文章编号:1001 - 9081(2007)08 - 1931 - 04

基于用户约束的关系数据库水印方法

邓海生[1],李军怀[1],张 璟[1],郑军平[2]
(1. 西安理工大学 计算机科学与工程学院, 西安 710048; 2. 宝鸡石油钢管公司, 陕西 宝鸡 721008)
(xaut0420@126.com)

摘 要:针对现有水印算法的不足,提出了在数值型属性最低有效位(LSB)嵌入水印信息的一种新方法。算法先筛选出可以嵌入水印信息的属性,将它们划分为若干个等大的子集,然后依据数据库所有者定义的约束对这些子集进一步筛选,以筛选后的每个子集为单位嵌入水印信息 0 或 1。水印检测算法依据的是中心极限定理,实现了水印信息的准确检测。SQL Server 仿真实验验证了此方法在现实应用中的有效性。

关键词:用户约束; 关系数据库; 水印信息; 水印检测
中图分类号: TP309.2 **文献标志码**:A

Relational database watermark method based on user restraint

DENG Hai-sheng[1], LI Jun-huai[1], ZHANG Jing[1], ZHENG Jun-ping[2]
(1. *School of Computer Science and Engineering, Xi'an University of Technology, Xi'an Shaanxi* 710048, *China*;
2. *Baoji Petroleum Steel Pipe Co., Ltd., Baoji Shaanxi* 721008, *China*)

Abstract: A new method that digital watermarking information was embedded in numerical attribute's Least Significant Bit (LSB) was put forward in view of the existing watermark algorithm's insufficiency. After sieving attributes that could embed watermarking information and dividing these attributes into some equal subsets that would be further sieved based on the binding defined by the database owner, watermark 0 or 1 was embedded into each sieved subset. The watermark detection algorithm based on the central limit theorem can test watermarking information accurately. In the end, the simulation experiment using SQL Server verified the validity of algorithm in practical application.

Key words: user restraint; relational database; watermarking information; checking watermarks

0 引言

数字水印技术[1]通过在数字产品中嵌入一定可感知或不可感知的信息,以确定数字产品的所有权或者检测数字内容的原始性[2]。目前,在关系数据库水印技术研究方法方面,文献[3,4]中进行了一些开创性研究。文献[5]则结合云理论来研究关系数据库水印技术。R. Agrawal 针对一个特定的数据库进行水印的嵌入和攻击试验,其中只包含数值型数据,且假定每个字段都能够添加水印,然后依据水印密钥和关键字确定需标记的字段及位置。该算法主要存在两个缺点:1)在关系数据中嵌入水印信息时,没有考虑被标记属性值的大小。事实上不同绝对值大小的属性值承受更改的能力是不一样的,绝对值大一些的属性值可以为水印嵌入算法提供较多的可标记位,而绝对值较小的属性值只能提供较少的,有时甚至不能提供可标记二进制位;2)对关系数据中属性的排列顺序十分敏感。也就是说攻击者只要改变属性顺序就能使水印检测算法无法从含有水印的数据库中检测出水印信息来,这就意味着该算法未能做到完全的检测,影响了其实用性。

本文基于用户约束的关系数据库水印方法不但克服了上述 R. Agrawal 水印算法的不足,而且具有简单实用、强鲁棒性等特点。该算法中引入了用户约束参数:数据库中数值型属性值允许变化范围 $b\%$ 和自定义子集约束条件 S,前者确保可标记位数"因值而异",后者实现了子集的合理筛选。该算法还通过记录相关属性的标识信息,实现了对水印信息完全、准确地检测。此外秘密排序和等子集划分等关键技术增强了该算法的鲁棒性。

1 基于用户约束的数据库水印算法

1.1 基本思想

首先根据用户定义的属性值允许变化范围 $b\%$ 筛选出数值型属性,并对这些数值型属性进行子集划分,然后根据自定义子集约束条件 S 进行子集筛选和水印信息的嵌入。而水印检测则根据所统计出水印信息匹配的属性值个数,计算水印存活率是否满足置信要求,以判断是否存在水印。

1.2 水印嵌入算法

1.2.1 子集划分

在嵌入水印信息之前,需要对关系数据库进行子集划分,然后再以每个子集为单位,完成水印信息的嵌入。子集划分通过以下步骤完成:1)对数值型属性值进行筛选;2)利用单向 Hash 函数对筛选后的属性值进行编码;3)按照编码由小到大的顺序将对应的属性值分组,完成子集划分。

设关系数据库为 $R(P,A_1,\cdots,A_i,\cdots,A_n)$,其中 P 为主键,

收稿日期:2007 - 02 - 02;**修回日期**:2007 - 04 - 10。
基金项目:国家 863 计划资助项目(2002AA414060);2005 年陕西省自然科学基金资助项目(2005F05)。
作者简介:邓海生(1980 -),男,山东人,硕士研究生,主要研究方向:数据库安全; 李军怀(1970 -),男,陕西人,副教授,博士,主要研究方向:网格和高性能计算; 张璟(1952 -),男,陕西人,教授,博士,主要研究方向:Web Services、SOA; 郑军平(1968 -),男,陕西人,高级工程师,主要研究方向:数据安全。

$A_1,\cdots,A_i,\cdots,A_n$ 为 n 个数值型属性列(不包括主键),R 由 m 个元组 $r_1,\cdots,r_j,\cdots r_m$ 组成,每个元组 r 都存在主键 $r.P$ 和 n 个数值型属性值 $r.A_1,\cdots,r.A_i,\cdots,r.A_n$,从而得到 $m\cdot n$ 个数值型属性值,对这 $m\cdot n$ 个属性值进行子集划分的具体步骤如下:

1)计算每个数值型属性值 $r.A_i$ 在约束 $b\%$ 下最低有效位(Least Significant Bit, LSB)可更改的范围 ε:

$$\varepsilon\lfloor\log_2(r.A_i\cdot b\%)\rfloor \quad (1)$$

其中,$\lfloor\ \rfloor$ 表示不超过某数的最大整数,$b\%$ 是由用户定义的数据库中数值型属性值允许变化的上限。

2)若 $\varepsilon<0$,即没有可以嵌入水印的位,返回 1),计算下一个属性值的 ε;若 $\varepsilon\geqslant 0$,即存在可嵌入水印的位信息,对此属性值 $r.A_i$ 通过 Hash 函数进行编码,其中 K 是密钥值:

$$\text{index}(r.A_i)=\text{hash}(K,r.P,r.A_i) \quad (2)$$

编码后返回步骤 1),计算下一属性值,直到完成所有数值型属性值的计算。

3)对所有编码由小到大进行排序。

4)按照编码由小到大的顺序将对应的属性值划分若干个等大子集。子集的个数取决于数值型属性值个数和子集的大小。例如,假设筛选后的数值型属性值有 100 000,子集大小若为 12 000 的话,可以得到 100 个等大子集。

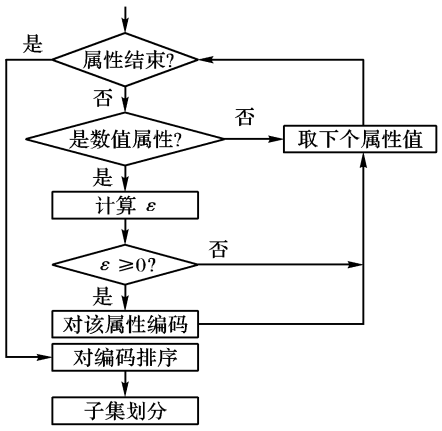

图 1 子集划分流程

1.2.2 水印信息的嵌入

水印信息嵌入过程以每个子集为操作对象嵌入一位水印信息。为了进一步控制水印信息的嵌入量,减少对于原始数据库的影响,需要做两方面的工作:1)筛选出符合要求的子集,嵌入水印信息;2)对于筛选后的子集,通过比较嵌入 0 和 1 对子集产生的影响,决定嵌入的水印信息是 0 还是 1。步骤如下:

1)确定子集中每个属性值的一位信息嵌入位置。方法是引入参数 j,用来确定水印信息的嵌入位置,方法如下:

$j=\text{index}(r.A_i)\bmod\varepsilon$ ($\text{index}(r.A_i)$ 见式(2),$\varepsilon>0$),或 $j=0(\varepsilon=0)$

属性值转换成二进制串后的第 j 位就是一位水印信息嵌入的位置。

2)计算嵌入水印信息 0 或 1 对于该子集的改变比例。

这里涉及到用户自定义约束 S 和方差改变比例 S_f。S 表示待嵌属性值允许变化的范围与全体待嵌属性值和的比,用来限定待嵌属性值允许变化的范围。

其中 $S_f=\left(\sum_{i=1}^{n}|A_i'-\bar{A}|\right)/(n\cdot\bar{A})$。$A_i'$ 表示嵌入水印信息后的属性值,$\bar{A}$ 表示属性值的平均值(嵌入水印信息前),n 表示待嵌属性个数。

3)通过比较 S_f 和 S 的大小关系,确定嵌入水印信息是 0 或 1 或不嵌入。方法是:计算嵌入水印信息 0 和 1 时的 S_f,即 $S_f(0)$,$S_f(1)$,然后比较 $S_f(0)$,$S_f(1)$ 和 S。若 $S_f(1)<S_f(0)$,且 $S_f(1)\leqslant S$,则将该子集每个属性值中相应二进制位设为 1;若 $S_f(0)<S_f(1)$,且 $S_f(0)\leqslant S$,则将该子集每个属性值中相应二进制位设为 0;其他则对该子集不予嵌入水印信息。

4)对下一个子集进行上述 1)~3)操作,直到最后一个子集。

1.3 水印信息检测算法

水印检测是指在检测过程中统计出水印信息匹配(包括位置匹配和水印信息值匹配)的属性值个数,计算出水印匹配率,然后依据中心极限定理计算水印存活率。若水印存活率满足置信度要求,则表示水印存在,反之认为水印不存在。

1.3.1 计算水印信息匹配率

当检测一个属性是否被嵌入水印信息时,要将这个属性值与原来数据库中的相应属性值进行比对,包括水印嵌入位置的比对和水印信息的比对。这就意味着在检测水印信息时需要备份原始数据库。本算法通过记录被嵌水印的属性值的相关信息,大大减少了数据备份量。

在子集划分中得到表 Embedtable,其中记录了属性值的如下信息:子集号 G,所在记录的主键值 P、列名 A;在水印信息的嵌入过程中得到表 Embedded,其中记录了被嵌入水印属性的相关信息:子集号 G,水印信息嵌入位 J、水印信息,记作 B。通过子集号 G 将表 Embedtable 和表 Embedded 进行关联,得到表 Embeddedtable,该表记录了所有被嵌入水印的属性值的完整信息:G、P、A、J、B。然后将表 Embeddedtable 中信息逐条和待测数据库中的相应属性值进行比较,如果水印嵌入位置和水印信息均匹配则累加器 matchcount 加 1,否则 matchcount 不变。用 totalcount 表示被嵌入水印信息属性个数(totalcount 等于表 Embeddedtable 的元组数),那么 matchcount/totalcount 就是水印信息的匹配率。整个检测过程中用户只需存储表 Embeddedtable,而不必备份海量的原始数据库。

基于以下思想将表 Embeddedtable 中信息和待测数据库中的相应属性值进行比较:为了保证原始数据库的可用性,元组主键 P 和列名 A 一般不会被更改,并且通过主键和属性名可以确定属性值,因此,首先通过 P_i,A_i 获得待测数据库中相应的属性值,方法是通过 SQL 语句"select Ai from tablename where $P=P_i$"检索得到(tablename:待测数据表名),然后计算该属性值第 J_i 位的二进制值,并和 B_i 比较,如果相同则匹配,否则不匹配。

计算水印信息的存活率具体算法的伪代码如下:

```
int matchcount = 0;                                   //累加器初始为0
for(int i = 1; i ++;i < = totalcount)
                          // totalcount: 被嵌入水印信息属性个数
{ get{G_i,P_i,A_i,j_i,B_i};                           //取得一组备份信息
  if (! exist (P_i,A_i)) continue;
          //待测数据库中如果没有 P_i 或 A_i, 提前结束本次循环
  value = get_value (P_i,A_i); //取得待测数据库中相应的属性值
  bit = get_bit (value,j_i); // 得到该属性值第 j_i 位的二进制的值
  if (bit = = B_i)                                    //水印信息一致,累加器加1
  { matchcount + = 1;}
}
```

上述算法可以得到匹配个数 *matchcount*,从而得到水息匹配率:*matchcount/totalcoun*。值得注意的是如果被测数据库中主键和属性名较原始数据库有变化,那么将无法通过 get_value(P_i,A_i)) 获得被检测数据库中的对应属性值,这是

本算法一个有待解决的问题。

1.3.2　检测水印信息是否存在

对于嵌入水印信息的情况,水印信息存活率理论上应该为100%,但对于关系数据库而言,存在着诸如子集数据扰乱、线形数据变换、随机项目改变等攻击,水印信息很难完全存活下来。因此,水印信息的存活率一般小于100%。不过只要存活率能满足一定的范围(例如大于95%),是可以认为存在水印信息的。

水印信息的存活率不能简单地理解为水印匹配率。例如,在1000个待测数值型属性值中,有600个经检测有水印匹配关系,水印存活率并非就是定值60%,而是以60%,即0.6为中心的一个小的区间。其区间的大小不但与水印匹配率有关,还有赖于置信因子 a,具体区间的计算可以依据以下的方法:

设 θ 为水印信息存活率,$1-2\times\alpha$ 为置信区间,n 为待测属性值的个数,待测属性值 $A_i(1\leqslant i\leqslant n)$。引入参数 x_i,并令:

$$x_i(1\leqslant i\leqslant n)=\begin{cases}0, & A_i \text{ 水印信息位不匹配}\\ 1, & A_i \text{ 水印信息位匹配}\end{cases}$$

当 n 较大时,根据中心极限定理可知:

$$\left(\sum_{i=1}^{n}X_i-n\cdot\theta\right)/\sqrt{n\cdot p\cdot(1-p)}=(n\cdot\overline{A}-n\cdot\theta)/\sqrt{n\cdot p\cdot(1-p)}$$

近似服从 $N(0,1)$ 分布(注:$\overline{A}$ = 匹配个数/待测属性值个数),于是有以下关系:

$$P\{-Z_\alpha<(n\cdot\overline{A}-n\cdot\theta)/\sqrt{n\cdot p\cdot(1-p)}<Z_\alpha\}=1-2\cdot a$$

而不等式 $-Z_\alpha<(n\cdot\overline{A}-n\cdot\theta)/\sqrt{n\cdot p\cdot(1-p)}<Z_\alpha$ 等价于 $(n+(Z_\alpha)^2)\cdot\theta^2-(2\cdot n\cdot\overline{A}+(Z_\alpha)^2)\cdot\theta+n\cdot(\overline{A})^2<0$,从而可以计算出 θ 近似的,置信度为 $1-2\times\alpha$ 的置信区间。只要置信区间的下限不小于认为水印信息存在时所满足的最小匹配率,就可以断言水印存在;反之,水印不存在。例如水印存活率大于95%就认为水印信息存在,那么对于置信区间[95.1%,98.3%]可以断定水印存在,而对于置信区间[94.9%,98.1%]则认为水印不存在。

2　仿真实验

仿真实验所用到的数据库是某发动机制造厂的生产数据,选取其中的90092个数值型属性做嵌入水印信息实验。后台数据库采用SQL Server 2000,编程环境为c#.net,数据库连接方式采用ADO.NET。

2.1　鲁棒性分析

当选取的约束参数 $b\%=0.00005$ 时,测得28650个数值型属性满足水印嵌入的条件,将28650个属性值分为15组,即15个子集。令用户约束参数 $S=5.0\mathrm{E}-08$,得到可以嵌入水印信息的5个子集并嵌入水印信息到这5个子集中。笔者针对嵌入水印后的数据库进行了鲁棒性仿真实验。

在仿真实验中,笔者模拟了子集线性变化,子集选取攻击和子集增加攻击。结果表明本文提出的水印嵌入算法对于以上攻击有很好的鲁棒性。

在子集更改攻击中令置信因子 $a=0.00005$。随机将这90092个数值型属性中一部分进行线性变化。仿真试验中将这部分随机子集变化0.0001,然后利用检测方法对线性变化后的数据库表进行水印信息检测。其部分测试结果如图1。在随机子集被线性变化30%的情况下,检测匹配率仍保持在80%以上,说明该算法对于子集更改攻击具有很好的鲁棒性。

由1.3节中的水印检测算法可知,水印信息的存活率和子集增加、子集选取没有关系,而在水印嵌入过程中又记录了被嵌入水印信息的属性标识信息,因此理论上只要在被测数据库中含有已经标识过的水印信息,检测匹配率即可达到100%,在笔者相关仿真试验中也证实了这一点。仿真图3、4显示在子集选取40%和子集增加30%的情况下,检测出的水印匹配率都在98%以上。因此本算法对于子集选取攻击和子集增加攻击同样具有很强的鲁棒性。

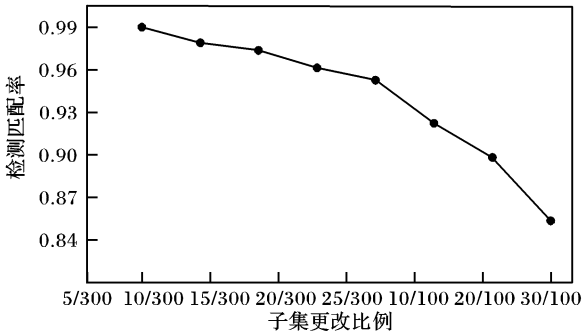

图2　子集更改检测匹配率

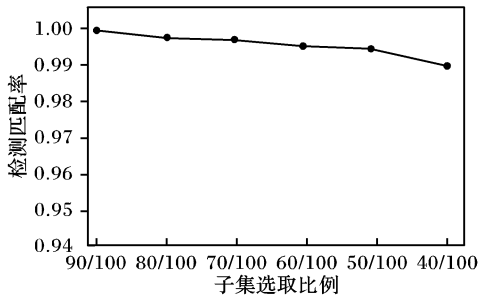

图3　子集选取检测匹配率

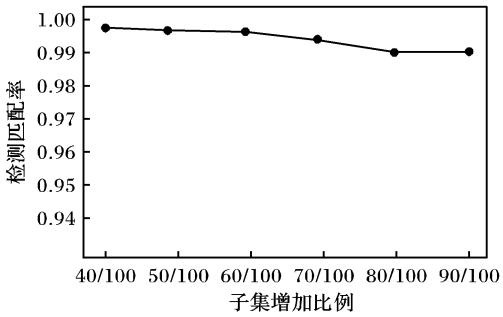

图4　子集增加检测匹配率

2.2　参数选取

水印信息的嵌入量一方面影响原始数据库的使用价值,另一方面影响版权的维护。嵌入的水印信息量越大,应用中心极限定理越客观,使得版权维护更具科学性,但对原始数据库使用价值的影响程度越大。同时,嵌入少量的水印信息,虽然对原始数据库的使用价值不构成影响,但不利于版权维护。本文的算法中,其影响数据库使用价值和版权维护的程度主要取决于用户约束参数 $b\%$、S 和置信度因子 a。笔者对参数 $b\%$ 做如下分析,同时为了客观分析参数 $b\%$,仿真试验中没有直接利用用户约束参数 S,而是选取变化率最小的若干组,同时令置信因子 $a=0.00005$。

当 $b\%=0.00005$ 时,可以嵌入水印信息的有28656个数值型属性,分10组,嵌入水印信息到变化率最小的5组共14355个属性中。当 $b\%=0.0001$ 时,可以嵌入水印信息的有34350个数值型属性,分10组,选择变化率最小的5组共

17 175 个属性嵌入水印信息。具体数据对照如表 1。

表 1 参数选取实验

$b\%$ = 0.000 05		$b\%$ = 0.000 1	
子集号	变化比率	子集号	变化比率
3	0	9	6.0037E - 06
7	0	2	6.1910E - 06
9	0	1	6.7077E - 06
4	3.0626E - 09	6	6.7564E - 06
6	4.7346E - 08	3	6.7607E - 06

分别对 $b\%$ = 0.000 05 和 $b\%$ = 0.000 1 时的数据库(嵌入水印信息后)做线性变化攻击,攻击后的水印检测结果如图 5 所示。

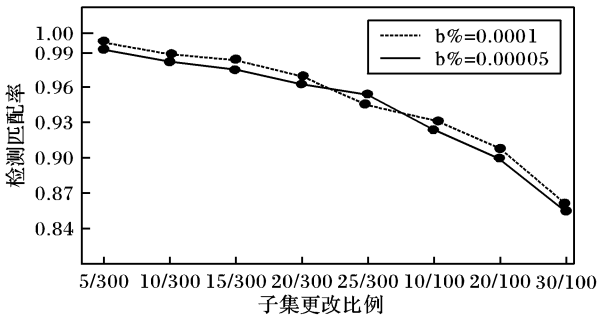

图 5 线性变化攻击

由上可以看出,增加水印信息嵌入量在一定范围内有利于水印的检测,这是以增大对原始数据库的影响程度为代价的,因此适当的选取参数很有必要。

3 结语

本文所介绍的数字水印技术是基于关系数据库的,主要围绕水印信息的嵌入方法和提取检测算法进行了讨论。算法通过等子集划分经由约束参数 $b\%$ 筛选后的数值型属性,真正意义上实现了水印信息的均匀分布;通过用户自定义约束 S 进一步筛选子集,确保了水印信息的嵌入不会破坏关系数据库原来的价值;在水印提取过程中通过备份属性值信息实现了水印信息的准确检测并依据中心极限定理思想判断水印信息存在与否。

下一步主要工作是在确保水印信息均匀分布的前提下,如何在关系数据库中的数值属性中嵌入有意义水印信息,如何应对更改数据表主键和列名的攻击等两方面。

参考文献:

[1] KATZENBEISSE S, PETITCOLAS F A P. Information hiding techniques for steganography and digital watermarking[M]. Boston, London: Artech House, 1999: 10 - 16.

[2] 王秋生. 变换域数字水印嵌入算法研究[D]. 哈尔滨: 哈尔滨工业大学, 2001.

[3] AGRAWAL R, KIERNAN J. Watermarking relational databases [C]// Proceeding of the 28th VLDB Conference. Hong Kong: Morgan Kaufmann, 2002: 155 - 166.

[4] SION R, ATALLAH M, PRABHAKAR S. On watermarking numeric sets[DB/OL]. (2003 - 06)[2007 - 01 - 18]. http://www.cs.purdu.edu/homes/sion.

[5] 李德毅, 史雪梅, 孟海军. 隶属云和隶属云发生器[J]. 计算机研究和发展, 1995, 32(6): 15 - 20.

(上接第 1930 页)

器分配安全资源。如果协商失败(无法取得统一的等级),则放弃通信。安全等级协商流程如图 3 所示。

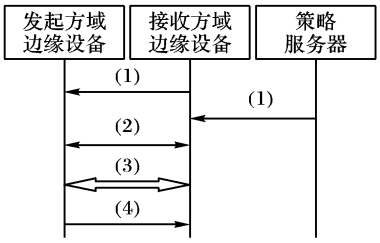

图 3 安全等级协商流程

图 3 中,协商所涉及的网元包括发起方域边缘设备、接收方域边缘设备和策略服务器三部分。域边缘设备是移动网络的接入设备,如前述移动终端或安全网关。

协商流程如下:

1) 若安全策略服务器首次对域边缘设备提供安全服务,域边缘设备从策略服务器下载安全等级配置信息。

2) 在用户根据当前服务和业务类别为发起方选择适当的安全等级后,发起方域边缘设备代表用户与接收方域边缘设备进行安全协商,协商原则如前所述,协商内容包括安全等级、算法参数等。

3) 如以上安全协商成功,则以协商结果建立由发起方到接收方的通信安全信道。

4) 用户使用安全服务结束,发起方域边缘设备代表用户通知安全服务结束。

该协商流程中,通信双方选择安全等级,协商过程按照前述安全策略和系统架构,充分考虑双方的安全需求,不同安全等级的通信双方能够对采用的安全协议协商一致。

4 结语

安全是 3G 发展中需要面临的一个重要问题,满足用户在安全方面的差异性需求,对移动业务安全实行分级管理,是实现移动网络安全的一个重要手段。本文提出的基于安全等级协商的移动网络安全系统,克服了传统的移动通信系统只能为用户提供固定模式的安全服务的缺点,用户能够根据自身安全需要,灵活地定制安全等级,从而选择性使用移动网络提供的安全服务,满足了用户在安全服务方面的特殊需求,为移动运营商实现增值安全服务提供技术保障。

未来的主要工作包括:将提出的系统模型应用在 3G 环境中,结合 3G 提供的安全技术,实现不同安全等级的协议和算法的协商,这对于提升 3G 安全服务能力,增强安全控制粒度具有重要意义。

参考文献:

[1] (芬) CAMARILLO G, GARCÍA-MARTIN M. 3G IP 多媒体子系统 IMS: 融合移动网与因特网[M]. 张同须, 译. 北京: 人民邮电出版社, 2006.

[2] 隋爱芬, 杨义先. 第三代移动通信系统的安全[J]. 世界电信, 2003, 16(5): 37 - 40.

[3] 刘东苏, 韦宝典, 王新梅, 等. 第三代移动通信系统的安全体系结构[J]. 西安电子科技大学学报, 2002, 29(3): 351 - 354.

[4] 李睿, 曾德贤. 3G 系统安全技术研究[J]. 中兴通讯技术, 2003, 9(6): 25 - 27.

[5] 李世鸿, 李方伟. 3G 移动通信中的安全改进[J]. 重庆邮电学院学报: 自然科学版, 2002, 14(4): 24 - 27.

[6] 陈剑勇, 彭志威, 罗忠生. 一种安全等级握手协商方法和系统: 中国, B, 200410070653.3[P]. 2006.

[7] ITU-T. Security architecture for systems providing End-to-End communications: ITU-T recommendation X.805[S]. 2003.