除的安全问题，他们描述了可能应用于云存储服务中的 3 种攻击：识别客户端的个人身份信息、窃取文件内容和产生隐信道。实际上，Mulazzani 等[4]已经利用客户端重复数据删除系统的漏洞，将云存储服务 Dropbox 滥用为文件分发系统，非法分发受版权保护的数据。不仅如此，Dropbox 软件中的一个新漏洞，将用户的私人数据向公众暴露长达 4 h；甚至有报道称攻击者利用 Twitter 官方客户端软件的漏洞可以非法访问用户的账户。因此，对于用户来说，实现安全有效的文件去重复化删除，并且保护数据的机密性和用户的隐私是非常必要和紧迫的，而且这是新型云存储大规模应用和推广前亟需解决的问题。

关于安全的客户端重复数据删除，早先的一个基本的方案是“hash-as-a-proof”，即利用文件散列值的唯一性进行重复数据删除。然而，Halevi 等[5]发现该方案中攻击者容易获得文件散列值，并导致文件的非授权访问。因此，针对该攻击首次提出了一种跨客户端的去重复化删除技术——文件所有权认证，其核心思想是在文件上建立散列树，服务器与客户端进行协议交互，认证文件的所有权，完成重复数据删除。

接着，Roberto 等[6,7]提出一个基于明文的客户端重复数据删除方案，服务端请求客户端返回文件中随机 k 个位置的 k bit，服务端可以根据其所拥有的明文文件验证客户是否真正拥有该文件。但该方案存在以下缺点：1) 方案中没有考虑诚实但好奇的服务器将侵犯数据的保密性；2) 如果文件具有较低的信息量，则文件所有权证明的证据将会经常发生碰撞。文献[8]首次提出针对密文的去重删除方案——收敛加密。该方案以文件散列值作为文件的加密密钥，相同的明文能够生成系统范围内相同的密文，然后在收敛加密生成的密文上进行文件所有权认证。然而，文件所有权认证的安全模型假定，一定量的文件信息可以被泄露，这将会导致收敛加密不安全。因为在收敛加密中的加密密钥是通过确定的方法由文件生成且很容易被泄露，因此该方案也是存在安全漏洞的[9]。

接下来，文献[10]针对密文去重[11]问题首次提出了用于敏感数据的多客户端交叉的重复数据删除方案 Xu-CDE。其主要特点是能够在外部攻击者和诚实而好奇的服务器存在的场景下，保护数据的隐私。然而该方案在安全性及实用效率方面存在 2 个缺陷：1) 认证缺乏实时性不能抗重放攻击；2) 文件加密低效。

针对上述缺陷，本文提出一种新的去重删除方案——MRN-CDE，其改进思想是：引入随机数，保证每一次文件所有权认证过程的及时有效性，即使攻击者截获密文散列值，没有随机数，也不能计算出实时有效的证据，不能通过文件所有权认证，达到避免重放攻击的目的；利用 MLE（message-locked encryption）方案中的 K_P 算法[12]从原始文件中提取出密钥代替用文件 F 本身作为加密密钥，MLE-K_P 算法的安全性在文献[12]中已证明。因此，MRN-CDE 方案不仅提高了安全性，而且大大减小了加解密过程的运算量，使用户可以更方便高效地使用云存储服务。所以本文的目的是保留方案原本的特点，修补方案，使其更加安全高效。

2 系统与攻击者模型

2.1 系统模型

服务器，即云存储服务提供者，拥有巨大的存储空间为用户提供数据备份服务。由于用户数量巨大，服务器使用重复数据删除技术提高存储资源的利用率。服务器拒绝已存在文件的重复上传，因此，在服务器范围内，数据具有唯一性。

客户端用户。用户向服务器注册后，能够合法使用存储服务。他们向服务器上传并下载文件。在用户上传的数据当中，包含隐私数据。

在本系统中，云存储服务使用者的角色分为 2 类。

1) 首位上传者(FU, first uploader)，即第一位上

传文件的用户。FU 根据手中的原文件计算后续文件所有权证明过程所需要的中间证据(IP, intermediate proofs)，并传给服务器。

2) 后续上传者(SU, subsequent uploader)。SU 所上传的文件已存在于服务器中，因此他需要向服务器证明他对文件的所有权，即与服务器进行文件所有权证明交互。

2.2 攻击者模型

在客户端重复数据删除系统中，有 2 类攻击者需要考虑。

外部攻击者（EA, external attacker）。该类攻击者可能获得部分用户数据，也可截获网络中传输的数据，进而发动重放攻击。

内部攻击者（IA, internal attacker）。能够直接访问文件的攻击者称为内部攻击者。存储服务器、服务器的管理者和实现存储服务的客户端软件属于此类。诚实但好奇的服务器会维护数据的完整性和可用性，但对敏感数据具有一定的好奇，它能够在用户无法获知的情况下，非法访问用户的数据。

3 Xu-CDE 方案的缺陷分析与改进

文献[10]首次提出了抗 Poison 攻击的客户端加密重复数据删除方案——Xu-CDE。该方案的特点是针对可能“偷窥”文件内容的服务器，提供密文上的客户端去重方案，同时抵抗去重复删除的“Poison 攻击”。然而通过分析发现，文献[10]的方案不能保证文件所有权认证的实时性，以致不能抵抗重放攻击，并且加解密过程运算量大。

3.1 系统模型

在 Xu-CDE 方案中，FU 选取随机密钥 τ 对文件 F 进行加密得到密文文件 C_F，用文件 F 对密钥 τ 进行加密得到密钥密文 C_τ，计算文件的散列值 $hash(F)$，向服务器发送数据分组，服务器存储并计算 C_F 的散列值 $hash(C_F)$。SU 计算 $hash(F)$，发送至服务器，服务器查找到对应条目则说明已经有人上传了该文件，服务器端已存储文件的密文 C_F。然后进行文件密钥的发放：将 C_τ 发送给 SU， SU 将待上传的文件 F' 解密 C_τ 得到 τ'，用 τ' 加密 F' 得到 $C_{F'}$，计算 $hash(C_{F'})$ 发送给服务器，服务器将其与自己计算的 $hash(C_F)$ 对比。若满足 $hash(C_{F'})$= $hash(C_F)$，则说明 SU 确实拥有该文件，并且没有 Poison 攻击的发生，通知 SU 将手中的文件删除，需要使用文件时，用手中密钥解密文件密文即可；若文件所有权通不过，说明 SU 并非真正拥有文件，仅拥有文件的散列值，或者是 FU 发动了“Poison 攻击”，FU 和 SU 需根据所拥有文件计算相关信息，甚至上传文件原文做进一步验证。

然而，经过分析发现，Xu-CDE 方案存在以下 2 个缺陷。

1) 认证缺乏实时性不能抗重放攻击

Xu-CDE 方案中同一文件密文的散列值相同，对于拥有该文件的多个用户，每次文件所有权认证的过程中向服务器传送的都是同一散列值，没有任何保证认证实时性的因子，不能抗重放攻击。

2) 文件加密低效

用一个比较大的文件作为密钥加解密一个比较短的密钥内容，运算量非常巨大，降低了云存储及文件去重效率，将对客户产生不可忍受的时延，甚至可能导致该系统存在巨大的拒绝服务攻击风险。

由缺陷 1)导致的攻击如图 1 所示，攻击者未能正确完成的步骤用虚线标出。

因为在上述文件所有权认证中，认证凭证都没有实时性的因子保护，所以，外部攻击者可轻易截获网络中的 $hash(C_{F'})$，并在新一轮的文件所有权证明中，冒充一个合法 SU，重放认证凭证 $hash(C_{F'})$，骗取服务器的信任，获得文件密文 C_F，进而非法获取文件拥有者的所有权限，甚至包括修改或删除原始文件。

Xu-CDE 方案本质是一个认证的过程，认证协议的基本原则是需要确认认证凭证的实时性。之所以存在上述重放攻击，是因为不论 FU 还是 SU，对于同一文件 F 计算出的密文散列值是相同的，对于拥有该文件的多个用户，每次文件所有权的认证过程中向服务器发送的密文散列值是同一个值，没有添加可保证认证凭据实时性的因子，因此，外部攻击者 EA 可发动重放攻击，导致原始文件所有权的非法认证和权限滥用。

3.2 新的改进方案-MRN-CDE

针对 Xu-CDE 方案的缺陷 1)中文件所有权的认证过程必须保证每一次认证的实时性，在改进方案 MRN-CDE 中引入随机数，保证每一次的认证凭据的实时性，即使攻击者同时截获，和随机数也不能伪造，达到抵抗重放攻击的目的。针对 Xu-CDE 方案的缺陷 2)，利用 MLE-K_P 算法[12]，通过原始文件提取出密钥来加密密钥，代替用文件本身作为加密

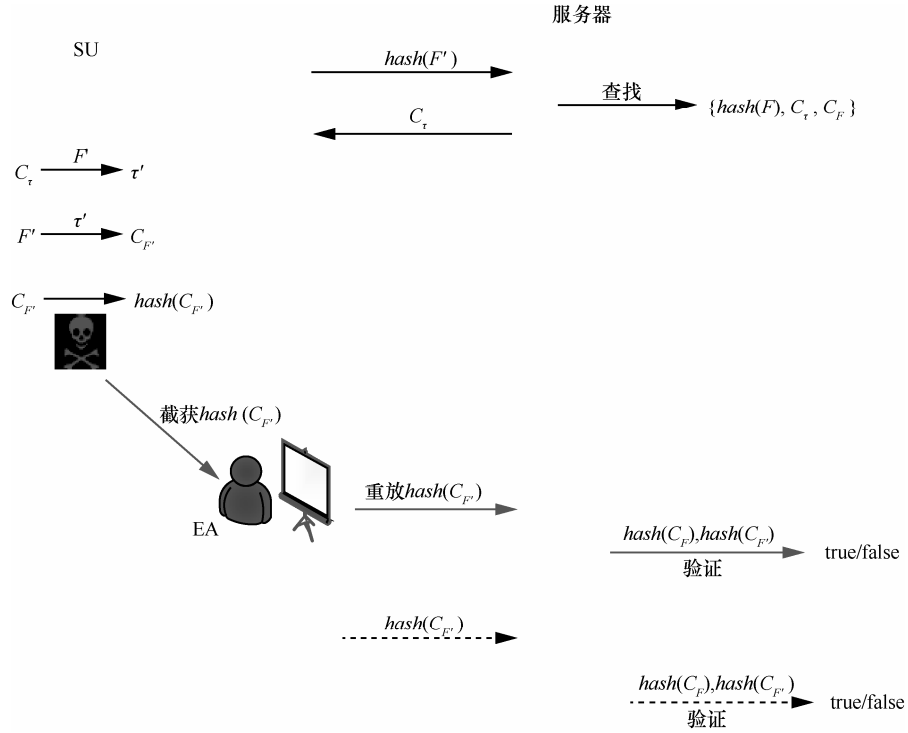


图 1　Xu-CDE 方案的一种攻击

密钥，提高安全性且大大减小加解密过程运算量。新的改进方案 MRN-CDE 如图 2 所示。

4　改进方案 MRN-CDE 的安全性分析

本节将对所提出的新方案 MRN-CDE 进行安全性证明。由于文件是加密后存储，服务器存储的是密文，并且服务器手中没有密钥，几乎不可能解密得到明文，因此可以抵抗内部攻击。下面，主要针对外部攻击进行安全性分析，建立一个安全模型游戏 Game。

在 MRN-CDE 的安全游戏 $G_A^{\mathrm{CDE}}(t,s)$ 中，除了存在 MRN-CDE 协议中所规定的客户端和服务器端，同时还存在一个多项式时间 PPT 的攻击者 A。其中，$t>s>\lambda$，λ 是系统的安全参数，t 是协议中所有权证明的文件 F 所包含信息量的下界，s 是固定参数。

假定攻击者 A 在安全游戏的交互中，至多能从文件 F 中获取 $(t-s)$ bit 的信息量。不失一般性，假定文件 F 是从任意分布 $\{0,1\}^M$ 中抽取的，其最小信息量不小于 t，其中，公开的整数参数 $M\geqslant t$ 且以安全参数 λ 为约束界。

1) 学习阶段

多项式时间 PPT 的攻击者 A 可以被动窃听客户端和服务器端的所有协议交互，也可以主动进行询问，主要包括以下询问类型，并且可以在进入下阶段前进行多次询问。

E-Query：攻击者可以询问一个随机预言机 O^F，O^F 通过在文件 F 上运行一个概率的编码算法获得 3 个参数值 $(hash(F),\beta,hash(\beta\|C_F))$，其中，$\beta$ 是协议执行中服务器端生成的一次性随机数。

V-Query：攻击者模拟服务器端，以 $hash(F)$ 为输入与一个正常的客户端进行协议交互，向客户端返回伪造的密钥加密文件 C_τ' 和随机数 β'，并在得到客户端的返回值后得到服务器端的内部状态 (y_1,y_2)，其中，y_1 表示是否接受用户的文件所有权证明，y_2 表示所获得的文件密文 C_F 和随机数 β 的散列值。

P-Query：攻击者模拟客户端，以随机数 β 和密钥密文 C_τ 为输入与正常的服务器端进行协议交互，通过向随机预言机 O^F 询问获得密钥加密密钥 δ'、文件密钥 τ、文件密文 $C_{F'}$ 和 $hash(\beta\|C_{F'})$，并与服务器端完成协议交互。

2) 质证阶段

在质证阶段，攻击者 A 基于学习阶段获取的文件 F 的 $(t-s)$ bit 的信息量，构造一个文件 F'，并利用获取的真实 $hash(F)$ 与服务器进行完整的 MRN-CDE 协议交互，企图通过服务器对他的文件所有权的验证。在该阶段，攻击者 A 只能与服务器

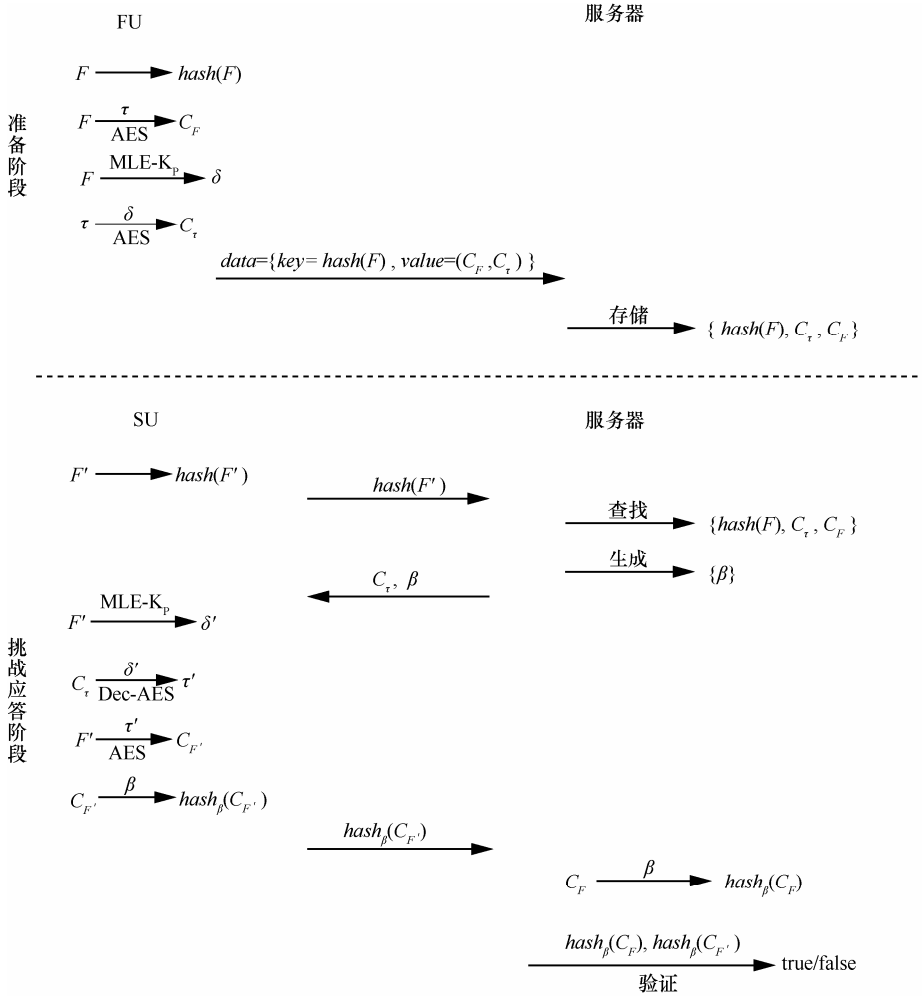


图 2　改进方案 MRN-CDE

进行正常协议交互，不能进行任何学习阶段的查询操作。在质证阶段结束后，如果攻击者 A 通过质证，则服务器输出 $b=1$，称之为“游戏赢家”，否则输出 $b=0$，其中，$b\in\{0,1\}$。

定义 1　MRN-CDE 安全性。λ 是系统的安全参数，t 是协议中所有权证明的文件 F 所包含信息量的下界，s 是固定参数，且 $t>s>\lambda$，协议 MRN-CDE 是 (t,s) 安全的，当且仅当对于任意的 $t\in N$ 和任意的文件来源分布 $\{0,1\}^M$，攻击者在上述安全模型游戏中没有文件 F 且最多能获得文件 (t,s) bit 信息量的情况下，攻击者能否成功说服服务器其拥有文件 F 所有权的概率是可忽略的，即 $P_r[A_{\text{wins}}]\leqslant negl(\lambda)$。

定理 1　假设 $hash(\cdot)$ 是一类抗强碰撞的散列函数，加密算法 E_{AES} 具有语义安全的，密钥提取算法 MLE-K$_P$ 是 PRV\$-CDA 安全的，则本文所提出的 MRN-CDE 是安全的。

证明　假设攻击者 A 在安全模型游戏中没有文件 F 的情况下，能否以不可忽略的概率 $P_r[A_{\text{wins}}]\cdot negl(\lambda)$ 通过服务器对文件所有权的验证，则利用攻击者 A 的能力可以攻破 $hash(\cdot)$ 的抗强碰撞性和加密算法 E_{AES} 的语义安全，从而反证得证。具体来讲，攻击者 A 在没有原始文件 F 的情况下，为了赢得安全模型游戏，其需要构造一个合适的值 x，使 $hash(x)=hash(\beta\|C_F)$，这意味着 $x=(\beta\|C_F)$ 或 $x\neq(\beta\|C_F)$。当 $x\neq(\beta\|C_F)$，则使 $hash(x)=hash(\beta\|C_F)$ 成立的值 x 就是散列函数 $hash(\cdot)$ 的一个强碰撞，这与 $hash(\cdot)$ 是一类抗强碰撞的散列函数的假设矛盾；当 $x=(\beta\|C_F)$，则攻击者 A 需要直接猜测 C_F，其猜测概率为 $\dfrac{1}{|C_F|}=\dfrac{1}{\{0,1\}^M}\leqslant negl(\lambda)$，显然与攻击者的能力假设矛盾；或者，攻击者 A 需要在没有文件 F 的情况下计算 $E_{\text{AES}-\tau}(F)$ 的值，在此情况下，攻击者 A 需要在最多获得文件 (t,s) bit

信息量的情况下构造文件 F，并且同时猜测加密密钥 τ，则此情况下的攻击者 A 成功的概率为 $\frac{1}{|F-(t-s)|}=\frac{1}{\{0,1\}^{M-(s-t)}}\times\frac{1}{|\tau|}\leqslant negl(\lambda)$，通常情况下，文件来源分布 $\{0,1\}^M$ 至少是 1 KB 以上，即 $\frac{1}{\{0,1\}^M}$ 相当于 2^{-1024}，认为该值是可以忽略的，即这显然与攻击者的能力假设矛盾；或者，攻击者 A 需要计算 $D_{\mathrm{AES}-\delta}(C_\tau)$，要么攻击者 A 在没有文件的情况下构造密钥解密密钥 δ，则要么攻击者 A 攻破密钥提取算法 MLE-K_P，而这与假设矛盾，要么攻击者直接猜测 δ，则其成功的概率为 $\frac{1}{\{0,1\}^{M-(s-t)}}\times\frac{1}{|\delta|}$，通常情况下文件来源分布 $\{0,1\}^M$ 至少是 1 KB 以上，则该概率也小于可忽略的概率 $negl(\lambda)$，得证。

综上所述，提出的 MRN-CDE 方案，利用随机数保证了文件所有权每一次认证过程的实时性，可抵抗重放攻击，不仅达到了可证明的安全级别，更满足了上述重要的安全需求，在安全性上优于现有的 Xu-CDE 方案，如表 1 所示。

表 1 安全属性比较

方案	原始数据参与计算	动态证据	抵抗重放攻击
MRN-CDE	√	√	√
Xu-CDE	NONE	NONE	NONE

5 改进方案 MRN-CDE 的性能测试与分析

5.1 测试方案与测试场景

本节旨在对改进方案 MRN-CDE 和 Xu-CDE 方案进行性能方面的测试与对比，测试方案如下：选取 3 台测试机，依次配置为服务器、FU 客户端、SU 客户端，针对不同大小的文件：1 MB、16 MB、128 MB、2 GB，分别按照 MRN-CDE 和 Xu-CDE 方案的算法步骤各进行 200 次测试，其中，MLE-K_P 算法使用 MD5 从文件本身提取出固定长度的密钥，记录 2 个方案各个阶段所需时间，通过时间比较 2 个方案的性能。

租用云服务器，配置参数如表 2 所示。

表 2 客户端配置参数

硬盘	CPU	内存	操作系统
500 GB	双核	4 GB	Windows 7（64 bit）

本地搭建客户端，配置参数如表 3 所示。

使用 C 语言编写测试程序，在相同的网络条件下，对 MRN-CDE 和 Xu-CDE 这 2 个方案进行测试。

5.2 测试数据与结果分析

针对不同大小的文件，从 1 MB、16 MB、128 MB 到 2 GB，分别按照 MRN-CDE 和 Xu-CDE 方案的步骤进行测试，根据 200 次实验统计的 200 组数据，记录 2 个方案各个阶段所需时间，通过时间比较 2 个方案的性能。如表 4 所示。

表 4 表示 Xu-CDE 准备阶段的时间分配，可以看出计算文件散列值 $hash(F)$ 与生成密钥密文所用时间相差不大；客户端处理时间包括 $T_{hash}(F)$，$T_{\mathrm{GET}}(\tau)$，$T_{\mathrm{AES}}(\tau,F)$ 和 T_{C_τ}，其中，文件加密时间 $T_{\mathrm{AES}}(\tau,F)$ 所占比例达 70%以上，而计算文件散列值 $hash(F)$ 时间 $T_{hash}(F)$ 与生成密钥密文 C_τ 的时间各

表 3 云服务器配置参数

服务商	位置	Server ID	硬盘	CPU	内存	操作系统	带宽
阿里云	青岛	i-28vj83Web	20 GB	双核	4 GB	Windows Server 2012 标准版 64 位中文版	5 Mbit/s

表 4 Xu-CDE 准备阶段时间

文件大小/MB	准备阶段				阶段总时间/s
	$T_{hash}(F)$/s	$T_{\mathrm{AES}}(\tau, F)$/s	T_{C_τ} /s	文件上传时间/s	
1	0.020 200	0.084 400	0.015 000	16.76	16.88
16	0.260 500	1.251 200	0.254 500	155.36	157.13
128	2.087 700	10.165 000	2.025 500	1 151.93	1 180.49
256	4.347 500	19.927 500	3.869 000	2 630.20	2 686.49
512	8.846 500	52.702 000	9.646 500	5 460.40	5 531.60
1024	18.279 00	104.290 00	18.707 00	11 120.80	11 262.08

占 15%；文件上传所用时间要比客户端处理时间大 2 个数量级。

表 5 表示 Xu-CDE 挑战应答阶段的时间分配，V_1 阶段表示客户端计算 *hash*(*F*)发送给服务器，服务器收到后查找数据库，找到匹配的 *hash*(*F*)值后，发送该散列值对应的密钥密文；还原文件加密密钥与计算文件密文散列值所用时间几乎相同，其中，文件加密时间 $T_{\mathrm{AES}}(\tau,F)$ 占用 70%以上的比例，还原文件加密密钥时间 $T_{\mathrm{GET}}(\tau)$ 与计算文件密文散列值时间 $T_{hash}(C_F)$ 各占 15%。

表 6 表示 MRN-CDE 准备阶段的时间分配，可以看出客户端处理时间主要用于计算文件散列值 $T_{hash}(F)$ 和文件加密 $T_{\mathrm{AES}}(\tau,F)$，生成密钥密文的时间 T_{C_τ} 可忽略不计，准备阶段总时间较 Xu-CDE 方案有所节省，文件上传时间占总时间的 99%以上。

表 7 表示 MRN-CDE 挑战应答阶段的时间分配，V_1 阶段表示客户端计算 *hash*(*F*)发送给服务器，服务器收到后查找数据库，找到匹配的 *hash*(*F*)值后，产生随机数并发送该随机数及对应的密钥密文；该阶段的时间主要用于计算文件密文 $T_{\mathrm{AES}}(\tau, F)$和带随机参数的密文的散列值 $T_{hash}(\beta \| C_F)$，获取密钥过程 $T_{\mathrm{GET}}(\tau)$可忽略不计，

表 5　　Xu-CDE 挑战应答阶段时间

大小/MB	V_1 阶段/s	V_2 阶段			阶段总时间/s
		$T_{\mathrm{GET}}(\tau)$/s	$T_{\mathrm{AES}}(F,\tau)$/s	$T_{hash}(C_F)$/s	
1	0.558 2	0.022 000	0.075 000	0.016 000	0.647
16	0.798 5	0.251 000	1.248 000	0.259 000	2.209
128	2.625 7	2.079 000	9.943 000	2.073 000	24.040
256	4.885 5	4.044 00	20.168 000	4.120 000	27.784
512	9.384 5	8.161 000	40.494 000	8.197 000	54.257
1024	18.817 0	16.218 000	92.701 000	16.504 000	121.388

表 6　　MRN-CDE 准备阶段时间

文件大小/MB	准备阶段				阶段总时间/s
	$T_{hash}(F)$ /s	$T_{\mathrm{AES}}(\tau,F)$/s	T_{C_τ} /s（可忽略）	文件上传时间/s	
1	0.022 000	0.087 000	0.000 002	16.76	16.86
16	0.350 00	1.233 000	0.000 001	155.36	156.94
128	2.256 000	10.361 000	0.000 001	1 151.93	1 164.55
256	4.428 000	20.593 000	0.000 001	2 630.20	2 655.22
512	9.090 000	46.071 000	0.000 001	5 460.40	5 515.56
1024	17.624 00	95.479 000	0.000 001	11 120.80	11 233.90

表 7　　MRN-CDE 挑战应答阶段时间

文件大小/MB	V_1 阶段/s	V_2 阶段			阶段总时间/s
		$T_{\mathrm{GET}}(\tau)$/s（可忽略）	$T_{\mathrm{AES}}(F,\tau)$/s	$T_{hash}(\beta \| C_F)$ /s	
1	0.570 0	0.000 001	0.073 000	0.016 000	0.627
16	1.188 0	0.000 001	1.247 000	0.248 000	2.033
128	2.794 0	0.000 001	10.235 000	2.042 000	12.815
256	4.966 0	0.000 001	20.254 000	4.034 000	24.826
512	9.628 0	0.000 001	39.470 000	8.200 000	48.208
1024	18.162 0	0.000 001	91.090 00	16.213 000	107.841

表8 Xu-CDE与MRN-CDE对比时间

文件大小/MB	准备阶段/s		挑战应答阶段/s		总时间/s	
	Xu-CDE	MRN-CDE	Xu-CDE	MRN-CDE	Xu-CDE	MRN-CDE
1	16.88	16.86	0.647	0.627	17.527	17.487
16	157.13	156.94	2.209	2.033	159.339	158.973
128	1 180.49	1 164.55	24.040	22.815	1204.53	1 187.365
256	2 686.49	2 655.22	27.784	24.826	2 714.274	2 680.046
512	5 531.60	5 515.56	54.257	48.208	5 585.857	5 563.768
1024	11 262.08	11 223.90	121.388	107.841	11 383.468	11 341.741

对于通过验证的用户，无需再次上传文件，时间节省99%以上。

从表4和表5可以看出，散列算法，AES（128位密钥）加密算法、文件上传3个操作所用的时间与文件的大小成正比关系，耗时由大到小排序：文件上传>AES加密>散列算法；采用客户端数据去重方法，可以大大减少用户的备份等待时间并节省大量带宽资源，提高了用户体验指数。

表8中将Xu-CDE与MRN-CDE的2个阶段及去重过程总时间进行对比准备阶段利用MD5算法，通过原始文件提取出密钥δ来加密密钥τ，代替用文件F本身作为加密密钥，实际测试结果也说明准备阶段MRN-CDE方案所需的时间小于Xu-CDE方案，效率更高，说明通过原始文件提取密钥代替文件本身来加密密钥是可行且有效的。挑战应答阶段的实际测试结果表明MRN-CDE方案明显优于Xu-CDE方案，且随着文件增大，2个方案时间差越大，即说明MRN-CDE效率更高，说明通过原始文件提取密钥代替文件本身来加密密钥是可行且有效的。

文件去重总时间对比说明，对于小文件进行完整的文件去重过程，改进方案MRN-CDE与Xu-CDE方案所需的总时间相差不多，效率相当，随着文件增大改进方案MRN-CDE所需的时间比Xu-CDE方案少，即效率更高，并且文件越大，时间差异越明显，说明对于大文件的文件去重改进方案MRN-CDE更佳。

综上所述，改进方案MRN-CDE降低了去重过程的总时间，效率更高，不仅提高了去重过程的安全性，在时间效率上也有所提高，对于大文件的文件去重效果尤其好，具有较好的应用价值。

6 结束语

在信息化时代，企业及个人用户纷纷使用云存储服务来存储数据，重复数据删除技术被广泛地应用于云存储服务中以节省带宽和存储资源，然而，云存储重复数据删除技术的安全问题是用户最大的质疑和担心。客户端重复数据删除中存在安全缺陷，使外部攻击者能够访问用户私有数据[13]。文献[12]首次提出用于加密数据的多客户端交叉的重复数据删除方案Xu-CDE，该方案能在外部攻击者和诚实而好奇的服务器存在的场景下保护数据的隐私，具有较好的理论意义和一定的代表性。然而该方案中的用户所有权认证凭据缺乏实时性保护，以致不能抵抗重放攻击。针对该缺陷，本文提出改进方案MRN-CDE，引入随机数以保证认证凭据的实时性，并利用MLE-K_P算法通过原始文件提取密钥代替用文件本身作为加密密钥，在提高安全性的同时大大降低了运算量。经过安全性分析与实际测试，结果表明，本文所提出的改进方案MRN-CDE比Xu-CDE方案不仅增强了所有权认证的安全性，而且在时间效率上也有所提高，对于云端大文件的文件去重效果尤其佳，具有一定的应用价值。

参考文献：

[1] BARACALDO N, ANDROULAKI E, GLIDER J, et al. Reconciling end-to-end confidentiality and data reduction in cloud storage[C]// Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security. 2014.

[2] 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 Communi-

cations, 2016, 37(8): 167-184.

[3] HARNIK D, PINKAS B, SHULMAN-PELEG A. Side channels in cloud services, the case of deduplication in cloud storage[J]. IEEE Security and Privacy, 2010, 8(6): 40-47.

[4] MULAZZANI M, SCHRITTWIESER S, LEITHNER M, et al. Dark clouds on the horizon: using cloud storage as attack vector and online slack space[C]// USENIX Security Symposium. 2011.

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

[6] ROBERTO D P, ALESSANDRO S. Boosting efficiency and security in proof of ownership for deduplication[C]//ACM Symposium on Information, Computer and Communications Security. 2012: 81-82.

[7] PIETRO D, ROBERTO, SORNIOTTI A. Proof of ownership for deduplication systems: a secure, scalable, and efficient solution[J]. Computer Communications, 2016, 82(2): 71-82.

[8] DOUCEUR J, BOLOSKY W, THEIMER M. US Patent 7266689: encryption systems and methods for identifying and coalescing identical objects encrypted with different keys[P]. 2007.

[9] GONZÁLEZ-MANZANO L, ORFILA A. An efficient confidentiality-preserving proof of ownership for deduplication[J]. Journal of Network and Computer Applications, 2015,50(1): 49-59.

[10] XU J, CHANG E C, ZHOU J Y. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage[C]//The 8th ACM Symposium on Information, Computer and Communications Security. 2013: 195-206.

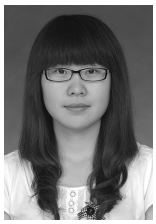
[11] TANG H Y, CUI Y, GUAN C W, et al. Enabling ciphertext deduplication for secure cloud storage and access control[C]//The 11th ACM on Asia Conference on Computer and Communications Security. 2016: 59-70.

[12] BELLARE M, KEELVEEDHI S, RISTENPART T. Message-locked encryption and secure deduplication[C]//Advances in Cryptology-CRYPTO 2013, Lecture Notes in Computer Science. 2013: 374-391.

[13] RASHID F, MIRI A, WOUNGANG I. Proof of retrieval and ownership protocols for enterprise-level data deduplication[C]//The 2013 Conference of the Center for Advanced Studies on Collaborative Research. 2013:81-90.

作者简介：

杨超（1979-），男，陕西西安人，西安电子科技大学副教授，主要研究方向为密码学与网络安全、云计算及移动智能计算安全。



纪倩（1989-），女，山西大同人，西安电子科技大学硕士生，主要研究方向为云计算和存储安全。

熊思纯（1992-），女，湖南娄底人，主要研究方向为云计算和网络安全。

刘茂珍（1993-），女，山东临沂人，主要研究方向为云存储安全。

马建峰（1963-），男，陕西西安人，西安电子科技大学教授、博士生导师，主要研究方向为计算机系统安全、移动与无线安全、系统可生存性和可信计算。

姜奇（1983-），男，安徽全椒人，西安电子科技大学副教授，主要研究方向为无线网络安全、安全协议。

白琳（1980-），女，陕西商州人，西安邮电大学副教授，主要研究方向为网络安全与智能信息处理。