

# 机密数据库泄漏源检测与量化评估方案

咸鹤群

(中国科学院软件研究所信息安全国家重点实验室 北京 100190)  
(信息安全共性技术国家工程研究中心 北京 100190)

**摘 要** 针对机密数据库泄漏源检测问题以及现有技术存在的不足,提出了一种基于复合型数据库水印的检测方法与量化评估方案.设计了复合型数据库水印的添加与检测算法,利用水印添加与数据分发协议,将机密数据与其所有者和用户的身份绑定,实现了泄漏源检测与追查过程中对参与双方的公平保护.使用概率分析的方法实现了检测结果的可靠性量化评估.最后通过实验分析了算法的计算性能,验证了方案的有效性和可行性.实验结果表明该方案具有较高的实用价值.

**关键词** 数据库水印;泄漏源检测;复合水印;量化评估;数据库安全

**中图法分类号** TP309 **DOI号**: 10.3724/SP.J.1016.2009.00721

## Leakage Identification and Quantitative Evaluation Scheme for Confidential Databases

XIAN He-Qun

(State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100190)  
(National Engineering Research Center of Information Security, Beijing 100190)

**Abstract** The problem of leakage identification for confidential database is a small-scaled pirate identification problem. After analyzing current solutions and their drawbacks, the author proposes a leakage identification and quantitative evaluation scheme for confidential databases, which is based on a compound database watermark technique. The principle of the novel database watermark technique and the watermark generation and detection algorithms are presented in the paper. By embedding a compound watermark in the data and applying a data watermarking and distribution protocol, the confidential database is bound to the identities of its owner and the user, so the interests of both parties can be fairly protected in the leakage identification process. Probabilistic analyses are adopted to provide a quantitative evaluation of the identification result. Experiments on the algorithms show that the proposed scheme is efficient and effective, and it is highly qualified for real world applications.

**Keywords** database watermark; leakage identification; compound watermark; quantitative evaluation; database security

## 1 引 言

为了保护存储在数据库中的机密信息,人们通

常采用诸如数据库加密和访问控制之类的机密性保护技术.这些技术的安全目标是防止非授权主体访问机密数据,所谓非授权主体主要指的是外部入侵者或系统内部未经授权的用户,而来自于授权用户

的潜在安全隐患则是普通机密性保护技术无法应对的. 由于数据合法持有者的失误或变节造成的机密数据泄露事故往往会导致更为严重的后果. 一旦此类事故发生, 如何准确有效地检测数据泄露来源, 并且通过技术手段提供不可否认的证据, 是责任认定和事故追查需要解决的首要问题. 尤其是在机密数据已经被分发给其合法使用者的情况下, 由于无法借助传统的数据库审计技术, 泄漏源的检测与追查变得更加困难.

机密数据库泄漏源检测的主要任务是鉴定泄露数据的最初来源. 当所有者将一份机密的关系数据与多个使用者共享时, 使用者必须首先承诺不泄漏数据内容. 一旦由于某个使用者变节或失误导致数据泄漏, 所有者不仅需要检测并追踪到数据泄漏的源头, 而且需要向仲裁机构证明泄漏者的泄密行为. 另一方面, 在泄漏发生后, 一个清白的使用者要能够证明自己的清白, 从而防止所有者错误地甚至故意地制造用于指控清白者的证据. 实际上, 机密数据库泄漏源检测问题可以视为小规模的数据库盗版识别问题. 数据库盗版识别的主要任务是保护关系数据库的版权, 同时追查盗版者. 两者的区别在于, 泄漏源检测问题中持有机密数据的用户数量远小于盗版识别问题中持有普通商业数据的用户数量.

现有的数据库盗版识别技术不能直接用于机密数据库泄漏源检测. 目前的数据库版权保护与盗版识别主要依赖于数据库水印技术<sup>[1]</sup>. 利用此类技术, 所有者可以将自己的水印标志以不可察觉的方式添加到数据库中, 并在必要时公开展示水印的存在, 从而证明自己对数据的所有权<sup>[1]</sup>. 在此基础上, 数字指纹式的水印方案增加了识别盗版者的功能<sup>[2]</sup>, 所有者在每个购买者的数据中添加一个特有的水印标志, 一旦发生盗版, 所有者在证明所有权的同时, 可以根据数据中的水印标志查明盗版数据来源于哪个购买者. 不论是普通数据库水印还是指纹式数据库水印, 都不能直接被用来解决机密数据库泄漏源检测问题, 因为在现有的方案中, 生成水印所使用的密钥和指纹字符串都是由数据所有者选定的, 整个水印添加和数据分发过程也由所有者控制, 数据的使用者无从验证水印的正确性. 因此, 所有者完全有可能错误地、甚至故意地将其它使用者的水印添加到一个清白的使用者的数据中. 所有者虽然可以通过展示水印的存在来证明自己对数据的所有权, 但不能仅凭自己制造的“证据”指控某个使用者.

本文针对机密数据库泄漏源检测问题, 改进并

扩展了 Agrawal 等人提出的水印方案<sup>[3]</sup>, 设计了一种复合型数据库水印, 用于机密数据库泄漏源检测, 同时在概率分析的基础上给出了泄漏源检测结果的可靠性量化评估方法, 从而形成了一套完整的机密数据库泄漏源检测与量化评估方案. 其主要创新之处在于: 在实现机密数据库的泄漏源检测与评估的同时, 该方案能够有效地防止所有者对清白的使用者进行错误或恶意的指控, 从而公平地保护双方的利益.

## 2 相关工作

数据库水印技术是在多媒体水印技术的基础上发展起来的. 研究者们沿用传统数字水印的原理, 结合关系型数据自身的特点, 提出了一系列用于数据库的水印技术和方法, 其中最具代表性的是 Agrawal 等人在 2003 年提出的适用于关系数据的数字水印方案<sup>[3]</sup>. 该方案利用数据库元组中的非关键比特 (Least Significant Bits, LSBs), 将水印标志以数据误差的形式添加到数据中. 水印的添加过程由所有者的密钥控制, 通过对部分元组的非关键比特的值进行修改, 实现水印标志的添加和隐藏. 任何人在不掌握该密钥的情况下无法检测或删除数据中的水印标志. 所有者在向公众证明所有权时, 需要出示该密钥并使用公开的水印检测算法证明水印的存在性. 在 Agrawal 等人的水印方案的基础上, Swarup 等人在 2004 年提出了一种可以用作数字指纹的水印方案<sup>[2]</sup>, 该方案能够把任意的比特串以水印方式添加到数据中. Agrawal 等人的水印方案中每个水印标志位的取值 (0 或 1) 是由密钥决定的, 而指纹式水印方案使用指纹比特串和密钥共同决定各个标志位的取值. Guo 等人 2005 年提出的指纹式水印方案能够实现与文献[2]中方案相同的安全功能<sup>[4]</sup>. Li 和 Deng 在 2006 年提出了一种可公开验证的水印方案<sup>[5]</sup>, 该方案使用了与 Agrawal 等人的水印方案类似的原理, 但水印的添加和检测所依赖的密钥可以公开, 因此任何人可以无限次地验证水印的存在, 其功能仅限于版权保护. Sion 等人提出的水印方案利用数据统计信息进行水印添加<sup>[1]</sup>, 但其数据分片的方法无法抵抗专门针对数据库水印的元组添加删除攻击<sup>[6]</sup>. Shehab 等人将数据库水印问题归结为带约束的优化问题, 在 2008 年提出了一种使用基因算法等最优求解方法的水印方案<sup>[6]</sup>. Gross-Amblard 首先提出了一种适用于关系数据和 XML 文档的水印

方案<sup>[7]</sup>. 由于普通数据库水印和指纹式水印方案在安全功能上存在局限性, 因此无法完全解决版权纠纷问题. Huang 等人在 2007 年提出了一种基于可信第三方的水印协议<sup>[8]</sup>, 该协议没有考虑具体的水印添加和检测机制, 仅从协议的角度讨论了纠纷各方的利益保护问题, 讨论了由可信第三方产生、保存并提供交易证据, 使用移动 Agent 实现水印操作等设计思想.

### 3 泄漏源检测方案

本文方案设计了一种复合型数据库水印, 并将其用于机密数据库的泄漏源检测. 一个复合水印 (compound watermark) 是由主水印和副水印构成的联合体, 它由随机分布在关系数据各个元组和属性上的比特标志位组成. 复合型数据库水印的基本原理与普通数据库水印类似, 主水印本身是一个普通数据库水印, 其生成由主密钥控制, 副水印的生成由副密钥和主密钥联合控制. 主水印的检测可以独立进行, 副水印的检测只能依赖于主水印进行. 从本质上讲, 副水印也可以被看作一个普通数据库水印, 但它的存在依赖于主水印, 其生成是伴随着主水印的生成过程进行的, 在没有主密钥的情况下, 副水印无法被检测到. 因此, 副水印既隶属于主水印, 同时又与主水印一同构成了复合水印这样一个整体.

本文提出的泄漏源检测方案使用复合型数据库水印作为信息隐藏的载体. 对于每个共享机密数据库的使用者, 所有者首先将数据上传到水印服务器 (Watermarking Server, WMS), 形成一份临时数据拷贝. WMS 向数据中添加复合水印, 并把带有水印的数据发送给该使用者, 同时删除临时数据拷贝. 水印添加所使用的主密钥由所有者提供, 每个使用者对应不同的主密钥. 副密钥由 WMS 随机生成并随后以加密的形式发送给使用者. 所有者和使用者各自对主密钥和副密钥进行数字签名, 并把签名结果作为非否认证据发送给对方. 一旦泄漏发生, 所有者用各个使用者对应的主密钥对被泄漏的数据进行水印检测, 确定泄漏源并给出可靠性的量化评估结果. 如果某个清白的使用者被指控, 他可以出具副密钥并证明被泄露的数据中不存在与其关联的复合水印. 我们的方案杜绝了所有者在不掌握副密钥的情况下成功伪造水印并指控使用者的可能. 由于双方都对密钥进行了签名, 因此, 任何一方都无法伪造密钥, 从而达到了相互制约的目的, 使双方的利益得到

公平保护. 本章节以下部分给出了基于复合水印的泄漏源检测方案的具体描述.

#### 3.1 基本假设

假设关系  $R$  中用作水印添加的属性包含若干个非关键比特 (LSBs), 如果所有这些 LSBs 被删除或修改, 该属性上的数据将失去使用价值. 但如果只有其中少数比特被修改, 由此造成的数据误差是可接受的, 并且不会影响数据的正常使用<sup>[3]</sup>. 实际应用中有许多数据具有上述特性, 比如一些机械、电子、化学等方面的工业参数数据, 再比如基因表达式之类的自然科学数据等.

我们假设各个参与方都有自己用于数字签名的私钥, 并且对应的公钥证书可以借助于一个 PKI 系统<sup>[9]</sup> 获取和验证. 参与方之间可以使用非对称密钥体制互相认证并协商会话密钥<sup>[10]</sup>, 建立安全的通信信道. 我们假设所使用的 Hash 函数  $H(x)$  是安全的, 该函数具有抗原像攻击和抗碰撞攻击的特性. 给定一个值  $y$ , 寻找原像  $x$  使得  $H(x) = y$  是计算上不可行的. 寻找两个值  $x$  和  $x'$ , 使其满足条件  $x' \neq x$  且  $H(x') = H(x)$  是计算上不可行的. 我们假设使用一个安全的数字签名体制  $\langle \text{Sign}, \text{Verify} \rangle$ . 使用私钥  $sk$  对消息  $m$  应用签名函数  $\text{Sign}()$  可以获得签名  $\sigma = \text{Sign}_{sk}(m)$ . 验证函数  $\text{Verify}_{pk}(m, \sigma)$  使用  $sk$  对应的公钥  $pk$  验证签名的合法性. 在不掌握私钥的前提下伪造一个能够通过验证的签名是计算上不可行的.

#### 3.2 水印服务器

水印服务器 (WMS) 在本文的方案中被用作可信第三方, 其接受的输入是水印密钥和需要添加水印的关系数据, 在使用公开的水印添加算法将水印添加到数据中之后, WMS 将带有水印的数据输出给指定的使用者. WMS 的结构和功能都非常简单, 它不包含用于长期保存数据或密钥的永久性存储设备, 因此 WMS 仅被用来处理数据, WMS 不具有数据保存或密钥保存功能, 其作用仅限于提供水印添加处理服务和数据中继服务, 一旦接收者成功下载了带有水印的数据, WMS 将立即删除该数据并同时销毁所有相关的密钥, 它不保存任何用作纠纷判定证据的交易记录或数字签名结果. 在水印添加过程中, 所使用的密钥以及数据本身的机密性由 WMS 保证, 密钥和数据不会向任何未授权的一方泄漏.

#### 3.3 复合水印添加算法

复合水印添加算法的作用是将水印标志以确定

性的、不可察觉的方式添加到数据中,作为泄漏源检测的依据. 我们定义要添加水印的数据库关系为  $R(P, A_1, \dots, A_v)$ , 其中  $P$  是可以唯一标识元组的主键属性.  $A_1, \dots, A_v$  是关系  $R$  中用来添加水印的  $v$  个属性, 其中属性  $A_i$  中包含有  $\xi_i$  个 LSB. 对于没有主键的关系, 我们使用文献[3]中的方法构建虚拟主键属性. 复合水印的添加需要使用主密钥  $K_M$  和副密钥  $K_S$ . 我们用  $r.X$  表示元组  $r \in R$  中属性  $X$  的值. 算法需要使用一种伪随机数序列发生器  $G$ , 它能确定地产生一系列随机数. 该随机数序列由初始化  $G$  的种子决定. 给定一个种子, 反复执行  $G$  将得到同一个随机数序列. 预测序列中的下一个随机数或通过序列中的随机数推算种子的取值都是计算上不可行的<sup>[11]</sup>.  $G_X::initialize(S_Y)$  表示用种子  $S_Y$  初始化伪随机数序列发生器  $G_X$ ,  $G_X::next$  表示  $G_X$  产生的随机序列中的下一个随机数. 表 1 给出了算法使用的其它符号和标记.

表 1 算法用到的标记和符号	
符号	具体含义
$R$	需要添加水印的数据库关系
$\gamma$	添加水印的元组比例
$v$	关系 $R$ 中属性的数量
$\xi_i$	属性 $A_i$ 中 LSB 的个数
$G_M, G_S$	两个伪随机数序列发生器

算法需要使用函数  $mark(\text{bit\_value } p, \text{value } v, \text{bit\_index } j)$ , 该函数将数值  $v$  的第  $j$  个非关键比特设为  $p$ , 其中  $p \in \{0, 1\}$ . 算法 1 给出了复合水印添加算法的主要步骤: 对于关系中的每个元组, 首先使用元组主键  $r.P$  和主密钥  $K_M$  构造种子并初始化伪随机数序列发生器  $G_M$ .  $G_M$  生成的随机数序列首先决定当前元组是否被用来进行水印添加, 然后选择添加主水印的属性和主水印标志比特的位置, 最后决定主水印标志位的取值. 当序列中的随机数是偶数时, 水印标志位被赋值为 0, 否则为 1. 伪随机数序列发生器  $G_S$  的种子由元组主键、主密钥和副密钥构造.  $G_S$  生成的随机数序列首先被用来选择用于添加副水印的属性和标志比特的位置. 一旦主水印和副水印的标志位发生重叠, 算法将继续利用  $G_S$  的序列重新寻找用于添加副水印的属性和标志比特的位置, 直到找到一个不重叠的位置为止. 最后, 使用  $G_S$  的序列决定副水印标志位的取值.

算法 1. 复合水印添加算法.

$CWGen(R, K_M, K_S)$

//  $R$  是数据库关系,  $K_M$  是主密钥,  $K_S$  是副密钥

```
1. for each tuple  $r \in R$  do
2.    $G_M::initialize(H(r.P | K_M))$ ;
      //使用主键和主密钥构造种子并初始化发生器
3.   if ( $G_M::next \bmod \gamma = 0$ ) then
      //该元组用作水印添加
4.      $im = G_M::next \bmod v$ ; //属性  $A_{im}$  用作水印添加
5.     if ( $r.A_{im}$  is NULL) then continue;
      //遇到空值,跳入下一个循环,处理下一个元组
6.      $jm = G_M::next \bmod \xi_{im}$ ;
      //主水印标志添加到属性  $A_{im}$  的第  $jm$  个比特上
7.      $r.A_{im} = mark(G_M::next \bmod 2, r.A_{im}, jm)$ ;
      //添加主水印标志
8.      $G_S::initialize(H(r.P | K_M | K_S))$ ;
      //使用主、副密钥和主键构造种子并初始化
9.      $is = G_S::next \bmod v$ ; //在属性  $A_{is}$  上添加副水印
10.     $js = G_S::next \bmod \xi_{is}$ ;
      //副水印标志添加到属性  $A_{is}$  的第  $js$  个比特上
11.    while (( $im == is$ ) and ( $jm == js$ )) do
      //保证主水印和副水印不重叠
12.       $is = G_S::next \bmod v$ ;
      //重新选择一个添加副水印的属性
13.       $js = G_S::next \bmod \xi_{is}$ ;
      //重新选择一个比特位置
14.    if ( $r.A_{is}$  is NULL) then continue;
      //遇到空值,跳入下一个循环,处理下一个元组
15.     $r.A_{is} = mark(G_S::next \bmod 2, r.A_{is}, js)$ ;
      //添加副水印标志
```

在上述算法中,  $G_S$  和  $G_M$  为每个元组生成的随机数序列与其它元组的随机数序列是无关的. 因此, 水印不受元组排列顺序的影响. 虽然伪随机数发生器是公开的, 但是, 要正确地 为某个元组生成对应的随机数序列, 必须掌握主密钥和副密钥. 因此, 密钥对  $\langle K_M, K_S \rangle$  与关系数据中的复合水印是一一对应的. 掌握正确的密钥, 是准确定位到每个水印标志位的充分必要条件. 另外, 副水印标志只能出现在有主水印标志的元组中, 这也是主、副水印从属关系的一个重要体现. 当一个元组用来添加主水印的属性值为 NULL 时, 算法直接跳过该元组并转入下一个元组的处理, 当用来添加副水印的属性值为 NULL 时, 同样跳过该元组并转入下一个循环.

3.4 水印检测算法

在本文的方案中, 泄漏源检测的依据是数据中是否存在某个特定的复合水印. 水印检测算法的功能是在水印密钥的控制下对数据进行扫描, 为水印存在性判定提供原始的检测结果. 在水印检测过程中, 判定元组是否被水印使用、选择水印属性和标志

位位置以及决定标志位取值的方法与水印添加过程是完全相同的. 要检测水印的存在, 我们必须拥有用于生产水印的密钥, 以便生成与水印添加时相同的随机数序列. 检测过程检查水印标志位的实际值与预期的计算值是否匹配, 匹配的次数被用于水印存在性的概率分析和评估. 算法 2 和算法 3 分别给出了主水印和副水印的检测算法. 只要拥有主密钥, 主水印是可以被单独检测的, 而副水印的检测需要同时使用主密钥和副密钥. 算法中用到函数  $match(bit\_value\ q, value\ v, bit\_index\ j)$ , 其作用是比较  $v$  的第  $j$  个非关键比特与  $q$  的取值, 如果相同, 则返回 1; 否则返回 0.

### 算法 2. 主水印检测算法.

```
MWDetect( $R, K_M, \&totalCount, \&matchCount$ )
//totalCount: 被检测总数, matchCount: 匹配总数
1.  $totalCount=0; matchCount=0;$  //计数器清零
2. for each tuple  $r \in R$  do
3.  $G_M::initialize(H(r.P|K_M));$ 
   //使用主键和主密钥构造种子并初始化发生器
4. if ( $G_M::next \bmod \gamma = 0$ ) then
   //该元组包含水印标志
5.  $im = G_M::next \bmod \nu;$ 
   //属性  $A_{im}$  包含主水印标志
6. if ( $r.A_{im}$  is NULL) then continue;
   //遇到空值, 跳入下一个循环, 检测下一个元组
7.  $jm = G_M::next \bmod \xi_m;$ 
   //主水印标志在属性  $A_{im}$  的第  $jm$  个比特上
8.  $totalCount = totalCount + 1;$  //检测数总数加 1
9.  $matchCount = matchCount +$ 
    $match(G_M::next \bmod 2, r.A_{im}, jm);$ 
   //更新匹配总数
```

### 算法 3. 副水印检测算法.

```
SWDetect( $R, K_M, K_S, \&totalCount, \&matchCount$ )
1.  $totalCount=0; matchCount=0;$  //计数器清零
2. for each tuple  $r \in R$  do
3.  $G_M::initialize(H(r.P|K_M));$ 
   //使用主键和主密钥构造种子并初始化发生器
4. if ( $G_M::next \bmod \gamma = 0$ ) then
   //该元组包含水印标志
5.  $im = G_M::next \bmod \nu;$  //属性  $A_{im}$  包含主水印标志
6. if ( $r.A_{im}$  is NULL) then continue;
   //遇到空值, 跳入下一个循环, 检测下一个元组
7.  $jm = G_M::next \bmod \xi_m;$ 
   //主水印标志在属性  $A_{im}$  的第  $jm$  个比特上
8.  $G_S::initialize(H(r.P|K_M|K_S));$ 
   //使用主、副密钥和主键构造种子并初始化
9.  $is = G_S::next \bmod \nu;$  //在属性  $A_{is}$  上添加副水印
```

```
10.  $js = G_S::next \bmod \xi_{is};$ 
   //副水印标志添加到属性  $A_{is}$  的第  $js$  个比特上
11. while (( $im == is$ ) and ( $jm == js$ )) do
   //处理水印标志位重叠情况
12.  $is = G_S::next \bmod \nu;$ 
   //重新选择一个添加副水印的属性
13.  $js = G_S::next \bmod \xi_{is};$ 
   //重新选择一个比特位置
14. if ( $r.A_{is}$  is NULL) then continue;
   //遇到空值, 跳入下一个循环, 检测下一个元组
15.  $totalCount = totalCount + 1;$  //检测数总数加 1
16.  $matchCount = matchCount +$ 
    $match(G_S::next \bmod 2, r.A_{is}, js);$ 
   //更新匹配总数
```

算法  $MWDetect$  和  $SWDetect$  在构造随机序列种子和水印标志定位的方法上与  $CWGen$  完全相同. 算法  $SWDetect$  中定位主水印位置是为了处理主水印与副水印标志位重合的情况, 其处理方法与算法  $CWGen$  相同. 在以上两个算法中, 每个包含水印标志元组会使得检测数总数加 1, 每次水印标志位匹配成功,  $match$  函数会返回 1, 从而使得匹配总数加 1;  $match$  函数在匹配失败的情况下返回 0, 匹配总数将不变. 检测算法在遇到空值时直接跳过该元组.

### 3.5 数据分发协议

数据分发协议由数据的所有者、使用者(即数据的接收者)和 WMS 三方共同参与. 协议使用水印添加算法实现复合水印的添加, 同时实现数据分发、密钥传送以及签名证据的交换. 数据分发协议由使用者发起, 使用者向所有者请求数据库关系  $R$ , 并将自己的公钥  $PK_U$  发给所有者. 所有者生成主密钥  $K_M$  并将  $R, K_M$  和  $PK_U$  发送给 WMS. WMS 收到数据后, 随机生成副密钥  $K_S$ , 然后将  $K_S$  和  $K_M$  作为输入参数, 使用算法  $CWGen$  为  $R$  添加复合水印. 水印添加完成后, WMS 将  $K_S$  的 Hash 值作为确认消息发送给所有者. 所有者收到确认消息后, 用自己的私钥  $SK_O$  对  $K_S$  和  $K_M$  的 Hash 值签名, 并将签名与 Hash 值发送给使用者. 使用者随后向 WMS 发送自己的公钥  $PK_U$  和对主、副密钥的数字签名  $Sign_{SK_U}(H(K_M)|H(K_S))$ .  $PK_U$  用于声明身份,  $Sign_{SK_U}(H(K_M)|H(K_S))$  为 WMS 提供身份验证依据. WMS 使用  $PK_U$  验证签名合法性, 从而判定该使用者是否为当前水印数据的合法接收者. 验证通过后, WMS 将带有水印的数据  $R^W$  发送给使用者, 同时发送的还有被使用者

公钥加密的副密钥. 该副密钥只有使用者能够解密, 使用者将其保存并在必要时用作辩护的证据. 协议的最后, WMS 将使用者对主、副密钥签名发送给所有者, 该签名具有双重作用, 即被用来证明使用者身份, 还为所有者提供了不可否认的证据. 协议 1 给出了数据分发协议的具体描述, 其中  $o$  表示数据所有者,  $u$  表示使用者.

**协议 1.** DDP(Data Distribution Protocol).

1.  $u \rightarrow o: PK_U$
2.  $o \rightarrow WMS: K_M, R, PK_U$
3.  $WMS: K_S, CWGen(R, K_M, K_S)$
4.  $WMS \rightarrow o: H(K_S)$
5.  $o \rightarrow u: Sign_{SK_O}(H(K_M) | H(K_S)), H(K_M), H(K_S)$
6.  $u \rightarrow WMS: PK_U, Sign_{SK_U}(H(K_M) | H(K_S))$
7.  $WMS: Ver_{PK_U}(H(K_M) | H(K_S), Sign_{SK_U}(H(K_M) | H(K_S)))$
8.  $WMS \rightarrow u: \{K_S\}_{PK_U}, R^w$
9.  $WMS \rightarrow o: Sign_{SK_U}(H(K_M) | H(K_S)).$

### 3.6 泄漏源的检测过程

所有者通过执行数据分发协议将机密数据库与使用者共享. 在协议的执行过程中, 所有者为每个使用者生成唯一的主密钥, WMS 随机生成唯一的副密钥, 并在这两个密钥的控制下, 使用复合水印添加算法将水印添加到数据中. 通过数据分发协议的执行, 使用者接收到带有水印标志的数据, 同时所有者和使用者各自获得了对方对于该份数据及其密钥的签名证据. 在向多个使用者进行数据分发之后, 所有者保存一份持有机密数据的使用者列表, 列表中同时保存了使用者对应的主密钥及其对两个密钥的签名. 一旦泄漏发生, 所有者用列表中的主密钥逐一一对被泄漏的数据进行水印检测, 通过下文中的概率分析我们可以看到, 如果数据中不包含由某个主密钥生成的水印, 则主水印检测算法返回结果的匹配率约为 50%. 如果检测结果的匹配总数没有落入由总检测数和阈值参数计算得到的可接受区间, 则认定被检测数据中含有由该主密钥生成的水印, 从而说明其对应的使用者是泄漏的源头. 利用概率分析的方法, 我们对检测结果的可靠性进行量化评估, 为争议仲裁机构提供决策参考. 另外, 如果某个清白的使用者遭到错误的指控, 他可以出具副密钥并使用副水印检测算法对被泄露数据进行水印检测, 如果检测算法返回的匹配总数未能落入可接受区间, 则可以认定被泄露的数据中不存在与其关联的复合水印, 从而证明使用者的清白. 第 4 节中给出了两个检

测算法的可接受区间定义和概率分析的推导过程.

## 4 概率分析与量化评估

由于伪随机数序列发生器具有的随机特性, 复合水印生成算法中标志位的取值符合随机掷币模型. 假如我们用水印的  $W$  的密钥检测一份不包含  $W$  的数据, 则匹配总数服从概率参数  $p=1/2$  的贝努利分布. 我们将会以很高的概率获得大约 50% 的匹配率. 式(1)给出的是匹配总数为  $m$  的发生概率, 其中  $\eta$  为检测总数.

$$Prob[m \text{ matches}] = b(m; \eta, 1/2) = \left( \frac{\eta!}{m!(\eta-m)!} \right) (1/2)^\eta \quad (1)$$

用算法  $MWDetect$  返回的总检测数和匹配总数对式(1)进行计算, 如果结果几乎为零, 那么该数据不包含  $W$  的假设被推翻, 从而说明  $W$  是存在的. 上述概率分布的中心项为  $b(\lfloor (\eta+1)/2 \rfloor; \eta, 1/2)$ , 当  $\eta$  为奇数时, 中心项为  $b((\eta+1)/2; \eta, 1/2)$  和  $b((\eta-1)/2; \eta, 1/2)$  两项, 当  $\eta$  为偶数时, 中心项为  $b(\eta/2; \eta, 1/2)$  一项. 以下我们只讨论  $\eta$  为偶数的情况,  $\eta$  为奇数的计算方法与之类似. 我们定义  $\lambda$  为可接受的概率区间半径阈值. 如果数据中不包含水印, 则匹配总数  $m$  的取值落入区间  $[\eta/2 - \lambda, \eta/2 + \lambda]$  的概率为

$$\begin{aligned} Prob[|m - \eta/2| \leq \lambda] &= \sum_{i=(\eta/2)-\lambda}^{(\eta/2)+\lambda} b(i; \eta, 1/2) \\ &= \sum_{i=(\eta/2)-\lambda}^{(\eta/2)+\lambda} \frac{\eta!}{i!(\eta-i)!} (1/2)^\eta \quad (2) \end{aligned}$$

随着  $\lambda$  的增大,  $Prob[|m - \eta/2| \leq \lambda]$  趋近于 1, 对于不同的总检测数,  $Prob[|m - \eta/2| \leq \lambda]$  趋近于 1 的速度不同, 它与  $\lambda$  的关系如图 1 所示.

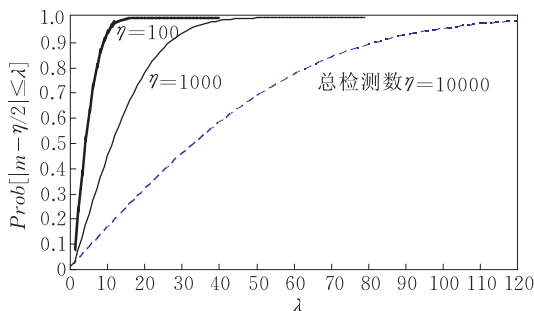


图 1 匹配总数  $m$  落入区间  $[\eta/2 - \lambda, \eta/2 + \lambda]$  的概率与阈值  $\lambda$  的关系

当  $Prob[|m - \eta/2| \leq \lambda]$  趋近于 1 时, 实际匹配总数未落入可接受区间的发生概率将趋近于 0, 我

们定义此小概率事件的概率为  $\epsilon$ . 选取适当的  $\lambda$  将获得足够小的  $\epsilon$ , 如果检测结果说明水印存在, 我们将  $1-\epsilon$  作为可靠性量化评估的结果. 因此, 水印存在性检测结果和所使用的参数共同决定了结果的可靠性.

由于副水印与主水印存在的伴随关系, 同一份数据的副水印匹配总数应当与主水印的匹配总数相近. 唯一可能造成差别的是数据中存在的空值属性. 从 *MWDetect* 和 *SWDetect* 两个算法可以看出, 如果在某个含有主水印标志的元组中, 副水印所在的属性值为 NULL, 则该元组会造成 *SWDetect* 算法的匹配总数  $mS$  比 *MWDetect* 算法的匹配总数  $mM$  少 1. 对于随机选取的一个元组  $r$ , 定义概率  $p_{\text{NULL}} = \text{Prob}[r.A_{\text{random}} \text{ is NULL} | r.A_{\text{random}} \text{ 是 } r \text{ 中随机选择的一个属性}]$ .  $p_{\text{NULL}}$  是一个平均概率, 由于随机序列的随机特性, 各个候选属性被选中用作副水印添加的概率是相等的. 若关系  $R$  的  $v$  个属性中有  $k$  个允许空值, 各个属性上的空值比例分别为  $1/L_i (1 \leq i \leq k)$ , 且空值随机分布, 则有  $p_{\text{NULL}} = (1/v) \sum 1/L_i (1 \leq i \leq k)$ . 在已知主水印存在的前提下, 假设副水印存在, 则使用正确的副密钥进行检测后, 所得的匹配总数  $mS$  服从概率参数  $p = 1 - p_{\text{NULL}}$  的贝努利分布. 令  $mM$  为主水印检测匹配总数, 则  $\text{Prob}[mS \text{ matches}] = b(mS; mM, 1 - p_{\text{NULL}})$ , 中心项为  $b(\lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor; mM, 1 - p_{\text{NULL}})$ . 我们定义可接受的概率区间半径阈值为  $\sigma$ , 则  $mS$  落入  $[\lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor - \sigma, \lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor + \sigma]$  的概率为

$$\begin{aligned} & \text{Prob}[\lfloor mS - \lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor \rfloor \leq \sigma] \\ &= \sum_{i=\lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor - \sigma}^{\lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor + \sigma} b(i; mM, 1 - p_{\text{NULL}}) \\ &= \sum_{i=\lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor - \sigma}^{\lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor + \sigma} \frac{mM!}{i!(mM-i)!} \cdot (1 - p_{\text{NULL}})^i (p_{\text{NULL}})^{mM-i} \end{aligned} \quad (3)$$

根据  $mM$  和  $p_{\text{NULL}}$  的值, 我们选择适当的  $\sigma$ , 使式(3)中的概率趋近于 1, 匹配总数未落入区间  $[\lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor - \sigma, \lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor + \sigma]$  的发生概率则极小, 而一旦出现这样的情况, 则说明副水印存在的假设不成立, 从而达到了使用正确的副密钥证明副水印不存在的目的. 例如: 某个关系具有 8 个候选属性, 若其中 2 个属性包含空值, 空值比例均为 10%, 则可计算  $p_{\text{NULL}} = 0.025$ . 当  $mM = 1000$  时, 计算得到  $\lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor = 975$ . 取  $\sigma = 20$ , 则  $\text{Prob}[\lfloor mS - \lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor \rfloor \leq \sigma] \approx$

$0.9998$ , 随着  $\sigma$  取值的增加,  $\text{Prob}[\lfloor mS - \lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor \rfloor \leq \sigma]$  趋近于 1. 因此, 对于副水印不存在的检测结果, 我们使用  $\text{Prob}[\lfloor mS - \lfloor (mM+1)(1 - p_{\text{NULL}}) \rfloor \rfloor \leq \sigma]$  作为其可靠性的量化评估值. 也就是说, 结果的可靠性同样可由结果数据本身及所使用的参数进行评估.

## 5 安全性分析与比较

对于既需要公平保护双方利益, 同时又要防止双方作弊行为的问题, 引入第三方是最好的解决方法. 但第三方本身实现难度以及安全性会对整个方案的可用性和安全性造成影响, 与文献[7]中的可信第三方相比, WMS 仅在水印添加过程中被使用, WMS 不存储交易记录或签名证据, 不提供纠纷仲裁依据, 功能单一, 安全要求低, 容易实现. WMS 不提供持久性存储的特性, 大大降低了针对数据或密钥的潜在威胁. WMS 通过使用非对称密钥体制的身份认证方式, 将水印数据的安全规约到使用者私钥的安全. WMS 本身的作用仅限于数据处理和安全的数据中继. 整个方案对 WMS 的依赖程度远远低于文献[7]中对可信第三方的依赖, 因此具有更高的实用价值.

本文方案与指纹式水印方案相比的优越性主要表现在能够同时保护数据所有者和使用者双方的利益, 首先是能够防止所有者错误地、甚至恶意指控清白的使用者. 在指纹式水印方案中, 所有者掌握水印密钥和指纹字串, 因此, 所有者可以任意地向数据中添加水印, 而使用者无法验证水印的正确性. 如果使用者获知了检测水印需要使用的密钥, 就能够将水印从数据中清除或者向数据中添加另外的指纹, 因为检测水印需要使用的密钥与生成水印的密钥相同. 在本文的方案中, 所有者始终不掌握副密钥, 副密钥由 WMS 随机生产, 并由使用者最终保存. 所有者即使自行添加主水印, 也无法生成正确的副水印. 安全的哈希函数和签名体制保证了所有者无法从 DDP 协议中获得副密钥. 当所有者出示主密钥指控使用者时, 使用者出示其副密钥并检测副水印的存在, 如果与所有者声称的主水印对应的副水印不存在, 则可以证明自己是清白的. 这就有效地防止了所有者使用错误的或伪造的密钥指控清白的使用者. 双方利益保护的另一方面是泄密行为的不可否认性. 在 DDP 协议中, 使用者只有使用其私钥对副密钥进行正确的签名才能从 WMS 处获得数据. 由于



所有者持有该签名,泄密的使用者无法伪造副密钥来证明数据中不存在对应的副水印,因此也就无法否认其泄密的行为。

6 实验结果

我们将复合水印添加和检测算法的执行时间与普通的数据库读写操作进行了对比实验,衡量水印添加和检测算法的计算性能和执行代价.实验数据来源是加州大学 KDD 数据中心的森林覆盖类型数据集<sup>[12]</sup>,该数据集是有一个包含 61 个属性,581012 条记录的数据表.我们使用 Microsoft SQL Server 2005 数据库,并在数据导入过程中对该表进行改造,添加了名为 ID 的主键属性.我们选取前 10 个整数类型的属性用于水印添加,并从原始数据中选取不同数量的元组进行 6 组实验,使用的水印百分比参数  $\gamma=4$ ,LSB 数量参数  $\xi=5$ .我们首先将水印添加算法的执行时间与普通数据库读写操作的执行时间进行对比.对于包含  $\eta$  条元组的关系,我们将水印添加执行时间与读取  $\eta$  条元组并改写  $\eta/4$  条元组的操作时间进行比较.每组实验重复 5 次,图 2 所示的是实验结果的平均值.图中  $x$  轴表示要添加水印的关系所含的元组数, $y$  轴表示执行时间。

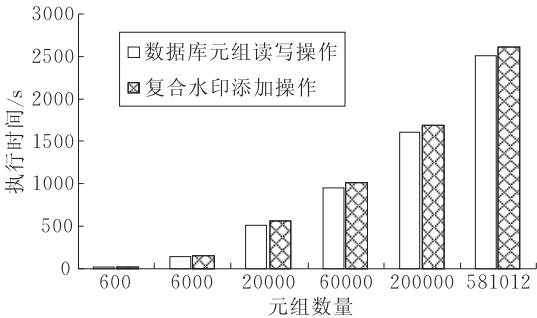


图 2 复合水印添加与普通数据库读写操作执行时间的比较

水印检测算法不涉及数据库写操作,因此我们将检测算法的执行时间与普通数据库读操作进行比较.图 3 所示的是重复 10 次实验结果的平均值.通过对多次实验的结果进行平均值计算,我们还得到如下结果:与普通数据库操作相比,复合水印添加算法造成的额外执行时间增长约为 7.2%,复合水印检测算法造成的额外执行时间增长约为 11.2%.这些的时间增长主要是由算法中随机数生成和计算水印标志位等操作造成的,但其平均增长比例都比较小,可以满足绝大多数应用场景的要求,因此该方案

在性能方面具有较强的实用性优势。

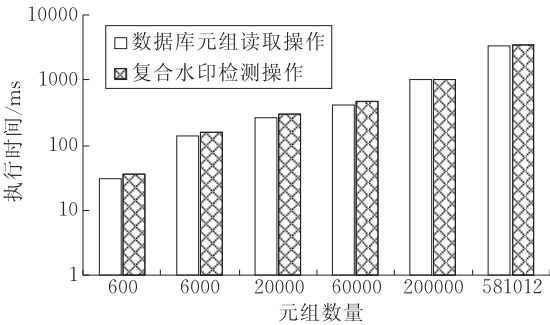


图 3 复合水印检测与普通数据库读操作执行时间的比较

7 结束语

机密数据库泄漏源检测的主要任务是在泄密事故发生后,对泄漏源头进行检测和鉴定,并使用技术手段为责任认定和事故追查工作提供证据.现有的数据库版权保护和盗版源追查技术在公平性保护方面具有较大缺陷,不能被直接用于泄漏源检测.本文对 Agrawal 等人的数据库水印方案进行改进和扩展,提出一种复合型数据库水印,在此基础上给出了完整的泄漏源检测及可靠性量化评估方案.该方案克服了现有技术的缺陷,能够在准确检测泄漏源的同时,对所有者的行为进行限制,防止其使用错误的或者伪造的检测结果对清白的使用者进行指控,从而使双方的利益得到公平的保护.本文给出了复合水印的添加与检测算法以及水印添加与数据分发协议,同时给出了检测结果可靠性量化评估的概率分析方法.实验分析表明,复合型数据库水印算法的执行效率较高,具有较强的实用性。

参 考 文 献

[1] Sion R, Atallah M, Prabhakar S. Rights protection for relational data. *IEEE Transactions on Knowledge and Data Engineering*, 2004, 16(12): 1509-1525

[2] Swarup V, Li Y, Jajodia S. Fingerprinting relational databases: Schemes and specialties. *IEEE Transactions on Dependable and Secure Computing*, 2005, 2(1): 34-45

[3] Agrawal R, Kiernan J. Watermarking relational databases: Framework, algorithms and analysis. *The International Journal on Very Large Data Bases*, 2003, 12(2): 157-169

[4] Guo F, Wang J M, Li D Y. Fingerprinting relational databases//*Proceedings of the 2006 ACM Symposium on Applied computing*. Dijon, France, 2006: 487-492



[5]

Li Y, Deng R H. Publicly verifiable ownership protection for relational databases//Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security. New York, NY, USA: ACM, 2006: 78-89

[6]

Shehab M, Bertino E, Ghafoor A. Watermarking relational databases using optimization-based techniques. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(1): 1041-4347

[7]

Gross-Amblard D. Query-preserving watermarking of relational databases and XML documents//Proceedings of the 2003 ACM Symposium on Principles of Database Systems. San Diego, California, USA, 2003: 191-201

[8]

Huang Min, Cao Jia-Heng, Peng Zhi-Yong. Database watermark security protocol based on trusted third party. Journal of Wuhan University: Natural Science Edition, 2007, 53(3): 297-300(in Chinese)  
(黄敏, 曹加恒, 彭智勇. 基于可信第三方的数据库水印安全协议. 武汉大学学报(理学版), 2007, 53(3): 297-300)

[9]

Housley R, Ford W, Polk W, Solo D. Internet X. 509 public key infrastructure certificate and crl profile. July 2, 2005

[10]

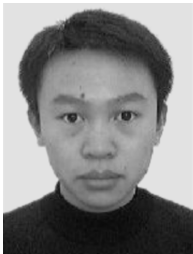
Standard Specifications for Public Key Cryptography, IEEE Std. 2000, 1363-2000

[11]

Schneier B. Applied Cryptography: Protocols, Algorithms and Source Code in C. Hoboken, NJ, USA: Wiley & Sons, 1996

[12]

Craver S, Memon N, Yeo B L, Yeung M M. Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications. IEEE Journal on Selected Areas in Communications, 1998, 16(4): 573-586



**XIAN He-Qun**, born in 1979, Ph.D. candidate. His main research interests include network and information system security.

Background

This paper presents a novel watermark based scheme to solve the problem of leakage identification for confidential databases, which is also a small-scaled pirate identification problem. When a security breach occurs to a group sharing confidential relational data, the proposed scheme can be used to identify the leakage source and provide non-repudiation evidence. Current solutions for database copyright protection and pirate identification cannot be directly applied to the leakage identification problem, because in most of those schemes, the data owner controls the key and the whole watermark generation process. So the other users are prone to false accusations. In this paper, a compound watermark technique and a data distribution protocol are presented, which provide fair protection for both the owner and the users

in the process of leakage identification. Quantitative evaluation methods are introduced to provide probabilistic analyses of the identification results.

The work in this paper is supported by the National High-Tech Research and Development Plan of China under grant Nos.2007AA120404, 2007AA120405. The projects are expected to solve challenging problems and to provide key techniques in building a highly trusted geographical and spatial database system. This paper focuses on the research of leakage identification and quantitative evaluation for databases containing confidential information. Several other papers on related works in the projects have been published on international journals or in proceedings of international conferences.