密文去重系统中的数据访问控制策略

贾春福[1,2]，哈冠雄[1,2]，李瑞琪[1,2]
（1. 南开大学网络空间安全学院，天津 300350；
2. 天津市网络与数据安全技术重点实验室，天津 300350）

摘 要：针对云存储中现有密文去重系统大多使用收敛加密，数据所有者无法对外包数据进行有效访问控制的问题，设计了支持身份认证、授权去重、权限更新等访问控制功能的密文去重系统。外包数据仅与授权用户去重，未授权用户无法获取数据信息；通过 CP-ABE 与 ElGamal 私钥的动态拆分更新数据的访问权限；使用自我控制对象封装用户数据及其访问策略，对数据访问者进行身份认证并确保访问控制策略有效执行。安全性分析与仿真实验表明，所提系统实现了数据访问控制且具有较高的执行效率。

关键词：安全策略更新；授权去重；自我控制对象；访问控制；云数据安全
中图分类号：TP309.2
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2020062

Data access control policy of encrypted deduplication system

JIA Chunfu[1,2], HA Guanxiong[1,2], LI Ruiqi[1,2]
1. College of Cyber Science, Nankai University, Tianjin 300350, China
2. Tianjin Key Laboratory of Network and Data Security Technology, Tianjin 300350, China

Abstract: To solve the problem that convergent encryption was commonly used in existing encrypted deduplication systems in cloud storage and data owner couldn't effectively enforce access control on their outsourced data, an encrypted deduplication system was proposed to support access control functions such as identity authentication, authorization deduplication and the update of access control policy. The outsourced data was only deduplicated with the authorized users, and the unauthorized users couldn't obtain any data information. CP-ABE and the partition of the ElGamal private key were used to update the access control policy of data. Self-control objects was used to encapsulate user's data and its access policy, providing authentication for data visitors and ensuring the access control policies enforced effectively. Security analysis and simulation results demonstrate that the proposed system enables data access control and executes efficiently.

Key words: update of security policy, authorized deduplication, self-control object, access control, cloud data security

1 引言

如今，全球每天都会产生大量数据。互联网数据中心（IDC, Internet Data Center）预计 2020 年的全球数据总量可能会达到 40 ZB。随着云计算的迅猛发展，大量的个人和企业都选择将数据外包给云服务提供商。有的云服务商为节省存储空间开始采用数据去重技术，即多个用户上传相同数据时服务器只存储一份。

近年来，云数据安全问题频出，用户为保证数

收稿日期：2019−11−14；**修回日期**：2020−03−03
基金项目：国家重点研发计划基金资助项目（No.2018YFA0704703）；国家自然科学基金资助项目（No.61972215, No.61772291, No.61702399, No.61972073）；天津市自然科学基金资助项目（No.17JCZDJC30500）
Foundation Items: The National Key Research and Development Program of China (No.2018YFA0704703), The National Natural Science Foundation of China(No.61972215, No.61772291, No.61702399, No.61972073), The Natural Science Foundation of TianJin (No.17JCZDJC30500)

据隐私，需要将数据先加密再上传至云存储服务器[1]。然而，传统密码学安全的加密算法与数据去重技术难以结合，数据加密后云存储服务器无法对用户上传的密文进行重复检测。文献[2]提出了收敛加密（CE, convergent encryption），CE 使用基于文件内容生成的哈希值作为对称加密的密钥加密文件，相同明文加密后将得到相同密文，因此，云存储服务器能够对密文进行重复检测。但 CE 无法保证可预测（即明文空间$\{M_i\}$有限，M_i 表示文件）文件的安全性，攻击者可以通过猜测有限明文空间中的 M_i，依次加密$\{M_i\}$中的明文并与截获的密文对比，通过字典攻击破解出密文对应的明文。DupLESS[3]中使用专门的密钥服务器生成密钥，通过在密钥服务器端进行速率限制抵抗攻击者的字典攻击。但 CE 和 DupLESS 都属于消息锁定加密（MLE, message locked encryption）[4]的范畴。MLE 中相同明文被加密成相同密文，这严重泄露了数据频率信息，Li 等[5]提出了基于数据块局部性的频率分析攻击，攻击者可通过统计明文块和密文块的频率信息有效推测出大量密文数据块所对应的明文信息。此外，现有的客户端去重系统容易受到攻击者的侧信道攻击[6]和所有权欺骗攻击[7]。在侧信道攻击中，攻击者可以通过观察客户端上传文件哈希值后服务器端是否去重来判断该文件的存储情况，威胁其他用户的数据隐私；所有权欺骗攻击是指攻击者在没有完整文件的情况下，通过文件的部分信息（如文件哈希值）从服务器端获取整个文件的访问权限。Li 等[8]和 Shin 等[9]分别提出使用收敛扩散和差分隐私抵抗侧信道攻击。Halevi 等[7]提出了所有权证明（PoW, proof of ownership）方案，抵抗恶意攻击者的所有权欺骗攻击。在 PoW[7,10]方案中，文件所有者能够高效安全地向云存储服务器证明其拥有完整文件内容。文献[11]提出了差异化去重的思想，设计了基于混合云模型的授权去重概念，授权去重中每个用户被赋予一系列权限。云存储服务器中的文件都对应一组访问权限，满足访问权限的用户才能够进行去重检测。客户端提交去重检测前，用户需要输入文件及自己的属性，当属性满足权限要求时用户才能够检测到文件去重。文献[12]通过结合 MLE 和 CAONT（convergent all-or-nothing transform）实现了密文去重中的密钥更新，对数据进行加密后得到存根信息$(st_1,st_2,\cdots,st_n)$和 TP 分组信息$(TP_1,TP_2,\cdots,TP_n)$，通过重新加密体量较小的存根信息实现外包数据的密钥更新。但文献[11-12]都是基于 MLE 的确定性加密，会泄露用户数据的频率信息，并且单纯利用密码学技术实现去重数据的机密性，缺乏对数据的分布式管理，无法确保用户访问策略的有效执行。

密文去重系统属于云存储系统，而云存储系统中的虚拟化和多租户特征导致了用户数据的所有权与管理权分离，用户不知道与其共享资源的实体身份，这一特性给用户数据带来了巨大威胁。因此，在云数据管理中，身份认证和访问控制是保护用户数据隐私性和机密性的重要防线。Squicciarini 等[13]提出了自我控制对象（SCO, self-controlling object）的概念，通过面向对象编程封装用户的敏感数据和访问策略，确保用户为数据设置的访问策略能够有效地被执行。文献[14]提出了基于 SCO 技术设计的敏感数据保护方案，防止用户数据被其他用户发布，确保用户对外包数据的访问控制。Zafar 等[15]提出 SCO 技术可以拓展到数据可信删除和数据去重领域。

本文关注去重数据访问控制中的身份认证、授权去重与权限更新问题，这些问题在很多实际场景中都具有很大的实用价值。例如，公司选择外包数据至云存储服务器端，而公司内的部分数据可能仅允许具有某些权限的用户进行去重检测，或者某些数据仅在部门内去重，其他部门的用户不应得到该数据的任何信息，上述情况均需要应用授权去重。例如公司的某个项目由 A、B 这 2 个团队共同开发，但项目进行到中期时公司决定由团队 C 代替团队 B，项目数据的访问权限应从 A、B 属性修改为 A、C 属性，此时便需要更新应用数据的访问权限。

2 本文工作

引言中的实际场景充分体现了外包数据访问控制的应用需求，而目前的密文去重系统大多基于 MLE，难以提供外包数据的高效访问控制。密文去重中的侧信道攻击和所有权欺骗攻击等安全威胁均是基于去重系统中多个用户共享同一密文这一特点。如果系统中不能提供外包数据的高效安全的访问控制，用户将无法控制与其共享同一数据的实体身份，这将造成严重的数据安全问题。现有密文去重系统中的访问控制[11-12]主要基于确定性加密方案，并且在权限更新时需要下载并重新加密数据后上传。这种方式具有以下缺点：首先，确定性加密容易使数据受到字典攻击并且泄露数据的频率信息；其次，频繁的数据传输与加解密大大增加了

客户端的计算开销和系统的带宽开销。此外，现有方案一般仅使用密码学方法提供数据的访问控制，缺乏数据的分布式管理，难以保证当数据位于云存储服务器或其他用户的客户端时访问策略仍能有效地被执行。因此，实际场景中的应用需求和密文去重系统在访问控制方面的功能限制是本文工作要解决的主要问题。

针对上述问题，结合 SCO 技术、多种密码算法与密码协议，本文设计并实现了一种支持身份认证、授权去重和访问权限更新的客户端密文去重系统。该系统中，用户可为外包数据设定访问控制策略，确定数据与哪些属性的用户去重，并能够高效地更新数据的访问权限。

本文系统在去重标签中嵌入了属性密钥，基于消息认证码的安全性，实现了安全的授权去重，未授权用户无法进行侧信道攻击，数据对于未授权用户而言是语义安全的。客户端使用 SCO 技术封装用户数据和访问策略，提供了对数据访问者的身份认证并实现了用户数据的分布式管理，确保数据的访问策略无论在云存储服务器还是其他客户端中均能有效执行。本文系统利用密文策略属性基加密（CP-ABE, ciphertext-policy attribute-based encryption）[16]和 ElGamal 加密算法等密码学技术实现了高效的数据访问权限更新，更新过程中仅需重新拆分 ElGamal 私钥，不需要用户数据文件的任何传输或加解密，系统的带宽开销和客户端的计算开销均可以忽略。使用随机密钥加密取代确定性加密，混淆了数据的频率信息，通过在满足访问权限的用户间共享密钥来实现密文去重，同时有效抵制了攻击者的字典攻击和频率分析攻击。最后，安全性分析和仿真实验证明本文系统在有效实现去重数据访问控制的同时仍具有较高的执行效率。

3 相关知识和技术概述

3.1 SCO 技术

SCO 是一种策略执行技术，将用户数据和访问策略封装在可执行 JAR（Java archive）包当中，通过适应性的安全策略保证数据安全。SCO 可能会被分布式地存储在多个服务器中，数据接收者可随时下载 SCO，其工作流程如图 1 所示[13]。

当接收者试图访问 SCO 中的数据内容时，其内部封装的程序将验证用户输入的身份证书是否满足访问策略，若该证书满足访问策略，SCO 的应用程序部件可为用户解密其中封装的密文。基于 SCO 的数据共享方案不需要可信第三方完成身份认证和访问控制，所有核心的安全部件均封装在 SCO 中。

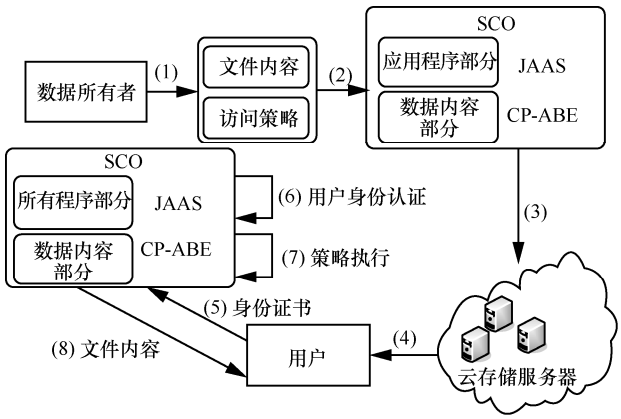

图 1 SCO 工作流程

客户端可使用 SCO 封装数据和安全策略，其中安全策略包括访问控制、认证、用法控制等。客户端将封装完成的 SCO 发送至云存储服务器，数据接收者访问 SCO 时通过输入身份证书进行认证。

SCO 的内部架构[13]如图 2 所示，其中包括数据内容和应用程序两部分。

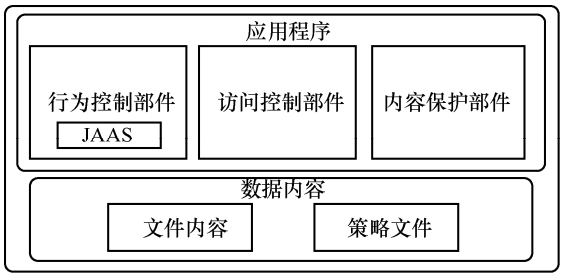

图 2 SCO 内部架构

数据内容部分存储用户封装的文件内容和策略文件；应用程序部分是一组软件部件，负责验证用户身份证书及访问 SCO 内的数据内容。应用程序部分包括行为控制部件、访问控制部件和内容保护部件。行为控制部件使用 Java 中的 JAAS（Java authentication and authorization service）实现，验证用户输入的 X.509 证书；访问控制部件根据身份验证和授权的结果访问数据内容；内容保护部件管理数据内容部分，处理数据内容的更新。

3.2 密文策略属性基加密

CP-ABE 是一种公钥加密算法。加密方可以在密文中嵌入所需的访问控制策略，只有当解密方的

属性满足密文中的访问策略时才能够解密数据。CP-ABE 加密包括以下 4 个算法。

Setup。输入安全参数 L，输出公钥 PK 和主密钥 MK。PK 用来加密数据，MK 用来生成用户的属性私钥 UK。

KeyGen。输入用户的属性集合 UA 和主密钥 MK，输出属性私钥 UK。

Encryption。输入明文 M、访问控制策略 ACP 和公共参数 PK，输出密文 C。

Decryption。输入密文 C 和用户属性私钥 UK，如果用户属性 UA 满足密文 C 中的访问策略，输出明文 M；否则输出 $\perp$。

3.3　ElGamal 加密

ElGamal 加密[17]是一种基于离散对数的公钥加密算法，其安全性基于判定性 Diffie-Hellman 问题的困难性。ElGamal 加密算法由以下步骤组成。

1) Initialization。选择素数 p，计算 p 的原根 c，计算 $b = c^x \bmod p$，其中，x 是随机选择的私钥，公钥为 $\{p, b, c\}$。

2) Encryption。生成随机值 r，计算消息 m 的密文 $E(m) = mb^r \bmod p = mc^{rx} \bmod p$，计算 $g = c^r \bmod p$。

3) Decryption。解密时使用私钥 x 解密数据 $D_x(E(m)) = g^{-x}E(m) \bmod p = (c^r)^{-x} mc^{rx} \bmod p = m \bmod p$。

3.4　所有权证明

所有权证明协议使客户端能够向云存储服务器证明其确实具有完整文件。假设云存储服务器拥有完整数据 M，并根据数据内容计算认证信息 $w = f(M)$。客户端为证明其具有完整文件，需要发送所计算的认证信息 v 给云存储服务器，云存储服务器检查 v 是否与 w 相等，判断客户端是否具有完整文件，具体过程如图 3 所示。

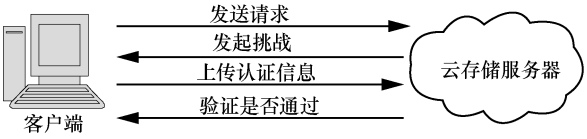

图 3　所有权证明过程

4　系统模型

4.1　系统架构

本文方案中设计的密文去重系统架构如图 4 所示，由客户端、云存储服务器和分布式密钥服务器组成，每个用户都拥有自己的属性及与该属性对应的权限私钥和属性密钥。权限私钥安全地存储在客户端，用来解密 CP-ABE 密文，当用户的权限私钥满足密文的访问结构时可解密出明文；属性密钥存储在分布式密钥服务器，帮助客户端生成有效的去重标签。

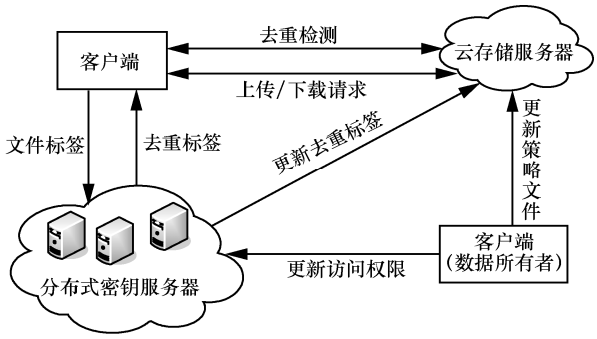

图 4　密文去重系统架构

1) 云存储服务器。为个人或企业提供云存储服务，用户可将需要外包的数据及其相关信息上传至云存储服务器，并能够随时访问和下载这些数据。为提高存储效率和降低通信开销，云存储服务器将根据用户的去重标签等信息对数据进行去重。

2) 分布式密钥服务器。存储系统中使用 ElGamal 私钥 x 和 CP-ABE 主密钥 MK，帮助用户生成去重标签，并且可通过动态拆分 x 和 CP-ABE 加密为用户更新数据的访问策略。

3) 客户端。帮助用户外包数据至云存储服务器，节省用户的本地数据存储和管理开销。系统中的每个数据都对应一个数据所有者，其可通过客户端对外包数据设定访问策略，仅当其他用户的属性满足数据访问策略时才可与此数据进行去重检测并下载解密数据。当需要修改数据访问策略时，数据所有者可通过与云存储服务器和分布式密钥服务器的交互来更新数据的访问策略。

4.2　安全模型

为规范本文系统的安全性分析，本文提出以下安全性假设。

1) 云存储服务器是诚实但好奇的，会按照既定协议存储用户外包的数据，但会尝试获取用户的数据信息。

2) 分布式密钥服务器是诚实但好奇的，会按照既定协议完成生成去重标签、拆分密钥成分、权限更新等操作，但会试图获取用户的数据信息。

3) 系统中存在不可信的恶意攻击者，他们会通

过截获客户端与云存储服务器及分布式密钥服务器之间的通信数据或攻击云存储服务器中的数据内容，尝试获取合法用户的数据信息。

4) 部分合法用户可能会与攻击者发起联合攻击，通过伪造去重标签或身份证书等方式获取自身权限外的数据信息。

4.3 设计目标

本文关注去重数据的访问控制问题，提出了一种新的云数据密文去重系统，支持以下功能。

1) 身份认证。当实体试图访问 SCO 内封装的数据时，SCO 将基于实体输入的身份证书和 SCO 内部的访问控制策略决定该实体是否有权限访问数据内容。身份证书无效或不符合访问控制策略的实体将无法获取到任何数据信息。

2) 授权去重。系统中的不同用户拥有不同的权限，每个用户能够基于自己的权限执行去重检查。用户外包数据至云存储服务器时可选择该数据与哪些属性的用户去重，未授权用户无法进行去重检测。

3) 权限更新。数据所有者可更新云存储服务器端的数据访问权限、撤销或添加访问策略中的属性。权限更新后，被撤销访问权限的用户将不能与该数据进行去重检测，且不能下载解密数据；只有被授予访问权限的用户才可以检测到数据在服务器端的存储情况。

本文系统的设计目标包括性能目标和安全目标。性能目标包括以下几点。

1) 较高的加密/解密效率。系统中客户端对数据使用高效的对称加密算法，而公钥加密算法仅用于加密密钥和策略文件。

2) 较高的权限更新效率。系统中的权限更新仅需重新拆分 ElGamal 密钥及执行 CP-ABE 加密算法，不需要重新下载与解密文件，具有较高的执行效率。

安全目标包括以下几点。

1) 授权去重安全性。未授权用户无法与其权限外的数据进行去重检测，这需要去重标签的不可伪造性和不可区分性。不可伪造性表示任何用户不能伪造其权限外的任何有效去重标签；不可区分性表示用户不能够通过去重标签得到任何额外信息，例如该标签代表哪些属性或文件。

2) 数据机密性。恶意攻击者无法通过截获客户端发送的 SCO 或攻击服务器端存储的 SCO 获取到用户数据的任何信息。即使攻击者与内部用户进行联合攻击，也无法通过伪造去重标签和身份证书等方式获取到自己权限外的数据信息。

3) 权限更新安全性。数据所有者更新数据权限后，被撤销权限的用户不能与服务器进行去重检测或下载解密数据；被授予访问权限的用户可针对该数据进行去重检测，通过所有权证明即可下载解密数据。

5 方案设计

5.1 方案设计思想

本文设计了一种支持身份认证、授权去重和权限更新的密文去重系统。基本思想是，客户端首先通过与分布式密钥服务器（DKS, distributed key server）的认证和交互得到去重标签 d_token。去重标签中不仅包含数据信息，还包含表示用户属性的属性密钥。客户端将 d_token 发送至云服务提供商（CSP, cloud server provider）进行去重检测，若云存储服务器检测到数据已存储，则与客户端进行所有权证明。所有权证明通过后，该客户端具有下载文件的权限，不需要上传完整文件；若检测数据尚未存储，则要求客户端上传完整文件。客户端将加密后的密文数据和访问策略封装在 SCO 中上传至云存储服务器。后续用户上传数据时，由于去重标签中带有用户属性，云存储服务器可根据去重标签检测该用户是否具有数据去重的权限，实现授权去重。未授权用户将无法进行去重检测，数据对于他们而言是语义安全的。

当用户需要更新数据的访问权限时，客户端与云存储服务器和分布式密钥服务器交互，通过动态拆分 ElGamal 私钥和 CP-ABE 加密实现数据访问策略的更新。

5.2 方案详细设计

系统可分为加密数据上传、密文授权去重、数据下载解密和访问权限更新 4 个模块。

5.2.1 *加密数据上传*

假设 A 属性用户需要上传文件 F 至云存储服务器，客户端首先计算文件的哈希值 $H(F)$作为文件标签，其中，$H(\cdot)$为密码学安全的哈希函数。客户端与分布式密钥服务器进行身份认证，认证后客户端将文件标签 $H(F)$发送至分布式密钥服务器。分布式密钥服务器验证用户信息后，找到 A 属性用户对应的属性密钥 k_a，计算文件去重标签 d_token=$H(H(F),k_a)$并将其返回给客户端，客户端将 d_token 发送至云存储服务器进行去重检测，具体过程如图 5 所示。

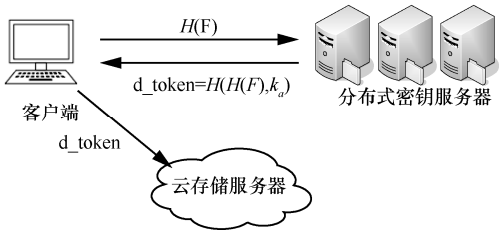

图 5　获取去重标签

云存储服务器通过去重标签检测该数据是否已存储。若已存储，与客户端进行所有权证明协议，证明通过则允许客户端下载数据；若未存储，要求客户端上传完整文件。

若需要客户端上传完整文件，客户端可以先设置该数据的访问权限。假设客户端希望和 B 属性用户共享密文，即指定 A（上传者的属性）、B 属性用户可与数据进行去重检测，客户端将共享属性 B 发送至分布式密钥服务器。密钥服务器找到 B 属性对应的属性密钥 k_b，计算 B 属性用户的去重标签 d_token=$H(H(F),k_b)$。随后，密钥服务器将存储的 ElGamal 私钥 x 随机拆分成 n+1 份，分别为 $\{x_1,x_2,\cdots,x_n\}$和 x'。密钥服务器将$\{x_1,x_2,\cdots,x_n\}$分布式地存储在 n 个密钥服务器中，使用 CP-ABE 加密 x' 得到 $C_{x'}$，访问结构为 $A\cup B$，即 A、B 属性用户的权限私钥均可解密 $C_{x'}$。密钥服务器将 B 属性的去重标签 d_token 和 CP-ABE 密文 $C_{x'}$发送至云存储服务器。

客户端封装 SCO 步骤如下。

1) 客户端选择随机密钥 key，使用 AES 加密算法加密文件 F 为密文 C。

2) 用户构建访问策略文件 policy，将访问结构 $\{A,B\}$写入 policy。数据解密中过程，仅当访问者提供的身份证书满足访问策略时，SCO 才执行解密程序。客户端使用 ElGamal 公钥加密策略文件 policy 和 AES 密钥 key，得到密文 C_{policy} 和 C_{key}。

3) 客户端将 C、C_{key} 和 C_{policy} 封装在 SCO 的数据内容部分，本文方案的 SCO 结构如图 6 所示，包括身份认证、部分解密、密文更新和数据解密。其中，身份认证在实体访问 SCO 时执行，部分解密和密文更新由云存储服务器端执行，数据解密在客户端解密数据时执行。SCO 根据身份认证的结果为不同实体执行不同操作。

4) 为了防止攻击者截获 SCO 后分析其内部的程序部分，客户端生成 SCO 后对其内部的程序部分进行代码混淆。

客户端生成 SCO 后将其上传至云存储服务器，如图 7 所示。云存储服务器调用 SCO 中的密文更新程序，将分布式密钥服务器发送来的 CP-ABE 密文 $C_{x'}$封装进 SCO 中。然后，云存储服务器将 SCO 数据的去重标签设定为 d_token=$\{H(H(F),\ k_a),\ H(H(F),k_b)\}$，后续用户上传的去重标签在此集合中才可进行去重检测。

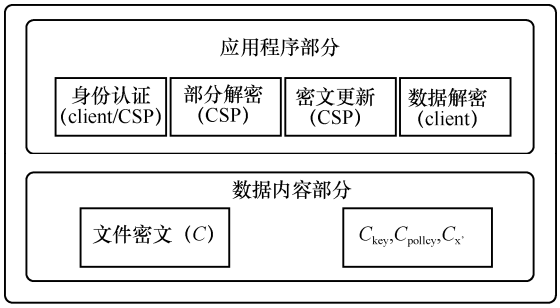

图 6　本文方案的 SCO 结构

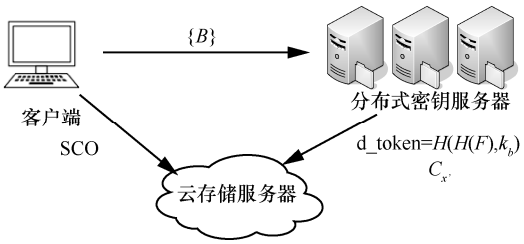

图 7　数据加密上传

5.2.2　密文授权去重

假设 B 属性用户同样外包文件 F 至云存储服务器，此时系统将进行密文授权去重过程，如图 8 所示。

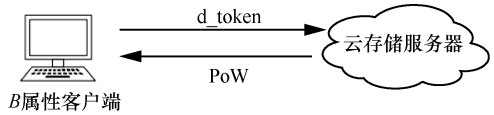

图 8　授权去重

B 属性客户端从分布式密钥服务器得到自己的去重标签 d_token=$H(H(F),\ k_b)$并发送至云存储服务器，云存储服务器发现客户端发送的去重标签已在云端存储（本文所述场景中为 A 属性用户上传）。此时，云存储服务器将告知 B 属性用户文件已存储，需要进行所有权证明。证明通过后，B 属性用户即可从服务器下载完整数据。

A、B 属性外的用户无法检测到数据在服务器端的存储情况。如图 9 所示，B'属性用户外包文件 F，上传去重标签 d_token=$H(H(F),\ k_{b'})$。云存储服务器并未存储过该去重标签，无法检测到数据重复。

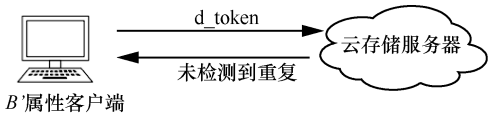

图 9　未授权无法去重

通过密文授权去重过程可以看出，数据所有者设定的数据访问策略为{*A*,*B*}，只有 *A*、*B* 这 2 个属性的用户能够识别数据重复。未授权用户即使上传相同文件，云存储服务器仍无法与该数据进行去重，数据对于未授权用户来说是语义安全的。

5.2.3 数据下载解密

客户端向云存储服务器提出下载请求，如图 10 所示，云存储服务器从 *n* 个密钥服务器中检索 *n* 个密钥成分$\{x_1,x_2,\cdots,x_n\}$，并向 SCO 申请部分解密。SCO 通过身份认证程序验证云存储服务器提交的证书后，使用$\{x_1,x_2,\cdots,x_n\}$对 SCO 内部的 C_{policy} 和 C_{key} 调用部分解密程序。C_{policy} 和 C_{key} 是 ElGamal 密文，密文形式为$(c^r,\text{key}c^{rx}\bmod p)$和$(c^r,\text{policy}c^{rx}\bmod p)$。SCO 中的部分解密程序使用$\{x_1,x_2,\cdots,x_n\}$进行部分解密，得到更新的密文为$(c^r,\text{key}(c^r)^{x-x_1-x_2-\cdots-x_n}\bmod p)$和$(c^r,\text{policy}(c^r)^{x-x_1-x_2-\cdots-x_n}\bmod p)$，云存储服务器将部分解密后的 SCO 发送至客户端。

客户端收到 SCO 后，输入自己的 X.509 证书和权限私钥 sk。SCO 若验证证书有效，则使用 sk 解密 CP-ABE 密文 $C_{x'}$，若权限私钥 sk 满足密文的访问结构，则可解密出 *x'*。SCO 使用 *x'*解密云存储服务器部分解密后的 policy 的密文，得到$(c^r,(c^r)^{x-x_1-x_2-\cdots-x_n-x'}\text{policy}\bmod p)=\text{policy}\bmod p=\text{policy}$。解密得到访问策略 policy 后，SCO 验证用户的 X.509 证书是否满足策略文件 policy 中设定的访问权限。在上述例子中，SCO 的验证程序将判断用户证书是否来自 *A* 属性或 *B* 属性客户端。若用户证书满足访问权限，SCO 使用 *x′*解密 C_{key}，得到$(c^r,(c^r)^{x-x_1-x_2-\cdots-x_n-x'}\text{key}\bmod p)=\text{key}\bmod p=\text{key}$。得到 AES 密钥 key 后，SCO 使用 key 解密密文 *C* 得到明文 *M*。

在数据下载解密过程中，用户只有能提供有效且满足访问策略的证书并拥有满足密文访问结构的权限私钥 sk 时，SCO 才能解密出明文。任何未经授权的用户均无法得到明文数据。

5.2.4 访问权限更新

数据所有者可能会更改数据的访问策略，例如授予新属性用户访问权限或撤销旧属性用户访问权限，这时就需要使用访问权限更新模块。访问权限更新时，某个数据所有者（代表全体的数据所有者）通过与云存储服务器和分布式密钥服务器的交互更新数据的访问权限。例如，数据所有者希望将文件 *F* 的访问权限{*A*,*B*}修改为{*A*,*B′*}，客户端发送新设定的访问权限{*A*,*B′*}至分布式密钥服务器。分布式密钥服务器将此前存储的{*A*,*B*}的密钥成分$\{x_1,x_2,\cdots,x_n\}$删除，重新拆分 ElGamal 私钥 *x* 为$\{x_1',x_2',\cdots,x_n'\}$和 *x″*，分别把$\{x_1',x_2',\cdots,x_n'\}$存储在 *n* 个密钥服务器中，使用新的访问结构 $A\cup B'$ 加密 *x″*得到新的 CP-ABE 密文 $C_{x''}$。密钥服务器还需计算新的去重标签 d_token=$H(H(F),k_a)$和 d_token=$H(H(F),k_c)$，并将新的去重标签同 $C_{x''}$一起发送至云存储服务器。云存储服务器使用新去重标签取代旧标签，此前存储的旧去重标签将不能再进行去重检测，在本例中 *B* 属性用户的去重标签将失效，而 *B′* 属性用户的去重标签将生效。客户端生成新的访问策略文件 policy′，进行 ElGamal 加密后得到 $C_{\text{policy}'}$并将其发送至云存储服务器。云存储服务器调用 SCO 内部的密文更新程序，使用 $C_{x''}$和 $C_{\text{policy}'}$替换原有的 $C_{x'}$ 和 C_{policy}，完成数据的访问权限更新。

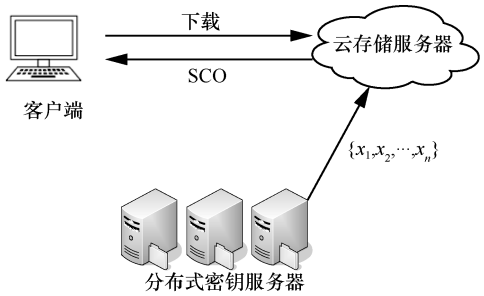

图 10　数据下载解密

访问权限更新后云存储服务器存储的去重标签中已无 *B* 属性的去重标签，*B* 属性用户将无法进行去重检测；并且由于 SCO 中的 C_{policy} 和 $C_{x'}$已替换，被撤销访问权限的 *B* 属性用户的身份证书将无法通过策略文件 policy′的访问控制，也无法使用权限私钥 sk 解密密文 $C_{x''}$。而 *B′*属性用户可以成功地与云存储服务器进行去重检测并下载解密 SCO 中的明文。

上述的访问权限更新中不需要完整文件数据的网络传输和加解密，系统带宽开销较低，且客户端仅需加密体量很小的策略文件，客户端计算开销较低。

5.3　安全性分析

本文方案的安全性分析主要考虑系统中数据机密性、授权去重安全性和权限更新安全性三方面。假设方案中使用的密码原语均是安全的，包括消息认证码、CP-ABE、AES 和 ElGamal 加密算法。假设方案中的安全参数为 n，存在可忽略的函数 negl_1、negl_2 和 negl_3，使 AES、CP-ABE 和 ElGamal 加密算法被攻破的概率分别为 $\mathrm{negl}_1(n)$、$\mathrm{negl}_2(n)$和 $\mathrm{negl}_3(n)$。

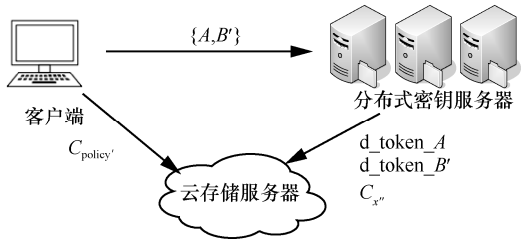

图 11　权限更新

5.3.1　授权去重安全性

定理 1　本文方案满足授权去重安全性，当且仅当方案中的去重标签满足不可伪造性和不可区分性。

标签的不可伪造性指的是恶意用户不能伪造出其自身属性外的去重标签。若 A 属性用户上传文件 F，相应的文件去重标签应为 d_token=$H(H(F), k_A)$。不可伪造性要求 A 属性的攻击者不能伪造出 A'属性的文件 F 的有效去重标签 d_token=$H(H(F), k_{A'})$（$A' \neq A$）。不可区分性表示假设用户具有属性 A，他不能通过 d_token=$H(H(F), k_{A'})$（$A' \neq A$）区分标签中包含了哪个文件或哪个属性。

授权去重的关键在于去重标签的生成，若保证了去重标签的安全性，也就保证了授权去重的安全性。去重标签安全性可分为标签的不可伪造性和不可区分性。下面将证明本文方案中去重标签的不可伪造性和不可区分性，从而证明本文方案具有授权去重安全性。

引理 1　假设方案中使用的消息认证码是密码学安全的，其基于的哈希函数可抵抗碰撞攻击和原像攻击，则攻击者成功伪造出其权限外的去重标签的概率是可忽略的，攻击者成功区分去重标签中的属性信息的概率也是可忽略的。

证明　使用 2 个安全游戏来说明方案中去重标签的不可伪造性和不可区分性，安全游戏描述了系统中攻击者的攻击能力。下面证明攻击者在 2 个给定安全游戏中获得成功的概率均为可忽略的。

1) 去重标签不可伪造性安全游戏

准备阶段　假设游戏中存在挑战者和攻击者，攻击者向挑战者询问去重标签，并试图生成自身权限外的有效去重标签。挑战者可生成去重标签并可利用函数 Verify(d_token)验证去重标签的有效性。攻击者具有文件集合$\{F_1,\cdots,F_s\}$和属性 A_p，挑战者拥有文件集合 $\{F_1,\cdots,F_s\}$ 和属性密钥集合 $\{k_1,\cdots,k_m\}$ $k \xleftarrow{R} \{0,1\}^n$。

询问阶段　攻击者可以从文件集合中选取任意文件 $F_i(i=1,2,\cdots,s)$，然后计算文件标签 $H(F_i)$，并发送 $H(F_i)$和属性 A_p 给挑战者。挑战者根据属性 A_p 找到其相应的属性密钥 k_p，并计算去重标签 d_token= $H(H(F_i),k_p)$发送给攻击者。

挑战阶段　攻击者根据询问阶段得到的信息，计算一个新的去重标签 d_token′=$H(H(F_j), k_q)$并发送给挑战者。挑战者根据文件集合$\{F_1,\cdots,F_s\}$和属性密钥集合$\{k_1,\cdots,k_m\}$检测 d_token′是否为攻击者权限外的有效去重标签，即 d_token′= $H(H(F_j), k_q)$ $(F_j \in \{F_1,\cdots,F_s\}, k_q \in \{k_1,\cdots,k_m\}, q \neq p)$。若 d_token′满足上述条件，则 Verify(d_token′)=1，攻击者赢得游戏。否则，Verify(d_token′)=0，攻击者输掉游戏。

由于攻击者在询问阶段不能获知其他属性的密钥 k_q，因此攻击者若生成有效的 d_token= $H(H(F),k_q)$，则存在 2 种情况，即攻击者未破解出 k_q 或者破解出 k_q。当攻击者未破解出 k_q 时，说明攻击者找到了 k_l，使 $H(H(F),k_l)=H(H(F),k_q)\ (l \neq q)$。但此时消息认证码中的哈希函数不满足抗碰撞特性，与假设中消息认证码是密码学安全的这一点相矛盾。若攻击者通过暴力破解获得了其他属性的密钥 k_q，由于属性密钥 k_q 是从$\{0,1\}^n$中随机选取，暴力猜测成功的概率为$\left(\frac{1}{2}\right)^n$，而此概率是可忽略的。

证毕。

2) 去重标签不可区分性安全游戏

本节的去重标签不可区分性安全游戏是针对属性的不可区分性证明，针对文件的不可区分证明与之类似，这里不再赘述。

准备阶段　假设游戏中存在挑战者和攻击者，攻击者具有 2 个属性 A_0 和 A_1，请求挑战者为其生成去重标签。挑战者拥有属性密钥集合$\{k_1,\cdots,k_m\}$，$k \xleftarrow{R} \{0,1\}^n$，可为攻击者生成去重标签。

询问阶段　攻击者任意选择文件 F_i，计算文件标签 $H(F_i)$，并将其与 2 个属性 A_0 和 A_1 共同发送至

挑战者。挑战者随机选择 $b \xleftarrow{R} \{0,1\}$，计算去重标签 d_token=$H(H(F_i),k_b)$，将去重标签发送给攻击者。

挑战阶段 攻击者输出 $b' \xleftarrow{R} \{0,1\}$。若 $b'=b$，则攻击者赢得游戏；否则攻击者失败。

由于攻击者在安全游戏的询问阶段不能获知属性密钥 k_b，而消息验证码中的哈希函数是抗碰撞的且哈希函数的输出结果是随机的，攻击者只能通过暴力破解猜测密钥 k_b 以区分 b 和 b'，而 k_b 是在 $\{0,1\}^n$ 中随机选取的。因此，$P_r[b=b']=\frac{1}{2}+\frac{1}{2^n}$，其中 $\frac{1}{2^n}$ 是可忽略的。

证毕。

由定理 1 和引理 1 可知，本文方案实现了去重标签的安全性。此外，由于本文方案中实现了所有权证明协议，只有满足访问策略且拥有完整文件的用户才能够使用有效地去重标签检测云存储服务器中的数据存储情况。攻击者不能通过去重标签检测到自身权限外数据的存储情况，无法进行侧信道攻击。

5.3.2 数据机密性

基于 4.2 节安全模型中的假设，即云存储服务器和分布式密钥服务器均为诚实但好奇的，本文提出引理 2，证明方案的数据机密性。

引理 2 假设方案中使用的密码学工具均是密码学安全的。攻击者通过截获通信数据或攻击云存储服务器得到 SCO 时，破解自身权限外的数据明文信息的概率 p_1 是可忽略的；攻击者通过与内部用户发起联合攻击，伪造满足 SCO 中访问策略的身份证书或篡改 SCO 中的身份认证程序，破解自身权限外的明文信息的概率 p_2 是可忽略的。此时，称方案满足数据机密性。

证明

情况 1 攻击者通过截获通信数据或攻击云存储服务器得到 SCO。由于数据均以 AES 密文形式存储，攻击者无法提供有效的身份证书和权限私钥 sk，不能解密出 AES 密钥 key。而攻击者在不知道 key 的情况下，成功破解数据的概率小于或等于破解 AES 的概率，即 $p_1 \leqslant \mathrm{negl}_1(n)$，这一概率是可忽略的。

情况 2 攻击者与内部用户发起联合攻击，伪造身份证书或篡改身份认证程序，试图破解权限外的数据信息。本文方案对 SCO 中的代码进行了代码混淆，攻击者难以篡改 SCO 中的认证程序绕过身份认证。假使攻击者通过伪造身份证书进入解密程序，由于攻击者没有满足访问策略的权限私钥 sk，破解密文 $C_{x'}$ 的概率 p_3 小于或等于破解 CP-ABE 加密算法，即 $p_3 \leqslant \mathrm{negl}_2(n)$，这一概率是可忽略的。而在没有解密 $C_{x'}$ 的情况下，试图解密 ElGamal 密文 C_{key} 的概率 p_4 小于或等于攻破 ElGamal 算法，即 $p_4 \leqslant \mathrm{negl}_3(n)$，此概率可忽略。最终，攻击者只能以可忽略的概率破解出 C_{key}。因此，攻击者破解出明文的概率 p_5 仍小于或等于破解 AES 的概率，即 $p_5 \leqslant \mathrm{negl}_1(n)$。因此，攻击者只能以可忽略的概率破解权限外明文数据。证毕。

由于使用概率性随机加密而非类似收敛加密的确定性加密，本文方案不会暴露数据的频率信息。并且，由于本文方案中使用了随机密钥而非基于数据内容产生的收敛密钥，攻击者在不知道密钥的情况下无法通过遍历明文集合的方式对截获的密文进行字典攻击。5.3.1 节中说明了攻击者无法对其权限外的数据进行侧信道攻击，结合本节中论述的数据机密性，方案中的外包密文数据对未授权用户实现了语义安全。数据在云存储服务器和分布式密钥服务器中均是以密文形式存储的，因此诚实但好奇的云存储服务器和分布式密钥服务器无法获取数据的任何明文信息。

5.3.3 权限更新安全性

权限更新安全性在于保证被撤销权限的用户不能再访问数据内容。

引理 3 假设方案中使用的密码学工具均是密码学安全的，数据所有者更新权限后，被撤销权限的攻击者仍可访问明文数据的概率 p_6 是可忽略的。

证明 权限更新后，密钥服务器将重新拆分 ElGamal 私钥 x 并基于新的访问策略对 x'进行 CP-ABE 加密，被撤销权限的攻击者存储的私钥成分 x'失效，其权限私钥 sk 将无法解密出 CP-ABE 密文。攻击者即使截获密文数据，由于 AES 密钥 key 以 ElGamal 密文的形式存储，其破解出 key 的概率 p_7 小于或等于破解 ElGamal 算法，即 $p_7 \leqslant \mathrm{negl}_3(n)$。因此，攻击者只能以可忽略的概率得到 AES 密钥 key，那么其破解出明文的概率 p_8 小于或等于破解 AES 的概率，即 $p_8 \leqslant \mathrm{negl}_1(n)$，此概率是可忽略的。

此外，由于本文方案实现了去重标签安全性，且在权限更新阶段云存储服务器的去重标签也完成了更新，被撤销权限的用户无法通过自己的去重

标签检测到数据在云端的存储情况，无法进行侧信道攻击。攻击者还可能在自己未被撤销权限前分析 SCO 内部的解密程序或通过攻击分布式密钥服务器试图获取密钥信息。本文方案中对 SCO 的内部程序进行了代码混淆，可以进一步采用白盒加密[18]或将程序放在 Internet SGX（software guard extensions）[19]等可信执行环境中运行以缓解攻击者分析 SCO 解密程序的攻击。由于 ElGamal 私钥 x 拆分后的 $\{x_1,x_2,\cdots,x_n\}$ 分布式地存储在了 n 个不同的密钥服务器中，攻击者要得到 ElGamal 私钥 x 需要在其未被撤销权限的情况下攻击 n 个密钥服务器，得到 $\{x_1,x_2,\cdots,x_n\}$，再加上自身存储的 x' 才能得到完整的 ElGamal 私钥 x，但攻击者同时攻击 n 个密钥服务器是困难的。

结合上述的相关引理，可以得到安全性结论如结论 1 所述。

结论 1　假设方案中使用的密码学工具均是密码学安全的，那么本文方案中攻击者破坏授权去重安全性、数据机密性和权限更新安全性的概率均是可忽略的。本文方案实现了预设的安全目标。

6　仿真与性能分析

本文实现了方案中的密文去重系统，系统采用 C++与 Java 语言编写。为使仿真实验更加接近真实场景，实验中租用了云存储服务器作为实验的服务器端，客户端是个人 PC 平台，共同完成方案中的数据加密上传、下载解密与权限更新过程。搭建的服务器环境如下：云存储服务器位于上海，2 核 4 GB 内存，操作系统为 Windows Server 2016 版 64 位，带宽为 5 Mbit/s；客户端位于杭州，操作系统为 Windows10，处理器为 Inter Core i7-8550U，内存为 16 GB。

系统中的 SCO 应用程序部分、AES 加密和 CP-ABE 加密由 Java 语言编写，数据传输和数据存储由 C++语言编写。方案的实现技术包括 jPBC 库，Proguard、JAAS 和可执行 JAR 包。

本节对所设计的系统给出了运行性能的评估，包括数据加密上传、密文下载解密以及访问权限更新的时间开销。在测试中，关注去重数据访问控制功能所带来的时间开销以及权限更新过程中各部分的时间开销，并与基于收敛加密的密文去重方案进行了性能对比。本节中的所有实验数据均为 10 次以上实验的平均值，除文件的上传下载过程易受网络情况影响外，其他实验过程在多次实验中数值均较为稳定。

6.1　加密上传

1) 首次数据上传

测试 1 MB、5 MB、10 MB、50 MB、100 MB、200 MB、300 MB、400 MB、500 MB、600 MB、700 MB、800 MB 和 1 000 MB 的文件，统计系统中加密数据、封装 SCO 和上传数据的时间开销，测试结果如图 12 所示。由图 12 可以看出，系统的主要时间开销来自数据的上传时间，而数据加密时间和封装 SCO 时间占比较低。图 13 中将本文方案与 CE 在加密过程中产生的时间开销进行了对比，可以看出本文方案在实现外包数据访问控制的同时仅增加了有限开销，系统中为访问控制所添加的封装 SCO 和 ElGamal 加密等步骤并未引起过多开销。

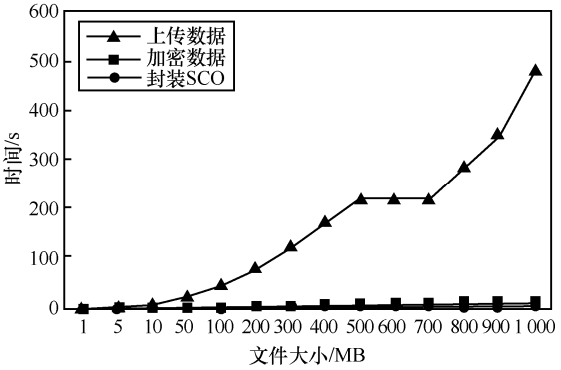

图 12　数据加密上传

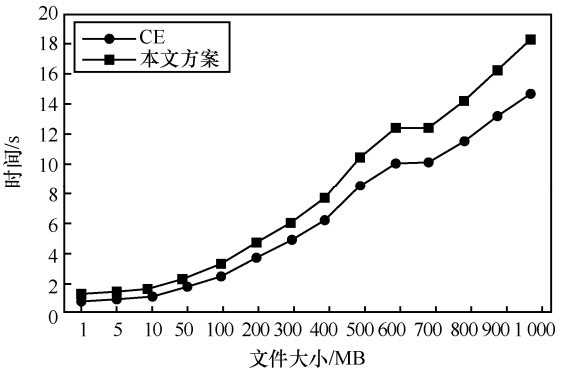

图 13　本文方案与 CE 对比

2）后续数据上传

本文方案为了实现去重数据的访问控制，增加了将数据封装进 SCO 的开销。但由于方案实现了客户端去重，所有重复数据仅需加密上传一次。后续用户上传重复数据时，若该用户符合访问权限，则仅需与云存储服务器进行数据所有权证明，不需要再次加密上传数据。本文测试了当数据已在服务器端存储时，后续用户上传数据的时间开销，测试中

所有权证明协议使用了 POEF（provable ownership of encrypted file）方案[20]，实验结果如表 1 所示。

由表 1 可以看出，当后续用户上传重复数据时，系统的时间开销明显减少。因此，由于系统实现了跨用户客户端去重，多个用户上传相同数据时，仅数据的首次上传需要一定的时间开销，后续上传时开销很小，系统具有较高的执行效率。

表 1 后续数据上传

文件大小/MB	文件读取用时/ms	算法用时/ms	总用时/ms
1	0.838	0.102	1.487
5	1.107	0.653	1.760
10	1.093	0.676	1.769
50	1.273	0.765	2.038
100	1.941	0.635	2.576
200	1.891	0.737	2.628
300	1.667	0.603	2.270
400	1.769	0.747	2.516
500	1.769	0.742	2.511
600	1.782	0.653	2.435
700	1.734	0.654	2.388
800	1.660	0.650	2.310
900	1.129	0.718	1.847
1000	1.104	0.619	1.723

6.2 下载解密

系统中不同大小数据的下载解密时间如图 14 所示，系统的下载解密时间随着文件大小的增长逐步增加。

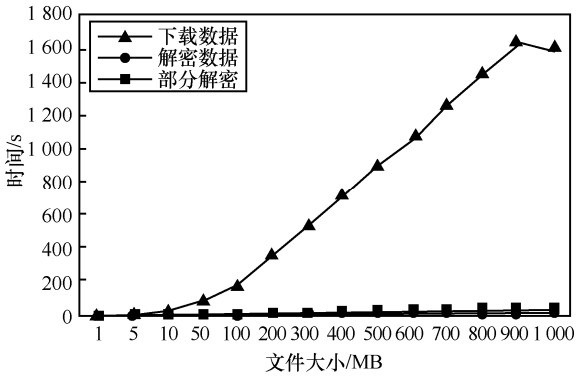

图 14 数据下载解密

系统的主要时间开销依然集中在数据的下载过程，客户端的数据解密时间与服务器端的部分解密时间基本相当。

6.3 权限更新

本节测试了系统中不同大小文件的权限更新性能。图 15 显示了随文件大小的增长，权限更新过程中的运行时间变化情况。更新过程中客户端的主要开销为对新的策略文件进行 ElGamal 加密；云存储服务器端的主要开销为运行 SCO 的密文更新函数，将新加密的策略文件封装进 SCO 中；分布式密钥服务器的主要开销为对重新拆分的 ElGamal 私钥成分并进行 CP-ABE 加密。

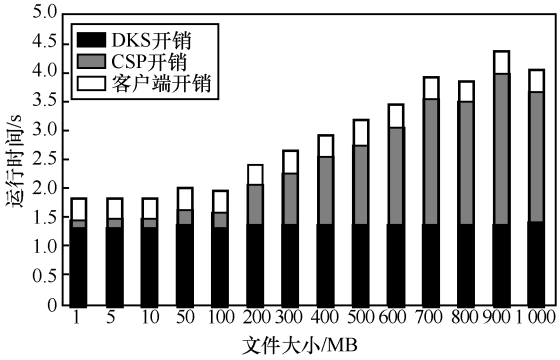

图 15 数据权限更新中运行时间变化情况

由图 15 可以看出客户端与分布式密钥服务器的开销较为固定，不会随着文件大小的增长而增加。而云存储服务器需要更新 SCO 中的内容，开销会随着 SCO 中数据文件大小的增长而增加。

在权限更新过程中，由于客户端不需要进行数据的下载及重新加密等复杂过程，客户端的计算开销是非常低的。并且由于客户端开销不随文件大小而变化，其在整体开销中的占比随着文件大小的增长而不断降低。当更新 1 MB 文件权限时，客户端开销占比总体开销为 20.40%；而当文件大小为 1 000 MB 时，客户端开销占比仅为 9.18%。

无论数据所有者需要更新权限的文件大小为多少，客户端的计算开销都是较低的。因此，本文方案实现了低客户端开销的外包数据权限更新。

7 结束语

在现有的云数据密文去重系统中，用户普遍缺乏数据的管理权，无法确定数据与哪些实体共享。恶意用户可能会利用去重中共享密文的特点威胁用户的数据隐私。本文提出密文去重场景下的数据访问控制方案，数据所有者可对外包数据进行访问控制，被授权的用户才能够与云存储服务器进行去重检测，数据对于未授权用户而言是语义安全的；当用户的身份证书和权限私钥均满足访问策略时才可下载解密数据；数据所有者可通过与密钥服务

器和云存储服务器的交互动态更新数据的访问控制策略。系统中通过 SCO 技术封装用户数据和访问策略，验证用户身份证书并确保访问策略有效执行；通过 CP-ABE 和动态拆分 ElGamal 私钥实现权限更新。分析表明，本文方案在实现去重数据访问控制的同时仅增加了有限开销，具有较高的执行效率和良好的应用前景。

参考文献：

[1] 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.

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

[3] BELLARE M, KEELVEEDHI S, RISTENPART T. DupLESS: server-aided encryption for deduplicated storage[C]//22nd USENIX Security Symposium. Berkeley: USENIX Association, 2013: 179-194.

[4] BELLARE M, KEELVEEDHI S, RISTENPART T. Message-locked encryption and secure deduplication [M]. Berlin: Springer, 2013: 296-312.

[5] LI J, QIN C, LEE P, et al. Information leakage in encrypted deduplication via frequency analysis[C]//47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Piscataway: IEEE Press, 2017: 2110-2118.

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

[7] 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 (CCS 2011). New York: ACM Press, 2011: 491-500.

[8] LI M, QIN C, LEE P. CDStore: toward reliable, secure, and cost-efficient cloud storage via convergent dispersal [C]//USENIX Annual Technical Conference. Berkeley: USENIX Association, 2015: 111-124.

[9] SHIN Y, KIM K. Differentially private client-side data deduplication protocol for cloud storage services[J]. Security and Communication Networks, 2015, 8(12): 2114-2123.

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

[11] 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.

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

[13] SQUICCIARINI A, PETRACCA G, BERTINO E. Adaptive data protection in distributed systems[C]//Third ACM Conference on Data and Application Security and Privacy. New York: ACM Press, 2013: 365-376.

[14] THILAKANATHAN D, CALVO R, CHEN S, et al. Secure and controlled sharing of data in distributed computing[C]//Proceedings of the 16th IEEE International Conference on Computational Science and Engineering. Piscataway: IEEE Press, 2013: 825-832.

[15] ZAFAR F, KHAN A, MALIK S U R, et al. A survey of cloud computing data integrity schemes: design challenges, taxonomy and future trends[J]. Computers & Security, 2017(65): 29-49.

[16] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[C]//2007 IEEE Symposium on Security and Privacy (S&P 2007). Piscataway: IEEE Press, 2007: 321-334.

[17] GAMAL T E. A public key cryptosystem and a signature scheme based on discrete logarithms[J]. IEEE Transactions on Information Theory, 1985, 31(4): 469-472.

[18] 林婷婷, 来学嘉. 白盒密码研究[J]. 密码学报, 2015, 2(3): 258-267.
LIN T T, LAI X J. Research on white-box cryptography[J]. Journal of Cryptologic Research, 2015, 2(3): 258-267.

[19] 王鹃，樊成阳，程越强，等. SGX 技术的分析和研究[J]. 软件学报，2018, 29(9): 2778-2798.
WANG J, FAN C Y, CHENG Y Q, et al. Analysis and research on SGX technology[J]. Journal of Software, 2018, 29(9): 2778-2798.

[20] 杨超, 张俊伟, 董学文, 等. 云存储加密数据去重删除所有权证明方法[J]. 计算机研究与发展, 2015, 52(1): 248-258.
YANG C, ZHANG J W, DONG X W, et al. Proving method of ownership of encrypted files in cloud de-duplication deletion[J]. Journal of Computer Research and Development, 2015, 52(1): 248-258.

[作者简介]

贾春福（1967− ），男，河北文安人，博士，南开大学教授、博士生导师，主要研究方向为计算机网络与信息安全、可信计算、恶意代码分析等。

哈冠雄（1995− ），男，回族，天津人，南开大学硕士生，主要研究方向为云数据安全、密码学应用等。

李瑞琪（1993− ），男，黑龙江尚志人，南开大学博士生，主要研究方向为同态加密、格密码学等。