

文章编号:1001-9081(2009)06-1636-04

基于信任访问控制中信任评估与追溯方法研究

郎波,雷凯,江川,张静辉

(北京航空航天大学 软件开发环境国家重点实验室,北京 100191)

(langbo@buaa.edu.cn)

摘要:信任的量化表达与信任关系发现是基于信任访问控制中的难点问题。首先介绍了一种基于模糊数学的信任评估模型。通过定义时间衰退函数,该模型表达了时间对信任的影响。时间因素的引入使得该信任评估模型更具实用价值。同时,还提出了一种信任关系的分布式信任追溯算法,有效避免了访问控制中效率瓶颈的产生。

关键词:信任评估;时间衰退函数;分布式追溯

中图分类号: TP393.08 **文献标志码:** A

Research on access control oriented trust evaluation and tracing methods

LANG Bo, LEI Kai, JIANG Chuan, ZHANG Jing-hui

(State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China)

Abstract: Quantificational trust expression and trust relation discovery are difficulties that are not well tackled in trust based on access control. A trust evaluation model based on fuzzy set theory was introduced. By defining a decay function, the effect of time on trust was considered in the model. Then an algorithm of distributed trust discovery was proposed and discussed with some examples. The algorithm can improve the efficiency of trust relation discovery in access control.

Key words: trust evaluation; decay function; distributed trust tracing

0 引言

随着 Internet 技术与应用的飞速发展,基于 Internet 的分布式应用的规模也在不断扩大,已经突破了各种管理组织和地理位置等的局限,形成了全球性分布式计算环境,电子商务、网格计算等都是目前蓬勃发展的大规模分布式应用^[1]。目前,在分布式系统中仍然广泛采用传统的访问控制模型。这些访问控制模型如 RBAC、DAC 是基于标识的、封闭式的,即访问控制机制只对系统已经存在的用户定义访问权限,在用户提出访问请求时,只有通过权限验证的用户才能对资源进行访问,不支持系统未定义用户的授权^[2]。而在大规模分布式应用中,必须对大量事先无法知晓、无法预先定义的“陌生”用户进行授权,因此传统的访问控制模型已经不能支持大规模分布式应用的这种新需求。

文献[3]提出了信任管理的概念,信任管理将传统安全研究中,尤其是安全授权机制研究中隐含的信任概念抽取出来,并以此为中心加以研究,为解决新环境下的安全问题提供了新的思路。信任管理的最大特点是通过权限传递使资源的用户群能够自动扩展,非常适合解决大规模分布式应用某些特有的安全问题。并且有研究表明,信任管理的能力涵盖并扩展了某些传统访问控制模型如访问控制矩阵、访问控制列表 ACL 等^[4]。因此基于信任管理的访问控制是一种很有发展潜力的新型访问控制方法。

本文主要针对协同环境中面向访问控制的信任评估与追溯问题进行研究,首先介绍分析了一种基于模糊数学的信任评估模型,随后把时间因素引入到该模型中,对其做了必要的完善;最后本文提出了一种信任关系的分布式追溯算法。

1 信任评估模型

合理的、有效的信任评估机制是基于信任的安全机制的基础。

目前,信任模型的主要研究方法有概率论、Dempster-Shafer 理论、模糊数学等,各国研究学者基于这些理论相继提出了几种信任评估模型。其中,比较典型的有 Jøsang 信任模型,文献[5]提出的基于模糊集合理论的主观信任评估模型。文献[6]引入了事实空间和观念空间的概念来描述和度量信任关系,并提供了一套主观逻辑运算符用于信任度的推导和综合计算。但该模型将信任的主观性和不确定性等同于随机性,实际上主观信任作为一种认知现象,其主观性和不确定性主要表现为模糊性,而不是一种随机性,因此这样对信任进行表达不符合信任的实际情况,同时该模型也无法有效地消除恶意推荐带来的影响。文献[5]认识到了信任的模糊性,其提出的主观信任模型把模糊集合理论引入到信任评估中,但此模型中信任值标度的定义没有结合现实世界的语义环境,在实用性和可操作性方面有待进一步研究^[7]。

下面,本文介绍一种以模糊数学为基础的信任评估模型^[7]:该模型在模糊数学的基础上,用多个模糊子集合 $T_j \in F(X)$ ($j = 1, 2, \dots, M$) 定义具有不同信任度的主体集合(简称信任集合)。即用离散标度 $\{1, 2, \dots, N\}$ 来描述主体信任的高低。同时,采用自然语言对 T_j 命名。信任用一个 M 维信任向量来表示,即 $V = \{v_1, v_2, \dots, v_M\}$,其中 v_j 表示 x_i 对 T_j 的隶属度,并且 $v_j \in [0, 1]$ 。

信任的评价过程是一个简单模糊综合评判的过程,有四

收稿日期:2009-01-04;修回日期:2009-03-02。

基金项目:国家自然科学基金资助项目(60573037);国家 863 计划项目(2006AA01Z441)。

作者简介:郎波(1968-),女,辽宁东港人,副教授,博士,CCF 高级会员,主要研究方向:分布式计算、信息安全、数据库;雷凯(1985-),男,重庆人,硕士研究生,主要研究方向:信息安全;江川(1986-),男,山东淄博人,硕士研究生,主要研究方向:信息安全;张静辉(1982-),男,黑龙江肇东人,硕士研究生,主要研究方向:信息安全。

个基本的要素:

- 1) 因素集 $E = \{e_1, e_2, \dots, e_n\}$;
- 2) 因素评价集 $D = \{d_1, d_2, \dots, d_M\}$;
- 3) 因素评价矩阵 $R = (r_{ij})_{n \times M}$;
- 4) 各因素的权重分配 $W = \{w_1, w_2, \dots, w_n\}$ 。

因素集 E 包含的是构成信任类型的所有属性。评价集 D 描述的是对特定主体的属性所作的不同等级的评价。从属性到评价的模糊关系 R 表示对各个因素 e_i 作各种评价的可能性。例如, r_{ij} 就表示对 e_i 做出 d_j 评价的可能性。 W 是一个权重分配, 它表示各因素在评价中的相对重要性。

评价的结果就是信任向量 $V = \{v_1, \dots, v_M\}$ 。信任的简单模糊综合评判就是进行如下的模糊变换:

$$(v_1, \dots, v_M) = (w_1, w_2, \dots, w_n) \circ (r_{ij})_{n \times M} \quad (1)$$

其中“ \circ ”为模糊关系的合成算子。

该信任评估模型充分认识到信任本质的主观性与模糊性, 成功把模糊集合理论应用到信任评估系统中, 具有一定的实用性和可操作性, 初步解决了对具有模糊性的主观信任进行建模的问题。

然而, 该评估模型尚有不足之处, 研究表明, 信任是时间相关的, 即: 随着时间的延续, 信任在逐渐衰退^[8]。这是信任的一个很重要的特征, 该模型没有体现出信任的时间相关性, 致使信任值稳定不变, 与实际情况不符。基于此, 本文对该信任评估模型作了适当的完善, 将时间因素引入到模型中, 并对时间因素的应用方式做了必要的阐述。

1.1 时间衰退函数

实践中, 信任值并不是一成不变的, 信任随着时间在逐渐衰退, 为了描述这样的衰退规律, 本文引进了时间衰退函数^[8]的概念, 并给出了具体的表达形式。

文献[9]认为信任值的衰退符合指数规律, 可以用 $Y(\Delta t) = s^{\Delta t}$ 表示, 其中 Δt 表示时间差, s 表示衰退幅度。例如: 设 $s = 90\%$, $\Delta t = 1$ 年, 则信任的衰退幅度为 $90\%^{1\text{年}}$ 。 s 值越小, 衰退幅度越大。并且随着时间的增加, 信任的衰退规律也在发生变化。时间间隔越长, 信任的衰退速度越快, 即: 随着 Δt 的增加, s 值在减小。为了反应这样的衰退规律, 本文提出采用分段指数函数作为时间衰退函数, 其表达形式如下:

$$Y(\Delta t) = \begin{cases} s_1^{\Delta t}, & 0 \leq \Delta t \leq t_1 \\ s_1^{t_1} \times s_2^{\Delta t - t_1}, & t_1 < \Delta t \leq t_2 \\ s_1^{t_1} \times s_2^{t_2 - t_1} \times s_3^{\Delta t - t_2}, & t_2 < \Delta t \end{cases} \quad (2)$$

其中: Δt 表示当前时间与最近一次交易时间的差值, 函数以此作为输入自变量; s 表示衰退幅度; t_1 和 t_2 分别代表不同的时间长度, 函数以 t_1 和 t_2 作为时间临界值, 区分不同的衰退表达式。该函数的图形曲线如图1所示。由图1可以看出, 随着时间间隔 Δt 的增加, 信任衰退的速度在加快。

1.2 时间衰退函数的作用方式

为了满足不同的计算需求, 时间衰退函数应既可以粗粒度的直接作用于信任向量, 也可以细粒度的单独作用于信任因素, 使时间衰退的计算更加精确。本文提出了两种使用时间衰退函数的方式, 第一种直接作用于信任向量, 第二种作用于信任向量的因素集, 下面分别介绍。

第一种作用方式相对简单, 公式如下:

$$\text{新信任向量} = \text{信任向量} \times \text{时间衰退函数} \quad (3)$$

例如, 设当前信任向量为 $\{0.201, 0.1, 0.075, 0.15, 0.075\}$, 时间衰退函数为 $Y(x)$, 经过时间 Δt 后, $Y(\Delta t) = 80\%$, 则新的信任向量为:

$$\text{新信任向量} = \text{信任向量} \times \text{时间衰退函数} =$$

$$\begin{aligned} & \{0.201, 0.1, 0.075, 0.15, 0.075\} \times Y(\Delta t) = \\ & \{0.201, 0.1, 0.075, 0.15, 0.075\} \times 80\% = \\ & \{0.1608, 0.08, 0.06, 0.12, 0.06\} \end{aligned} \quad (4)$$

第二种情况要相对复杂些, 信任的种类不同, 其随时间衰退的规律也不同, 因此不能用统一的时间衰退函数作用于所有的信任种类。信任评估模型中, 因素集共有5个安全因素, 不同的安全因素反映了信任的不同方面, 各种因素相互独立, 相互区别, 所以应有不同的时间衰退函数。本文提出了时间衰退函数矩阵的方法来满足这样的计算需求。

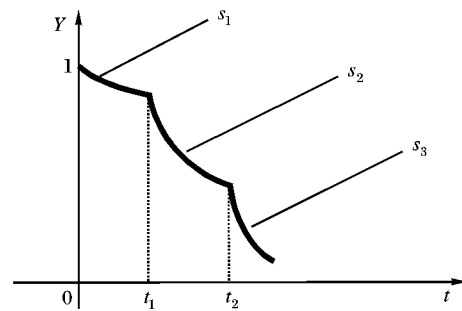


图1 时间衰退函数曲线

目前的信任向量计算公式为:

$$(v_1, \dots, v_M) = (w_1, w_2, \dots, w_n) \circ (r_{ij})_{n \times M} \quad (5)$$

其中: $\{w_1, w_2, \dots, w_n\}$ 是权重分配, $(r_{ij})_{n \times M}$ 为因素评价矩阵。为了让时间衰退函数可以分别作用于各个信任因素, 本文建立了时间衰退函数矩阵, 其形式为:

$$\begin{bmatrix} Y_1(x) & 0 & 0 & \dots & 0 \\ 0 & Y_2(x) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & Y_n(x) \end{bmatrix}$$

其中 $Y_1(x)$ 、 $Y_2(x)$ 与 $Y_n(x)$ 为时间衰退函数。则新的信任向量公式为:

$$(v_1, \dots, v_M) = (w_1, w_2, \dots, w_n) \times \begin{bmatrix} Y_1(x) & 0 & 0 & \dots & 0 \\ 0 & Y_2(x) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & Y_n(x) \end{bmatrix} \circ (r_{ij})_{n \times M} \quad (6)$$

其中: 衰退函数 $Y_n(x)$ 对应因素 r_n (即: $Y_1(x)$ 作用于 r_1 , $Y_2(x)$ 作用于 r_2)。

例如, 设当前的因素评价矩阵 R 为:

$$R = \begin{bmatrix} 0 & 0.25 & 0.5 & 0.25 & 0 \\ 0 & 0 & 0 & 0.33 & 0.67 \\ 0.25 & 0.5 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0.5 & 0.25 \\ 0.67 & 0.33 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

权重分配矩阵为: $\{0.15, 0.05, 0.2, 0.3, 0.3\}$ 。

时间衰退函数矩阵为:

$$\begin{bmatrix} Y_1(x) & 0 & 0 & 0 & 0 \\ 0 & Y_2(x) & 0 & 0 & 0 \\ 0 & 0 & Y_3(x) & 0 & 0 \\ 0 & 0 & 0 & Y_4(x) & 0 \\ 0 & 0 & 0 & 0 & Y_5(x) \end{bmatrix}$$

经过时间 Δt 后, 时间衰退函数分别为: $Y_1(\Delta t) = 0.90$,

$Y_2(\Delta t) = 0.95, Y_3(\Delta t) = 0.87, Y_4(\Delta t) = 0.90, Y_5(\Delta t) = 0.80$ 。则新的信任向量计算过程是:

$$\begin{aligned} (v_1, \dots, v_M) &= (w_1, w_2, \dots, w_n) \times \\ &\begin{bmatrix} Y_1(x) & 0 & 0 & \dots & 0 \\ 0 & Y_2(x) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & Y_n(x) \end{bmatrix} \circ \\ (r_{ij})_{n \times M} &= (0.15, 0.05, 0.2, 0.3, 0.3) \times \\ &\begin{bmatrix} 0.90 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.95 & 0 & 0 \\ 0 & 0 & 0.87 & 0 & 0 \\ 0 & 0 & 0 & 0.90 & 0 \\ 0 & 0 & 0 & 0 & 0.80 \end{bmatrix} \circ R = \\ &= \{0.161, 0.087, 0.067, 0.135, 0.067\} \end{aligned} \quad (8)$$

没有经过衰减计算所得的信任向量为 $\{0.201, 0.1, 0.075, 0.15, 0.075\}$, 比较上面的信任向量, 可以看出结果的衰减程度。再比较第一种作用方式下计算得到的信任向量 $\{0.1608, 0.08, 0.6, 0.12, 0.6\}$, 可以看出不同的计算方式得到的不同衰减结果。

2 信任追溯

信任关系的拓扑发现是进行信任评估和决策的基础。信任发现的实质就是寻找实体之间的权限传递关系。在现有的信任关系查找算法研究中, 主要采用从被请求者或请求者出发查找信任传递的路径集中式发现方法。这种集中式的信任关系发现算法的特点是简单直接, 但在大规模分布式协同系统中, 每个实体接收到的访问请求以及对外发出的访问请求的数目都比较庞大, 这种集中式方法将导致请求实体或被请求实体的负载过重, 从而影响整个系统的运行效率。

本文提出一种分布式信任追溯算法, 其基本思想是: 以访问请求者为出发点, 逐级按照信任传递的反向进行追溯, 直到信任传递的根节点——资源。该方法不仅符合现实世界基于信任的决策模式, 还将信任拓扑结构发现的复杂操作分布在多个相关实体上, 有效避免了访问控制中效率瓶颈的产生。

2.1 信任追溯算法

为了更好的分析和解决问题, 本文将协同环境划分为若干个独立的自治域, 每个自治域管理若干数量的实体, 这些实体被称为信任实体。同时, 每个自治域设立一个信任代理, 负责处理与信任机制相关的各种操作, 信任评估、信任追溯等工作都是由信任代理完成的, 信任代理是信任机制中的重要角色。

每个信任代理维护一张权限接收表, 在信任传播与追溯过程中, 权限接收表起着非常重要的作用。权限接收表记录了各种资源的授权情况, 资源提供者把访问权限提供给某些信任代理, 后者根据相应规则继续授权给其他的信任代理, 授权关系均被记录在权限接收表中。其表的结构如下所示:

资源标识	资源提供者	资源传递者

其中: 资源标识唯一确定了一个资源; 资源提供者记录了该资源所属的域的信任代理; 资源传递者记录了该资源使用权限的传递者。资源提供者与资源传递者相同时, 表示资源

提供者直接授权给本信任代理; 两者不同时, 表示权限是通过其他信任代理传递过来的。

为了说明算法的工作原理, 这里假设了如图2所示的协同环境, 图中每个节点代表一个自治域的信任代理, 箭头方向表示资源的授权方向。如: $F \rightarrow B, F$ 把资源的访问权限授权给 B , B 又根据相应的规则将该权限传递给 C , 在 B 的权限接收表中记录了权限的传递过程, 如图2所示。

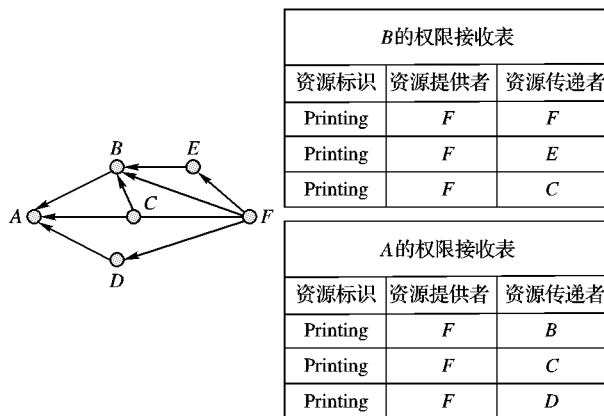


图2 协同环境信任传播示例

现在假设 A 是请求方, F 是资源方, A 需要访问 F 的资源。由图2可以看出, A 与 F 之间不存在直接联系, 因此 F 不能根据自身的经验判断 A 的信任水平, 然而 A 与 F 之间存在间接信任关系, F 可以根据其他自治域对 A 的评价来综合决定是否授权 A 的访问请求。信任追溯的任务, 就是把其他自治域对 A 的信任关系发掘并收集起来, 统一到 F 处, 由 F 根据相应规则决定是否授权。

根据分布式信任追溯的思想, 追溯将以访问请求者 A 为出发点, 逐级按照信任传递的反向进行追溯。 A 查询自身的权限接收表知, F 的访问权限通过 B, C, D 被传递过来, 于是 A 发送访问请求到 B, C, D , 收集它们对 A 的信任评价, 如图3(a)。

以 B 为例, B 收到 A 的请求信息后, 将 B 对 A 的信任评价整合到该请求中, 继续向下一级发送, 收集其他信任代理对 A 的评价。由 B 的权限接收表知, B 将请求信息发送到 C, E, F 处, 如图3(b)。

每个信任代理均执行与 B 相同的操作, 直到追溯的终点——资源方 F 。当所有的请求信息都到达 F 后, 便形成了如图3(d)所示的网状追溯轨迹。 F 会根据相应的规则推理计算出 A 的信任等级, 并以此为依据决定是否授权请求。

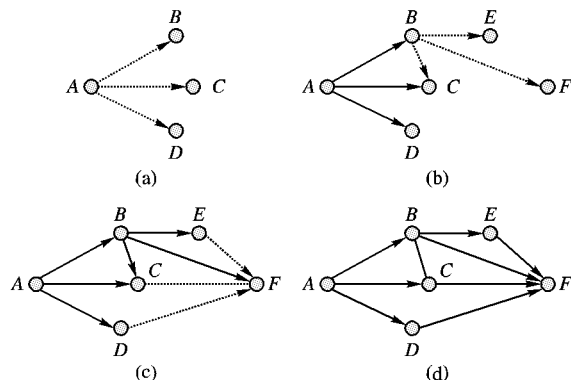


图3 信任追溯过程示例

2.2 信任追溯中的问题及解决办法

2.2.1 信任链环路

如上节介绍, 资源请求信息从 A 出发有许多路径可以到

达 F , $ABEF$ 就是其中之一,它标注了请求信息从请求方到资源方的一条链状轨迹,这样的链状轨迹称为信任链。 ACF 、 ADF 、 $ABCF$ 等都是图3(d)中存在的信任链。

信任链不仅给出了请求方到资源方的路径信息,还记录了沿途收集到的信任关系,如信任链 $ABEF$ 。

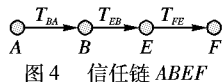


图4 信任链 $ABEF$

图4中: B 、 E 是信任链沿途经过的信任代理, T_{BA} 、 T_{EB} 、 T_{FE} 是信任代理之间信任关系的具体表达,即信任向量。

同一个资源请求,经历了不同的路径,形成了不同的信任链,收集了不同的信任关系,每条信任链都代表了不同的节点对请求方的信任评价,资源方正是根据这些不同的评价综合计算出请求方的信任等级,以此作为决策的依据。

信任链中有可能存在回路,如图5所示,这是一个信任追溯的过程图,其中 BDC 构成了环路,请求信息将永远在 BDC 之间循环发送。

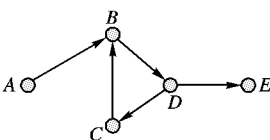


图5 信任链环路

为了避免信任链环路的出现,每个信任代理应判断到达的信任链是否已经经过自身,如果经过,则应将该信任链丢弃以避免环路的形成。

2.2.2 请求时间过长

如图6所示,信任链 $ABCDEH$ 可能需要经过很长时间才能到达最终的资源方 H ,在此之前,信任链 AFH 、 AGH 可能早已到达,那么 H 没有必要持续等待信任链 $ABCDEH$ 。过晚的信任链已经失去参考价值。

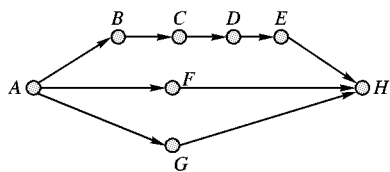


图6 $ABCDEH$ 请求时间过长

为了解决该问题,资源方在接到第一条信任链时,启动计时,设置必要的等待时间,超过该时间到达的所有同类信任链全部丢弃。同时,为了唯一区分信任链所属的访问请求,每个

访问请求生成一个随机序列,由请求方生成,包含在该请求所属的所有信任链中,资源方以此判断信任链的类别。

3 结语

信任评估与信任追溯是信任管理中两个重要的组成部分。信任评估是信任研究的基础,合理的、有效的信任评估机制是基于信任的安全机制的核心和引擎。本文介绍了一种基于模糊数学的信任评估模型,并对其进行了完善,将时间因素引入到该模型中,使其更符合实际应用。本文还根据信任关系发现已有的研究成果,提出了分布式信任追溯算法,其将追溯过程中的复杂操作分散到各个节点上,有效避免了访问控制中效率瓶颈的产生。

参考文献:

- [1] FOSTER I, KESSELMAN C, TUECKE S. The anatomy of the grid: Enabling scalable virtual organizations [C]// International Journal Supercomputing Applications, 2001, 15(3): 200 - 222.
- [2] ABADI M, BURROWS M, LAMPSON B. A calculus for access control in distributed systems [J]. ACM Transactions on Programming Languages and Systems, 1993, 15(4): 706 - 734.
- [3] BLAZE M, FEIGENBAUM J, LACY J. Decentralized trust management [EB/OL]. [2008 - 10 - 10]. <http://www.cs.utsa.edu/~winsboro/teaching/CS6463-S06/Papers/BFL96.pdf>.
- [4] CHANDER A, DEAN D, MITCHELL J C. A state-transition model of trust management and access control [C]// CSFW'01: Proceedings of 14th IEEE Computer Security Foundations Workshop. Cape Breton, Nova Scotia, Canada: IEEE Press, 2001: 27 - 43.
- [5] 唐文, 胡建斌, 陈钟. 基于模糊逻辑的主观信任管理模型研究 [J]. 计算机研究与发展, 2005, 42(10): 1654 - 1659.
- [6] JOSANG A, HAYWARD R, POPE S. Trust network analysis with subjective logic [C]// Proceedings of the Australasian Computer Science Conference: ACSC06. Hobart, Australia: Australian Computer Society, 2006: 85 - 94.
- [7] 赵莹. 面向访问控制的信任评估研究与实现 [D]. 北京: 北京航空航天大学, 2006.
- [8] AZZEDIN F, MAHESWARAN M. Evolving and managing trust in grid computing systems [C]// CCECE'02: IEEE Canadian Conference on Electrical and Computer Engineering. Winnipeg: IEEE Press, 2002: 1424 - 1429.
- [9] HUYNH T D, JENNINGS N R, SHADBOLT N. On handling inaccurate witness reports [C]// Proceedings of 8th International Workshop on Trust in Agent Societies. Utrecht, Netherlands: [s. n.], 2005: 63 - 77.

(上接第1635页)

5 结语

LSB 匹配数字隐写是隐写分析中的难点问题,尤其是对未压缩的高精度原始图像的检测。本文将空域图像的数字隐写过程看作是一种被加性噪声污染的图像退化过程,把自适应阈值小波滤波应用于 LSB 匹配隐写分析,并且结合图像滤波前后直方图/邻接直方图特征函数质心差异性特征,设计了一种不仅对经过 JPEG 压缩过的图像能够进行有效检测,而且对未压缩的高精度原始图像也具有一定检测能力的 LSB 匹配隐写分析方法。实验结果表明,本文所提出的方法在性能上要优于 Ker 的校准法,尤其对未压缩的高精度原始图像的检测更为有效。与文献[3]相比,克服了事先假定嵌入率得到噪声方差所带来的性能影响,方法简单,易于实现。

参考文献:

- [1] HARMSSEN J J, PEARLMAN W A. Steganalysis of additive noise modelable information hiding [EB/OL]. [2008 - 10 - 10]. http://www.cipr.rpi.edu/~harnsj/pubs/harmsen_ms.pdf.
- [2] KER A D. Steganalysis of LSB matching in grayscale images [J]. IEEE Signal Processing Letters, 2005, 12(6): 441 - 444.
- [3] 王国新. LSB 匹配隐写分析和隐藏信息盲检测技术研究 [D]. 郑州: 信息工程大学, 2007: 35 - 42.
- [4] 赵瑞珍, 宋国乡. 一种基于小波变换的白噪声消噪方法的改进 [J]. 西安电子科技大学学报, 2000, 27(5): 619 - 622.
- [5] University of Washington CBIR image database [DB/OL]. [2008 - 09 - 14]. <http://www.cs.washington.edu/research/magedatabase/roundtruth/>.
- [6] The USDA NRCS photo gallery [DB/OL]. [2008 - 09 - 14]. <http://photogallery.nrcs.usda.gov>.