数据库加密管理工具的设计与实现*

戴一奇 苏中民 陈 卫 尚 杰

(清华大学计算机系 北京 100084)

摘要 本文提出了一种新的密钥管理方法——两级转换表,并以此方案为基础,实现了数据库加密管理工具DBEMT. 文中详细介绍了DBEMT的总体设计和主要功能,根据不同方式的安全类划分,分别讨论了字段、记录和属性分类方式下密文数据库的设计. 与以往的方案相比,该方案具有安全性好、易于实现、运行效率高的特点,具有较好的实用前景.

关键词 数据库加密,密钥管理,密钥转换,数据加密,转换表.

当前在各个计算机应用部门中已存在或正在开发的数据库管理系统数不胜数. 这些系统的安全控制主要是通过口令和授权机制实现的,但在不少场合这种安全机制不能满足要求. 随着计算机应用的普及和深入,安全性问题将会成为许多系统的一个致命弱点,那些不设防的或只有简单控制机制的系统很容易被攻破,致使数据库中的数据内容被泄露甚至被篡改,从而造成严重后果.[1~3]

数据库加密管理工具DBEMT(database encryption management tool)就是针对这种状况而设计的. 它利用数据加密在信息安全保密中的特殊优势,为数据库系统提供一套统一的、便于使用的加解密和密钥管理工具,对数据库内容进行密文化. 用加密这种高度保密方式保护数据,对于运行在较差安全保密机制环境下的数据库系统尤其必要.

1 密钥管理方式

1.1 密钥转换

密钥转换是指合法用户在访问数据库中的某些数据时,将自己的用户密钥作为一个参数送入密钥管理系统,系统利用转换函数得到数据密钥,从而实现数据的加、解密.[4]

密文数据库的简化形式模型可描述如下:

数据库 $D=\{d_1,d_2,\ldots,d_n\}$,d_i 代表安全性要求相同且可用同一数据密钥加密的一类数据,它可以是记录级,也可以是字段级;相应的数据密钥集 $KD=\{kd_1,kd_2,\ldots,kd_n\}$;每类数据还对应一参数,称为多向锁,用集合 $ML=\{ml_1,ml_2,\ldots,ml_n\}$ 表示;加、解密算法为(E,

---

* 作者戴一奇,1946年生,副教授,主要研究领域为算法设计与分析,密码学. 苏中民,1970年生,博士生,主要研究领域为软件理论,软件工程. 陈卫,1969年生,助教,主要研究领域为算法分析,密码学. 尚杰,1972年生,助教,主要研究领域为算法设计,密码学.

本文通讯联系人:戴一奇,北京 100084,清华大学计算机系

本文1995-04-10收到修改稿

D);这样密文数据库可表示为 $C=\{c_i|c_i=E(kd_i,d_i),i=1,2,\ldots,n\}$. 用户集 $U=\{u_1,u_2,\ldots,u_m\}$有 m 个用户,相应地用户密钥集 $KU=\{ku_1,ku_2,\ldots,ku_m\}$,$u_i$ 可访问的数据是数据库 D 的一个子集 D_i. 这样,把用户密钥转换为数据密钥的转换函数 f 必须满足

$$\forall i\forall j(f(ku_i,ml_j)=kd_j \quad \text{当且仅当} \quad d_j\in D_i) \tag{1}$$

图 1 表示了用户访问数据库的过程,其主要步骤是:

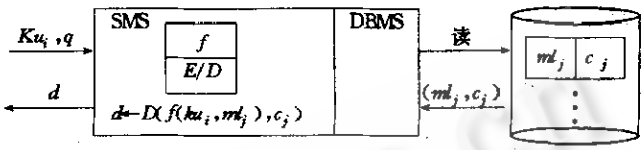

图1 密文数据库访问过程

〈1〉用户 u_i 向安全管理系统(SMS)提供用户密钥 ku_i 及查询请求 q.

〈2〉SMS 根据 q 从数据库中得到要查询的(ml_j,c_j).

〈3〉SMS 利用 ku_i,ml_j 做 $d\leftarrow D(f(ku_i,ml_j),c_j)$.

〈4〉SMS 将 d 返回给用户 u_i.

其中〈3〉是关键. 根据(1)式,当且仅当 u_i 可访问 d_i 时,$f(ku_i,ml_j)=kd_j$,从而 $d=d_j$. 这样保证了系统安全读取,这就是密钥转换的基本思想.

1.2　两级转换表方案

通常,数据库中数据安全类的数目较少,用户数量较多,但用户安全类的数目不会很多,也就是说有很多用户将属于同一用户安全类. 基于这种分析,我们提出了两级转换表方案.[5]

设 uc_i 是一个用户安全类,其中各用户可访问的数据类是相同的(仍用 D_i 表示),则用户类集可表示如下:　$UC=\{uc_i|$对所有 $u_j\in uc_i,u_k\in uc_i$,满足 $D_j=D_k\}$

每个用户安全类设定一个用户类密钥 kuc_j,这样密钥转换可以通过两级完成:第一级从用户密钥 ku_i 通过表 T_1 转换得到 kuc_j,第二级通过表 T_2 从 kuc_j 转换得到 kd_k. 因为一个用户只属于一个用户类,所以令表 $T_1=t_1[1..m,1..3]$,其中

$$\begin{aligned} &t_1[i,1]=E'(ku_i,kuc_j) && u_i\in uc_j\\ &t_1[i,2]=j && \text{用户类标志 } 1\leqslant i\leqslant m\\ &t_1[i,3]=E'(ku_i,c_0) && \text{用于身份验证,}C_0\text{ 为一常数} \end{aligned} \tag{2}$$

其中 $t_1[i,1]=ku_i\oplus kuc_j,u_i\in uc_j$.

这时对属于同一用户类的两个用户 u_i,u_j,可以用以下关系

$$ku_i\oplus t_1[i,1]=ku_j\oplus t_1[j,1]$$

因而 u_i,u_j 实际上可彼此导出对方的密钥,但由于他们本来就是同一类用户,因此即使相互知道密钥也不可能得到数据库系统中其它非授权信息. 而属于不同类的两用户,由于他们各自的用户类密钥是相互独立的,因而 $ku_i\oplus t_1[i,1]$与 $ku_j\oplus t_1[j,1]$毫无关系,不可能由此导出对方的密钥. 因此在一级表中算法采用异或方式仍是安全的. T_2 仍是二维表. 设用户安全类有 h 个,则 $T_2=t_2[1\ldots h,1\ldots n]$,其中

$$t_2[i,j]=\begin{cases}E'(kuc_i,kd_j) & d_j\in D_i\\ \text{无定义} & d_j\notin D_i\end{cases} \qquad 1\leqslant i\leqslant h,1\leqslant j\leqslant n \tag{3}$$

读取数据时用户仍提供(ku_i,i),得到后系统完成如下操作

〈1〉$a\leftarrow D'(ku_i,t_1[i,1])$,$b\leftarrow t_1[i,2]$

(2)$c \leftarrow D'(a, t_2[b,k])$

(3)$d \leftarrow D(c, c_k)$,送回 d

由(2),(3)可知,当且仅当 $d_k \in D_i$ 时,$d=d_k$. 因此两级转换表方案的安全性依赖于 T_2 表,亦即一级转换表的安全性,对此文献[5]中已经进行了证明,本文不再赘述. 两级转换表方案在常数时间内完成一次数据读取操作,而且管理操作十分简便,与以往的方案相比具有明显的优越性.

2 DBEMT 总体设计

DBEMT 以两级转换表方案为基础.

2.1 项目管理者

项目管理者是转换表的产生者和管理者,一个实用的数据库系统称为一个项目,由于一台机器的 DBEMT 中可能支持多个数据库系统,因此每个项目都有各自的转换表. 每个项目都有一个管理者(PM),他具有负责产生、管理转换表和密文数据库的特权,亦即他掌握有项目主密钥 mk. mk 在创造项目时产生. 在以后使用 DBEMT 时,项目管理者只有正确输入项目名称和 mk 后,通过系统的身份验证才能行使其特权.

转换表的产生由 PM 完成,PM 根据项目的安全性需求确定用户安全类和数据安全类个数,及读写权限关系,随机生成数据密钥和用户类密钥,形成第二级转换表,同时 PM 又要对用户进行注册,确定其所属安全类. 并根据用户提供的密钥维护第一级转换表. 另外,为保证完整性,在第一、二级转换表中每一行最后都要加上对该行的密码校验和,这样该行部分内容被改动后会被及时发现. 密文库的初始化也要由 PM 完成,在确定数据的安全类后,利用所产生的数据库密钥进行加密,并添加校验和,通过这一过程把原来的明文库加密为密文数据库. 转换表与密文库的管理也是 PM 的主要任务,它包括各类密钥增删、更换,用户权限变更等多项管理服务,这也是 DBEMT 所要完成的主要功能.

2.2 密文数据库

转换表方案中所讨论的数据库只是一个抽象的数据集合,在实现时必须将它与实际的数据库联系起来. DBEMT 是针对关系数据库而设计的.

关系数据库系统由若干关系组成,每个关系实际上可看成一个二维表格,它对应于一个库文件,其中每一行是相对独立的一条记录,每一列代表一个属性. 行列交叉点(亦称一个字段)上存放某条记录的某一属性值. 由于每个数据都要加一数据安全类标志(DSCT),因此关系数据库中每个字段也应有一个对应的 DSCT. 这必然使原关系(库文件)发生变动. 根据安全类划分的方式不同,原关系的变动形式可有以下几种:

(1)字段分类方式. 每个字段都有自己的 DSCT 值. 例如表 1 显示了学生档案关系加上 DSCT 以后的关系变动情况.

我们规定 DSCT=0 表示非保密数据,不对其进行加密,取 1,2,…,n 等值表示其相应的安全类,需用各自的密钥加密.

表1　加上DSCT后的学生档案关系

学号	DSCT−1	姓名	DSCT−2	性别	DSCT−3	成绩	DSCT−4
1101	0	李明	0	男	0	83.5	1
1102	0	王霞	0	女	0	92	1
1103	0	赵卫	0	男	0	78	1
…	…	…					

字段分类方式可以将安全类划分到最细的粒度,对数据实施不同程度的保护.由于一个字段的加解密独立于其它字段,它对数据库进行查询时对无关的数据可不必解密,进行联合、投影等关系运算时甚至可能不用解密操作.这是字段加密方式的最大优点.为方便起见,我们以DES作为密码算法,这时字段加密方法如下:

算法:字段加密(加密算法以DES为例)

输入:字段长度 l,明文字段 x,密钥 key.

输出:密文字段 y.

过程:

(a)将明文字段 x 分块,$x=x_1 \| x_2 \| \ldots \| x_{n-1} \| x_n$,其中$(n-1)*8+|x_n|=l$,$\|$表示字段中连接.$|x|$:$x$的字节数.

(b)若$|x_n|=8$则

for $i=1$ to n do

$y_i \leftarrow DES(key, x_i, \text{“encryption”})$;

$y=y_1 \| y_2 \| \ldots \| y_{n-1} \| y_n$.

输出 y,结束.

(c)若$|x_n|<8$且$n \geqslant 2$则

for $i=1$ to n do

$y_i \leftarrow DES(key, x_i, \text{“encryption”})$;

$z \leftarrow DES(key, y_{n-1}, \text{“encryption”})$;

$z' \leftarrow$取 Z 的前$|x_n|$个字节;

$y_n \leftarrow z' \oplus x_n$;

$y=y_1 \| y_2 \| \ldots \| y_{n-1} \| y_n$.

输出 y,结束.

(d)若$|x_n|<8$且$n=1$则

随机产生$8-|x_n|$个字节的串 z;

输出 y,结束.

从算法中可以看出,若明文中存在长度小于8字节的短块,当满足条件3时,则短块的加密用前一整块密文再加密结果与短块异或,这样密文无扩展,且安全性有保证,当按条件4处理时,由于采取补足8字节加密的方法,使得同一明文会对应不同密文,弥补了短字段易被破译的缺陷,当然这时是以一定的数据膨胀为代价的.但是,由于字段加密所需额外空间较多,而很多情况下又不要求这样细的粒度,因此还可以有下面两种分类方式.

(2)记录分类方式.[6]将一条记录作为一个整体有一个DSCT值,即只要在原关系的所

有属性后面增加一个 DSCT 属性.这时字段加解密密钥就是它所在记录安全类对应的密钥.

(3)属性分类方式.同一属性的所有字段安全类相同.属性的安全类值加入到属性说明之中,字段加解密时用它所在属性的安全类对应的密钥进行.

为了保证密文数据库的完整性,每个关系最后还要加一属性,称为密码校验和,一条记录完成加密后,要对该记录的所有内容用一个校验算法产生该记录 8 个字节的密码校验和,校验和内容填入记录最后的校验和字段中.其算法描述如下:

算法:产生密码校验和

输入:记录中字节串

输出:校验和

过程:

(a)将字节串分块 $x=x_1 \| x_2 \| \ldots \| x_{n-1} \| x_n$

(b)$Key \leftarrow C_0$,C_0 为 7 字节内部常量;

(c)for $i=1$ to $n-1$ do

{$t \leftarrow DES(Key, x_i, \text{“encryption”})$;

$Key \leftarrow$取 t 的前 7 个字节;}

(d)若$|x_n|=8$,则

$y \leftarrow DES(Key, x_i, \text{“encryption”})$;

输出 y,结束.

(e)若$|x_n|<8$,则

$t \leftarrow x_n \| (8-|x_n|$个字节的‘0’)

$y \leftarrow DES(Key, t, \text{“encryption”})$;

输出 y,结束.

这样,算法输出的密码校验和保证与字符串的所有字节有关,改动某一部分,校验和必然发生变化.

2.3 用 户

用户是 DBEMT 的服务对象.根据转换表方案要求及系统管理的方便,用户应掌握:用户身份 uid、身份号 i 及用户密钥 ku.当用户提供自己的身份,并自选一个密钥后,项目管理者则要根据用户情况确定其所属的用户安全类,DBEMT 根据这些信息将用户注册,并回送给用户一个身份号 i.此后,用户可以利用 DBEMT 读数据库中其权限所辖的内容;用户也可以利用 DBEMT 更换自己的密钥或自我注销.在使用 DBEMT 的服务或项目应用程序时,用户都应提供 uid、i、ku.由系统进行身份验证.

3 DBEMT 的主要功能

DBEMT 的主要功能分为项目创建、项目管理者的管理功能和用户服务 3 大类.

3.1 项目创建

项目创建是在系统中登记一个新项目的过程.新项目由项目管理者负责创建.创建时要输入项目名称,项目主密钥及项目所包括的库文件名.DBEMT 将项目名称和主密钥的密文

形式登记到项目信息文件中.以后只有正确输入项目主密钥的用户才被视为项目管理者.项目所包括的库文件名被存入该项目的库文件名数据文件中,其相应的加解密标记表示当前数据库文件的加密情况.以后对项目数据库的加解密只在这些库文件上进行.

3.2 项目管理者的管理功能

这是DBEMT提供的主要功能.通过它们,PM可以完成数据库加密中的密钥管理,密文库管理等主要任务,使密文数据库系统能适应多种变化的需求.

(1)项目管理功能.主要完成密钥的产生及项目状态的更改等.该功能包括用户、数据类密钥的产生;更换项目主密钥;增加项目中库文件和项目注销.PM可自我注销该项目.这时对数据库文件的处理可有两种选择方式.一种是删除该项目的所有数据库文件;另一种是保留它们的明文版本.即库文件全部解密后再做项目注销.

(2)密文库管理.主要子功能有:库文件加密、解密;显示密文库;更改密文库中数据安全类标志.在库文件加密时,项目管理者首先选出一个待加密的库文件,确定其安全类分类方式.DBEMT将会生成相应的密文库文件结构,然后要对库文件中已有的数据确定其安全类,DBEMT将选用对应的数据密钥,调用加密模块完成加密,并添加密码校验和.

(3)用户管理.主要功能包括新用户注册、用户注销、改变用户权限和显示用户权限.新用户注册时,应向PM提供身份及用户密钥,由DBEMT将用户信息添加到第一级转换表中,并计算相应的转换参数.最后将该用户在表中对应的行号作为用户身份号返回给用户.

(4)安全类管理.这里涉及类密钥更换和第二级转换表更改的一类重要管理功能.它们的使用使得转换机制有较大的灵活性,能够适应数据库系统安全需求的动态变化.主要功能包括:增加用户安全类;删除用户安全类;改变用户类权限;更换用户类密钥;显示用户类权限;增加数据安全类;删除数据安全类.

3.3 用户服务

DBEMT为用户提供的服务主要是涉及安全管理方面的服务.用户使用这些服务必须首先通过身份验证.它包括下面3种功能.

(1)密文库文件浏览.选用此功能时,用户可以读到他所能读到的密文库的部分内容.这些数据是相应的用户可读的数据类数据密钥解密后的结果.对于用户不可读数据不予显示.

(2)更换用户密钥.用户可自行更换密钥.首先输入原密钥,由DBEMT再次确认其身份,正确后他便可以输入一个新的自选密钥.以后只有新密钥才能通过身份验证.

(3)用户自我注销.用户想从项目中退出时,使用此功能.一旦注销后,用户对项目就不再有任何权利.

4 结　语

本文提出了实现数据库安全管理的一个模型.通过引入二级转换表给出了安全存取数据库的框架,并设计实现了数据库加密管理工具DBEMT.目前仍有许多困难有待解决[7,8],如查询处理、数据完整性等,我们力争使这个方案在不久的将来得以实用.

参考文献

1　费杭柏.数据库安全与加密.密码与信息,1991,(1):46~51.

2 Teresa F Lunt, Deborah M Cooper. Introduction: security and privacy. IEEE Transactions on Software Engineering, 1991, **17(11)**: 1145～1146.

3 Denning D E. Cryptography and data security. Massachusetts: Addison—Wesley, 1982.

4 陈卫. 数据库加密密钥的分配与管理技术. 清华大学学报, 1994, **34(1)**: 99～104.

5 戴一奇等. 一种新的数据库加密管理方案. 清华大学学报, 1995, **35(4)**.

6 Hardjono T. Record encryption in distributed databases. Advances in Cryptography—AUSCRYPT'90, Springer—Verlag, 1990. 386～397.

7 Fernandez E B, Summer R C, Wood C. Database security and integrity. Massachusetts: Addison—Wesley, 1981.

8 Dorthy E Denning, Selim G Akl, Matthew Morgenstern. Views for multilevel database security. IEEE Transactions on Software Engineering, 1987, **13(2)**: 129～140.

THE DESIGN AND IMPLEMENTATION OF DATABASE ENCRYPTION MANAGEMENT TOOL

Dai Yiqi Su Zhongmin Chen Wei Shang Jie

(*Department of Computer Science Tsinghua University Beijing* 100084)

Abstract In this paper a new scheme of key management is provided, two—level key transformation, which is considerably secure and easy to be implemented. Based on this scheme, a DBEMT (database encryption management tool) is achieved. The overall designs and functions of DBEMT are presented in detail, according to various ways for security class division, the designs of cipher databases using field classification, record classification and attribute classification methods are discussed respectively. Compared with conventional schemes, it has properties of good operability and high running speed. This scheme offers good prospect for utilization.

Key words Database encryption, key management, key transformation, data encryption, transformation table.