一种合谋安全的数据库指纹编码与盗版追踪算法*)

朱 勤[1,2] 陈继红[1] 乐嘉锦[2]

(南通大学计算机科学与技术学院 江苏南通 226019)[1] (东华大学计算机科学与技术学院 上海 200051)[2]

摘 要 在将数据库作为软件产品发行的应用场合,需要有相应的安全机制对叛逆用户的盗版行为进行约束。本文提出了一种基于数字指纹的关系数据库盗版追踪解决方案。以混沌二值序列将版权水印与用户指纹组合而成数据库指纹,在密钥的控制下嵌入数据库。基于混沌二值序列的随机性进行指纹提取与叛逆追踪。方案具有较高的合谋安全性,同时降低了指纹检测与叛逆追踪的运算复杂度。文中描述了数据库指纹编码与嵌入、指纹检测与提取算法,分析了算法的鲁棒性与叛逆追踪能力,并进行了实验验证。

关键词 关系数据库,数字指纹,合谋安全,叛逆追踪

Novel Algorithm of Collusion Secure Fingerprint Coding and Traitor Tracing for Relational Database

ZHU Qin[1,2] CHEN Ji-hong[1] LE Jia-jin[2]

(College of Computer Science and Technology, Nantong University, Nantong 226019, China)[1]

(School of Computer Science and Technology, Donghua University, Shanghai 200051, China)[2]

Abstract It is necessary to provide secure mechanism to resist the piracy from the renegade users when databases are distributed as software products. In this paper, a scheme of traitor tracing for rational database based on digital fingerprinting is presented. In the proposed scheme, the database fingerprints are composed of copyright watermarks and user's fingerprints by using the chaos binary sequence, and are embedded into the database under the control of the secret key. The fingerprints are extracted and the traitors are traced based on the randomicity of the chaos binary sequence. This scheme is collision secure and computationally simplified. The algorithms of the fingerprint coding, embedment, detection and extraction are described. The robustness and the traitor tracing performance of the algorithms are analyzed theoretically as well as validated practically by experiments.

Keywords Relational database, Digital fingerprinting, Collusion security, Traitor tracing

1 引言

目前,有不少应用系统将数据库作为软件的一部分分发给用户。如在车载 GPS 导航系统中,一般就含有地理信息数据库。再如,专注于地图搜索引擎开发的公司,一般会将地图数据库分发给各大网络搜索引擎网站,如百度、Google、Yahoo 等。数据发行者希望不但能对盗版数据库进行版权认证,而且能约束合谋盗版行为,鉴别出盗版来源,即能够根据盗版拷贝追踪到实施叛逆行为的协议用户。这就要求在载体数据中能对每一个协议用户进行标识,一旦发现盗版数据,能够根据提取出来的唯一性标识信息追踪盗版来源。具有抗合谋能力的数据库指纹技术能够满足这种安全需求。

数字指纹(Digital Fingerprinting)技术向被分发的每一份数字拷贝中嵌入与具体用户相对应的标志性识别代码——数字指纹,使得该拷贝是唯一的。当发现数字作品被非法传播时,可以通过提取盗版拷贝中的数字指纹,确定非法复制的来源,实现对协议用户的叛逆追踪。

基于数字指纹的数据库盗版追踪的工作原理如图 1 所示。

图 1 中,数据发行者将数据分发给三个合法用户 A,B,C,每个用户数据库中均含有各自的数字指纹。当发现盗版数据时,数据发行者启动数字指纹提取过程,追踪提供盗版数据的叛逆用户。

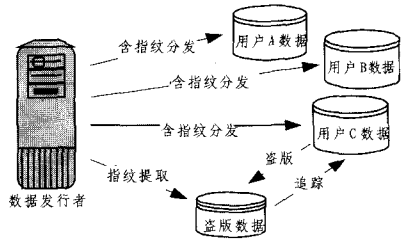

图 1 基于数字指纹的数据库盗版追踪工作原理

一般地,数字指纹具有隐蔽性、鲁棒性、确定性、合谋安全性等特征。区别于多媒体作品数字指纹,数据库指纹技术还应适应数据库频繁更新的特点,实现指纹的动态同步。同时,在对多媒体水印或指纹的攻击中不常见的直接对数据的增、删、改,在数据库中则被视为常规操作,这对数据库指纹的鲁棒性提出了更高的要求,也成为数据库指纹算法设计的一个难点。

本文提出了一种合谋安全的关系数据库指纹编码方案,设计了相应的指纹生成与嵌入、指纹检测及叛逆追踪算法。从理论上分析了数据库指纹的合谋安全性与叛逆追踪能力,

*)江苏省社会发展科技计划(BS2006524);江苏省高校自然科学研究计划(05KJD520168)。朱 勤 博士生,副教授,研究领域为数据库安全、数据库与信息系统;陈继红 副教授,研究领域为数据库与信息系统;乐嘉锦 教授,博士生导师,研究领域为数据库与数据仓库、软件工程技术。

并进行了实验验证。

2 相关工作

2.1 数字指纹编码

数字指纹编码主要解决用户信息的编码和跟踪问题，并针对合谋攻击增强指纹的合谋安全能力。早期的数字指纹研究集中于抗合谋指纹编码的理论研究。G. Blakley 等在 1985 年的 CRYPTO 会议上发表的论文《Fingerprinting Long Forgiving Messages》，是最早阐述抗合谋的数字指纹概念的文献之一。数字水印专家 I. Cox 等将数字指纹理论与数字水印相结合，最早提出了基于随机序列的连续指纹编码，并讨论了其合谋安全性能[1]。

离散指纹一般建立在代数结构编码基础上。最具代表性的离散数字指纹编码算法由 D. Boneh 和 J. Shaw 提出[2]。该方案建立在标记假设(Marking Assumption)的基础上，即假设参与合谋的敌手只能修改彼此作品码字中有不同值的对应位。同时，定义了一个对数长度 C-安全编码(Logarithmic Length C-secure Code)，给出了一种指纹码字长度与用户数目的对数及合谋容忍尺寸的四次方成正比的指纹编码方案，能够以较高的概率追踪到至少一名合谋者。该方案提出的抗合谋指纹编码及叛逆追踪的基本算法思想，为目前大多数数字指纹编码方案所沿用。

J. Ferrer 等在 2000 年提出了一种建立在鲁棒水印算法上的抗合谋编码，使用对偶二元汉明码来抵抗两个用户的合谋攻击[3]。F. Zane 提出了一种双层的 C-安全编码，将内层的 Cox 水印编码与外层的纠错码结合起来，并且采用码间最小距离以保持水印的抗合谋性能[4]。W. Trappe 等利用特殊的组合，基于 BIBD(平衡非完全区组设计)方法，提出了一种抗合谋指纹编码方案[5]。

目前，国内外学者对数字指纹编码的研究逐渐深入，研究热点主要是在码字长度、合谋容忍尺寸及算法效率之间寻找平衡点。为了提高嵌入算法的效率，需要在一定的合谋容忍尺寸下，减少用户的码长并尽量放宽标记假设。同时，改进追踪算法的效率也是当前数字指纹编码的研究热点。

2.2 数据库指纹

2004 年以来，一些学者开始关注关系数据库指纹技术的研究，取得了一些初步成果。

日本学者 K. Yoshioka 等在 2004 年首次报告了一种关系数据库数字指纹解决方案[6]。该方案在载体数据的每个分发拷贝中插入一些不同的“隐秘记录”(Stealth Records)作为数字指纹，隐秘记录的产生满足数据可用性约束条件，并使用二进制合谋安全编码实现对盗版者的追踪。法国学者 C. Constantin 等研究了保持数据完整性约束条件下适用于数据库与 XML 文档的数字指纹算法[7]，将载体数据的可用性条件抽象为一组约束规则，采用整数线性规划(Integer Linear Program，简称 ILP)方法来搜索最优的指纹嵌入载体，结合合谋安全编码，实现了对静态载体数据的指纹嵌入与盗版追踪。

迄今为止最为完整地讨论数据库指纹算法的文献，是新加坡学者 Y. Li 等在 2005 年发表的论文《Fingerprinting Relational Databases -Schemes and Specialties》[8]。该文借助 R. Agrawal 的关系数据库水印算法[9]来确定载体数据中的水印标记位置及标记值，由发行者密钥及用户序列号产生抗合谋指纹编码，将水印信号与用户指纹进行异或运算后嵌入关系数据库。该算法体现了一定的抗合谋性能，并且实现了一定条件下的数据库指纹增量更新。

从目前的研究现状来看，目前国内外对数据库指纹技术的研究还处于起步阶段。现有的数据库指纹方案仅限于对称指纹机制的研究。在系统的安全性与可用性之间寻找平衡点是数据库指纹技术的关键。

3 合谋安全的数据库指纹算法

3.1 算法框架

本文提出的关系数据库指纹算法的原理框架如图 2 所示。

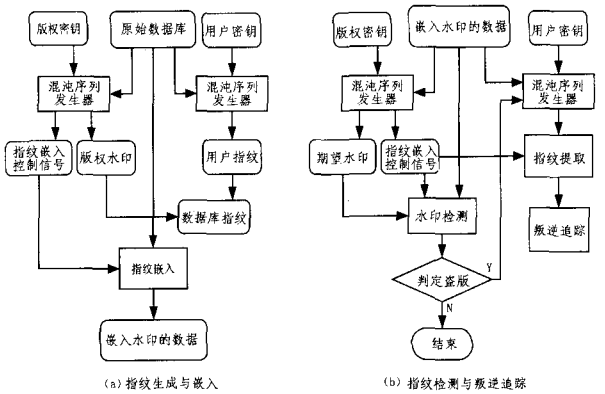

图 2 数据库指纹算法框架

以版权密钥生成混沌二值序列，分别作为版权水印信号及指纹嵌入的控制信号。以用户密钥生成混沌二值序列，作为用户指纹。版权水印与用户指纹组合而成数据库指纹，在指纹控制信号的控制下嵌入数据库。指纹检测分为两个过程：版权判定与指纹提取。对怀疑为盗版的数据库，首先使用版权密钥进行水印检测。然后对判定为盗版的数据库，使用用户密钥提取用户指纹，根据叛逆追踪算法追踪盗版来源。

3.2 合谋安全指纹编码

定义 1 令 Σ' 表示字符表 Σ 上长为 l 的字符集合，(l,n) 指纹码是指当函数 $E(u)$ 将编号 $u(1\leqslant u\leqslant n)$ 映射到 Σ' 中 n 个序列构成的码字集合。

定义 2(标记假设，Marking Assumption)[2] 设 $\Gamma=\{w^{(1)},w^{(2)},\cdots,w^{(n)}\}$ 是一个 (l,n) 指纹码，并且 $C=\{u_1,u_2,\cdots,u_c\}$ 是 c 个用户的合谋集合。若 C 中所有码字在位置 i 的值相同，即 $w_i^{(u1)}=w_i^{(u2)}=\cdots=w_i^{(uc)}$，则称位置 i 为不可侦测位且不能被修改，而只有非不可侦测位才能被修改。定义合谋集 C 所能产生码字的可行集为：

$$\Gamma(C)=\{(x_1,x_2,\cdots,x_l)\in\Sigma^l\mid x_j\in W_j,1\leqslant j\leqslant l\} \tag{1}$$

其中，

$$W_j=\begin{cases}\{w^{(ul)}{}_j\},w_j^{(ul)}=w_j^{(u2)}=\cdots=w_j^{(uc)}\\ \{w_j^{(ui)}\mid 1\leqslant i\leqslant c\}\cup\{\perp\},\text{otherwise}\end{cases} \tag{2}$$

其中，$\perp$ 为擦除标记。

由标记假设可知，合谋的用户通过对比他们的拷贝，只可能在拷贝相异之处(可侦测位)发现部分指纹，而对于其他任何位置，合谋用户侦测不到指纹。在不破坏数据可用性的情况下，合谋用户无法改变不可侦测位的指纹信息，这就为叛逆追踪提供了依据。

我们选择文献[10]提出的二进制合谋安全指纹作为数据库指纹的基本编码，与数据库版权水印相结合，获得了一种适用于数据库的指纹编码。

文献[10]的指纹编码基于标记假设，通过使用伪随机序列对指纹比特的重复嵌入进行控制，实现了一定错误概率下的抗合谋能力。该编码的算法思想简述如下：

(1)每一个用户被分配一个长为 l 的随机二进制码字，每个比特重复 m 次，每个用户的码字的总长度为 $L = l \times m$。称每个码元重复嵌入的 m 个比特为一个“块”。

(2)数据发行者为每个用户选择一个随机种子数，用伪随机数发生器生成的伪随机序列去控制重复的比特中哪些比特取反后嵌入。

(3)进行叛逆追踪时，数据发行者将非法拷贝中的指纹位置的比特提取出来，并用相应的取反控制伪随机序列进行还原。若该用户参与了合谋，因为非法拷贝中的比特含有合谋者在不可侦测位的信息，则在还原后的块中将会出现“0”占优势或者“1”占优势的情况。而对于无辜的用户，只要所使用的伪随机序列具有很好的随机性且 m 取值充分大，则在还原后的块中，“0”，“1”将是均衡的。

上述编码算法在适当的合谋尺寸下，能够对非法分发者进行有效跟踪。其主要不足是：在进行指纹检测时，需要对所有被检测拷贝遍历所有用户的密钥进行指纹提取，运算开销巨大。作为改进，我们将鲁棒水印算法引入指纹编码，利用两者的优点组合成兼具快速检测、合谋安全及叛逆追踪能力的数据库指纹编码。

定义 3 设数据发行者设定的数据库版权水印为二进制序列 $w[l_1]$，用户的个人指纹编码为二进制序列 $pp[l_2]$，定义向该用户发行的数据库拷贝的基本指纹码为

$$p[l]=w[l_1] \| pp[l_2] \quad l=l_1+l_2 \tag{3}$$

其中，$\|$ 表示位连接运算。

定义 4 将数据库拷贝的基本指纹码 $P[l]$ 重复 m 次，生成指纹码矩阵 $P[m,l]$。以长度为 m 的伪随机序列 $r[m]$ 作为指纹取反控制序列，对于 $r[m]$ 中为 1 的比特，基本指纹码 $P[l]$ 中对应位置的比特取反后作为指纹码矩阵 $P[m,l]$ 的值，即

$$P(i,j)=p(j) \oplus r(i) \quad 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant l \tag{4}$$

在本编码方案中，版权水印是所有数据库拷贝指纹中的共有部分。根据标记假设，对于合谋攻击者而言，版权水印的嵌入位置均是不可侦测位。因此，该方案比常用的 C-安全指纹编码具有更高的不可侦测位比例，从而具有更高的抗合谋攻击的能力。

另一方面，由于版权水印基于数据发行者的版权密钥产生，而与用户密钥无关，因此，当对可疑拷贝进行检测时，只需利用版权密钥运行鲁棒水印算法即可进行版权认证。对于在版权认证中确定为盗版的数据库拷贝，基于不同的用户密钥运行指纹检测及叛逆追踪算法。这种分步检测大大降低了指纹提取过程的运算复杂度。

需要说明的是，上述示例为了简化描述，取了较小的编码长度。当用户数为 n 时，基本指纹码的长度的基本条件为：$l > \log_2 n$。由于本编码方案本质上是一种随机二进制编码，因此在实际应用中，为了获得更好的随机性与均衡性，基本指纹码及指纹取反控制序列的长度，应在满足数据可用性的前提下尽量取大值。

3.3 指纹生成与嵌入

本算法由数据发行者的版权密钥及数据库元组主键共同确定指纹位置，由用户密钥生成用户指纹，并与版权水印组合成数据库指纹。由于所有数据库拷贝使用相同的版权密钥，因此各数据库拷贝中不同的数据库指纹具有相同的指纹嵌入位置，满足标记假设。通过修改元组数值型属性值 10 进制低位数字的奇偶性，在满足一定精度与可用性要求的前提下，获得指纹载体信道。

定义 5 设数据库关系为 $R(P, A_1, \cdots A_j, \cdots A_v)$，其中，$P$ 为主键，$A_j(1 \leqslant j \leqslant v)$ 为 R 中属性，$r_i(1 \leqslant i \leqslant n)$ 为 R 中元组，$r_i.A_j$ 为元组 r_i 中属性 A_j 的取值。若 $A_j(1 \leqslant j \leqslant v)$ 为 R 中的数值型属性，且 A_j 的取值精度在数值低位存在一定的冗余，称 A_j 为候选属性。

为简化问题的描述，以下假定 R 中的所有属性 $A_j(1 \leqslant j \leqslant v)$ 均为整数型候选属性。同时，假定 R 中主键 P 是不可修改的。

定义 6 称候选属性 $A_j(1 \leqslant j \leqslant v)$ 中存在精度冗余的 10 进制低位为候选位。设 A_j 中候选位位数为 ξ_j，各候选位由高到低分别为 $d_{\xi j}, d_{\xi j-1}, \cdots, d_1$。

由以上定义可知，数据库关系 $R(P, A_1, \cdots, A_j, \cdots, A_v)$ 中实际可用于嵌入水印的载体，是候选属性 $A_j(1 \leqslant j \leqslant v)$ 的 10 进制候选位 $d_{kj}(1 \leqslant k_j \leqslant \xi_j)$。

定义 7 候选属性 $A_j(1 \leqslant j \leqslant v)$ 在可用性范围内所能容忍的误差百分比称为 A_j 的允许误差，以 δ_j 表示。

规则 1 设二值序偶 (c_1, c) 为水印及嵌入控制信号，对于候选位 d，其取值 $VL(d)$ 以下式所示规则嵌入水印：

$$VL(d)=\begin{cases} VL(d)-1, & (w_1,w_2)=(0,0) \\ \quad \text{且 } VL(d) \text{为奇数} & \\ VL(d)+1, & (w_1,w_2)=(0,1) \\ \quad \text{且 } VL(d) \text{为不是 9 的奇数} & \\ VL(d)-1, & (w_1,w_2)=(1,0) \\ \quad \text{且 } VL(d) \text{为不是 0 的偶数} & \\ VL(d)+1, & (w_1,w_2)=(1,1) \\ \quad \text{且 } VL(d) \text{为偶数} & \end{cases} \tag{5}$$

算法 1 指纹生成与嵌入(以对用户 u 发行的数据库拷贝为例)

```
1)输入版权密钥 Kc
2)以 NRM (Kc)为初值,生成 Logistic 混沌二值序列 c[l1 + l + m]
3)w[l1] = c[1,l1]
  // 生成版权水印信号
4)e[l] = c[l1+1,l1 + l]
  // 生成指纹嵌入策略控制信号
5)r[m] = c[l1 + l +1,l1 + l + m]
  // 生成指纹取反控制信号
6)输入用户密钥 Ku
7)以 NRM (Ku)为初值,生成 Logistic 混沌二值序列 pp[l2]
  // 生成用户指纹码
8)p[l] = { w[l1],pp[l2] }
  // 生成数据库拷贝基本指纹码
9)对 R 中每个元组 ri(1 ≤ i ≤ n),重复:
10)  if LGS (NRM (Kc|| Pi) ) mod γ = 0,then
11)    k = NXT (LGS (NRM (Kc|| Pi))) mod l +1
       // 确定基本指纹码序列中的标记位
12)    q = NXT (LGS (NRM (Kc|| Pi))) mod m +1
       // 确定指纹取反控制序列中的对应位
13)    p(k) =p(k)⊕r (q)
       // 根据指纹取反控制信号调整基本指纹码的嵌入值
14)    j = NXT (LGS (NRM (Kc|| Pi))) mod v +1
       // 确定被标记的元组属性
15)    d = NXT (LGS (NRM (Kc|| Pi))) mod ξ (j) +1
       // 确定被标记的候选位
16)    if ri. Aj * δ ( j ) ≥ 10^(d-1),then
           根据规则 1,由 (P(k),e (k) ) 确定候选位 d 的取值,并
       更新 ri. Aj
```

算法 1 中，l_1 为版权水印长度，l_2 为用户指纹长度，m 为指纹取反控制序列长度，数据库拷贝指纹码长度 $l = l_1 + l_2$。γ 为元组标记间隔值，$\xi[v]$ 为各候选属性的候选位位数数组，$\delta[v]$ 为各候选属性允许误差数组。数组 $\xi[v]$，$\delta[v]$ 及标记间隔值 γ 与版权密钥 K_c 及用户密钥 K_u，共同组成了输入参

数，形成系统的完整密钥。函数 NMR（x）对 x 进行归一化处理，$0<$ NMR（x）<1。函数 LGS（x）以 x 作为 Logistic 方程的初值产生随机混沌序列，并去掉小数点，把序列值作为整数处理。函数 NXT（LGS（x））取混沌序列中的下一个值。Logistic 混沌序列的产生算法参见文献[11]。对混沌序列中小于 0.5 的值取 0，反之取 1，生成伪随机二值序列。

以版权密钥 K_c 生成混沌二值序列，取混沌序列中的子序列，分别作为版权水印信号 $w[l_1]$、指纹嵌入策略控制信号 $e[l]$，及指纹取反控制信号 $r[m]$。以用户密钥 K_u 生成混沌二值序列，作为用户个人指纹 $pp[l_2]$。版权水印 $w[l_1]$与用户个人指纹 $pp[l_2]$位连接而成数据库拷贝基本指纹码 $p[l]$。

指纹嵌入过程中，由版权密钥 K_c 和当前元组主键 P_i 共同确定要嵌入的基本指纹码 $p[l]$中的相应位及嵌入位置，根据指纹取反控制信号 $r[m]$ 调整基本指纹码嵌入值，同时根据允许误差的要求判断是否进行嵌入。对于确定的候选位，根据对应的嵌入策略控制信号位，修改候选位奇偶性以嵌入指纹。

容易知道，在所有候选位均满足嵌入条件的情况下，当可嵌入候选位大于拷贝指纹码矩阵 $P[m,l]$的元素数目时，该算法实现了指纹的重复嵌入。指纹矩阵被重复嵌入的次数为

$$rep=\mathrm{floor}\left(\frac{n}{\gamma\times m\times l}\right) \tag{6}$$

其中，floor(x)为取整函数，返回不大于 x 的整数；n 为关系数据集的元组总数；γ 为元组标记间隔值。

3.4 指纹检测与叛逆追踪

本方案将指纹检测分为两个过程：版权判定与指纹提取。其中，版权判定就是检测版权水印的过程，不涉及用户指纹，因此只需要数据库发行者的版权密钥 K_c 参与。指纹提取则只针对经水印检测判定为盗版的数据库进行，需要用户密钥 K_u 的参与，提取指纹并追踪叛逆者。

版权水印的检测过程是水印嵌入过程的逆过程，算法步骤类似。在版权密钥 K_c 的参与下生成混沌二值序列，对每个元组，以密钥与元组主键的位连接归一化值及允许误差确定检测位置，读取相应候选位的数值，根据其奇偶性匹配为二值序列。再根据相应的期望水印比特值及对应的指纹取反控制信号进行水印还原与比对。由于水印序列的同一比特可能被多次采样嵌入，通过“多数表决”算法[12]最终确定检测出二值序列中的每一位，并采用归一化相关检测来计算其与预期水印信号的相关程度[13]，实现版权判定。

对于判定为盗版的数据库拷贝，使用用户密钥 K_u 提取用户指纹。若该用户参与了合谋，在标记假设条件下，由于不可侦测位的存在，还原后的块中将会出现非全“0”或非全“1”的情况。对于提取出的指纹矩阵，逐个取指纹列向量，计算其中占优势的比特个数。当占优势的比特比例满足优势比特率阈值条件，则认为基本指纹码中对应的码元是占优势的，称该码元是可判定指纹码元[10]。对用户 u，如果该用户指纹中的可判定码元比例满足优势码元率阈值条件，则认为用户 u 是合谋分发者；否则，认为 u 是无辜用户。叛逆追踪的算法步骤如算法 2 所示。

算法 2 叛逆追踪（基于用户密钥 K_u 提取的指纹矩阵 $P'[m,l]$）

```
1)C_l = 0
// 可判定指纹码元计数器清零
2)for i = 1 --l_1 +1 to l
3)   C_m = 0
     // 优势比特计数器清零
4)   for j = 1 to m
5)     if P[ i,j ] =1,then
6)       C_m = C_m +1
7)     if C_m / m < 0.5
8)     C_m = m - C_m
9)   if C_m / m ≥ P_b_thre ,then
     // C_m / m 为优势比特率
10)      C_l = C_l +1
11)if C_l / l_2 ≥ P_c_h ,then
   // C_l / l_2 为可判定指纹码元率
12)  判定 u 为叛逆者
13)if C_l / l_2 ≤ P_c_l ,then
14)  判定 u 为无辜用户
```

实际应用中，需要对判定为盗版的数据库拷贝遍历所有用户的密钥。本方案将版权认证过程从指纹检测中独立开来，有效缩小了参与指纹提取运算的数据库拷贝范围，提高了运算效率。算法中优势比特率阈值 P_{b_thre} 及优势码元率阈值 P_{c_h} 与 P_{c_l} 的计算依据在下一节算法分析中给出。

3.5 算法分析与参数选择

下面通过对算法的抗合谋能力的分析，给出叛逆追踪过程中优势比特率阈值 P_{b_thre} 及优势码元率阈值 P_{c_h}（上界）与 P_{c_l}（下界）的计算依据。

设合谋用户数为 c。设对于一个盗版数据库拷贝，根据用户 u 的密钥 K_u 检测出的数据库指纹矩阵为 $P'[m,l]$。对于 $P'[m,l]$中的任一指纹列向量 $pc[m]$，由嵌入假设可知，$pc[m]$中的不可侦测位数为：

$$N_b=\frac{m}{2^{c-1}} \tag{7}$$

当使用指纹取反控制序列 $r[m]$从指纹矩阵 $P'[m,l]$还原基本指纹码时，$pc[m]$中的不可侦测位将被还原为全“0”或全“1”，其取值视基本指纹码中对应位置的原值而定。

对于 $pc[m]$中的可侦测位，假设合谋用户 u 已经以随机比特进行了重置攻击，则用 $r[m]$从 $P'[m,l]$中还原基本指纹码时，由于 $r[m]$具有随机性，因此，这些可侦测位经过还原后，“0”和“1”将是均衡分布的。所以，对指纹矩阵还原后提取出的基本指纹码列向量中，优势比特率的数学期望为：

$$p_b=\frac{N_b+(m-N_b)/2}{m}=\frac{1}{2}+\frac{1}{2^c} \tag{8}$$

定理 1 对定义 4-5 所示指纹矩阵还原后获得的基本指纹码列向量，设其优势比特率的数学期望为 P_b（$0<P_b<1$）；当指纹取反控制序列长度 m 充分大，对于给定的 α（$0<\alpha<1$），令 Z_α 为标准正态分布的上 α 分位点，

记 $a=m+Z_{\alpha/2}^2$，$b=2mP_b+Z_{\alpha/2}^2$，$c=mP_b^2$，则置信度为 $1-\alpha$ 的优势比特率阈值的置信区间为

$$\left[\frac{1}{2a}(b-\sqrt{b^2-4ac}),\frac{1}{2a}(b+\sqrt{b^2-4ac})\right] \tag{9}$$

证明：根据中心极限定理，当 m 充分大时，有

$$\frac{P_b-\mu}{\sigma/\sqrt{m}}\sim N(0,1),$$

根据标准正态分布的上 α 分位点的定义，设 p 为优势比特率，有

$$P\left(\left|\frac{P_b-p}{\sqrt{p(1-p)/m}}\right|\leqslant Z_{\alpha/2}\right)=1-\alpha,$$

将上式整理得

$$P((m+Z_{\alpha/2}^2)p^2-(2mP_b+Z_{\alpha/2}^2)p+m_b^2\leqslant 0)=1-\alpha,$$

由 $a=m+Z_{\alpha/2}^2$，$b=2mP_b+Z_{\alpha/2}^2$，$c=mP_b^2$，得

$P(ap^2-bp+c\leqslant 0)=1-\alpha$，有

$$P\left(\frac{1}{2a}(b-\sqrt{b^2-4ac})\leqslant p\leqslant\frac{1}{2a}(b+\sqrt{b^2-4ac})\right)=1-\alpha$$

可知，p 的置信度为 $1-\alpha$ 的置信区间为

$$[\frac{1}{2a}(b-\sqrt{b^2-4ac}),\frac{1}{2a}(b+\sqrt{b^2-4ac})]$$

证毕。

根据定理 1，取置信度为 $1-\alpha$ 的优势比特率阈值 P_{b_thre} 为 P_b 的下界，即：

$$P_{b\_thres}=\frac{1}{2a}(b-\sqrt{b^2-4ac}) \tag{10}$$

优势码元率阈值 P_{c_h} 与 P_{c_l} 的确定则与指纹检测的虚警概率 P_{fp}（即肯定错误概率）和漏检概率 P_{fn}（即否定错误概率）有关。

对于使用密钥 K_u 检测得到的用户指纹 $pp[\ l_2]$，若用户 u 为合谋用户，$pp[\ l_2]$ 中的优势码元数目服从二项分布：

$$B(l_2,1-P_{fn})=\binom{l_2}{k}(1-P_{fn})^k P_{fn}^{l_2-k},k=0,1,2,\cdots,l_2 \tag{11}$$

其数学期望为 $l_2(1-P_{fn})$，可取优势码元率阈值上界为：

$$P_{c\_h}=1-P_{fn} \tag{12}$$

若用户 u 为无辜用户，$pp[\ l_2]$ 中的优势码元数目服从二项分布 $B(l_2,P_{fP})$，其数学期望为 $l_2 P_{fp}$，可取优势码元率阈值下界为：

$$P_{c\_l}=P_{fp} \tag{13}$$

虚警概率 P_{fp} 与漏检概率 P_{fn} 的确定和系统的鲁棒性及安全性要求有关。根据文献[10]的定理 1，设合谋用户数为 c，当用户指纹长度 l_2 满足下式关系时，算法是合谋安全的：

$$l_2>2^{2c-1}(\sqrt{-\ln P_{fn}}+\sqrt{-\ln P_{fn}})^2) \tag{14}$$

4 实验

实验采用 Forest CoverType 数据集[9]。该数据集以二维表的形式描述了 1999 年美国的森林覆盖情况，取 581012 个观测点，共有 54 个属性，均为数值型。取前 10 个属性作为实验数据，转换至 MS SQL Server 数据库，并添加观测点编号字段为主键。

实验 1　指纹鲁棒性攻击。取指纹嵌入的元组标记间隔值 $\gamma=10$。分别以不同的指纹基本码长度指纹取反控制序列长度对数据库嵌入指纹。对生成的含指纹数据库拷贝，以随机值替换 50%的数据库元组。被攻击后数据库拷贝的指纹检出率如图 3 所示。

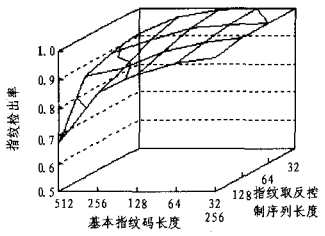

图 3　对子集替换攻击的指纹检出率

由实验结果可知，指纹系统对随机性子集替换攻击具有较高的鲁棒性。在子集替换攻击下，指纹检出率随所嵌入的指纹矩阵元素的增多而降低。注意到当指纹基本码长度为 512，指纹取反控制序列长度为 235 时，系统鲁棒性在子集替换攻击下大幅下降。这是由于此时指纹矩阵元素数目已超过实验数据集在元组标记间隔值为 10 的条件下所能提供的指纹嵌入位置数。这也从另一个角度给出了指纹载体数据选用及基本指纹码、指纹取反控制序列尺度选择的限制条件。

实验 2　合谋攻击与叛逆追踪。取数据库版权水印长度 $l_1=16$，用户指纹编码长度 $l_2=128$，指纹取反控制序列长度 $m=64$，指纹嵌入的元组标记间隔值 $\gamma=10$，以 4 个不同的用户密钥生成 4 个含指纹的数据库拷贝。分别以不同的用户数模拟合谋攻击，考察不同合谋人数下，优势比特率及可判定码元率的变化情况。实验结果如图 4 所示。

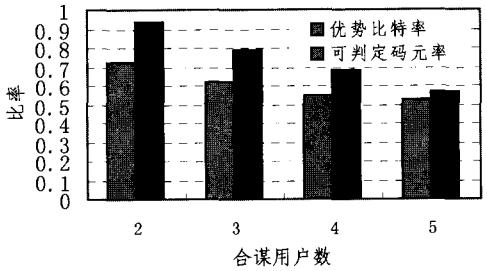

图 4　不同合谋用户数下的指纹检测

从图 4 可以看出，随着合谋用户数的增加，指纹检测的优势比特率逐渐下降，可判定码元率也随之下降。由实验结果可知，本指纹方案对合谋用户数在 4 以内的指纹攻击具有较高的指纹检出率与叛逆追踪的准确度。

结束语　本文提出了一种结合版权水印的数据库指纹编码方案及相关算法，实现了对盗版数据库拷贝的叛逆追踪。方案具有较高的合谋安全性，与现有的数据库指纹方案相比，在保持较高的指纹检出率与叛逆追踪的准确度的同时，降低了指纹检测与叛逆追踪的运算复杂度。

需要说明的是，本方案有一定的适用条件。首先，由于采用二进制随机指纹编码，其相应的指纹检测与叛逆追踪算法均建立在指纹信号的随机性与均衡性基础之上，因此，要求嵌入的指纹编码有足够的长度。其次，本方案的叛逆追踪能力受到合谋用户数的限制。

本文提出的数据库指纹解决方案中，指纹嵌入与指纹检测均由数据发行者使用同一组密钥完成，是一种对称指纹机制，不能实现对无辜用户的防陷害性。解决这一问题可以采用数据库非对称指纹机制。这也是我们下一步工作的研究方向。

参 考 文 献

[1] Cox I J, Kilian J, Leighton F T, et al. Secure spread spectrum watermarking for multimedia [J]. IEEE Trans. on Image Processing, 1997, 6(12): 1673-1687

[2] Boneh D, Shaw J. Collusion-Secure Fingerprinting for Digital Data. IEEE Trans. Information Theory, 1998, 44: 1897-1905

[3] Ferrer J D, Joancomarti J H. A Simple Collusion-Secure Fingerprinting Schemes for Images// Proceedings of the IEEE International Symposium on Information Technology: Coding and Computing (ITCC 2000). 2000: 128-132

[4] Zane F. Efficient watermark detection and collusion security// LNCS 1962. Springer-Verlag, 2001: 21-32

[5] Trappe W, Wu M, Wang Z J, et al. Anti-collusion Fingerprinting for Multimedia. IEEE Trans. Signal Processing, 2003, 51: 1069-1087

[6] Yoshioka K, Shikata J, Matsumoto T. A Method of Database Fingerprinting// The 2004 Workshop on Information Security Research. 2004

[7] Constantin C, Gross-Amblardet D, Guerrouani M. Watermill: an

Optimized Fingerprinting Tool for Highly Constrained Data // ACM Workshop on Multimedia and Security (MMSec). New-York, USA, August 2005:143-155

[8] Li Yingjiu, Swarup V. Fingerprinting Relational Databases -Schemes and Specialties. IEEE Transaction on Dependable and Secure Computing, 2005, 2(1):34-45

[9] Agrawal R, Kiernan J. Watermarking Relational Databases // The 28th VLDB Conference. Hong Kong, China, 2002

[10] 王彦,吕述望,徐汉良. 一种二进制数字指纹编码算法. 软件学报,2003,14(6):1171-1171

[11] Yen J C. Watermarks Embedded in the Permuted Image // The 2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001). Sydney, Australia, 2001

[12] Sion R, Atallah M, Prabhakar S. Rights Protection for Relational Data. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(6):1509-1525

[13] Le Jia-jin, Zhu Qin, Zhu Ying. A Novel Robust Scheme of Watermarking Database // The 2nd International Conference on Software and Data Technologies (ICSOFT). Barcelona, Spain, 2007

(上接第 251 页)

任务分解以及在处理器间重新划分数据库所需的时间。一旦任务分配方案确定后,算法才进行数据库的再划分,再划分的通信开销取决于每个处理器所要接收的数据库的规模。在理想状态下,由于每个处理器平均收到的数据量大小为数据库的$\frac{1}{p}$,因此通信时间为$O(\frac{|D|}{p})$。然而,由于总会有部分的数据重叠,实际的通信开销可能会高一些。为了便于分析,我们假定每个处理器需要接收的数据库序列的平均数为$\beta\frac{|D|}{p}$($\beta \geqslant 1$),于是任务分解所需的时间为 $T=O\left(\beta\frac{|D|}{p}\right)$。而每个处理器 P_i 生成子森林所需的时间 T_i 取决于本地数据库的大小$\beta\frac{|D|}{p}$、其后辈节点的总数 d_i 以及 $|I|$(I 表示数据库中不同项目的集合),推理可知:

$$T_i=f(\frac{\beta|D|}{p},|I|,d_i)=O((\frac{\beta|D|}{p})^{\alpha}|I|^2d_i)$$

若 $d_i|I|>>1$,则算法用于任务分解的时间远小于处理器生成本地子森林的时间,这时任务并行模式将是非常有效的。

6 实验结果与分析

6.1 实验环境

机器类型:曙光 TC1700 并行机;节点数:4;处理器个数:8(每个节点有 2 个处理器);处理器的内存为:512;操作系统:Linux;计算环境:MPI;数据库规模为:200k;最小支持度:1%;网络环境:千兆以太局域网。

6.2 实验结果及分析

本实验进行两组测试:一组是串行的数据挖掘算法,另一组是采取本文提出的并行的数据挖掘算法。具体实验结果如表 1 所示。

表 1 串行与并行数据挖掘算法的执行时间比较

处理器个数	串行数据挖掘算法		并行数据挖掘算法	
	数据执行时间	任务执行时间	数据执行时间	任务执行时间
2	34.69	32.17	34.69	32.17
4	34.67	32.17	19.49	16.82
6	34.67	32.16	14.09	12.16
8	34.66	32.16	10.76	9.12

由上表可以看出,随着处理器数量的增加,两种并行算法的执行时间都明显减少,并取得了较好的加速比。同时我们注意到,随着处理器数目的增加,数据并行模式的并行效率随着下降,这是因为分配给每个处理器的工作量在减少,而归约操作所带来的通信开销是固定的;而在任务并行模式中,只要有足够多的节点和项目集,就能获得较好的负载均衡。

结束语 序列模式挖掘有着广泛且重要的应用前景。因为序列模式挖掘所面临的数据量往往非常大,所以整个系统的存储容量和挖掘效率就显得至关重要。为了进行有效挖掘,高效的并行算法尤为必要。本文提出了一种并行的序列模式数据挖掘算法。通过理论分析与实验验证可知:本文提出的并行数据挖掘算法,在数据量较大、多个处理器计算的秉性环境下,能很好地提高数据的挖掘效率。

参 考 文 献

[1] Jovanovic N, Milutinovic V, Obradovic Z. Foundations of predictive data mining; Neural Network Applications in Electrical Engineering // 2002 6th Seminar on. 2002:53-58

[2] Riedmiller M, Braun H. A direct adaptive method for faster backpropagation learning: the RPRO Palgorithm // Neural Networks, 1993, IEEE International Conference on. 1993:586-591

[3] Onoda T. Neural network information criterion for the optimal number of hidden units. Neural

[4] Folino G, Pizzuti C, Spezzano G. Improving Induction Decision Trees with Parallel Genetic Programming[C] // Parallel, Distributed and Network-basedProcessing. Proceedings 10th Euromicro Workshop on. 2002

[5] Han E, Karypis G, Kumar V. Scalable parallel algorithms for mining association rules. IEEE Transactions on Knowledge and Data Eng., 2000, 12 (3)

[6] Zaki M J. An efficient algorithm for frequent mining sequences. Machine Learning Journal, 2001, 42:31-60

[7] Per J, Han J, Mao R. CLOSET: An efficient algorithm for mining frequent closed itemsets // Proc. 2000 ACM-SIGMOD Int. Workshop Data Mining and Knowledge Discovery (DMKD00). Dallas, TX, May 2000:11-20

[8] 王珊,等编著. 数据仓库技术与分联机分析处理. 北京:科学出版社,1998:1-17

[9] Wilkinson B, Allen M 著. 并行程序设计. 陆鑫达,等译. 机械工业出版社,2002:20-37

[10] 冯百鸣,经彤. BP 算法并行程序的自动生成与并行效率预测. 电光与控制,1997(2):1-5

[11] 任立勇,卢显良. 基于串-并行计算 BP 网络拓扑结构的研究与实现. 电子科技大学学报,2000,29(2):197-200