

基于 DFS 的最优路径数据迷乱算法

黄光球, 谭 蕾

(西安建筑科技大学 管理学院, 陕西 西安 710055)

摘 要: 提出一种基于深度优先搜索的数据迷乱算法, 将数据库中的原始数据通过 DFS 算法生成邻居关系树, 找出生成树中边缘距离最小的路径作为原始数据的替代. 对于原始数据中含有相同元素的特殊情况, 提出了多种数据迷乱算法并行使用的技术以实现数据项的迷乱. 此算法的提出突破了原有数据迷乱技术的简单数据变形, 大大增加了攻击者逆向工程的复杂度以达到联机信息保护的目的.

关键词: 数据迷乱; DFS; 最优路径; NEDO

中图分类号: TP393.08

文献标识码: A

文章编号: 1000-7180(2010)02-0049-06

Nearest Edge Data Obfuscation Algorithm Based on DFS

HUANG Guang-qiu, TAN Lei

(School of Management, Xi'an University of Architecture & Technology, Xi'an 710055, China)

Abstract: This paper put forward a data obfuscation algorithm based on Depth First Searching. This algorithm create neighborhood tree through DFS algorithm and choose the nearest edge distance path in DFS Spanning tree, than substitute the original data with the data from nearest path. With regard to the identical data items, this paper use wide ranges of approaches concurrently for obfuscating the database. This algorithm break through simple data distortion of original obfuscation techniques and strengthen the complexity of reverse engineering for protecting the information.

Key words: data obfuscation; DFS; optimum path; NEDO

1 引言

随着移动代理技术的广泛应用, 软件产权保护与移动代码保护的重要性日益突出. 作为一种程序保护方法, 迷乱变换指对拟发布的应用程序进行保持语义的变换, 使得变换后的程序和原来的程序在功能上相同或相近, 但是更难以被静态分析和逆向工程所攻击. 数据迷乱^[1]是迷乱变换的一个重要分支, 通过对抽象数据类型的迷乱算法实现原始数据的迷乱, 从而隐藏原始信息的真实性^[2-3].

文中提出一种基于数据结构中深度优先搜索算法(DFS)的最优路径数据迷乱算法(Nearest Edge Data Obfuscation Algorithm, NEDO), 将数据库中的原始数据生成邻居关系树, 找出树中边缘距离最小的路径作为原始数据的替代, 从而实现数据迷乱的

作用. NEDO 算法的提出突破了原有数据迷乱技术的简单数据变形, 大大提高了迷乱的复杂度, 其次在改变数据位置的基础上将原始数据信息隐藏, 基于 DFS 的迷乱算法可以保证在所有元素从其原始位置被替换掉的同时不会产生任意两个元素的交换, 这是其他数据迷乱方法无法保证的^[4].

2 最优路径数据迷乱算法

设 $P \xrightarrow{T} P'$ 是一个由源程序 P 到目标程序 P' 的一个迷乱变换, 如果 P 和 P' 具有相同的可观测行为, 则称 T 为一个从 P 到 P' 的迷乱变换. 更准确来说, 当同时满足以下两个条件时, T 才是一个合法的迷乱变换: ① 如果 P 无法终止或者以错误的状态终止, 则 P' 可终止也可不终止; ② 否则 P' 必须终止并且产生和 P 相同的输出结果.

最优路径数据迷乱技术是一种无损的数据迷乱技术,通过使用欧氏空间内的邻居替代原有的数据元素从而达到数据迷乱的目的.

2.1 算法的基本过程

输入: 数据库中的表 $\sum_{in} = [\sum_1, \sum_2, \dots, \sum_m]$, $\sum_i, i \in [1, m]$ 表示表中的 m 个属性, 每个属性中有 n 条记录 $v_0, v_1, \dots, v_n$.

输出: $\sum_{out} = [\sum'_1, \sum'_2, \dots, \sum'_m]$.

数据迷乱的过程被独立的应用到表中的每一个属性 $\sum_i$ 上, 对输入表中的其中一个含有 $n+1$ 条记录 $v_0, v_1, \dots, v_n$ 的属性 $\sum_i$ 的整个迷乱过程如图 1 所示.

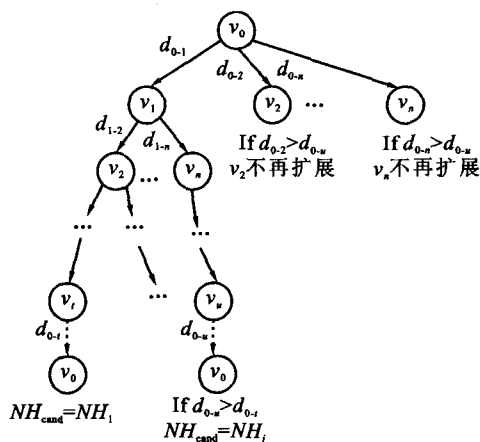


图1 NEDO算法生成树

独立的属性 $\sum_i$ 经过 DFS 算法迷乱后被划分为 NH_{size} 个邻居关系, 其中的第 j 个邻居关系记为 $NH_j, j \in [1, NH_{size}]$, 即图 1 中从根节点 v_0 到叶子节点 v_u 的一个路径 $v_0 \rightarrow v_1 \rightarrow v_n \rightarrow \dots \rightarrow v_u$. 每产生一个 NH_j 后便成为候选替代路径 NH_{cand} , 最终替代数据的产生由路径 NH_j 的叶子节点和根节点之间的最小边缘距离决定. 具体步骤如下:

Step 1: 将 $\sum_i$ 中的记录 $[v_0, v_1, \dots, v_n]$ 的第一个元素 v_0 作为根节点, 其余元素 $v_1, \dots, v_n$ 作为根节点的孩子节点由远及近排列, 图 1 中两节点间的 $d_{x-y}, x, y \in [0, n]$ 表示节点元素 v_x, v_y 间的边缘距离.

Step 2: 所有的节点根据 DFS 算法从左到右扩展到达叶子节点 v_t , 使路径 NH_i 遍历所有元素, 并且计算出此路径 NH_i 叶子节点与根节点的边缘距离 d_{0-t} , 此时 NH_i 成为候选路径.

Step 3: 如果新产生的路径 NH_j 的叶子节点 v_u

与根节点 v_0 的边缘距离小于当前的候选路径 NH_{cand} , 则边缘距离小的 NH_j 替代原来的候选路径成为新的 NH_{cand} .

Step 4: 当下一个要扩展的节点 v_2 的边缘距离与根节点间的边缘距离 d_{0-2} 大于候选路径叶子节点 v_u 与根节点 v_0 间的边缘距离 d_{0-u} , 此节点 v_2 不再扩展, 直到选出最小边缘距离的路径 (例如如图 1 中 $v_1 \rightarrow v_n \rightarrow \dots \rightarrow v_u \rightarrow v_0$) 作为原始数据迷乱后的结果^[5].

只考虑那些没有在树的根节点到子树根节点的路径上出现的节点可以减小生成树的大小, 同时也避免了数据项之间的交换. 最大边缘距离的开销将不是最优的路径移除从而减小搜索过程的复杂度. 只有可能产生最小边缘路径的节点进行扩展. 这种深度优先的方法根据 DFS 遍历算法实现, 由于树是有限的, DFS 的算法最终会产生最优的替代路径.

2.2 算法的特殊情况

最短路径数据迷乱算法生成邻居关系的相似数据项替代原始数据的位置关系. 在算法找出最短路径替代数据集时, 存在一些特殊的情况.

2.2.1 迷乱后仍存在相同数据项 (GT-NEDO)

在使用最短路径算法对原始数据库表中的每一个属性分别进行迷乱后, 产生的数据库结果有可能仍旧保持原来的数据对应关系, 攻击者通过逆向工程有可能截获到此真实信息. 如表 4 所示, 原始数据含有年龄和工资两个数据项, 分别经过最短路径迷乱后得到的数据仍保持原始记录的数据对应关系, 这种情况大大降低了数据迷乱的复杂性, 有可能被攻击者逆向工程分析得到真实数据. 因此, 将数据迷乱技术中的几何变换与最短路径算法结合使用, 原始数据在 NEDO 算法迷乱后, 再将数据项统一进行几何变化, 式 (1) 代表一般的几何变换, $[v_0, v_1, \dots, v_n]$ 为经过 DFS 最优路径算法迷乱后的数据, 经过式 (1) 的几何变换处理后得到的 $[v'_0, \dots, v'_n]$ 为最终的迷乱结果. 根据式 (1) 中矩阵 A, B 的取值实现不同的几何变换, 矩阵 (2)、(3) 分别代表对原始数据的平移和放缩, 相对比较简单. 矩阵 (4) 表示矩阵的旋转, 当 DFS 算法迷乱后仍存在相同元素, 将此相同元素的矩阵取出做旋转变换 θ 度角, 可得到与原相同元素构成的向量模相同但数值不同的数据. 当然, 在进行几何变换的时候可以将变化的复杂度

加大,也可以将几种不同的几何变换同时应用到一个数据集上.这样不但改变了数据项的存放位置,同时改变了数据项的真实值,大大增加了迷乱的复杂度.

$$\begin{bmatrix} v'_0 \\ \vdots \\ v'_n \end{bmatrix} = \mathbf{A}^* \begin{bmatrix} v_0 \\ \vdots \\ v_n \end{bmatrix} + \mathbf{B} \quad (1)$$

$$\mathbf{T}:\mathbf{A} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & 1 \end{bmatrix}_n \quad \mathbf{B} = \begin{bmatrix} t_0 \\ \vdots \\ t_n \end{bmatrix}_n \quad (2)$$

式中, $[t_0 \cdots t_n]$ 为任意一组常数.

$$\mathbf{S}:\mathbf{A} = \begin{bmatrix} S_x & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S_y \end{bmatrix}_n \quad \mathbf{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_n \quad (3)$$

式中, $[s_x \cdots s_y]$ 为任意一组常数.

$$\begin{bmatrix} v'_x \\ \vdots \\ v'_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (4)$$

式中, $[v_x, v_y]$ 为DFS算法迷乱后产生的相同数据, $[v'_x, v'_y]$ 为旋转变换后的数据.

2.2.2 原始数据集含有相同元素

在以上对最优路径数据迷乱算法的讨论中,原始数据集中的元素值均不同,但是数据库中可能出现数据值相同的情况,因此引入数组分裂^[6]的思想将相同的元素和不同的元素分开处理.数组分裂是一种强制性的迷乱抽象数据结构的方法,通常的迷乱技术只应用于面向对象的语言中,文中提出的数学方法可以将迷乱技术应用到更多领域.数组分裂的迷乱技术将数组元素划分到两个或者更多的数组中,从而破坏原有的数据结构.要进行这种变化首先要确定如何将数组分裂即选择函数 $\text{ch}(i)$ 的确定,例如选择函数要将数组中的奇数选出,满足选择函数的奇数元素划分到一个数组中,而不满足选择函数的元素即为偶数,划分到另一个数组,如式(5)所示. $A[i]$ 为原始数组,满足选择函数 $\text{ch}(i)$ 的数据被划分到数组 B_1 ,不满足选择函数的数据划分到数组 B_2 ,函数 $f_1(i)$ 和 $f_2(i)$ 分别用来确定数据元素在数组 B_1 和 B_2 中的位置.

$$A[i] = \begin{cases} B_1[f_1(i)] & \text{if } \text{ch}(i) = \text{Ture} \\ B_2[f_2(i)] & \text{Otherwise} \end{cases} \quad (5)$$

在NEDO算法中原始数据中含有多个相同元素时,用数组分裂中的选择函数将相同的元素和不同的元素分别放在两个数组中,不同的元素使用NEDO算法,相同元素用数据随机选择的方法进行迷乱.

2.3 时间复杂度分析

最优路径数据迷乱算法使用深度优先搜索遍历的方法在每一个数据项中选出最优的替代数据集.生成树中的每一个路径都成为替代数据集的候选路径.当选出最小的边缘距离时搜索结束,在以上年龄数据项的例子中,至少要扩展数据集中的一半节点才得到替代的数据集合.而最坏的情况下,生成树中的每一个节点都要被搜索才可以得到最优的替代数据集.搜索每一个节点所需要的时间为 $(n-1)!$,最坏的情况下算法的时间复杂度为 $O((n-1)!)$.

2.4 NEDO算法抗逆向攻击分析

数据迷乱技术的隐私性保护主要基于两个重要性质,迷乱后数据库逆向攻击的复杂性和迷乱结果和原始数据的不相似性.NEDO算法属于非线性双射转换,在这种转换中,可逆性取决于可以完成逆向工程的最小记录数量值 r .在NEDO中,要完成整个数据集的逆向工程,至少知道每一个邻居关系的 $r = c - 1$ 个数据元素,其中 c 是邻居关系的最小值. $(c_i - 1)/c_i$ 决定含有 c_i 个元素的特定邻居关系 NH_i 的逆向情况,攻击者的目的是要恢复出原始数据值.

定理1 设 X, Y 分别代表含有 n 个的元素的原始数据集和迷乱后的数据集, y_i 是属于第 p 个邻居关系中攻击者期望得到原始数据 x_i 的迷乱值.假设在第 p 个邻居关系中的所有 c 个数据项都是不同的值,且攻击者完全了解生成迷乱数据集 Y 的NEDO算法过程,除此之外没有任何有关原始数据集的信息,则攻击者需要获得至少 $c - 1$ 个原始数据项才可以恢复出全部原始数据.

证明 设 $[X_p, Y_p]$ 为第 p 个邻居关系中的原始数据项和迷乱数据项. $X_p = x_{p1}, x_{p2}, \cdots, x_{pc}$, $Y_p = y_{p1}, y_{p2}, \cdots, y_{pc}$, 要评价如何确定至多 $c - 2$ 个原始数据项,攻击者只知道以下信息:

$$X'_p = x_{p1}, x_{p2}, \cdots, U, \cdots, U, \cdots, x_{pc},$$

$$Y = y_1, y_2, \cdots, y_n$$

式中, X'_p 为包含 $c-2$ 个原始数据项的集合, 每一个 U 代表一个遗漏值. 而攻击者的目的是要确定这两个遗漏值并且确定其 y_i 的原始数据值.

(1) 在迷乱的数据集中存在两个数据项 $y_k, y_l \in [\min(X_p), \max(X_p)]$, 在这样的情况下, 攻击者知道 y_k, y_l 是在邻居关系 p 中的遗漏项, 这两个数据项在邻居关系中有以下两种放置方法: $X'_p = x_{p1}, x_{p2}, \dots, y_k, \dots, y_l, \dots, x_{pc}$ 或 $X''_p = x_{p1}, x_{p2}, \dots, y_l, \dots, y_k, \dots, x_{pc}$ 均可以产生相同的迷乱邻居关系. 因为没有附加的信息可以使攻击者准确的确定在原始数据邻居关系中是序列 X'_p, X''_p 中的哪一个, 因此无法确定迷乱数据 y_k, y_l 哪一个是对用原始数据项 x_i .

(2) 在迷乱数据集中没有任何数据项属于区间 $[\min(X_p), \max(X_p)]$, 这种情况下, 遗漏项是以下 3 种可能出现的组合: $\min(X_p) - 2, \min(X_p) - 1; \max(X_p) + 1, \max(X_p) + 2$ 或者 $\min(X_p) - 1, \max(X_p) + 1$. 对于每一种组合都有两种邻居关系的排序可能是原始邻居关系. 因此, y_i 的原始值可以是 6 个取值, 攻击者根本无法确定原始值 x_i .

(3) 在迷乱数据集中有一个项属于区间 $[\min(X_p), \max(X_p)]$, 这个项记为 y_{kl} . 这种情况下, 遗漏项是以下 2 种可能出现的组合: $\min(X_p) - 1, y_{kl}$ 或者 $y_{kl}, \max(X_p) + 1$, 每一种组合可以用两种方法填充遗漏位置. 因此, 将产生 y_i 原始值的 4 个候选项, 攻击者同样无法确定原始数据项的值 x_i .

基于以上证明分析, 文中提出的最短路径数据迷乱算法可以很好的实现对原始数据库的迷乱功能, 将数据信息隐藏, 同时对于攻击者的逆向工程也有极高的复杂性.

2.5 数据恢复方法

当接收方得到经过迷乱变化的数据后, 要恢复出原始数据. NEDO 算法在发送方将数据迷乱的过程以 DFS 生成树的形式存储, 在树中选出边缘距离最小的路径作为替代数据. 因此, 要将数据恢复必须根据树的存储结构执行不同的恢复方法. 当采取顺序存储结构时, 接收方将存储 DFS 树的数组中前 n 个元素依次取出即为原始数据, 如图 2 所示, 虽然显示迷乱的数据, 但存储结构本身仍保持原有的数据. 当采取多重链式存储结构时, 每个节点含有多个指针域, 如图 3 所示, 每个节点指针域中的第一个位置

存放元素本身, 第二个位置为指向此元素父节点的指针, 第三个位置以后依次存放指向此元素孩子节点的指针^[7-8].

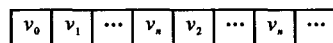


图2 顺序存储结构的数据恢复

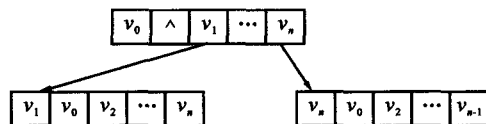


图3 链表存储结构的数据恢复

步骤如下所示.

Step 1: 判断进行数据迷乱时 DFS 生成树的存储形式; 如果是顺序存储结构, 执行 Step 2; 如果是链式存储结构, 执行 Step 3;

Step 2: 依次取出存放 DFS 树的数组中前 $n+1$ 个元素, 即为原始数据 $[v_0, v_1, \dots, v_n]$, 如图 2 所示;

Step 3: 首先找到指针域中第二个位置即父节点为空的节点, 取出此元素即为原始数据的第一个元素, 然后找出指针域中父节点位置指向根节点的元素, 再从左到右依次排列在首元素之后, 即可恢复出原始数据, 如图 3 所示.

3 算法举例

3.1 不同元素的 NEDO 算法

原始数据库如表 1 所示, 现将年龄数据项所为迷乱的原始数据 $[35, 37, 38, 40, 42]$. 35 作为树的根节点, $[37, 38, 40, 42]$ 为孩子节点, 它们之间用边缘距离用连接, 如图 4 所示.

表1 原始数据库

Number	Age	Salary
1	35	86 000
2	37	88 000
3	38	93 000
4	40	85 000
5	42	94 000

(1) 按照生成树扩展下一个节点, 即 38 的第二个子节点 42. 得到路径 $(35 \rightarrow 37 \rightarrow 38 \rightarrow 42 \rightarrow 40 \rightarrow 35)$, $d_{ME} = 5$, 此时边缘距离小于候选路径, 因此 $C_{ME} = 5$ 的路径成为迷乱候选路径.

(2) 算法回到树中下一个未被展开的节点, 即

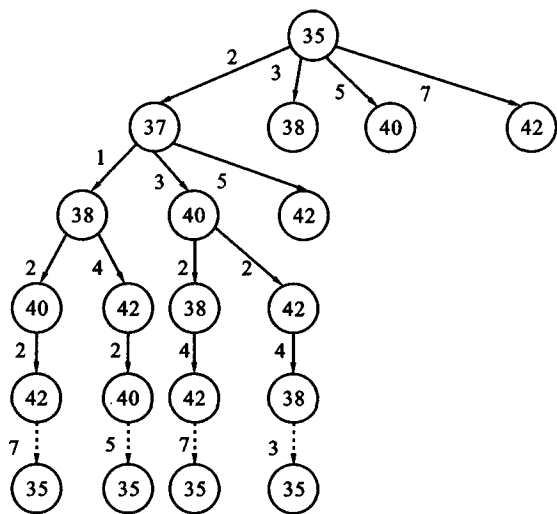


图4 年龄数据项DFS生成树

37的第二个子节点40. 扩展40这个节点后,它的最左路径的边缘距离为 $d_{ME}=7$,比候选路径的边缘距离大,因此这个路径被忽略. 而路径(35→37→40→42→38→35)的边缘距离为 $d_{ME}=3$,小于候选路径的边缘距离,成为新的迷乱候选路径.

(3) 下一个被扩展的节点为37的第三个子节点42. 此时 $d_{E_{37-42}}=5$,大于候选路径的边缘距离,放弃这个节点的扩展.

(4) 根节点的剩下子节点与根节点的边缘距离分别为 $d_{E_{35-38}}=3$, $d_{E_{35-40}}=5$, $d_{E_{35-42}}=7$,大于或者等于候选路径的边缘距离 $d_{ME}=3$. 因此,这些节点不再扩展. 最终的候选路径(35→37→40→42→38→35) $d_{ME}=3$ 成为年龄数据项的迷乱结果. 如表2所示.

表2 迷乱后的数据库

Number	Age	Salary
1	37	86 000
2	40	88 000
3	42	93 000
4	38	85 000
5	35	94 000

只考虑那些没有在树的根节点到子树根节点的路径上出现的节点可以减小生成树的大小,同时也避免了数据项之间的交换. 这种深度优先的方法根据DFS遍历算法实现,由于树是有限的,DFS的算法最终会产生最优的替代路径.

利用生成树的方法也可以将工资数据项进行迷乱,取出数据项后将其排序可以简化生成树的算法.

原始数据为[85,86,88,93,94],利用数据迷乱的方法可将其迷乱为[93,85,94,86,88], $d_{ME}=3$. 结果如表3所示.

表3 年龄和工资同时迷乱结果

Number	Age	Salary
1	37	93 000
2	40	85 000
3	42	94 000
4	38	86 000
5	35	88 000

3.2 迷乱后仍存在相同数据项

如表4所示,原始数据有年龄和工资两个数据项,分别经过最短路径迷乱后得到的数据中有一条记录(Age:40,Salary:7 800)原始数据相同,在表4的例子中,将迷乱后的工资数据项统一进行缩放,缩放比例为 $\delta=0.88$.

表4 GT-NEO算法的迷乱结果

NO	Original Database		NEDO Database		GT-NEO Database	
	Age	Salary	Age	Salary	Age	Salary
1	35	7 600	37	8 400	37	7 392
2	37	7 900	40	7 800	40	6 864
3	38	8 400	35	8 200	35	7 216
4	40	7 800	42	7 600	42	6 688
5	42	8 200	38	7 900	38	6 952

3.3 原始数据含有相同元素(R-NEDO)

将原始数组进行分裂,满足选择函数 $ch(i)$ 的元素放入一个数组中,剩下不同的元素放入另一个数组. 对于存放不同元素的数组可以直接使用NEDO算法进行数据迷乱;而对于存放相同元素的数组,利用前面提到的数据随机性方法进行迷乱,给相同的数据项中加入一个服从高斯分布 $[0, \delta]$ 的噪音因子,其中 δ 是与邻居关系中不同元素的偏差. 如表5所示,偏差 $\delta=3.89$.

表5 相同元素的迷乱结果

NO	Original Database	NEDO Database	R-NEDO Database
1	86 000	88 000	88 000
2	89 000	92 000	92 000
3	94 000	86 000	86 000
4	88 000	89 000	89 000
5	105 000	105 000	106 000
6	92 000	94 000	94 000
7	105 000	105 000	101 200
8	105 000	105 000	111 600
9	105 000	105 000	102 000

4 结束语

数据迷乱技术虽然能够提高原始数据的隐蔽性,单一的算法无法满足各类应用的安全需要,而还要和各类的代码迷乱算法结合使用,使其性能指标最佳.文中提出的混淆算法还停留在理论阶段,完善算法还需要大量的工作.

参考文献:

- [1] Bakken D, Paramesw A R, Blough D. Data obfuscation: anonymity and desensitization of usable data sets[J]. IEEE Security and Privacy, 2004, 13(2):34-41.
- [2] Agarwal C, Kiernan J, Srikant R, et al. Order preserving encryption for numeric data[J]. Special Interest Group for Management of Data, 2004, 34(8): 63-74.
- [3] Canny J. Collaborative filtering with privacy[J]. IEEE Symposium on Security and Privacy, 2002, 20(3):45-57.
- [4] Agarwal C, Yu P. A condensation approach to privacy preserving data mining[J]. Advances in Database Technology

- EDBT, 2004(2): 83-199.

- [5] Collber C, Thomborson C. Software watermarking: models and dynamic embeddings[C]//Proceedings of the 26th Annual SIGPLAN - AIGACT Symposium on Principles of Programming Languages. USA, New York, Association for Computing Machinery Press, 1999:311-324.
- [6] Oliveira S, Zang O. Achieving privacy preservation when sharing data for clustering[J]. Secure Data Management in conjunction with VLDB, 2004(3178):67-82.
- [7] 龚本灿,李腊元. 基于生成树的无线传感器网络分布式路由协议[J]. 微电子学与计算机, 2008, 25(11): 77-84.
- [8] 莫晓莉,王汝传,张琳. 基于移动代理和信任机制的网格作业管理[J]. 微电子学与计算机, 2008, 25(10): 159-161.

作者简介:

黄光球 男,(1964-),博士,教授. 研究方向为电子商务与网络安全.

谭 蕾 女,(1984-),硕士研究生. 研究方向为电子商务与网络安全.

(上接第 48 页)

作者简介:

兰晓明 男,(1984-),硕士研究生. 研究方向为锁相环频率综合器设计.

颜 峻 男,(1976-),博士. 研究方向为集成射频收发系统芯片设计.

马何平 男,(1980-),博士研究生. 研究方向为片内滤波器设计.

石 寅 男,(1951-),研究员,博士生导师. 研究方向为新型高速电路设计.