数据库加密中间件设计与实现

设计目标

- 安全模型：防范目前数据库系统面临的两类安全威胁
- 其一是存储介质丢失可能造成的机密数泄漏，
  其二是数据库管理员权力过大导致信息泄露危机。

设计目标

- 权限分离
  - 有专门的安全管理员负责数据访问控制
- 属性粒度加密
  - 安全管理员可以决定哪些属性需要加密哪些不需要
- 密钥独立于DBMS
  - 削弱DBA的权利
- 用户透明
  - 采用视图机制，让普通用户几乎无法察觉加密的存在，
  - 可以灵活更改加密方案，无需修改客户端代码
- 基于硬件安全
  - 密钥存在Ukey里

系统基本特点

１）低耦合，宿主数据库系统的耦合度最低，对不同类型数据库的插件产品之间具有大量的可共享模块和可重用代码；
（目前只针对SQLserver）

２）高度模块化，　　加密中间件的设计采用高度模块化的方法，能够灵活的使用各种类型的底层加密硬件和加密算法；

３）分权管理，安全管理员是能够操作加密插件的唯一用户，安全管理员通常不是数据库内部用户，因此无法登陆数据库直接操作或查看数据库对象，因此该角色与DBA有着本质的区别。安全管理员仅通过加密中间件管理软件实施对数据库的数据加密以及加密管理操作；

４）图形化管理界面，安全管理员通过图形化的软件界面操作加密中间件方便快捷。

实现环境

- 项目的实现主要是
- 基于SQL Server 2008 和 Visual Studio 2010 的连接，
- 通过程序对SQL Server 数据库进行操作；
- 例如：对表格进行操作
- 新建表格，修改名称，删除表格，
- 添加新列，删除列，更新列值……
- 也可以对视图进行操作
- 新建视图，修改名称，删除视图，
- 添加新列，删除列，更新列值……
- 还可以查询系统表中的数据库，表格，视图，
- 列，类型……
- 密钥：
- 借鉴KEY宝的方式，采用U盘内置文件中的数据来作为修改数
- 据的密钥，这样既加大了保密程度又保证了密钥的唯一性。

基本原理

- 用视图代替表接受用户的访问

视图S

ID	Name	age	specialty
1	张三丰	104	武当
2	周芷若	20	峨嵋
3	杨逍	42	明教

表T_S

ID	Name	age	specialty
1	张三丰	@##@￥#	武当
2	周芷若	&#$$A!k1	峨嵋
3	杨逍	U#93?~	明教

- Create view s
- As
- Select ID, dbo.encrypt(age) as name ,AGE,SPECIALTY from T_S

解密函数实现数据转换

- SQL 用户自定义函数
- CREATE FUNCTION dbo.encrypt (@plaintext int)
- RETURNS int
- AS BEGIN
- DECLARE @cipher int;
- IF (@plaintext >0) SET @cipher= @plaintext+1
- ElSE
- SET @cipher= @plaintext-1
- RETURN(@ cipher);
- END;

数据更新

- 使用instead of 触发器实现用户SQL语句的转换
  - Insert into s values(a1,a2,a3,a4) 变成了 Insert into t_s values(a1, encrypt(a2),a3,a4)
- Create trigger 。。。 instead of insert
- Create trigger 。。。 instead of update
- 脚本见本页备注区域

```
--创建表
create table T_S
( id int identity,
  sname varchar(20),
  sage int,
  sdept varchar(5)
)
--创建函数
create function encrypt(@a int)
returns int
as
begin
  return (@a+10000)
end

create function decrypt(@a int)
returns int
as
begin
  return (@a-10000)
end

--创建视图
create view S
as
select id,  sname,  decrypt(sage) as sage, sdept
from T_S

--插入操作触发器
create trigger ins_S on s
instead of insert
as
  insert into t_s(sname,sage,sdept)
  select sname,dbo.encrypt(sage),sdept from inserted

--更新多条元组只能用这个：
create trigger updt_S on s
instead of update
as
   update t_s
   if update(sname)
   set sname=(select sname from inserted where id=t_s.id),
       sage=(select dbo.encrypt(sage) from inserted where  id=t_s.id),
       sdept=(select sdept from inserted where id=t_s.id)
   where
    t_s.id in (select id from inserted)

--测试用脚本
update s set sage=101 where id=1
update s set sage=sage+1 where id=1
update s set sage=sage+1 where id>1
update s set sname= 'newN' where id=1
```

CRL扩展

- 通过程序集机制把SQL函数扩展到操作系统调用。
- 给SQL函数赋予读取U盘的权限

自定义程序集--原型（Visual Studio 2010）

```
//整形数据的加密实现
public static int encrypt(int plaintext)
{
    int key;
    try
    {
        FileInfo fi = new FileInfo("H:\\csc.txt");
        FileStream fs = fi.OpenRead();
        byte[] ByteArray = new byte[1];
    int nBytesRead = fs.Read(ByteArray, 0, 1);
          key = Convert.ToInt32(ByteArray[0]);
      }
      catch
      {
          key = 0;
      }
      return (plaintext + key);
    }
```

```
//整形数据的解密实现
 public static int decrypt(int cipher)
{
      int key = 1;
      try
      {
          FileInfo fi = new FileInfo("H:\\csc.txt");
          FileStream fs = fi.OpenRead();
          byte[] ByteArray = new byte[1];
      int nBytesRead = fs.Read(ByteArray, 0, 1);
           key = Convert.ToInt32(ByteArray[0]);
        }
        catch
        {
            key = 0;
        }
        return (cipher - key);
    }
```

自定义程序集--原型（Visual Studio 2010）

```
//字符形的加密函数

public static char[] encryptstr(string plainText)
{
        char[] r = plainText.ToCharArray();
        int len = plainText.Length;
        int c;
        int key;

        FileInfo fi = new FileInfo("H:\\csc.txt");
        FileStream fs = fi.OpenRead();
        byte[] ByteArray = new byte[1];
        int nBytesRead = fs.Read(ByteArray, 0, 1);
        key = Convert.ToInt32(ByteArray[0]);

        for (int i = 0; i < len; i++)
        {
            c = Convert.ToInt32(r[i]);
            c = c + key;
            r[i] = Convert.ToChar(c);
        }
        return r;
    }
```

```
//字符形的解密函数

public static char[] decryptstr(string plainText)
{
        char[] r = plainText.ToCharArray();
        int len = plainText.Length;
        int c;
        int key;

        FileInfo fi = new FileInfo("H:\\csc.txt");
        FileStream fs = fi.OpenRead();
        byte[] ByteArray = new byte[1];
        int nBytesRead = fs.Read(ByteArray, 0, 1);
        key = Convert.ToInt32(ByteArray[0]);

        for (int i = 0; i < len; i++)
        {
            c = Convert.ToInt32(r[i]);
            c = c - key;
            r[i] = Convert.ToChar(c);
        }
        return r;
    }
```

自定义函数—使用C#程序集 (SQL Server 2008)

- //整形加密函数的创建
- CREATEFUNCTION
- lencrypt(@plaintext int)
- RETURNS int
- AS EXTERNAL NAME MY.Lclass.encrypt
- GO

- //整形函数解密的创建
- CREATE FUNCTION xdecrypt (@plaintext int)
- RETURNS int
- AS EXTERNAL NAME MY.Lclass.decrypt
- GO

- //字符型加密函数的创建
- CREATE FUNCTION lencryptstr (@plainText nchar(8))
- RETURNS nchar(8)
- AS EXTERNAL NAME MY.Lclass.encryptstr
- GO

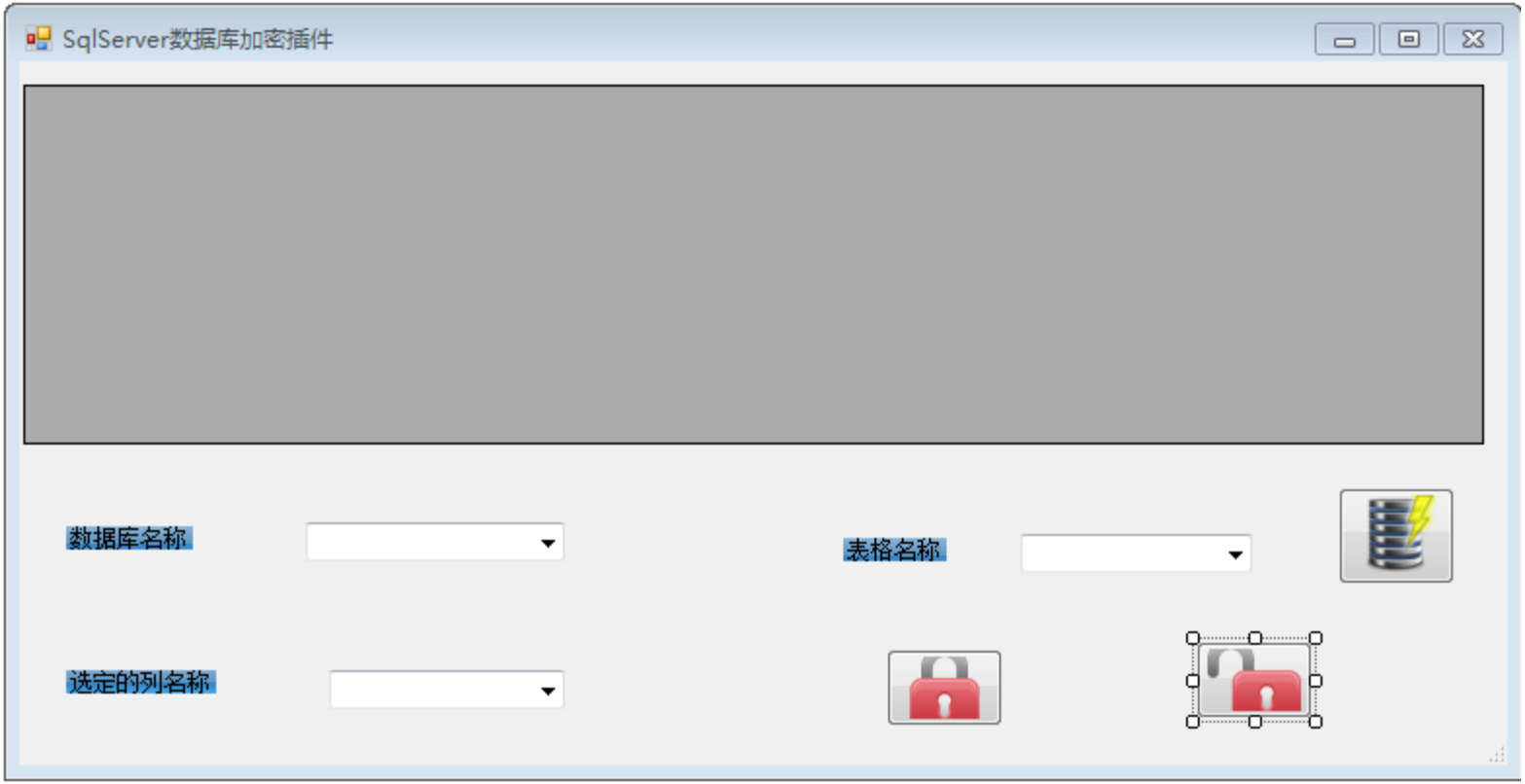
- //字符型解密函数的创建
- CREATE FUNCTION xdecryptstr (@plainText nchar(8))
- RETURNS nchar(8)
- AS EXTERNAL NAME MY.Lclass.decryptstr
- GO

实验过程

- 第一阶段：手工方式实现
  - 通过SQL语句创建函数、视图、触发器
  - Update语句加密数据
  - Select ,insert,update 语句验证效果
- 第二阶段：程序实现
  - 编程实现管理程序，实现UI驱动的加密管理
  - 查询元数据，封装关键SQL语句

界面设计

- 初始界面:



获得关系列表-方法1

- SELECT *
  ROM sysobjects
  WHERE xtype = 'u'
- Sysobjects-在数据库中创建的每个用户定义的架构作用域内的对象在该表中均对应一行。
- Xtype的定义可参考 http://msdn.microsoft.com/zh-cn/library/ms190324.aspx

获得关系列表-方法2

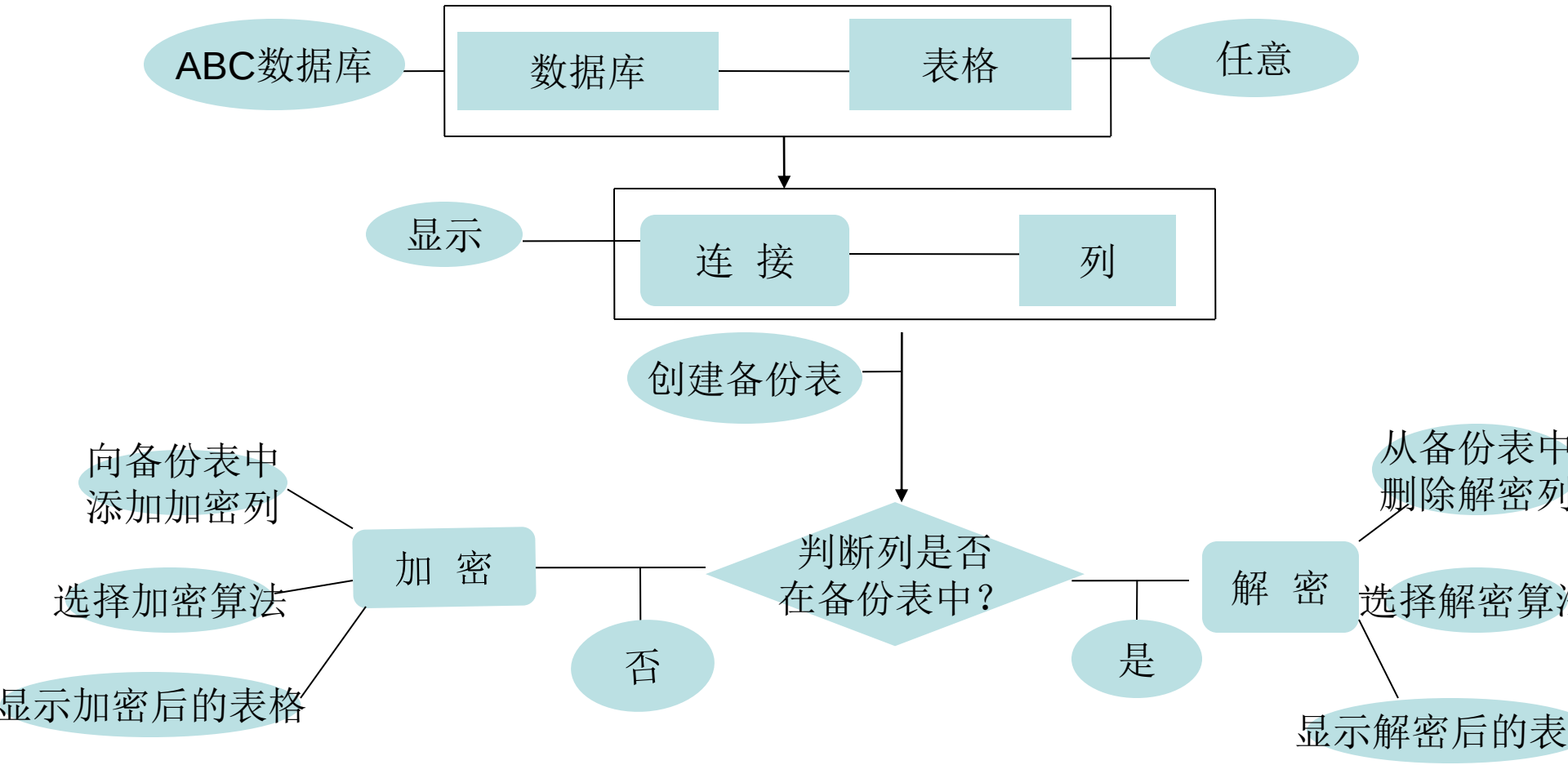
- adox是微软对ADO技术的扩展，使用它我们可以操作数据库

```
private void GetTables_ADOX()
        {
            //ADO的数据库连接
            ADODB.ConnectionClass cn=new
    ADODB.ConnectionClass();
            string ConnectionString="Provider=SQLOLEDB.1;Integrated
    Security=SSPI;Initial Catalog=Test;Data Source=HBXP";
            cn.Open(ConnectionString,"sa","",0);
            //操作ADOX的Catalog对象
            CatalogClass cat=new CatalogClass();
            cat.ActiveConnection=cn;
            for(int i=0;i<cat.Tables.Count;i++)
            {
                MessageBox.Show(cat.Tables[i].Name);
            }
        }
```

加密成功的界面


SqlServer数据库加密插件
床位号
所有者姓名
班级
专业
入住日期
1 余晗桀RRRRR 1班 网络工程 2011/9/1
2 宋孑孓RRRRR 2班 计算机科学与技术 2011/9/1
3 削星RRRRRR 1班 网络工程 2011/9/1
4 宽归茎RRRRR 2班 计算机科学与技术 2011/9/1
5 余薰RRRRRR 1班 网络工程 2011/9/1
6 险弼榭RRRRR 1班 网络工程 2011/9/1
7 了庱董RRRRR 2班 计算机科学与技术 2011/9/1
8 范霜蒌RRRRR 2班 计算机科学与技术 2011/9/1
信息提示
加密成功！！
确定
数据库名称
ABC
表格名称
宿舍信息
选定的列名称
所有者姓名

设计流程


ABC数据库
数据库
表格
任意
显示
连 接
列
创建备份表
向备份表中
添加加密列
选择加密算法
显示加密后的表格
加 密
否
判断列是否
在备份表中？
是
解 密
从备份表中
删除解密列
选择解密算
显示解密后的表

连接按钮

单击连接按钮主要完成的任务如下：

- 1）判断数据库和表格名称是否符合要求：i）特定的数据库ABC；
- ii）ABC数据库中的任意一个表格；
- 2）如果不符合上述两条要求则使用信息框提示出错信息；
- 3）如果符合要求则在datagird中显示出表格中的全部信息；
- 4）连接数据库的语句代码：

```
string consqlserver = @"Data Source = acer98\sqlexpress;
                        Integrated Security = True;Initial Catalog = " + 数据库.Text;
SqlConnection con = new SqlConnection(consqlserver);
```

5）显示出表格信息的语句代码：

```
string sql = "select * from " + 表括.Text;
SqlDataAdapter da = new SqlDataAdapter(sql, con);
da.Fill(ds);                    //填充数据
if (ds.Tables[0].Rows.Count > 1)
{
  dataGridView2.DataSource = ds.Tables[0];
}
```

加密按钮

- 单击加密按钮主要完成的任务：
- 1）判断是否有密钥，若不存在则提示不能操作；
- 若存在则继续执行以下操作：
- 2）为准备加密的表格创建一个备份表格，用来记录已经加过密的列值；
- 注意：一个表格只对应于一个与之对应的备份表格；
- 3）判断准备加密的列在备份表中是否存在，若存在则提示此列已加密
- 若不存在则继续加密：
- 4）判断准备加密的列值类型： 整形和字符型；
- 注意：我们加密的对象只针对整形的字符型两种数据；
- 5）根据列值类型选择相应的加密算法：
- i）整形数据----------lencrypt() 函数
- string sql = "update dbo." + 表格.Text + " set " + 列.Text +
- " = dbo.lencrypt(" + 列 .Text + ")";
- II）字符型数据----------lencryptstr()函数
- string sql11 = "update dbo." + 表格.Text + " set " + 列.Text +
- " = dbo.lencryptstr(" + 列.Text + ")";
- III）非以上两种类型数据则提示不能加密；
- 6）加密成功后将加密后的数据显示出来，把加密成功的列添加到备份表中，
- 并弹出对话框提示“加密成功！”

解密按钮

- 单击加密按钮主要完成的任务：
- 1）判断是否有密钥，若不存在则提示不能操作；
- 若存在则继续执行以下操作：
- 2）判断准备解密的列在备份表中是否存在，若不存在则提示此列未被加密
- 若存在则继续解密：
- 3）判断准备解密的列值类型： 整形和字符型；
- 4）根据列值类型选择相应的加密算法：
- i）整形数据----------lencrypt() 函数
- string sql = "update dbo." + 表格.Text + " set " + 列.Text +
- " = dbo.xencrypt(" + 列 .Text + ")";
- II）字符型数据----------lencryptstr()函数
- string sql11 = "update dbo." + 表格.Text + " set " + 列.Text +
- " = dbo.xencryptstr(" + 列.Text + ")";
- 5)加密成功后将解密后的数据显示出来，从备份表中删除已经解密成功的列；
- 判断备份表中是否还存在未被解密的列：如果存在则不进行任何操作，
- 如果不存在则删除备份表（因为此备份表已经没有意义）

6）最后弹出对话框提示“解密成功！”

后排使用

- 前排使用