面向云存储的高效动态密文访问控制方法

洪澄，张敏，冯登国
(中国科学院 软件研究所 信息安全国家重点实验室, 北京 100190)

摘 要：针对云存储中敏感数据的机密性保护问题，在基于属性的加密基础上提出了一种密文访问控制方法 HCRE。其思想是设计一种基于秘密共享方案的算法，将访问控制策略变更导致的重加密过程转移到云端执行，从而降低权限管理的复杂度，实现高效的动态密文访问控制。实验分析表明 HCRE 显著降低了权限管理的时间代价，而且没有向云端泄露额外的信息，保持了数据机密性。

关键词：云存储；云计算；密文访问控制；基于属性的加密；代理重加密
中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2011)07-0125-08

Achieving efficient dynamic cryptographic access control in cloud storage

HONG Cheng, ZHANG Min, FENG Deng-guo
(State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: To keep the data in the cloud confidential against unauthorized parties, a cryptographic access control solution called hybrid cloud re-encryption (HCRE) based on attribute-based encryption (ABE) was introduced. HCRE designed a secret sharing scheme to delegate the task of ABE re-encryption to the cloud service provider (CSP), which alleviates the administering burdens on the data owner. Experiments show that HCRE can handle dynamic access policies in a more efficient way. Additionally, HCRE does not reveal extra information of the plaintext to the CSP, thus it does no harm to the data confidentiality.

Key words: cloud storage; cloud computing; cryptographic access control; attributes-based encryption; proxy re-encryption

1 引言

借助于网络和存储技术的飞速发展，云计算已成为一种新兴的服务模式。云计算通过将计算和存储职责从本地转移到云中，为用户节省了大量成本，因此得到了广泛的支持和应用。但是这一模式也带来了新的安全隐患，其中一个重要问题是：云的存储介质位于用户控制之外，如何向用户保证数据的合法访问。

由于用户不一定完全信任云服务提供商（CSP），因此无法依赖 CSP 的服务器端访问控制，必须通过加密数据并控制用户的解密能力以实现密文访问控制。如何实现高效的密文访问控制是安全云存储的首要问题。

KP-ABE[1]、CP-ABE[2]等算法（以下简称 ABE）将解密规则蕴含在加密算法之中，可以免去密文访问控制中频繁出现的密钥分发代价。由于这一优良特性，当前已经有很多对使用 ABE 实现密文访问控制的研究[3~6]。这些方法在访问控制策略发生动态变更时，都要求数据所有者（DO）对数据进行

收稿日期：2011-02-28；修回日期：2011-06-10
基金项目：国家高技术研究发展计划(“863”计划)基金资助项目(2007AA120404)
Foundation Item: The National High Technology Research and Development Program of China (863 Program) (2007AA120404)

重加密。由于 ABE 算法的效率仍然不够高，一次 ABE 重加密的代价可能很大，这对于 DO 来说是难于接受的。因此使用 ABE 实现密文访问控制面临着一个严峻的问题：如何有效地支持动态策略。

一种解决方法是“懒惰重加密”，该方法认为在撤销权限这类策略变更中，由于被撤销者曾经对旧数据拥有访问能力，因此没有必要立即执行重加密，而是在数据内容被更新时才执行重加密。懒惰重加密的有效性取决于 2 点：1) 数据更新不频繁；2) DO 能够容忍懒惰重加密。事实上，很多数据的更新很频繁，而且对于一些高安全级的数据，DO 有立即重加密的需求，因此懒惰重加密并没有真正解决动态访问控制策略中存在的问题。

另一种解决方法是假设云服务提供商（CSP）是存在一定可信度的，DO 可以将一部分重加密工作交付 CSP 执行。S.Yu 等人[3]提出了一个结合 KP-ABE 和代理重加密的密文访问控制方法，但是该方法没有提及 ABE 随机参数的更换，因此其安全性难于保证（详细说明见 4.3 节）。CP-ABPRE[7] 提出了一种支持代理重加密的 CP-ABE 算法，但是使用 CP-ABPRE 生成代理重加密密钥的代价等同于一次 CP-ABE 加密，这与本文的目的相悖；而且 CP-ABPRE 每一次代理重加密都会增大密文体积，不适合进行频繁的重加密，因此该算法并不适用于云存储中的密文访问控制。目前还没有能够有效支持动态策略的 ABE 重加密方法。

根据实际场景中访问控制的特点，基于 CP-ABE 设计了一种有效的云端重加密方法 Hybrid Cloud Re-Encryption (HCRE)，该方法在不损失安全性的前提下将一部分重加密代价转移到云端，大大减少了 DO 端的权限管理代价，因此实现了高效地动态密文访问控制。

下文将首先介绍所需的预备知识和安全假定，然后描述 HCRE 的详细算法和实现。最后证明了 HCRE 的安全性，并通过实验测试了 HCRE 的有效性。

2 预备知识

本节描述本文需要的预备知识。其中 CP-ABE 算法是本文使用算法的基础，而代理重加密虽然没有被直接应用，但是其思想与 HCRE 甚为接近，所以也一并描述。

2.1 CP-ABE

基于属性的加密(ABE)最初由 Sahai 和 Waters[8] 提出。其基本思想是，密文与私钥分别与一组属性关联，当用户的私钥属性与密文属性相互匹配度达到一个门限值时，该用户才能解密密文。随后 ABE 算法发展为密钥策略(KP-ABE)[1]以及密文策略(CP-ABE)[2]2 类。其中 KP-ABE 的密钥与访问控制策略关联，而 CP-ABE 的密文与访问控制策略关联，因此后者更贴近实用环境。

访问控制策略可体现为一个由属性值和门限关系组成的访问控制结构，一个 CP-ABE 访问控制结构示意图如图 1 所示。CP-ABE 算法包括以下 4 个组成部分。

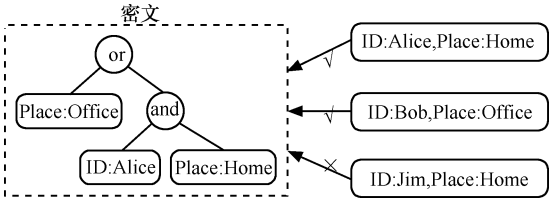

图 1 CP-ABE 访问控制结构示意图

1) *Setup*。生成主密钥 *MK* 和公开参数 *PK*。

2) CT_T=*Encrypt*(*PK*,*M*, *T*)。使用 *PK*、访问控制结构 *T* 将数据明文 *M* 加密为密文 CT_T。

3) SK_s=*KeyGen*(*MK*, *S*)。使用 *MK*、用户属性值 *S* 生成用户的私钥 SK_s。

4) *M*=*Decrypt*(CT_T, SK_s)。使用私钥 SK_s 解密密文 CT_T 得到明文 *M*。只有 *S* 满足 *T* 的条件下，*Decrypt*() 操作才能成功。

CP-ABE 具有上述优良的特性，因此可以被用于实现密文访问控制。本文的目标是在使用 CP-ABE 的同时，以尽量低的客户端代价更改服务器端密文的访问控制策略。

2.2 代理重加密

在 1998 年的欧洲密码学年会上，Blaze 等[9] 提出了代理重加密(PRE, proxy re-encryption)的概念。PRE 允许一个半可信的代理者(proxy)将 Alice 可解密的密文转换为 Bob 可解密的同一明文的密文，并且保证该代理者无法获知该明文的任何信息。

PRE 是一种非常有用的密码学工具，在加密邮件的转发等场景中得到了较为广泛的应用。HCRE 将利用与 PRE 类似的思想在半可信服务器端实现重加密，把一组属性集可解密的密文转换为另一组属性集可解密的同一明文的密文，如图 2 所示。

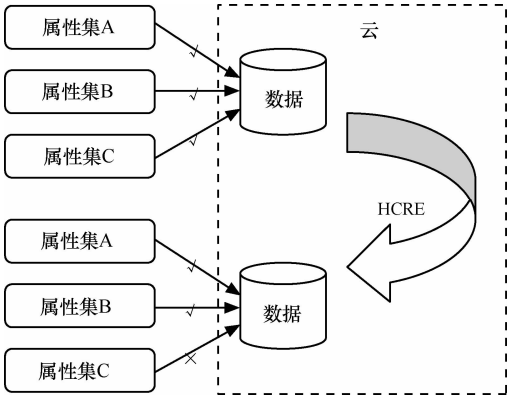

图 2　HCRE 示意图

3　安全假定

完全不可信的服务器是无法委托其重加密的，因此有必要对 CSP 的可信度进行限定。本文中要求 CSP 必须是“Honest but curious”的，这是 HCRE 实现的基本前提。

1) Honest: CSP 必须能够忠实的执行 DO 预设的重加密程序。

2) Curious: CSP 可能会窥探数据内容，因此重加密过程不应向 CSP 披露任何数据明文信息。

“Honest but curious”假定适用于大部分场景。对于那些安全性弱于该假定的场景（例：CSP 出于牟利目的而与非法用户合谋），DO 必须亲自进行权限管理和数据重加密，之前的工作[4]讨论了这种场景下的密文访问控制方法，因此不在本文中赘述。

另外，服务器端重加密不可避免的导致访问控制策略被泄露给 CSP，因此必须能够容忍这一点。

4　HCRE：实现

本节描述了 HCRE 的完整实现。首先描述 HCRE 的实现思路，然后设计 HCRE 需要的算法，最后描述具体操作。

4.1　思想

HCRE 的底层构造方法与大多数相关工作相同：由于 ABE 算法的效率不适合直接用于加密大型数据文件，因此首先使用对称加密算法（例如 AES, 3DES 等）加密数据文件得到数据密文，然后使用 ABE 加密该对称密钥得到密钥密文，用户通过依次解密密钥密文和数据密文访问数据文件。

数据密文和密钥密文在访问控制策略发生改变时可能需要被更新。对于对称加密，当前尚不存在安全有效的代理重加密算法，因此数据密文必须由 DO 亲自更新。HCRE 的优势在于允许将密钥密文的更新移交给 CSP: DO 在更新数据密文之后生成一份重加密信息发送给 CSP，然后下线，而由 CSP 完成余下的工作，从而降低了 DO 的重加密代价。

4.2　算法设计

HCRE 的有效性取决于 2 点：1) DO 生成重加密信息的代价远低于更新密钥密文的代价；2) 授予 CSP 重加密信息不会增加向 CSP 泄漏数据明文的风险。

如前所述，现有的方法均难于满足这 2 个要求。通过对 CP-ABE[2]中的访问控制结构作出变换，设计了一种满足要求的算法，称为 SCP-ABE。SCP-ABE 是实现 HCRE 的基础。在描述 SCP-ABE 之前，需要作出下列定义。

定义 1　简单访问控制结构(SAS, simple access structure)。若一个访问控制结构满足以下 3 个特征：

1) 只包含 and 和 or 关系；

2) 是析取范式；

3) 每个 and 节点的度都为 2。

则称该结构为简单访问控制结构。

易见任意的单调访问控制结构都能够通过 2 步转化为 SAS：1) 转化为析取范式； 2) 对于每个度大于 2 的 and 节点，通过自左向右优先嵌套将度降低到 2（如图 3 所示）。注意 SCP-ABE 不支持非单调访问控制结构 [10](如 Manager and Not (Shanghai))。

定义 2　简单访问控制结构秘密共享(SSS, SAS secret sharing)。设置 SAS 的根节点的值为待共享的值 s。由根节点开始，对每个值为 s_i 的非叶节点 i，自顶向下赋值如下。

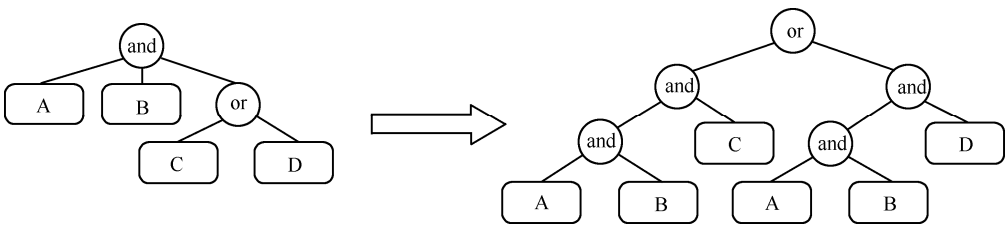

图 3　非 SAS 转换为 SAS

若 i 对应的关系是 or: 设置 i 的每个子节点的值为 s_i。

若 i 对应的关系是 and: 取随机数 s'，设置 i 的左子节点的值为 s'，而设置 i 的右子节点的值为 s_i+s'。

例如使用结构(A and B) or (C and D) 共享值 s 的过程如下：1) 设置 or 节点对应的值为 s; 2) 设置 2 个 and 节点对应的值为 s；3) 为 A 赋值 S_A，为 B 赋值 S_B，为 C 赋值 S_C，为 D 赋值 S_D，其中，$s=S_B-S_A=S_D-S_C$。可见掌握 A、B 或 C、D 2 对节点中的任一对都能够还原出 s。这种秘密共享方法与 Benaloh and Leichter[11]类似，不同的是对 and 节点使用减法而不是加法进行秘密共享。这样做是为了更简单的实现 4.3 节中描述的权限撤消。

定义 3 简单密文策略的基于属性的加密(SCP-ABE, simple ciphertext-policy attribute-based encryption) 算法：SCP-ABE 包含以下 4 个子函数。

1) *Setup*(): 构造一个双线性群 G_0，记 G_0 的生成元为 g，对应的双线性映射为 $e: G_0\times G_0\to G_T$。定义系统所需的属性空间为 $A=\{a_1,a_2,\cdots,a_k\}$，对每个属性 $a_i\in A$ $(1\leqslant i\leqslant k)$，随机选择 $x_i\in G_0$ $(1\leqslant i\leqslant k)$。最后随机选择 α、$\beta\in Z_P^*$，公开发布公钥如下：

$$PK=\{G_0,g,g^{\beta},e(g,g)^{\alpha},\{x_i\}_{i=1}^{k}\}$$

生成主密钥如下并秘密保存。

$$MK=\{\alpha,\beta\}$$

2) *KeyGen*(*w*,*MK*): 构造属性集 w 对应的私钥。首先随机选择 $r\in Z_P^*$，然后对每个属性 $a_i\in w$ $(1\leqslant i\leqslant k)$，随机选择 $r_i\in Z_P^*$。最后生成私钥 SK_w 如下：

$$SK_w=\{SK^{(1)}=g^{\frac{\alpha+r}{\beta}},SK^{(2)}=\forall a_i\in w: g^{r_i},$$
$$SK^{(3)}=\forall a_i\in w: g^{r}\cdot x_i^{r_i}\}$$

3) *Encrypt*(*m*, *T*, *PK*): 使用访问控制结构 T 加密明文 m。首先随机选择 $s\in Z_P^*$，然后按照 SSE 方法对 T 的每个叶节点赋值 $\{s_i\}_{a_i\in Y_T}$ (Y_T 表示 T 的叶节点对应属性的集合)。 最后构造密文如下：

$$CT_T=\{CT^{(1)}=g^{\beta s},CT^{(2)}=m\cdot e(g,g)^{\alpha s},$$
$$CT^{(3)}=\forall a_i\in Y_T: g^{s_i},$$
$$CT^{(4)}=\forall a_i\in Y_T: x_i^{s_i}\}$$

4) *Decrypt*(CT_T, SK_w): 使用私钥 SK_w 解密密文 CT_T。对 T 的叶节点 n,设其对应的属性为 a_i，首先定义函数 *DecNode*(n)如下:

$$DecNode(n)=\begin{cases}\dfrac{e\left(SK_i^{(3)},CT_i^{(3)}\right)}{e\left(SK_i^{(2)},CT_i^{(4)}\right)}=e\left(g,\ g\right)^{rs_i},\ a_i\in w\\ \perp,\ a_i\notin w\end{cases}$$

对 T 的内部节点 N，定义函数如下：

$$DecNode(N)=\begin{cases}DecNode(RChild(N))\\ \quad DecNode(LChild(N)),\ Op(N)=\text{'and'}\\ DecNode(LChild(N)),\ Op(N)=\text{'or'}\end{cases}$$

因为 s_i 是使用 SSE 方法赋值，所以由上述定义可得

$$w\ \ satisfies\ \ T\to DecNode(Root(T))=e(g,g)^{rs}$$

则明文可计算如下：

$$Decrypt(CT_T,\ SK_w)=\begin{cases}\dfrac{CT^{(2)}\cdot DecNode(Root\left(T\right))}{e\left(CT^{(1)},SK^{(1)}\right)}=m,\ w\ satisfies\ T\\ \perp,\ \text{otherwise}\end{cases}$$

SCP-ABE 对访问控制结构进行了规范化，以便于实现重加密。但是 SCP-ABE 并不像 CP-ABPRE[2] 一样是一个全功能的 PRE 算法。因此并没有像描述一个 PRE 算法一样将重加密函数包括在 SCP-ABE 中，而是在 4.3 节具体操作中描述如何实现重加密。

4.3 HCRE：体系结构

本节具体介绍了整个 HCRE 方案，包括文件的创建、访问等基本功能、以及授权和撤消等访问控制功能的实现过程。

4.3.1 系统初始化

DO 运行 *Setup* 以产生公钥 *PK* 和主密钥 *MK*。DO 对 *PK* 签名，并将 *PK* 和签名一起发送给 CSP，而 *MK* 自行秘密保存。

其次需要定义用户属性空间 $\{a_i\}_{i=1}^{k}$，特别的，对 DO 本人，生成一个独立的属性 a_o。然后 DO 为每个用户设置属性集，并分发私钥。例如若一个用户 u 的属性集合为 A_u={*Dept:Sales*，*Position: Manager*}，DO 为其生成 SK_u=*KeyGen*(A_u, *MK*), 并将 SK_u 通过安全信道发送给 u。CP-ABE 的优势在于密钥分发是一次性的，此后访问控制策略的变动将

不需要重新分发密钥。

4.3.2 文件创建

DO 首先随机生成一个对称密钥 k_f，使用 k_f 加密文件 f 并签名得到数据密文 C_f=(*Enc*(*f*, k_f),*Sig*(*Enc*(*f*, k_f))) (此处假设 *Enc* 是任一种合乎要求的对称加密算法)，然后使用 SCP-ABE 以 DO 本人的属性 a_o 加密 k_f 得到密钥密文 C_k=*Encrypt*(k_f , a_o,*PK*)，最后将 C_f 和 C_k 一起发送给 CSP。

在实际场景中，经常将访问权限相同的若干个文件划分为一个文件组，并通过 CP-ABE 管理文件组密钥以便高效的处理多个文件的访问控制，此处为简洁起见暂只描述对单个文件的处理。

4.3.3 权限授予

根据被授权者的不同，一次授权操作可能将权限授予一个或多个用户（例，DO 可以为 (*Dept: Sales*)∧(*Position:Manager*) 或者 *Position:Manager* 授权。设被授权者 u 的属性结构为 T_u，f 当前的访问控制结构为 T, 则 DO 授权 u 访问 f 的一种直观实现方法如下：1) 从 CSP 获得 C_k=*Encrypt*(k_f, T, *PK*)并解密 C_k 获得 k_f ；2) 使用 T'=$T\vee T_u$ 加密 k_f 得到新的密钥密文 C'_k=*Encrypt*(k_f ,T',*PK*)并发送给 CSP。

上述步骤包括一次解密和一次加密，因此时间代价不够理想。得益于 SCP-ABE 的构造特点，可以对其进行改进。观察 C_k 和 C'_k 的构造：

$$C_k=\{g^{\beta s},k_f\cdot e(g,g)^{\alpha s},\forall a_i\in Y_T: g^{s_i},\ \forall a_i\in Y_T: x_i^{s_i}\}$$

$$C'_k=\{g^{\beta s'},k_f\cdot e(g,g)^{\alpha s'},\forall a_i\in Y_{T'}: g^{s'_i},\ \forall a_i\in Y_{T'}: x_i^{s'_i}\}$$

因为 T'=$T\vee T_u$，根据 SSE 规则，若 s=s'，对 $\forall a_i\in Y_T$ ，有 s_i =s'_i, 因此不需要重新生成整个 C'_k，只需要生成 $\{\forall a_i\in Y_{T_u}: g^{s'_i},\forall a_i\in Y_{T_u}: x_i^{s'_i}\}$ 即可。

假定 DO 能够保存文件 f 对应的 s，则 DO 授权 u 访问 f 的过程可被优化如下：1) 将 T_u 转化为 SAS；2) 查找 f 对应的 s，按照 SSE 规则对 T_u 的每个节点赋值 s'_i；3) 生成 $C_u=\{\forall a_i\in Y_{T_u}: g^{s'_i},\ \forall a_i\in Y_{T_u}: x_i^{s'_i}\}$ 并发送给 CSP；4) CSP 生成 $C'_k=C_k\cup C_u$。一个授权示例如图 4 所示。

4.3.4 文件访问

为简便起见，假设只有 DO 能创建/更新文件，其他用户具有只读权限。用户 u 访问文件 f 的过程如下：u 向 CSP 发起对 f 的访问请求，CSP 向其返回 f 对应的 C_f 和 C_k。若 A_u 满足 f 的访问控制结构，则 u 能解密 C_k 得到 k_f =*Decrypt*(C_k , SK_u)，然后使用 k_f 解密 C_f 获得数据明文。

4.3.5 权限撤销

设 f 当前的访问控制结构为 T,撤消 u 访问 f 的权限之后，f 的访问控制结构为 T'。DO 授权 u 访问 f 的一种直观过程如下：1) 按照文件访问的步骤获取明文 f; 2) 随机生成一个新的对称密钥 k'_f ；3) 生成新的数据密文 C'_f=(*Enc*(*f*, k'_f),*Sig*(*Enc*(*f*, k'_f)))；4) 生成新的密钥密文 C'_k=*Encrypt*(k'_f , T', *PK*)；5) 将 C'_f 和 C'_k 发送给 CSP。

直接执行上述步骤的时间代价不够理想。可以利用 SCP-ABE 的特性以将第 4 步的重加密代价转移给 CSP，实现方法如下。

1) DO 生成新的随机参数 s'，(有效的撤销操作必须使用新的随机参数。S.Yu[3]一文没有讨论撤销过程中的随机参数的更换，因此存在安全隐患)，查找 f 对应的 s，将 T'、$k'_f\cdot k^{-1}_f$ 和 $s'-s$ 发送给 CSP。

2) 由于 T 是 SAS，因此 T 必定可以表示为 $T'\vee Tu$ 的形式。因此 CSP 可以基于现有的密钥密文：

$$C_k=\{C^{(1)}=g^{\beta s},C^{(2)}=k_f\cdot e(g,g)^{\alpha s},\ C^{(3)}=\forall a_i\in Y_T: g^{s_i},C^{(4)}=\forall a_i\in Y_T: x_i^{s_i}\},$$

对所有 $\forall a_i\in Y_T$ 自顶向下运算如下：

若 a_i 所在节点是 root：

$$C_i^{(3)'}=C_i^{(3)}\cdot g^{s'-s},C_i^{(4)'}=C_i^{(4)}\cdot x_i^{s'-s}$$

若 a_i 所在节点是 or：对 a_i 的所有子节点 a_j：

图 4　授权过程中的访问控制结构变化示意图

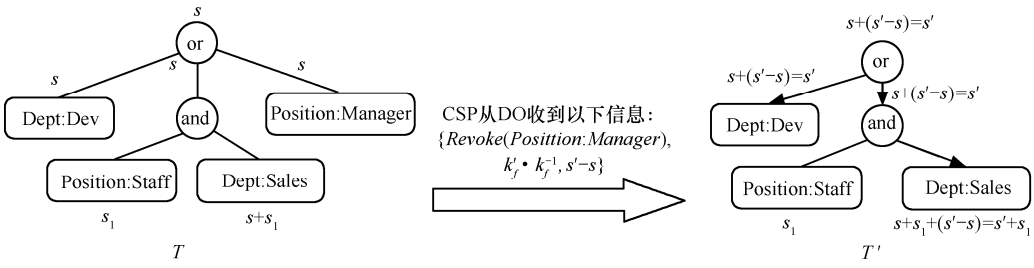

图 5 撤销过程的访问控制结构变化示意图

$$C_j^{(3)'} = C_j^{(3)} \cdot g^{s'-s}, C_j^{(4)'} = C_j^{(4)} \cdot x_i^{s'-s}$$

若 a_i 所在节点是 and：对 a_i 的右子节点 a_j：

$$C_j^{(3)'} = C_j^{(3)} \cdot g^{s'-s}, C_j^{(4)'} = C_j^{(4)} \cdot x_i^{s'-s}$$

若 $a_i \notin Y_{T'}$：删除 $C_i^{(3)}, C_i^{(4)}$

上述步骤中的访问控制结构结构变化示意图如图 5 所示。完成整个结构的更新后 CSP 生成新的密钥密文如下：

$$\begin{aligned} C'_k = \{&C^{(1)'} = C^{(1)} \cdot g^{\beta(s'-s)}, \\ &C^{(2)'} = C^{(2)} \cdot k'_f \cdot k^{-1}{}_f \cdot e(g,g)^{\alpha(s'-s)}, \\ &C^{(3)'} = \forall a_i \in Y_{T'} : C_i^{(3)'}, \\ &C^{(4)'} = \forall a_i \in Y_{T'} : C_i^{(4)'}\} \end{aligned}$$

易验证 C'_k=*Encrypt*(k'_f, T', *PK*)，这表明通过该方法可以成功实现 CSP 端重加密。

5 安全性分析

本节对 HCRE 的安全性进行分析。具体内容可以分成 2 部分：对 SCP-ABE 算法的安全性分析和对重加密操作过程的安全性分析。

5.1 算法安全性

SCP-ABE 与 CP-ABE 的主要区别在于加密阶段：CP-ABE 使用图 6 所示的多项式构造方法，而 SCP-ABE 使用的则是 SSE 方法。

假设图 6 中的 T 是 SAS：则 i 只有 2 种可能：

i 为 or 节点（门限值=1），此时 $q_i(x)$是一个常数，设 $q_i(x)=b_i$，则对 i 的子节点 j，$q_j(0)=b_i$。

i 为 and 节点（门限值=2），此时 $q_i(x)$阶为 1，设 $q_i(x)=a_ix+b_i$。则对 i 的左子节点 j，$q_j(0)=b_i$；对 i 的右子节点 k，$q_k(0)=a_i+b_i$。

可见 $q_i(0)$与 SSE 方法中的 s_i 赋值方法完全等价，因此 CP-ABE 在访问控制结构满足 SAS 条件时等价于 SCP-ABE。由于文献[2]已经证明了 CP-ABE 是安全的，所以 SCP-ABE 也是安全的。

Encrypt(*m*,*T*,*PK*): 使用访问控制结构 T 加密明文 m。对每个 T 中的节点 i，设 k_i 是节点 i 的门限值，选择一个 k_i-1 阶多项式 $q_i(x)$. 随机选择 $s \in Z_P^*$，$q_i(x)$的选择需要满足下述条件：

1) 对根节点 R，$q_R(0)=s$；

2) 对非根节点 i, $q_i(0)=q_{parent(i)}(index(i))$，$index(i)$ 含义为 i 在其兄弟节点中的序号。

最后构造密文如下：

$$\begin{aligned} CT_T = \{&CT^{(1)} = g^{\beta s}, CT^{(2)} = me(g,g)^{\alpha s}, \\ &CT^{(3)} = \forall a_i \in Y_T : g^{q_i(0)}, \\ &CT^{(4)} = \forall a_i \in Y_T : x_i^{q_i(0)}\} \end{aligned}$$

图 6 CP-ABE 的加密阶段

5.2 操作安全性

HCRE 的权限授予操作并没有向 CSP 泄露额外的信息，但是在权限撤销过程中，DO 需要将 $k'_f k_f^{-1}$ 和 $s'-s$ 发送给 CSP。因此必须证明 CSP 掌握 $k'_f k_f^{-1}$ 和 $s'-s$ 不会影响数据机密性。现讨论如下。

1) 首先分析密钥密文的安全性。对于密钥密文 C_k，由于 k'_f 和 s'由 DO 随机生成，与 C_k 毫不相关，因此发送 $k'_f \cdot k^{-1}{}_f$ 和 $s'-s$ 给 CSP 仅仅相当于 CSP 从随机预言机处获得了 2 个随机数。按照文献[2]的证明和前文的等价证明，SCP-ABE 在随机预言机模型下是安全的，因此 CSP 对 C_k 不会获得额外的攻击优势。

2) 其次是数据密文的安全性。对于数据密文 C_f，CSP 拥有的额外信息是新的数据密文 C'_f 和 2 份密文使用的密钥之间的关系 $k'_f k_f^{-1}$。这种情形下的安全问题可归结于一类密码学问题，称为 Related-Key Attack[12]。研究表明[13]，Related-Key Attack 确实会对很多加密算法形成威胁。但是如文献[14]分析，AES 算法对 Related-Key Attack 具有足够的混淆强度，迄今为止密码学界对 AES 提出的 Related-Key Attack 方法也都远远超出了现有的计算水平[15]。

综上所述，只要在对称加密阶段使用 AES 算法，认为 HCRE 是足够安全的。

6　分析与实验

本节将对 HCRE 的有效性进行分析，并设计实验以对结论进行验证。

6.1　性能分析

按照 4.3 节描述，一般的基于 ABE 算法的授权操作需要进行一次完整的密钥密文重加密，因此其时间复杂度与数据访问控制结构的大小（即访问控制结构树的叶节点数目）有关。而 HCRE 的授权只需针对被授权者构造授权信息，因此其时间代价仅与被授权者的属性数目有关。类似的，一般的撤销授权操作需要进行一次密钥密文重加密，而 HCRE 的撤销授权只需由 DO 告知 CSP 被撤销者的属性名称即可。此外，二者的撤销授权都必须进行数据密文重加密，因此其时间代价还与数据文件大小有关。综上所述，HCRE 对权限管理的效率的提升如表 1 所示。

表 1　　DO 实施权限管理的时间复杂度

时间复杂度	No-HCRE	HCRE
权限授予	$O(Y)$	$O(A)$
权限撤销	$O(F)+O(Y)$	$O(F)+O(1)$

*　Y 表示数据文件访问控制结构的大小，A 表示被授权/撤销者的属性数目，F 表示文件的大小。一般情况下，$Y>>A$。

6.2　实验环境

下面将设计实验以对上述分析进行验证。一般场景中，单个用户拥有的属性数目是相对有限、固定的，而单个数据文件访问控制结构的大小可能随其访问权限复杂度而增长。不失一般性，假设用户拥有的平均属性数目为 5，而数据访问控制结构大小在 0~50 波动。对于单个数据文件大小，分别对 1MB、10MB、50MB 3 种典型情形进行测试。

实验设备为 Intel Xeon 2.4 GHz, 2GB 内存，操作系统为 Windows Server 2003, 实验环境是构造于 VMware Workstation 6.5.1 虚拟机上的 Red Hat Enterprise 5，分配有 1GB 内存。实验代码基于 cpabe-0.10 库[16]编写，对称加密阶段使用 192bit AES 密钥。

6.3　实验结果

6.3.1　权限授予性能比较

图 7 展示了使用 HCRE 前后的授权时间代价对比。由于无论是否使用 HCRE，授权操作都只与密钥密文有关，不受文件大小影响，为简洁可读起见，图中只绘出了对 1MB 文件授权的时间代价曲线。

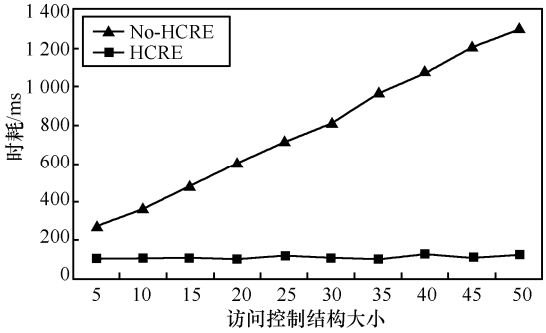

图 7　单次授权耗时(ms)与文件访问控制结构大小的关系

可以看出， 未使用 HCRE 时，授权花费的时间代价与文件访问控制结构大小成线性关系，而 HCRE 授权花费的时间代价则基本与其无关。

6.3.2　权限撤销性能比较

图 8 展示了使用 HCRE 前后的权限撤销时间代价对比。

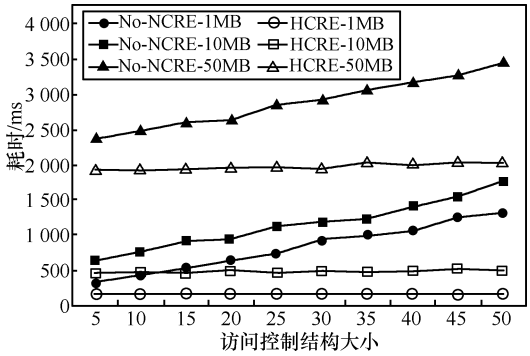

图 8　单次撤销耗时(ms)与文件访问控制结构大小的关系

与授权操作类似，HCRE 撤销花费的时间与文件访问控制结构大小无关。 不同点在于，权限撤销要求数据密文重加密，该操作不可避免的需要 DO 亲自执行，因此撤销时间代价还与文件大小成线性关系。

整体上，实验结果证明了 HCRE 成功达到了预期目的，大大缩短了授权、撤销授权操作的时间代价，访问控制策略越复杂、文件访问控制结构越大时，这一优势尤为明显。

7　结束语

很多相关工作利用 ABE 系列算法以实现云存储中的密文访问控制，而密文访问控制中的动态权限变更要求对数据进行重新加密。针对 ABE 重加密代价过高的问题本文提出了 HCRE 方法，该方法通过对访问控制结构进行变换，将密钥重加密工作转移到云端，从而降低了数据所有者的权限管理代

价。实验数据表明 HCRE 有效的达成了既定目标，且优化效果随着访问控制策略复杂度的增大而尤为明显。安全分析表明 HCRE 没有破坏数据的机密性。在后续工作中将对该方法进行改进，以支持更灵活的访问控制策略。

参考文献：

[1] GOYAL V, PANDEY O, SAHAI A, *et al*. Attribute-based encryption for fine-grained access control of encrypted data[A]. Proceedings of the 13th ACM Conference on Computer and Communications Security[C]. New York, USA, 2006.

[2] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[A]. Proceedings of 2007 IEEE Symposium on Security and Privacy[C]. Berkeley, USA, 2007.

[3] YU S, WANG C, REN K, *et al*. Achieving secure, scalable, and fine-grained data access control in cloud computing[A]. Proceedings of IEEE INFOCOM 2010[C]. San Diego, CA, 2010.

[4] HONG C, ZHANG M, FENG D G. AB-ACCS: a cryptographic access control scheme for cloud storage[J]. Journal of Computer Research And Development, 2010, 47(z1)

[5] MALEK B, MIRI A. Combining attribute-based and access systems[A]. Proceedings of 12th IEEE Int'l Conf on Computational Science and Engineering[C]. Vancouver, Canada, 2009.

[6] ECHEVERRIA V, LIEBROCK L M, SHIN D. Permission management system: permission as a service in cloud computing[A]. Proceedings of the 1st IEEE International Workshop on Emerging Applications for Cloud Computing[C]. Seoul, South Korea, 2010.

[7] LUAN I, MUHAMMAD A, PETKOVIC. An encryption scheme for a secure policy updating[A]. Proceedings of International Conference on Security and Cryptography (SECRYPT 2010)[C]. Athens, Greece, 2010.

[8] SAHAI A, WATERS B, Fuzzy identity-based encryption[A]. Proceedings of EUROCRYPT 2005[C]. Berlin, 2005.

[9] BLAZE M, BLEUMER G, STRAUSS M. Divertible Protocols and Atomic Proxy Cryptography[M]. Advances in Cryptology Springer-Verlag, 1998.127-144.

[10] OSTROVSKY R, SAHAI A, WATERS B. Attribute-based encryption with non-monotonic access structures[A]. Proceedings of the 14th ACM Conference on Computer and Communications Security[C]. New York, USA, 2007.

[11] BENALOH J, LEICHTER J. Generalized Secret Sharing and Monotone Functions[M]. Springer-Verlag, 1990.27-35.

[12] BIHAM E. New Types of Cryptanalytic Attacks Using Related Keys[M]. Springer-Verlag, 1993. 398-409.

[13] KELSEY J, SCHNEIER B, WAGNER D. Related-key Cryptanalysis of 3-way, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA[M]. Springer-Verlag, 1997. 203-207.

[14] DAEMEN J, RIJMEN V. AES proposal: rijndael, technical report, national institute of standards and technology[EB/OL]. http://csrc.nist.gov/encryption/aes/round2/r2algs.htm.

[15] BIRYUKOV A, KHOVRATOVICH D. Related-key Cryptanalysis of the Full AES-192 and AES-256[M]. Springer-Verlag, 2009. 1-18.

[16] BETHENCOURT J, SAHAI A, WATERS B. Advanced crypto software collection : the cpabe toolkit[EB/OL]. http://acsc.cs.utexas. edu/cpabe/

作者简介：

洪澄（1985-），男，江西余干人，中国科学院博士生，主要研究方向为数据库安全理论与技术。

张敏（1975-），女，安徽萧县人，中国科学院副研究员，主要研究方向为数据安全与隐私保护、数据库安全理论与技术。

冯登国（1965-），男，陕西靖边人，中国科学院研究员、博士生导师，主要研究方向为信息安全和密码学。