Journal of Computer Applications
计算机应用,2013,33(10):2832-2834,2877
ISSN 1001-9081
CODEN JYIIDU
2013-10-01
http://www.joca.cn

文章编号:1001-9081(2013)10-2832-03
doi:10.11772/j.issn.1001-9081.2013.10.2832

# 云环境下的基于属性和重加密的密钥管理

罗文俊,徐　敏*
(重庆邮电大学 计算机科学与技术学院,重庆 400065)
(*通信作者电子邮箱 xuminmail@163.com)

**摘　要**:在云计算环境中如何安全地存储数据是云计算面临的挑战之一。加密是解决云计算中数据存储安全问题最主要的方法,而加密的一个保密性问题是密钥管理。提出了云环境下的基于属性和重加密的密钥管理方案。云服务提供商对不同用户进行重加密时,可以一次为一组用户重加密,从而减少了重加密的个数。数据拥有者可以对组用户生成和发送重加密密钥,而数据请求者可以使用属性集对应的一个密钥解密多个数据拥有者的数据,从而能减少两者的密钥发送量,降低密钥管理的难度,提高方案的效率。最后,对方案的安全性和性能进行了分析。

**关键词**:云计算;云安全;基于属性的加密;重加密;密钥管理
**中图分类号**:TP309.7;TP309.2　　**文献标志码**:A

# Attribute-based encryption and re-encryption key management in cloud computing

LUO Wenjun, XU Min*
(*School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*)

**Abstract:** The security problem of the data stored in the cloud is a challenge for cloud computing. Encryption is the main method to solve the problem of data storage security in cloud computing. The confidentiality issue of the encryption is the key management. The attribute-based encryption and re-encryption scheme in cloud computing was proposed. It combined attribute-based encryption and re-encryption technology. The cloud server could re-encrypt for different users, and re-encrypt for a group of users at a time. Thereby the scheme reduced the number of re-encryption keys. The data owner could generate and send the keys of re-encryption for a group of users and the data requester could use the key corresponding to the attribute set to decrypt the data of several data owners, which reduced the amount of keys' transmission. The scheme reduced the difficulty of key management and improved the efficiency of the scheme. In the end, the security and efficiency of the scheme were discussed.

**Key words:** cloud computing; cloud security; attribute-based encryption; re-encryption; key management

## 0　引言

云计算是当前信息技术领域的热门话题之一,是产业界、学术界等各界均十分关注的焦点[1]。云计算是一种新近提出的计算模型[2],是分布式计算、并行计算和网格计算的发展。当前,云计算发展面临许多关键性问题,而安全问题首当其冲,并且随着云计算的不断普及,安全问题的重要性呈现逐步上升趋势,已成为制约其发展的重要因素。云计算已成为一种崭新的服务模式,针对云计算环境中的数据完整性和保密性等问题,都需要符合自身特点的安全模型[3]。为了避免敏感数据被恶意用户轻易拿到,除了对数据的存取和访问做严格的限制以外,还需要对其进行加密。加密提供了资源的保护功能,而密钥管理提供了对受保护资源的访问控制。密钥管理作为云计算环境下的数据安全问题之一,已成为云计算发展无法回避的重要问题。

国内外的云环境下的密钥管理的研究仍处于初级阶段。北京大学在云计算安全方面的标志性工作之一是研究、发展适合于物联网和云计算应用的下一代密码和密钥管理技术——基于标识的组合公钥密码[4]。在 Bennani 等[5]的云外包数据库的密钥管理方案中,根据角色分发密钥,能有效解决角色的撤销即密钥的撤销问题,实现了动态的密钥分发,能够有效地废除与用户相关的密钥而保护与角色相关的密钥。美国的微软公司也对云计算环境的密钥管理进行了探讨,其云环境下的公共可验证的密钥分享密钥管理方案[6]主要将云环境下的密钥存储在云环境中进行管理,以及处理密钥恢复问题。该方案具有非交互性,并且在标准模型下证明是安全的。在 Fakhar 等[7]提出的云环境中的对称密钥的管理方案中,主要对存储在云环境中的加密数据的搜索或操作等过程的密钥进行管理,使用增强的 Shamir 算法分离密钥将对称密钥存储在云环境中。Yang 等[8]提出了基于可信平台的云存储密钥管理方案,这个方案基于可信平台模块,方便管理对称密钥和非对称密钥。这些密钥管理机制包括密钥的存储、密钥的备份和密钥的备份。Sun 等[9]提出了云计算环境下的密钥管理框架,该框架面向各种云计算应用,可以实现统一、标准的密钥管理,同时该框架还具有简单和可扩展性的特点。在 Tysowski 等[10]提出的云计算下的安全可扩展的移动应用中的密钥管理方案中,主要使用了重加密的技术进行数据请求过程中的密钥管理,但是该方案存在重加密密钥和解密密钥量数量大,难于管理,存在安全隐患和效率瓶颈。本文主要针对 Tysowski 的方案进行改进,保留使用重加密技术的优点,结合基于属性的加密技术,从而减少重加密密钥产生的个数并降低其管理难度;同时重加密密钥由数据拥有者产生,增强

**收稿日期**:2013-04-17;**修回日期**:2013-06-24。　**基金项目**:重庆市自然科学基金资助项目(2010BB2402)。
**作者简介**:罗文俊(1966-)男,贵州绥阳人,教授,主要研究方向:信息安全、应用密码学;　徐敏(1985-)女,湖北仙桃人,硕士研究生,主要研究方向:信息安全、应用密码学。

了方案的安全性,提高了方案的效率。

## 1　预备知识

1)双线性映射:设 $q$ 为一个大素数,$G_1$ 是阶为 $q$ 的加法循环群,$G_2$ 是阶为 $q$ 的乘法循环群,对于所有的 $P,Q \in G_1$,双线性映射称为 $\hat{e}:G_1 \times G_1 \rightarrow G_2$。

2)两个群之间的双线性映射 $\hat{e}:G_1 \times G_1 \rightarrow G_2$ 满足以下的性质:

双映射性　对所有的 $\alpha,\beta \in \mathbf{Z}_p^*$,$e(g^\alpha,g^\beta) = e(g,g)^{\alpha\beta}$。

非退化性　$e(g,g) \neq 1$。

可计算性　对任意 $P,Q \in G_1$,存在有效算法计算 $e(P,Q)$。

3)离散对数问题(Discrete Logarithm Problem,DLP)。

给定一大素数 $p$,乘法群 $\mathbf{Z}_P^*$ 是域 $\mathbf{Z}_p$ 中的一个循环群。令 $g$ 是 $\mathbf{Z}_p^*$ 的一个生成元。已知 $x$,容易计算出 $y = g^x \bmod p$。域 $Z_p$ 上的离散对数问题(DLP)是指给定 $y,g \in \mathbf{Z}_p^*$,求 $x$ 使得 $y = g^x \bmod p$ 是困难的事情。

4)大整数分解问题(Factorization problem,FAC)。

已知两个大素数 $p$ 和 $q$,求 $n = pq$ 十分容易,但是若已知 $n$,求 $p$ 和 $q$ 是十分困难的事情,这就是大整数分解困难性问题。

## 2　本文方案

设有一个大素数 $q$,阶为 $q$ 的加法循环群 $G_0$,阶为 $q$ 的乘法循环群 $G_1$,则双线性对为 $e:G_0 \times G_0 \rightarrow G_1$,$g$ 为 $G_0$ 的生成元。定义拉格朗日系数 $\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$,其中 $i \in \mathbf{Z}_q$,$S$ 是 $\mathbf{Z}_q$ 中的元素集。此外,定义一个哈希函数 $H:\{0,1\}^* \rightarrow G_0$。此函数的功能是将属性映射为 $G_0$ 中的一个群元素。

1)系统建立:属性授权机构选择两个随机 $\alpha,\beta \in \mathbf{Z}_q$,并计算 $g^\alpha$,$h = g^\beta$,$g^{\frac{1}{\beta}}$ 和 $e(g,g)^\alpha$。将 $G_0,g,g^\beta,g^{\frac{1}{\beta}},e(g,g)^\alpha$ 作为公共参数公布,$\beta,g^\alpha$ 作为私钥保存起来。

2)数据拥有者将数据存入云环境之前,需要执行加密操作。现有数据 $m_1$,数据拥有者根据授权用户的属性集构造数据 $m_1$ 的访问树结构 $T_1$,其中访问结构 $T_1$ 的根为 $R$。定义一个哈希函数 $H_1:G_1 \rightarrow G_0$。数据拥有者选择随机数 $\omega,b,k,\delta \in \mathbf{Z}_q^*$,并计算 $g^{-\omega}$,$g^k$,$e(g^k,g^\omega)$ 以及 $b$ 所对应的 $H_1(b)$,$H_1(b+\delta)$ 和 $g^{-\omega} \cdot H_1(b)$,然后对数据 $m_1$ 加密得到密文 $C_1 = m_1 \cdot e(g^k,g^\omega)$。加密之后,数据拥有者将密文 $C_1$,$T_1$,$g^{-\omega} \cdot H_1(b)$,$g^k$,$H_1(b+\delta)$ 发送到云服务器中,并将随机数 $\omega,b$ 安全保存。

3)云服务提供商将对密文 $C_1$ 进行重加密操作。首先,云服务提供商为访问树 $T_1$ 上的每个节点 $x$(包括叶子节点)选择一个多项式 $q_x$。从根节点 $R$ 开始按从上往下的方式为每个节点 $x$ 选择一个多项式 $q_x$。设定多项式 $q_x$ 上的阶为 $d_x$,其与这个节点的门限值 $k_x$ 之间的关系为:$d_x = k_x - 1$。从根节点 $R$ 开始选择一个随机数 $s \in \mathbf{Z}_p$,并设定 $q_R(0) = s$。然后,随机选择多项式 $q_R$ 上的其他 $d_R$ 个点的值。对于其他的节点 $x$,设定 $q_x(0) = q_{parent(x)}(index(x))$,并随机地选择其他 $d_x$ 个点完成对 $q_x$ 的定义,其中函数 $parent(x)$ 表示节点 $x$ 在访问结构树 $T$ 上的父母节点。在访问结构树 $T$ 上的每个节点的孩子节点 $x$ 从 1 到 $n$ 编号,则函数 $index(x)$ 表示与节点相关的编号[11]。

$Y$ 表示 $T$ 上的叶子节点集合。最后,对密文 $C_1$ 进行重加密得密文 $\tilde{C}$:

$$\tilde{C} = C_1 \cdot e(g^k,g^{-\omega} \cdot H_1(b)) \cdot e(g,g)^{\alpha s} = m_1 \cdot e(g^k,g^\omega) \cdot e(g^k,g^{-\omega} \cdot H_1(b)) \cdot e(g,g)^{\alpha s} = m_1 \cdot e(g^k,H_1(b)) \cdot e(g,g)^{\alpha s}$$

并计算 $\tilde{H} = H_1(b+\delta) \cdot e(g,g)^{\alpha s}$,则重加密后的密文

$$CT = (T_1,\tilde{C},C = h^s,\tilde{H},\forall y \in Y:C_y = g^{q_y(0)},C'_y = H(att(y))^{q_y(0)})$$

其中:如果 $x$ 是一个叶子节点,$att(x)$ 表示结构访问树 $T$ 上的与 $x$ 相关的属性。

当用户向云服务商请求信息 $m_1$ 时,云服务提供商将 $m_1$ 对应的密文 $CT$ 发送给用户。在此过程中,云服务提供商也可以不需要对用户的身份进行认证。

4)属性授权机构监管属性并为用户分发属性密钥。首先,随机选择一个数 $r \in \mathbf{Z}_p$,然后为用户属性集 $S$ 中的每个属性 $j$ 选择一个随机数 $r_j \in \mathbf{Z}_q$。计算属性密钥为 $SK = (D = g^{\frac{(\alpha+r)}{\beta}},\forall_j \in S:D_j = g^r \cdot H(j)^{r_j},D'_j = g^{r_j})$,最后,将不同的用户的属性密钥通过安全通道发送给对应的用户。

5)用户对从云服务提供商得到的密文 $CT$ 进行解密。定义一个递归算法 $DecryptNode(CT,SK,x)$。其中:$SK$ 是与属性集 $S$ 相关的属性密钥,$x$ 是访问结构树上的一个节点。

如果 $x$ 是一个叶子节点,令 $i = att(x)$。如果 $i \in S$,可以得到

$$DecryptNode(CT,SK,x) = \frac{e(D_i,C_x)}{e(D'_i,C'_x)} = \frac{e(g^r \cdot H(i)^{r_i},g^{q_x(0)})}{e(g^{r_i},H(i)^{q_x(0)})} = e(g,g)^{rq_x(0)}$$

如果 $i \notin S$,定义 $DecryptNode(CT,SK,x) = \perp$。

当 $x$ 是非叶子节点时,对于 $x$ 所有孩子节点集 $Z$,令 $F_Z = DecryptNode(CT,SK,x)$。如果孩子节点集 $Z$ 中存在大小为 $k_x$ 的子集 $S_k$,那么 $F_Z \neq \perp$;否则,$F_Z = \perp$。

接下来,令 $i = index(z)$,$S'_x = \{index(z):z \in S_x\}$。

$$F_x = \prod_{Z \in S_x} F_z^{\Delta_{i,S'_x(0)}} = \prod_{Z \in S_x} (e(g,g)^{r \cdot q_z(0)})^{\Delta_{i,S'_x(0)}} = \prod_{Z \in S_x} (e(g,g)^{r \cdot q_{parent(z)(index(z))}})^{\Delta_{i,S'_x(0)}} = \prod_{Z \in S_x} (e(g,g)^{r \cdot q_x(i)})^{\Delta_{i,S'_x(0)}} = e(g,g)^{rq_x(0)}$$

$DecryptNode(CT,SK,x)$ 函数定义完成后,解密算法只要调用该函数在根节点 $R$ 上的值。如果属性集 $S$ 满足访问结构树 $T$,则 $A = DecryptNode(CT,SK,r) = e(g,g)^{rs}$,对密文 $CT$ 进行解密得:

$$\frac{\tilde{C} \cdot A}{e(C,D)} = \frac{\tilde{C} \cdot e(g,g)^{rs}}{e\left(g^{\beta s},g^{\frac{(\alpha+r)}{\beta}}\right)} = m_1 \cdot e(g^k,H_1(b))$$

同理,用户解密 $\tilde{H}$ 得 $H_1(b+\delta)$,数据请求者向数据拥有者发送 $H_1(b+\delta)$ 请求第二层解密密钥,数据拥有者返回 $H_1(b)^k$ 值。最后,计算得到数据明文 $m_1$:

$$m_1 = \frac{m_1 \cdot e(g^k,H_1(b))}{e(g,H_1(b)^k)}$$

## 3　数据拥有者对密钥的管理

不同的数据者拥有不同的访问权限。首先,数据拥有者为数据设置访问权限。例如 $m_1$、$m_2$、$m_2$ 产生对应的访问控制结构为 $T_1$、$T_2$、$T_3$,依此类推,最后形成一个访问控制结构列表,通过安全通道上传给云服务提供商。此时形成的访问控制列表相当于第一层重加密密钥。

为了实现数据加密的主密钥的安全性,数据拥有者可以产生常数个相应密文的第二层重加密密钥,只有授权的用户

才能得到第二层解密密钥。例如:数据拥用者产生四个第二层重加密密钥和对应的解密密钥过程如下:选择 $b,c,d,t \in \mathbf{Z}_q^*$,以及哈希值Ⅰ:$H_1(b),H_1(c),H_1(d),H_1(t)$ 和哈希值Ⅱ:$H_1(b+\delta),H_1(c+\delta),H_1(d+\delta),H_1(t+\delta)$,则第二层重加密密钥 $g^{-\omega}\cdot H_1(b),g^{-\omega}\cdot H_1(c),g^{-\omega}\cdot H_1(d),g^{-\omega}\cdot H_1(t)$ 以及解密密钥为 $H_1(b)^k H_1(c)^k,H_1(d)^k,H_1(t)^k$。数据拥有者将第二层重加密密钥和哈希值Ⅱ形成列表发送给云服务提供商。云服务提供商在执行重加密时,可以在上述列表中随机选一个值进行重加密,也可以为加密数据形成多种重加密密文。

在用户解密过程中,请求第二层解密密钥时,数据请求者将得到的哈希值如 $H_1(c+\delta)$ 发送给数据拥有者,数据拥有者将相应的第二层解密密钥 $H_1(c)^k$ 通过安全通道发送给解密数据请求者。因为第二层解密密钥不是根据不同用户分配不同的解密密钥,而是根据第一层中的解密密钥得到哈希值分发解密密钥,这样可以减少数据拥用者产生解密密钥个数。当某权限级别较高,可以对多个数据进行授权访问时,在其进行多次访问之后,可以形成第二层解密密钥和对应的哈希值列表,这样可以减少第二层解密密钥的请求次数。由于第一层的密文只有授权用户才能解开,所以即使未授权得到了第二层密文的解密密钥,最后也无法得到数据明文。

本方案中用户撤销的过程中,对密钥的更新即间接地通过对数据访问控制结构的更新和修改,将修改后的访问控制结构发送给云服务提供商,则实现了用户的更新。当对第二层解密密钥更新和撤销时,只需要产生新的私钥和公钥替换旧的私钥和公钥,就可以实现第二层解密密钥的更新和撤销。本方案假设半抵抗云服务提供商和用户的合谋,即信任云服务提供商不会修改数据拥有者的访问控制结构。

由于本方案实现了加密数据的主密钥的安全性,则对数据拥有者的所有数据使用一个密钥加密,但是在用户解密的过程中,只能解密授权访问的数据,而对未授权访问的数据,虽然是使用相同的密钥加密的,但是仍不能得到未授权访问的数据的明文。在本方案中,第一层重加密密钥和第二重加密密钥结合可以实现对组用户的重加密,减少了重加密产生的工作量和重加密密钥的个数,从而大量减少了重加密密钥的产生、发送和管理工作以及用户解密密钥的发送数量,提高了云环境中的密钥管理的效率。

## 4 安全性分析

**定理1** 在DLP和FAC假设下,本方案具有主密钥抵抗合谋安全性。

**证明** 对于数据 $m_1$,数据拥有者发送重加密密钥 $g^{-\omega}\cdot H_1(b)$ 和 $H_1(b+\delta)$ 给云服务提供商,同时授权的用户 $B$ 获得数据 $m_1$ 对应的第二层解密密钥 $H_1(b)^k$。现在假设未授权用户(攻击者)$A$ 通过非法途径获得了用户 $B$ 的私钥 $H_1(b)^k$,并与云服务提供商勾结合谋获得加密数据 $C_1$ 的主密钥,从而获得存储在云计算环境的加密数据 $C_1$ 的明文 $m_1$。由于离散对数困难问题(DLP),攻击者 $A$ 和云服务提供商通过 $H_1(b)^k$ 无法获得随机数 $k$,因此无法获得 $g^k$;同理,两者也无法获得随机数 $b$ 所对应的哈希值 $H_1(b)$。由于大整数分解困难问题(FAC),已知 $g^{-\omega}\cdot H_1(b)$,攻击者 $A$ 和云服务提供商得到 $g^{-\omega}$ 和 $H_1(b)$ 是困难的,从而很难得到 $g^{\omega}$,故两者无法得到主密钥 $\omega,k$ 而无法得到 $e(g,g)^{\omega k}$。因此本方案主密钥可以抵抗云服务提供商和未授权的用户的合谋攻击,保证了云环境下加密数据的安全性。

**定理2** 本方案具有前向保密性。

**证明** 假设数据拥有者对数据 $m_1$ 修改后得到 $m'_1$,并加密得到密文 $C'_1$。在数据拥有者修改数据后用户 $B$ 失去了对数据 $m'_1$ 的访问权限,则数据 $m'_1$ 的访问权限变为 $T'_1$,云服务提供商对密文 $C'_1$ 重加密得到 $\tilde{C}'_1 = m'_1\cdot e(g^k,H_1(b))\cdot e(g,g)^{\omega' s}$,其中的 $H_1(b)$ 可以是 $H_1(c),H_1(d),H_1(t)$。而用户 $B$ 拥有所有的第二层解密密钥如 $H_1(b)^k,H_1(c)^k,H_1(d)^k,H_1(t)^k$ 和 $e(g,g)^{rs}$。由于用户 $B$ 没有授权,虽然其可以解密第二层密文,但是无法使用解密密钥 $g^{\frac{(a+r)}{\beta}}$ 解密第一层密文 $m'_1\cdot e(g,g)^{\omega' s}$,故本方案具有前向保密性。

## 5 性能分析

在椭圆曲线密码体制,密钥长度为200位左右就可以满足一般安全要求,则令密钥长度为160 b、224 b、256 b、384 b、512 b。在本方案中,重加密密钥的存储量为常值,而在文献[10]中的两方案(简称第二方案)中,重加密密钥存储量随授权用户的增加而增加。假设数据拥有者 $A$ 有 $n$ 个数据 $m_1,m_2,\cdots,m_n$,取 $n$ 为 $2^{10}$,而每个数据块授权用户数为100、150、200、250、300。令密钥长度为256 b(32 B)时,在本方案中,$A$ 设定10个第二层重加密密钥,则需要的重加密密钥存储量为 $2^{10}\times 256\times 10 = 320$ KB。在第二方案中,当每个数据块授权用户为100时,重加密密钥存储量为 $256\times 2^{10} = 32$ KB 到 $256\times 2^{10}\times 100 = 3\,200$ KB 之间的值;同理,授权用户为150、200、250、300时,重加密密钥存储量分别为4 800 KB、6 400 KB、8 000 KB、9 600 KB(第二方案中的存储量取平均值)。在文献[12]方案中(简称第三方案),重加密密钥存储量为3 200 KB、4 800 KB、6 400 KB、8 000 KB、9 600 KB。重加密密钥的对比如图1所示。

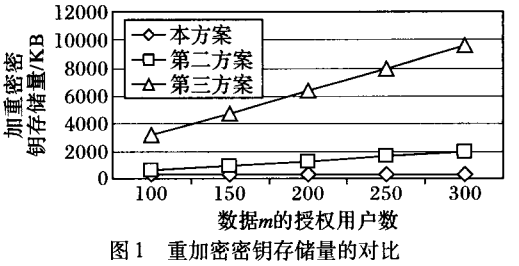


图1 重加密密钥存储量的对比

## 6 结语

本文提出了一种云环境下的基于属性加密和重加密混合的密钥管理方案,该方案可以实现对加密数据的组重加密,从而减少了重加密密钥的个数和用户解密密钥的个数。下一阶段的工作包括解决该方案中云服务提供商和用户的合谋问题,以及多个属性机构下的用户授权以及具有全安全性的基于属性的分层方案,可以满足云环境的高性能、可扩展性等特点。

**参考文献:**

[1] 冯登国, 张敏, 张妍, 等. 云计算安全研究[J]. 软件学报, 2011, 22(1): 71-83.

[2] 陈全, 邓倩妮. 云计算及其关键技术[J]. 计算机应用, 2009, 29(9): 2562-2567.

[3] 林果园, 贺珊, 黄皓, 等. 基于行为的云计算访问控制安全模型[J]. 通信学报, 2012, 33(3): 59-66.

[4] 云端安全——云计算面临的安全挑战[EB/OL].[2013-01-20]. http://labs.chinamobile.com/news/40509.2010.

(下转第2877页)

### 5.2 算法参数选择

使用本文算法进行训练集特征分布建模时,需要对混合高斯模型中高斯函数的个数 $l$ 进行设定。$l$ 的选择对训练集特征分布的描述和最后的识别率有直接的影响,$l$ 过小有可能对训练集特征的分布信息反映得不够充分,$l$ 过大则会引入一些不必要的干扰信息。设训练集中样本个数为 $n$,令

$$C = l/n \tag{17}$$

实验中发现,当 $C > 0.2$ 时,算法的识别效果会有明显的下降。图 4 给出在一次实验中算法识别率随 $C$ 的变化曲线。在使用本文算法时,建议训练集中的样本个数 $n > 10$,$l$ 在 $[0.06n, 0.17n]$ 范围内选取。

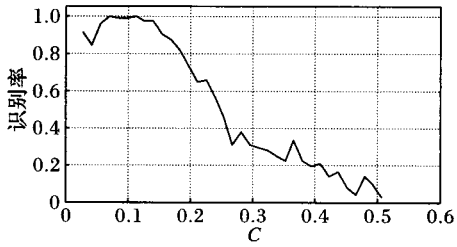


图 4　算法识别率随 $C$ 的变化曲线

## 6 结语

本文提出一种基于投影熵特征的图像识别算法。首先,针对原始定义下投影熵特征的不足,从"扩展"和"规范化"两个方面给出了改进的投影熵特征的定义,将图像的局部投影熵特征向量用于图像识别;在进行图像识别时,求取由训练集图像的局部投影熵特征得到的混合高斯模型和目标图像局部投影熵特征的 Mahalanobis 距离,根据距离判别法原理得到目标图像所属类别。实验表明:1) 与传统的基于不变矩的识别算法相比,本文算法具有更好的识别效果;2) 使用本文算法时候应保证训练集样本数目 $n > 10$,混合高斯模型中高斯函数的个数应在 $[0.06n, 0.17n]$;3) 本文算法具有良好的并行运算特性,采用多通道并行处理器结构的硬件体系来更好地体现出该算法的优势。

**参考文献:**


[1] 余瑞星, 孟立勋. 一种新的 ICM 模型参数设置方法[J]. 西北工业大学学报, 2012, 30(2):201 - 205.

[2] 刘雅轩, 苏秀琴, 王萍. 一种基于局部投影熵的图像匹配新算法[J]. 光子学报, 2004, 33(1):105 - 108.

[3] 王红梅, 李言俊, 张科. 一种基于 Contourlet 变换的图像匹配算法[J]. 宇航学报, 2008, 29(5):1643 - 1647.

[4] GUO X J, WANG W. Image matching algorithm based on subdivision wavelet and local projection entropy[C]// Proceedings of the World Congress on Intelligent Control and Automation. Piscataway: IEEE, 2006: 10380 - 10383.

[5] HUANG Y Y, LI J P, LIN J, *et al*. Robust face recognition by combining wavelet decomposition and local hybrid projection entropy [C]// Proceedings of the 2009 International Conference on Apperceiving Computing and Intelligence Analysis. Washington, DC: IEEE Computer Society, 2009:325 - 328.

[6] ZHANG J S, CHEN C S. Local variance projection log energy entropy features for illumination robust face recognition[EB/OL]. [2013-03-20]. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4547649.

[7] 盛业华, 张卡, 叶春, 等. 基于灰度投影的数字近景摄影立体影像匹配[J]. 光学学报, 2005, 25(12):1623 - 1628.

[8] ZHAN X L, MA B. Gaussian mixture model on tensor field for visual tracking[J]. IEEE Signal Processing Letters, 2012, 19(11): 733 - 736.

[9] FERNANDO B, FROMONT E, MUSELET D, *et al*. Supervised learning of Gaussian mixture models for visual vocabulary generation [J]. Pattern Recognition, 2012, 45(2): 897 - 907.

[10] DEMPSTER A P, LAIRD N M, RUBIN D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Royal Statistical Society, 1977, 39(1):1 - 38.

[11] TANG D J, ZHANG W S, WANG C, *et al*. Image recognition technology based on projection entropy[EB/OL]. [2013-02-17]. http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1267038.

[12] 师义民, 徐伟, 秦超英, 等. 数理统计[M]. 北京: 科学出版社, 2009: 227 - 243.

[13] Columbia University. Computer vision laboratory databases[DB/OL]. [2013-04-10]. http://www.cs.columbia.edu/CA-VE/databases/.

[14] NENE S A, NAYAR S K, MURASE H. Columbia Object Image Library, COIL-20[R]. New York: Columbia University, Department of Computer Science, 1996.

[15] SHI J F, SUN B. Research on image recognition based on invariant moment and SVM[C]// Proceedings of 1st International Conference on Pervasive Computing, Signal Processing and Applications. Piscataway: IEEE, 2010: 598 - 602.


(上接第 2834 页)


[5] BENNANI N, DAMIANI E, CIMATO S. Toward cloud-based key management for outsourced databases[C]// Proceedings of the 34th Annual IEEE Computer Software and Applications Conference Workshops. Piscataway: IEEE, 2010: 232 - 236.

[6] D'SOUZA R, JAO D, MIRONOV I, *et al*. Publicly verifiable secret sharing for cloud-based key management[C]// Proceedings of the 12th International Conference on Cryptology. Berlin: Springer-Verlag, 2011: 290 - 309.

[7] FAKHAR F. Management of symmetric cryptographic keys in cloud based environment [C]// Proceedings of the 15th International Conference on Advanced Communication Technology. Piscataway: IEEE, 2013: 39 - 44.

[8] YANG X, SHEN Q N, YANG Y H, *et al*. A way of key management in cloud storage based on trusted computing[C]// Proceedings of the 8th IFIP International Conference on Network and Parallel Computing. Berlin: Springer-Verlag, 2011: 134 - 145.

[9] SUN L, DAI Z S, GUO J D. Research on key management infrastructure in cloud computing environment[C]// Proceedings of the 9th International Conference on Grid and Cloud Computing. Washington, DC: IEEE Computer Society, 2010: 404 - 407.

[10] TYSOWSKI P K, HASAN M A. Re-encryption-based key management towards secure and scalable mobile applications in clouds[C]// IACR Cryptology ePrint Archive. Berlin: DBLP, 2011: 668 - 678.

[11] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext - policy attribute-based encryption[C]// Proceedings of the 2007 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2007: 321 - 334.

[12] TYSOWSKI P K, HASAN M A. Hybird attribute-based encryption and re-encryption for scalable mobile applications in clouds[EB/OL]. [2013-01-20]. http://cacr.uwaterloo.ca/techreports/2013/cacr2013-13.pdf.

# 云环境下的基于属性和重加密的密钥管理

作者： 罗文俊， 徐敏， LUO Wenjun， XU Min
作者单位： 重庆邮电大学计算机科学与技术学院,重庆,400065
刊名： 计算机应用 ISTIC PKU
英文刊名： Journal of Computer Applications
年，卷(期)： 2013,33(10)