数据库中基于密文角色的密钥管理方案①

肖 飞，黄正东，王光华

(广州军区武汉总医院 信息科，武汉 430070)

摘 要：设计一种高效安全的管理方案来解决数据库中加密密钥的分配与管理问题。通过对多种密钥管理机制的分析，结合数据库角色管理设计了一种基于密文角色的密钥管理方案。分析了基于密文角色的密钥管理方案的安全性，并在数据库中对其进行了应用实现。基于密文角色的密钥管理方案极大方便了密文访问权限的管理，极大提高了密钥管理的效率。

关键词：数据库加密；密文角色；密钥管理

Design of Security Key Management Scheme Based on Encryption Role in Database

XIAO Fei, HUANG Zheng-Dong, WANG Guang-Hhua

(Information Department,Wuhan General Hospital of Guangzhou Command,Wuhan 430070, China)

Abstract: Designing an effective safe management scheme to resolve the database encryption key for the allocation and management issues. By analyzing various key management mechanism,a key management scheme based on encryption role is designed. Security of key management scheme based on encryption role was analyzed,key management scheme based on encryption role was actualizd in database. Key management scheme based on encryption role greatly facilitated access jurisdiction management and greatly improved the key management efficiency.

Keywords: database encryption; encryption role; security key management

1 引言

在密码体制中，密钥是控制明文与密文之间加密与解密变换的关键[1]，也是其安全性的度量[2]，对数据库采取加密技术来保证数据库安全性时，如何保证密钥本身的安全性又是一件非常困难的事情[3]。在数据库管理系统中，由于数据的共享性和存储数据的持久性等原因，要求灵活和安全的加密机制[4,5]及密钥管理机制[6]。

为解决数据库中加密密钥的分配与管理问题，G.I.DaVid 提出了子密钥管理技术[7]，但此技术的安全性一直未得到充分的证明。清华大学的陈卫提出了可变密钥管理方案[8]，其原理是基于一种多向锁的机制将用户密钥转换为所需的数据密钥，但其未能给出具体的实现方法。随后，清华大学的戴一奇等提出了一种基于转换表的密钥管理方案[9]，其实现方式是将用户密钥通过查询转换表的方式得到加密数据的工作密钥，从而来实现对数据的解密。后来，他们又对此密钥管理方法进行了改进，提出了二级转换表密钥管理方案[10]，通过将用户分类，每类用户有一个类密钥，用户在访问密文数据时，首先由用户密钥得到类密钥(此过程对应于一级转换表)，再由类密钥获取工作密钥(此过程对应于二级转换表)，从而来实现对数据的加解密。之后，华中科技大学的余祥宣等针对分布式环境下数据库加密密钥管理问题提出了一种基于矩阵模型的密钥转换表方案[11]，但此方案的实现极其复杂，效率也未得到充分论证。中山大学张立秋等人则提出了一种可用性与易用性更高的密钥管理方式[12]，在基于字段加密的前提下，由数据密钥对敏感数据进行加密保护，再由数据库用户公钥对数据密钥进行加密保护，最后再用数据库用户口令对用户私钥进行加密保护。

角色是数据库中一些操作权限的集合，角色的引

① 收稿时间:2010-04-21;收到修改稿时间:2010-06-05

入极大的方便了数据库中用户访问授权的管理[13]，密文角色就是数据库中涉及到密文操作的权限的集合，密文角色的引入，使得对密文数据的精细访问控制成为现实。针对数据库加密的需要，结合数据库角色管理及上述多种密钥管理机制，本文设计了一种基于密文角色的密钥管理方案并在华中医院的 HIS 系统中进行了实现。

2 密钥管理方案模型

方案模型描述如下：

(1) 加解密算法分别为 E(K,D)和 D(K,E)。

(2) 密钥生成函数为 KR()。

(3) 明文数据库为 D={d_i|i=1,2,…n}，其中 di 代表安全性要求相同可用同一数据密钥加密的字段。数据密钥集为 KD= {kd_i|i=1,2,…n}，其中，kd_i 为 d_i 所对应的数据密钥。密文数据库为 C={c_i|c_i=E(kd_i,d_i),i=1,2,…n}。

(4) 密文用户集为 CU={(cu_j,p_j)|j=1,2,…m}，其中 cu_j 为密文用户的用户名，p_j 为用户 cu_j 的口令。用户密钥集为 KCU={kcu_j|j=1,2,…m}，其中 kcu_j 为密文用户 cu_j 所对应的用户密钥。

(5) 密文角色集为 CR={cr_k|k=1,2,…l}。密文角色密钥集为 KCR={kcr_k| k=1,2,…l}，其中 kcrk 为密文角色 crk 所对应的角色密钥。

(6) 安全密钥为 KS。

(7) 密文数据字典用于记录各密文字段及其对应密钥信息，其结构为一维表 Td=td[1…n]，其中 td[i]={di,kdiS}，kd_i^S =E(KS,kd_i)，i=1,2,…n。

(8) 密文用户字典用于记录密文用户及其对应用户密钥信息，其结构为一维表 Tcu=tcu[1…m]，其中 tcu[j] ={cu_j,kcu_j^U,kcu_j^S}，kcu_j^U=E(H(cu_j,p_j),kcu_j)，H 为哈希函数，kcu_j^S= E(KS,kcu_j)，j=1,2,…m。

(9) 密文角色字典用于记录密文角色及其对应角色密钥信息，其结构为一维表 Tcr= tcr[1…l]，其中 tcr[k]={cr_k,kcr_k^S}，kcrkS=E(KS,kcr_k)，k=1,2,…l。

(10) 密文用户与角色对应关系字典的结构为二维表 Tcu_cr=tcu_cr[1 … m,1 … l] ，其中 tcu_cr[j,k]={cu_j,cr_k,kcr_k^E}，kcr_k^E=E(kcu_j,kcr_k)，j=1,2,…m，k=1,2,…l。

(11) 密文角色与数据对应关系字典的结构为二维表 Tcr_d=tcr_d[1…l,1…n],其中 tcr_d[k,i]={cr_k,d_i,kd_i^E}，kd_i^E= E(kcr_k,kd_i)，k=1,2,…l，i=1,2,…n。

方案中共有四种密钥：用户密钥、密文角色密钥、数据密钥和安全密钥，四种密钥均由密钥生成函数随机生成，前三者保存于数据库中，安全密钥则由安全管理员保存于安全介质中。由方案模型可知，数据库中保存的前三种密钥均由其它密钥(或散列码)加密[14]，只要加密算法及哈希函数的安全性足够高[15]，它们的安全性即可保障。而安全密钥也单独保存于安全介质中，从而保证了此密钥管理方案安全性。

3 安全管理系统

安全管理系统是整个数据库加密系统的控制中心，用于完成密文用户、密文角色及各种密钥的管理，同时还对用户的密文访问进行审核、解析并提交数据库系统执行。

3.1 安全密钥的管理

密钥的生成：由 KR()随机生成 KS，生成后保存于安全介质中，至更换前其值保持不变。

密钥的更换：

(1) 由 KR()随机生成新密钥 KS’。

(2) 更新 Td，先由旧密钥 KS 解密 Td 各记录中加密后的数据密钥，再由 KS’对各数据密钥重新加密。设更新后的记录为 td’[i]，则有 td’[i]={d_i,E(KS’,D(KS,kd_i^S))}={d_i, E(KS’,D(KS,E(KS,kd_i)))}={d_i,E(KS’,kd_i)}，i=1,2,…n。

(3) 更新 Tcr，先由旧密钥 KS 解密 Tcr 各记录中加密后的角色密钥，再由 KS’对各角色密钥重新加密。设更新后的记录为 tcr’[k]，则有 tcr’[k]={cr_k,E(KS’,D(KS,kcr_k^S))}={cr_k, E(KS’,D(KS,E(KS,kcr_k)))}= {cr_k,E(KS’,kcr_k)}，k=1,2,…l。

(4) 由 KS’更换安全介质中 KS。

3.2 数据密钥的管理

密钥的生成：根据 d_i 由 KR()随机生成 kd_i，然后向 Td 中插入新纪录 td[i]= {d_i,kd_i^S}={d_i,E(KS,kd_i)}。

密钥的更换：

(1) 根据 d_i 由 KR()随机生成新密钥 kd_i’。

(2) 使用 kd_i 将加密数据列解密，解密出 d_i，再使用 kd_i’对其加密。

(3) 更新 Tcr_d 中与 d_i 相关的记录。首先由 Tcr_d 中查询出与 d_i 相关的记录的 cr_k，再从 Tcr 中查询出 crk 对应的 kcr_kS=E(KS,kcr_k)，解密出 kcr_k=D(KS,E(KS,kcr_k))= D(KS,kcr_k^S)，然后更新 tcr_d[k,i]={cr_k,d_i,kd_i^E}

={cr_k,d_i, E(kcr_k,kd_i)} 为 tcr_d ’ [k,i]={cr_k,d_i,kd_i ’ E}={cr_k,d_i,E(kcr_k,kd_i’)}。

(4) 更新 Td 中记录 td[i]={d_i,kd_i^S}={d_i,E(KS,kd_i)} 为 td’[i] ={d_i,kd_i’^{S}}={d_i,E(KS,kd_i’)}。

3.3 密文角色的管理

密文角色的建立：

(1) 建立角色 cr_k。

(2) 根据 cr_k 由 KR()随机生成 kcr_k，然后向 Tcr 中插入新纪录 tcr[k]={cr_k,kcr_k^S}={cr_k, E(KS,kcr_k)}。

密文角色密钥的更换：

(1) 根据 cr_k 由 KR()随机生成新角色密钥 kcr_k’。

(2) 更新 Tcr_d 中与 crk 相关的记录。首先由 Tcr_d 中查询出与 crk 相关的记录的 d_i，再从 Td 中查询出 di 对应的 kd_i^S=E(KS,kd_i)，解密出 kd_i=D(KS,E(KS,kd_i))=D(KS,kd_i^S)，然后更新 tcr_d[k,i]={cr_k,d_i,kd_i^E}={cr_k,d_i,E(kcr_k,kd_i)}为 tcr_d’[k,i] ={cr_k,d_i,kd_i’^{E}}={cr_k,d_i,E(kcr_k’,kd_i)}。

(3) 更新 Tcu_cr 中与 cr_k 相关的记录。首先由 Tcu_cr 中查询出与 cr_k 相关的记录的 cu_j，再从 Tcu 中查询出 cu_j 对应的 kcu_j^S=E(KS,kcu_j)，解密出 kcu_j=D(KS,E(KS,kcu_j))= D(KS,kcu_j^S)，再将 tcu_cr[j,k]={cu_j,cr_k,kcr_k^E}={cu_j,cr_k, E(kcu_j,kcr_k)}更新为 tcu_cr’[j,k]={cu_j,cr_k,kcr_k’^{E}}={cu_j,cr_k, E(kcu_j,kcr_k’)}。

(4) 更新 Tcr 中记录 tcr[k]={cr_k,kcr_k^S}={cr_k, E(KS,kcr_k)}为 tcr’[k]={cr_k,kcr_k’^{S}}={cr_k,^{E}(KS, kcr_k’)}。

授予密文角色密文操作权限：授予密文角色 cr_k 密文数据列 d_i 操作授权时，首先由 Tcr 中查询出 kcr_k^S，解密出 kcr_k=D(KS,E(KS,kcr_k))=D(KS,kcr_k^S)，再从 Td 中查询出 d_i 对应的 kd_i^S=E(KS,kd_i)，解密出 kdi=D(KS,E(KS,kdi))= D(KS,kdi^S)，最后向 Tcr_d 中插入记录 tcr_d[k,i]={cr_k,d_i,kd_i^E} ={cr_k,d_i, E(kcr_k,kd_i)}。

撤销密文角色密文操作权限：撤销密文角色 cr_k 密文数据列 d_i 操作授权时，从 Tcr_d 中删除记录 tcr_d[k,i]={cr_k,d_i,kd_i^E}。

3.4 密文用户的管理

密文用户的建立：

(1) 建立用户 cu_j，为用户分配口令 p_j。

(2) 根据 cu_j 由 KR()随机生成 kcu_j，然后向 Tcu 中插入纪录 tcu[j]={cu_j,kcu_j^U,kcu_j^S}={cu_j,E(H(cu_j, p_j),kcuj),(KS,kcu_j)}。

密文用户密钥的更换：

(1) 根据 cu_j 由 KR()随机生成新用户密钥 kcu_j’。

(2) 更新 Tcu_cr 中与 cu_j 相关的记录。首先由 Tcu_cr 中查询出与 cu_j 相关的记录的 cr_k，再从 Tcr 中查询出 cr_k 对应的 kcr_k^S=E(KS,kcr_k)，解密出 kcr_k=D(KS,E(KS,kcr_k))= D(KS,kcr_k^S)，再将 tcu_cr[j,k]={cu_j,cr_k,kcr_k^E}={cu_j,cr_k, E(kcu_j,kcr_k)}更新为 tcu_cr’[j,k]={cu_j,cr_k,kcr_k’^{E}}={cu^j,cr_k ’^{E}(kcu_j’,kcr_k)}。

(3) 更新 Tcu 中记录 tcu[j]={cu_j,kcu_j^U,kcu_j^S}={cu_j,^{E}(H(cu_j,p_j),kcu_j),E(KS,kcu_j)}为 tcu’[j] ={cu_j,kcu_j’^{U},kcu_j’^{S}} ={cu_j,^{E}(H(cu_j,p_j),kcu_j’),E(KS,kcu_j’)}。

密文用户口令的更改：用户口令由 p_j 更新为 p_j’，更新 Tcu 中记录 tcu[j]={cu_j,kcu_j^U,kcu_j^S}={cu_j,E(H(cu_j,p_j),kcu_j), E(KS,kcu_j)}为 tcu’[j]={cu_j,kcu_j’^{U},kcu_j^S} ={cu_j,E(H(cu_j,p_j’), kcu_j),E(KS,kcu_j)}。

授予密文用户密文访问权限：授予密文用户 cuj 密文角色 cr_k 时，首先由 Tcu 中查询出 kcu_j^S，解密出 kcu_j= D(KS,E(KS,kcu_j))=D(KS,kcu_j^S)，再由 Tcr 中查询出 kcr_k^S，解密出 kcr_k=D(KS,E(KS,kcr_k))=D(KS, kcr_k^S)，最后向 Tcu_cr 中插入记录 tcu_cr[j,k] ={cu_j,cr_k,kcr_k^E} ={cu_j,cr_k, E(kcu_j,kcr_k)}。

撤销密文用户密文访问权限：撤销密文用户 cu_j 密文角色 cr_k 时，从 Tcu_cr 中删除记录 tcu_cr[j,k]={cu_j,cr_k,kcr_k^E}。

4 用户对密文数据的访问

用户访问数据库中密文数据的主要过程如图 1 所示。其主要步骤为：

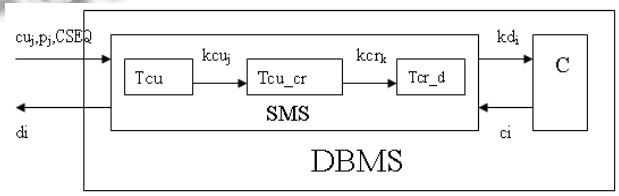

图 1　用户访问数据库中密文数据的主要过程

(1) 用户 cu_j 向数据库管理系统(DBMS)发出密文查询请求 CSEQ，DBMS 首先验证用户所提交的用户名 cu_j 和口令 p_j，通过数据库验证后 DBMS 将此用户的查询请求转给安全管理系统(SMS)处理；

(2) SMS 验证用户 cu_j 是否有 CSEQ 中涉及密文的访问权限，若有，则由密文用户字典 Tcu 中取出 cu_j 被加密的用户密钥 kcu_j^U=E(H(cu_j,p_j),kcu_j)，并根据用户提供的用

$R_n(u, j, t)$

户名 cuj 和口令 pj 对其进行解密，得到 $kcu_j=D(H(cu_j,p_j), E(H(cu_j,p_j),kcu_j)) =D(H(cu_j,p_j), kcu_j^U)$，否则，返回；

(3) SMS 根据用户 cu_j、查询请求 CSEQ 及用户密钥 kcu_j 由密文用户与角色对应关系字典 Tcu_cr 查询到用户所具有的密文角色 cr_k 及 cr_k 对应的密文角色密钥 $kcr_k=D(kcu_j,kcr_k^E)=D(kcu_j, E(kcu_j,kcr_k))$；

(4) SMS 根据密文角色 cr_k、查询请求 CSEQ 及密文角色密钥 kcrk 由密文角色与数据对应关系字典 Tcr_d 查询到其所对应的数据密钥 $kd_i=D(kcr_k,kd_i^E)=D(kcr_k,E(kcr_k,kd_i))$；

(5) SMS 根据 CSEQ 由数据库中查询出密文 c_i，使用数据密钥 kd_i 将其解密为明文 d_i 并将其返回给用户 cu_j。

5 安全性分析

密码体制的安全性主要取决于攻击者获得密文后解密出明文的计算复杂度。本密钥管理方案面临的安全攻击主要有下面几种。

(1) 密文统计学攻击。系统中使用一个数据密钥加密一类安全要求相同的字段，攻击者可能试图通过对这一类密文数据进行统计学分析，期望根据相同的密文结合已知的明文统计信息(或经验信息)来破译出其中的明文值，针对此种攻击，只需在选取数据密钥时将明文信息更加细分，使明文信息与加密后的密文不具有相同的统计学规律，这样，此种攻击就无法成功。

(2) 对比明文攻击。在已获取若干明文及其对应密文情况下，攻击者试图从中分析出此类数据的加密密钥，此时，数据的安全性就依赖于加密采用的密码算法的强度，只要密码算法的密钥空间足够大，就可充分抵御此类攻击。

(3) 基于字典表中已知密钥的攻击。本密钥管理方案包含了四种密钥：用户密钥、密文角色密钥、数据密钥和安全密钥，前三者保存于数据库中并采用逐级加密机制加密，安全密钥则由安全管理员保存于安全介质中。即使攻击者获取了数据库表中某种密钥，仍需对此密钥进行解密，根据方案中的设定，要对密钥进行解密只能依赖于对密文用户的登录名和口令的获取或对安全密钥的获取，在密文用户安全保护自己口令及安全管理员妥善保管安全密钥的前提下，此种攻击无法成功。

6 应用实现

基于密文角色的密钥管理方案将密文访问控制与角色管理相结合，对于大量的具有相同密文访问权限的用户，只需建立一个密文角色，而不必将每个用户与其所能访问的密文数据相关联，极大的方便了密文访问权限的管理。通过在数据库中创建相应数据字典并开发相应的安全管理系统，本密钥管理方案已在基于华中医院经济核算与绩效管理系统实施的 oracle 数据库内核层加密系统中应用实现，其中的安全密钥、数据密钥、密文角色、密文用户的管理均由使用 JAVA 和 PB 语言开发的安全管理系统完成，系统自 2007 年 10 月实施以来，运行良好，极大的提高了密钥管理的效率。

参考文献

1 胡向东,魏琴芳.应用密码学教程.北京:电子工业出版社, 2005.4－138.

2 胡志奇.数据库安全与加密技术研究.计算机与现代化, 2003,11:42－46.

3 Fernandez EB, Summers RC, Wood C. Database security and Integrity. Boston: Addison-Wesley Longman Publishing Co. Inc, 1981.97－231.

4 崔国华,汤学明.数据库中加密机制的实施研究.密码与信息, 2000,1:17－21.

5 崔国华,洪帆,付小青,等.数据库系统中一种更安全的加密机制.华中理工大学学报,2000,28(7):29－31.

6 Jansen CJA. On the key Storage Requirements for Secure Terminals. Computers and Security, 1986,5(2):145－149.

7 DaVid GI, Wells DL, Kam JB. A Database Encryption System with Subkeys. ACM Trans. on Database Systems, 1981,6(2): 312－328.

8 陈卫.数据库加密密钥的分配与管理技术.清华大学学报: 自然科学版, 1994,34(1):99－104.

9 戴一奇,尚杰,陈卫,等.一种新的数据库加密密钥管理方案. 清华大学学报,1995,35(4):43－47.

10 尚杰,戴一奇,李向阳.密文数据库及其密钥管理.计算机应用研究,1996,3:98－100.

11 余祥宣,崔永泉,崔国华.分布式环境下数据库加密密钥管理方案.华中科技大学学报(自然科学版),2002,30(4):43－45.

12 张立秋,常会友,刘翔.一种支持共享的高可用数据库加密机制.计算机工程与应用,2005,20:189－191.

13 David, Ferrailolo F, Sandhu R, et al. Proposed NIST Standard for Role-Based Access Control. ACM Trans. on Information and System Security, 2001,4(3):224－274.

14 李哲,方勇,陈淑敏,等.数据库加密技术中散列函数的应用. 计算机工程,2003,29(17):68－69.

15 Agrawal R, Kirenan J, Srikant R, et al. Order Preserving Encryption for Numeric Data. Proc. of the 2004 ACM SIGMOD Int'l Conf. on Management of data. Paris, France. 2004. New York: ACM Press, 2004.563－574.