巧用数据库的登录口令实现 MIS 中的电子签名

李 捷 (南京蔡家巷 24 号电力自动化研究院 210003)
徐嗣鑫 (南京东南大学自动化所 210096)

摘要:本文介绍了作者设计的一种管理信息系统中的电子签名方案。该方案将数据库登录口令和电子签名结合起来,采用了已签名数据保护和签名通知等功能,具有安全、可靠、简单和使用方便的特点,在江苏利港电力有限公司管理信息系统中得到了成功应用。

关键词:管理信息系统 电子签名 数据库 客户/服务器体系

在 MIS 中,许多数据都要确定责任人,即需要责任人对数据签名确认,并对其正确性负责。签名在业务处理流程中也十分重要,只有上一个人处理完毕并签名确认后才能转入下一个处理流程,签名是上一处理流程结束和下一处理流程开始的标志。可见,设计一个安全、合理、易于操作的电子签名机制对 MIS 来说是非常重要的。

1.签名口令

在电子签名中,用户需要输入一个签名口令,系统则根据输入口令的正确性来判断签名的合法性,并在口令正确的情况下,通过口令检索出用户的姓名写入数据库中,从而实现签名的功能。并且用户不能通过其他任何方法将姓名写入数据库中从而达到签名的效果。

因为电子签名涉及到重要数据正确性的责任的确定和重要业务流程的走向,一旦安全性存在问题,将对整个系统的安全运行构成极大威胁。因此,安全性是电子签名机制首先需要考虑的因素。而签名口令的选择合适与否是安全性的核心问题。可选择的方案有两种:

(1)签名口令和登录口令分开,为用户单独设置一个签名口令。

(2)签名口令与登录口令统一,签名口令即登录口令。

在第一种方案中,需要建立一个名为"口令"的基表,其建表 DDL 为:

```
CREATE TABLE 口令
( 工号 VARCHAR2(12),
口令 VARCHAR2(10));
```

用户的签名口令即存储在这张表中,每次签名须检索此表,判输入的口令是否正确,并根据检索出的工号继续查出对应的姓名填入签名项中。这种方案的优点是将登录和签名两个概念分开,登录用户和签名用户不一定是同一个概念;缺点是由于"口令"基表是公用查询表,必须对其进行加密处理。加密方案必须严谨、不易破译而又要避免烦琐,避免加密、解密算法占用过多的系统时间。这种选择是比较困难的;用户不仅需要记住自己的登录口令,而且还要记住自己的签名口令,这对于用户来说是相当烦琐的。

经过比较,我们认为采用数据库用户的登录口令作为签名口令比较合适。ORACLE 等数据库为每个用户提供了一个登录口令,用户只有输入正确的登录口令才能实现对数据库的成功登录,并且用户可以随时对自己的登录口令进行修改。采用登录口令作为签名口令,所具有的优点是:由于数据库管理系统为登录口令实施了严密的保密机制,口令的安全性得到了保障;对口令正确性的判断方法比较简便;用户只需记住自己的登录口令,就不仅可以对数据库进行登录,而且还可以进行所有的签名操作,用户操作相对比较简便。

这个签名方案在江苏利港电力有限公司管理信息系统中得到了成功应用。该系统以 ORACLE 网络数据库为核心,采用客户/服务器体系,以 ORACLE DEVELOPER/2000 为开发工具。下面结合系统中的调度指令模块详细介绍签名方案。调度指令下发是发电厂生产管理的一个重要环节,首先作为发令人的调度员根据生产计划和生产的实际情况填写调度指令并签名下发,当作为受令人的班组长查询到该调度指令并签名接受后,该调度指令便正式生效,任何人不得修改,且必须严格按照它进行生产。调度指令的表结构为:

```
create table 调度指令
(编号 varchar2(6),日期 date,当值 varchar2(2),班组 varchar2(2),发令人 varchar2(12), 发令人签名时间 date 接收人 varchar2(12),接收人签名时间 date, 指令内容 varchar2(600) );
```

首先为签名人在数据库中建立一个用户,并为其分配一个登录口令。在签名时签名人只需输入登录口令,

然后由系统判断口令的正确性，最后在口令正确的情况下进行签名操作，即将签名人的姓名填入数据库中。

口令正确性判断可分为两种情况：

第一种情况，签名人是当前的登录用户，这种情况下的判断方法比较简单。只需要使用 FORMS 的内置函数 GET-APPLICATION-PROPERTY(PASSWORD)取得当前登录用户的登录口令，和签名人输入的口令相比较，若相同则判断通过；若不同，则可能签名人不是当前登录用户，进行第二种情况的判断。具体的 PL/SQL 代码如下：

```
pw := get-application-property(password);/* 取得登录口令 */
if upper(pw) = upper(:签名块.password) then /* 和签名人输入的口令相比较 */
/* 签名成功，取得签名人姓名填入数据库 */
:调度指令.发令人 := get-application-property(username);
:调度指令.发令人签名时间 := sysdate;
else /* 可能签名人不是当前登录用户，进行第二种情况的判断 */
……
end if;
```

第二种情况，签名人不是当前登录用户，这种情况下的判断方法相对比较复杂，因为没有直接的方法来判断口令是否正确。这里我们采用了试登录的方法来解决这个问题。

FORMS 提供了两个内置函数：LOGON 和 LOGOUT。

LOGON 函数根据提供的用户名和口令完成对数据库的登录操作，它的语法为：

LOGON(username, password, logon-screen-on-error);

username 为合法的用户名；passoword 为合法的口令；

logon-screen-on-error 若为 FALSE，则调用 LOGON(username, password, FALSE)后，若登录失败，FORMS 的内置变量 FORMS-FAILURE 被自动设置为 TRUE。我们就以这种机制来实现对口令正确性的判断。

LOGOUT 则完成退出数据库登录的操作。

首先，签名人在输入签名口令的同时必须提供自己的用户名。程序根据输入的用户名和口令使用 LOGON 进行登录判断，若变量 FORMS-FAILURE 为 TRUE，则说明签名人提供的口令为非法，否则判断口令是正确的，签名成功。具体的 PL/SQL 代码如下：

```
logout;/* 退出当前登录 */
logon(:签名块.usename,:签名块.password,FALSE);/* 根据签名人提供的用户名和口令试登录 */
if form-failure then /* 试登录失败 */
message('不正确的口令和用户名，请重输入');
else /* 口令正确，签名成功，取得签名人姓名填入数据库 */
:调度指令.发令人 := :签名块.username;
:调度指令.发令人签名时间 := sysdate;
end if;
logon(:global.username,:global.password,FALSE);/* 恢复原来的登录用户 */
```

程序流程图如图 1 所示.

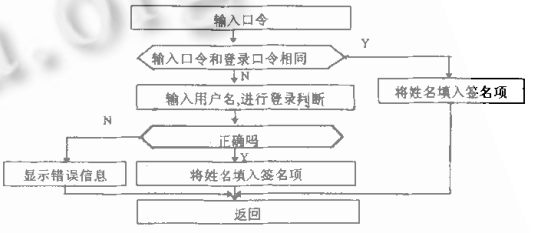

图 1　口令正确性判断的程序流程图

2. 已签名数据的保护

电子签名的另一个重要方面是保证已签名数据可不可更改性，即使用者无法对已经进行了签名操作的记录进行更新、删除等操作。这里我们可以利用 FORMS 中项的 PRE-TEXT-ITEM 触发器，这个触发器在导航到这一项前触发。可以将一段代码放入基表块的第一个导航项的 PRE-TEXT-ITEM 中，这段代码判断当前记录的签名是否为空，若不空，则这条数据表示已经进行了签名，那么就将基表块的 UPDATE-ALOWED、DELETE-ALOWED 属性设置为 FALSE；若为空，则表示这条数据还没有进行签名，那么就将基表块的 UPDATE-ALOWED、DELETE-ALOWED 属性设置为 TRUE。这段代码当导航到每一条记录时都随着触发器被触发而执行，这样就可以对每一条记录进行判断。具体的 PL/SQL 代码为：

```
if block1.签名项 is not null then
  set-block-property('block1', DELETE-ALOWED, FALSE);
  set-block-property('block1', UPDATE-ALOWED, FALSE);
else
  set-block-property('block1', DELETE-ALOWED, TRUE);
  set-block-property('block1', UPDATE-ALOWED,
```

```
TRUE);
    end if;
```

3. **签名通知**

电子签名方案还涉及到一个签名通知的问题。例如生产调度指令需要发令人和受令人都签名才能生效，发令人在输入生产调度指令并签名后，希望尽快通知受令人，使其能够查看生产调度指令并签名确认。这需要有一个签名通知机制。

我们认为一个签名通知机制应达到以下要求：

(1)被通知人应该尽可能及时地得到通知；

(2)通知内容对无关的人员应具有保密性；

(3)被通知人可以通过通知便捷地切换到进行签名操作的界面，并检索到需要进行签名的记录。

作者设计了一个基于客户/服务器体系的签名通知机制，该签名通知机制的主要设计思想是：利用客户/服务器体系，在数据库服务器上设置公告牌，所有的签名通知都存储在公告牌中。在所有的客户机上运行一个监视程序，完成对公告牌内容的定时检索，若检测出有符合要求的通知到达，则以一定的形式告知用户。

公告牌的实现方式有多种，其中比较可行的是利用数据库的基表形式，基表如下：

```
create table 签名通知
(编号 number(4) primary key,通知人 varchar2(12),
被通知人 varchar2(12),
通知内容 varchar2(100),签名入口 varchar2(30),签名关键字 varchar2(30),
发通知时间 varchar2(12));
```

当用户需要进行签名通知，可转入发签名通知模块，在该模块中，用户录入“签名通知”基表的内容。其中用户只需要输入通知人，被通知人和通知内容即可。编号可以由系统自动生成，签名入口取为需要进行签名操作的模块的可执行程序名，签名关键字取为需要进行签名操作的记录的主键，发通知时间取为当前系统时间。具体的 PL/SQL 程序可如下编写：

```
:签名通知.签名入口 := 'c:\lgmis\ddzl.fmx'; /*ddzl.fmx 为调度指令模块的可执行程序名*/
:签名通知.签名关键字 := :调度指令.编号;
:签名通知.发通知时间 := sysdate;
```

当监视程序检索到公告牌上有指定用户的签名通知时，就会以蜂鸣声和对话框给用户以通知。用户得到通知后，就可以进入通知查看模块查看通知的内容。为了确保通知内容的保密性，用户需要输入自己的登录口令。系统对口令的正确性进行校验，校验的方法和前面所叙述的签名口令校验的方法完全一致，这里不再详述。在口令正确的情况下，系统自动检索出发给这个用户的签名通知。用户在查看完通知之后，可以按一个“处理”按钮，系统就可以切换到需要签名的模块，并自动检索出需要签名的记录，用户便可以很方便地完成审核和签名操作。

“处理”按钮的 WHEN-BUTTON-PRESSED 触发器可如下编写：

```
DECLARE
pl-id ParamList;
BEGIN
/*调用签名模块，将签名关键字作为参数传给签名模块*/
pl-id := Create-Parameter-List('my-param');
Add-Parameter(pl-id,'key',TEXT-PARAMETER,:签名通知.签名关键字);
New-Form(:签名通知.签名入口,TO-SAVE-POINT,pl-id);
END;
```

调度指令模块的 PRE-BLOCK 触发器可如下编写：

```
/*将查询条件设置为主键和作为参数传递过来的签名关键字相等*/
Set-Block-Property('调度指令',DEFAULT-WHERE,'编号=:key');
Execute-Query; /*执行查询*/
Set-Block-Property('调度指令',DEFAULT-WHERE,''); /*恢复查询条件*/
```

这样，在切换到调度指令模块时，就会自动检索出需要签名的模块记录供用户进行签名，这样在很大程度上方便了用户。

4. **结论**

本文以在调度指令中的应用为例介绍了电子签名方案，事实上在江苏利港电力有限公司管理信息系统的各个需要签名的模块均采用了这个方案，证明了它是安全、可靠、简单并且使用方便。

参考文献

[1] 俞盘祥，《ORACLE 数据库系统基础》，清华大学出版社，1995 年 4 月.

[2] 《ORACLE Developer/2000 使用技术与方法》(上、下册)，北方交通大学、自动化系统研究所 编著，1996 年 1 月.

(来稿时间：1998 年 11 月)