信任模型中搭便车节点的抑制

许晓东[1] 邹宝军[2] 朱士瑞[1]
(江苏大学网络中心 镇江 212013)[1] (江苏大学计算机与通信工程学院 镇江 212013)[2]

摘 要 现有的信任机制虽然有效地遏制了P2P系统中节点的恶意攻击,但未考虑如何抑制内在的大量搭便车节点的存在,即高信任值节点向搭便车节点的转变。大量搭便车节点的存在,降低了P2P网络的健壮性及可用性,为此设计了基于时间窗口的信任模型,对距现在时刻越近的时间窗口给予的权重值越大。仿真结果表明,机制不仅能够有效抑制大量高信任值节点向搭便车节点的转变,而且与提出的其它方案相比,能够有效遏制节点近期内进行恶意攻击,并且节点的信任值会更高。

关键词 信任模型,节点抑制,搭便车,P2P网络,恶意攻击
中图法分类号 TP393.08 **文献标识码** A

Reducing Number of Free Riders in Trust Model

XU Xiao-dong[1] ZOU Bao-jun[2] ZHU Shi-rui[1]
(Network Center, Jiangsu University, Zhenjiang 212013, China)[1]
(College of Computer and Communication Engineering, Jiangsu University, Zhenjiang 212013, China)[2]

Abstract Although current existing trust model reduces the impact of peers' malicious attack in P2P system, these reputation systems do not concern with reducing free riders in P2P networks, namely, peers with high trust value tansform to free riders. Because lots of free riders exist, robust and avaliable of P2P system are degraded. So, a trust model based on time slots was designed, and the nearer time from now the greater time slot's weight was gaven. Theoretical analysis and simulation show that this proposal not only prevents peers with high trust value to free riders, but also limits peers to make malicious attack recently, and peers' trust value are more and more hight in compared with other proposals.

Keywords Trust model, Peer reduction, Free rider, P2P networks, Malicious attack

1 引言

在P2P文件系统中,节点通过共享文件等方式,从其他节点获得所需的文件。然而大量节点只下载所需的文件,而不上传文件,即所谓的搭便车节点。这些节点的大量存在,使得文件增长及更新速度缓慢。随着时间的流逝,诚实节点发现有价值的文件越来越少,最终离开系统,导致网络的可用性变差,甚至崩溃。

搭便车节点不愿共享资源,主要原因有:

1)带宽的消耗。

2)自身硬件资源的消耗。例如CPU需要不断地处理请求,甚至占用大量的内存及硬盘空间。

其根本原因就是自私行为。

现有的信任模型通过节点的历史服务,估计节点的信任值,把恶意节点与诚实节点区分开来。节点获取资源前,首先获得所有服务提供者的信任值,通常选择信任值较高的节点来作为服务提供者,以此来避免恶意节点的攻击。然而,此过程存在一个严重的问题:节点享受服务会选择信任值较高或最高的节点,信任值越高的节点处理其他节点的请求会越来越繁重,并且不间断地提供服务,最终导致此节点的带宽或硬件资源被耗尽,对部分节点提供的服务可能中断,造成这些节点不能获得所需的服务,这些节点会对此节点提供不满意的评价,导致此节点的信任值会有相应程度的降低,迫使这些具有高信任值的节点不再提供服务或只响应少量的服务请求。然而,这些高信任值节点仍具有较高的信任值,享受任何服务都没有问题,此时这些高信任值节点转变为搭便车节点。因此,P2P信任模型的主要问题不仅仅是安全问题,还包括高信任值节点向搭便车节点的转变(含部分节点达到享受服务的信任值底限,直接成为搭便车节点)。这种现象的存在违背了P2P系统当初的设计理念。

针对上述问题,本文设计了基于时间窗口的信任模型TMTS(Trust Model Based on Time Slots),它不仅能够有效遏制高信任值节点向搭便车节点的转变,而且节点的信任值更高。

2 相关工作

EigenTrust[1]算法是一种利用信任的传递特性,由直接信任值计算全局信任值的信任算法。EigenTrust提出,直接

到稿日期:2011-04-22 返修日期:2011-07-14 本文受国家自然科学基金(61005017)资助。
许晓东(1965-),男,副教授,主要研究方向为网络管理、系统集成,E-mail:xdxu@ujs.edu.cn;邹宝军(1985-),男,硕士生,主要研究方向为信任模型、网络安全,E-mail:baojun666@126.com(通信作者)。

信任值越高的节点推荐的信任值越可信，在计算全局信任时赋予较大权重[10]。但 EigenTrust 系统可扩展性差，全局信任值收敛速度较慢，还需要预置可信节点，这在 P2P 系统中是不可行的；并且没有对恶意节点作出相应的惩罚。文献[3]对提供不好的服务进行了惩罚，促使节点提供良好的服务。文献[2]采用 EigenTrust 信任模型，根据节点信任值分配相应的带宽和 TTL 等激励机制，来遏制搭便车节点，但不能解决本文提到的问题。

PeerTrust 信任模型[4]引入了更多的信誉评价因素，对于信誉的计算表现得更为合理；使用平方根方法计算该节点与另外一个节点评价的相似程度，并作为对另一个节点的反馈评价的权重因子，降低恶意节点提供不诚实反馈评价的影响。此方法会面临公共交互节点集合很小的问题，在计算节点可信度时会引起较大的误差。文献[5]提出了基于向量相似性的信任模型，它遏制了恶意团体提供不诚实评价的问题，解决了公共交互节点集合很小的问题。TrustGuard[6]提出信誉值的计算需同时考虑节点的历史信息和节点近期行为的突然改变，并采用记忆衰减来降低节点的历史信息维护代价。PowerTrust[7]算法通过分析 eBay 中的评价信息发现节点间的评价存在的幂律关系，即存在少数 Power 节点，它们得到的评价数量显著地多于其它节点。PowerTrust 把这些 Power 节点组成可信节点集合[10]，证明了幂律分布可应用于任何动态增长的 P2P 系统；并采用向前看随机游走(LRW)策略，来聚合近期信任值和更新节点的全局信任值。PowerTrust 在计算全局信任值的精确性和收敛速度方面有着重大的改进。然而，这些信任模型都没有考虑高信任值节点向搭便车节点转变的问题。

文献[8]指出信任模型应该考虑时间因素，采用时间窗口机制计算节点的直接信任值，来激励不活跃的节点。但直接信任值的计算存在严重的偏差，并且没有仿真实验对其验证。

本文采用了上述模型的优点，并对其不足之处进行了改进，以解决引言提出的问题。信任数据存储基于 DHT，其安全存放问题参见文献[9]，本文不述。

3 模型设计

3.1 TMTS 设计

本文采用相同大小的时间窗口，其大小可根据具体的应用场景来确定，即 Δ_t，$1\leqslant t\leqslant n$，时间窗口按次序从 1 开始，数字越大表示距离现在越近。

在第 t 个时间窗口中，节点 i 对节点 j 的信任评价通过直接信任和间接信任获得。直接信任通过两个节点直接交互的经验获得，间接信任从其它节点对节点 j 的反馈评价获得，节点 i 对节点 j 的信任评价记为 T_{ij}，使用下式进行定义：

$$T_{ij}=\alpha\times D_{ij}+\beta\times R_{ij}+\gamma\times\frac{B}{N_B},\alpha+\beta+\gamma=1 \tag{1}$$

式(1)由 3 部分组成：第一部分是自己和节点 j 所提供服务的直接信任值 D_{ij} 的计算，第二部分 R_{ij} 是其他节点与节点 j 提供服务的直接信任值 D_{kj} 的计算，最后一部分对积极提供评价的节点给予一定的奖励。其中，α,β,γ 为权重因子，通常 $\alpha>\beta>\gamma$；B 指节点 j 提供评价的次数，N_B 指节点 j 享受服务的总次数。

节点通常不愿提供一次较长时间的服务，因为只能得到一次评价，所以需对提供较长一次良好服务的节点进行奖励，如式(2)中满意的情况。

定义 1 A_{ij}^k 表示在第 k 次服务中，节点 i 对节点 j 所提供的服务是否满意，即

$$A_{ij}^k=\begin{cases}\left\lfloor\frac{H}{L}\right\rfloor+1, & \text{满意}\\ -1, & \text{不满意}\end{cases} \tag{2}$$

式中，L 为系统指定的最长服务时间，H 指节点 j 对节点 i 的一次服务时间。

定义 2 $D_{ij,t}^{(T)}$ 表示在第 t 个时间窗口中节点 i 对节点 j 的直接信任值计算，即

$$D_{ij,t}^{(T)}=\frac{S_{ij,t}}{S_{ij,t}+P\times F_{ij,t}+C},P<-1,C>1 \tag{3}$$

式中，$S_{ij,t}$ 指在第 t 个时间窗口中节点 i 对节点 j 所提供良好的服务次数的累加(含奖励次数)，$F_{ij,t}$ 指在第 t 个时间窗口中节点 i 对节点 j 不满意次数的累加；P 为惩罚因子，对不诚实的服务作出惩罚；C 为常数，制止节点在每个时间窗口中提供一次及几次良好的服务，其信任值立即为 1。C 值较大时，限制节点信任值较快地增长而接近为 1，因此节点需要一定数量的良好服务才能获得信任值的持续增加。

对离现在越近的时间窗口给予的权重值越大。例如，在 P2P 文件共享系统中，节点提供的文件资源不断地完善，提供越来越可靠的带宽。相反，近期内节点提供的文件中含有木马或提供不可靠的带宽，节点此时所提供的服务能够与过去相同吗？再举一通俗的例子：若干年前，某人与他的朋友各自开了一公司，两公司进行业务往来，完全互相信任。此后不再进行业务合作，近期两公司再次相互合作，能够完全相互信任吗？因此，对近期的时间窗口给予的权重值较大，一方面可以有效遏制搭便车节点的大量存在，迫使搭便车节点需要经常提供良好的服务，信任值才能维持在一定的范围内。另一方面可以有效抵御恶意节点的攻击，即抵御恶意节点积累一定的信任值后于近期内进行恶意攻击。

定义 3 D_{ij} 表示节点 i 对节点 j 的直接信任值计算，即

$$D_{ij}=t\times\sum_{t=1}^{n}D_{ij,t}^{(T)}\times\theta_t,\sum_t\theta_t=1 \tag{4}$$

式中，θ_t 为第 t 个时间窗口的权重因子。近期的时间窗口对应的 θ_t 值较大，并且所有权重因子的和为 1。n 值较大时，将长期存在的时间窗口的权重因子值置为 0，其内的信任值不再存储，并释放存储空间。本文实验中时间窗口对应的权重因子的取值如算法 1。

算法 1 weightCreate(N)

```
输入:N
输出:权重因子序列
begin
    for i=0 to i<N-1 do
        Que[i]=Random()/(N-1);
        sumQue=sumQue+Que[i];
    end for
    Que[N-1]=1-sumQue;
    tempRand=Que[N-1]/N;
    for j=0 to j<(N-1)/2-1 do
        Que[j]=Que[j]+ 2 * tempRand;
```

```
    end for
    for k=0 to <N-1 do
        sumTemp=sumTemp+Que[k];
    end for
    Que[N-1]=1-sumTemp;
    Sort(Que); //由小到大排序
end
```

定义 4 R_{ij} 指节点 i 对节点 j 的间接信任值计算，即

$$R_{ij}=\frac{1}{|S_j^{(\alpha)}|}\sum_{k\in S_j^{(\alpha)}且k\neq j} D_{kj}\times S_{ik}^{(im)} \tag{5}$$

式中，$S_j^{(\alpha)}$ 指与节点 j 交互的节点集合不包括 i 节点；$S_{ik}^{(im)}$ 指节点 i 与节点 k 评价的相似程度，其值越大，表示评价的相似程度越接近，节点 k 越可信，使用下式进行定义：

$$S_{ik}^{(im)}=\vec{I}_{set}\times\vec{K}_{set}=\frac{\sum_{\forall C_n\in U}(\vec{V}_{IC_n}\times\vec{V}_{KC_n})}{\sqrt{\sum_{\forall C_n\in U}(\vec{V}_{IC_n})^2}\times\sqrt{\sum_{\forall C_n\in U}(\vec{V}_{KC_n})^2}} \tag{6}$$

式中，$U=\{C_1,C_2,C_3,\cdots,C_n\}$ 指节点 i 和节点 k 的并集，采用并集的证明参见文献[5]；$\vec{I}_{set}=[\vec{V}_{IC_1},\vec{V}_{IC_2},\vec{V}_{IC_3},\cdots,\vec{V}_{IC_n}]$ 指节点 i 对每个 C_i 的评价值所构成的向量；$\vec{K}_{set}=[\vec{V}_{KC_1},\vec{V}_{KC_2},\vec{V}_{KC_3},\cdots,\vec{V}_{KC_n}]$ 指节点 k 对每个 C_i 的评价值所构成的向量。

3.2 其它降低搭便车节点的方案描述

方案 1　引入时间衰减因子的信任模型

信任模型引入时间衰减因子后，随着时间的流逝，所有节点的信任值会逐渐缓慢地降低。如果节点经常地提供良好的服务，其信任值呈缓慢增长趋势，迫使节点增加服务次数，而搭便车节点的信任值将处于一直缓慢的降低状态。此方案设计如下：只需不考虑 TMTS 中式(3)的参数 t，并重新定义式(4)，即

$$D_{ij}=\varphi\times D_{ij}^{(T)},0<\varphi<1 \tag{7}$$

式中，φ 为时间衰减因子。

缺点：正常节点需要额外提供几次或数次良好的服务，才能达到不引入时间衰减因子的信任模型中同一节点的信任值。

方案 2　基于单支付的信任模型

节点每享受一次服务，需要扣除一定的信任值。被扣除的信任值因所享受的服务不同而异，而提供服务的节点不能得到此享受服务的节点所扣除的信任值，与微支付有本质的区别；被扣除的信任值与评价信息一块存储。此方案设计如下：对 TMTS 中的式(3)修改如下：

$$D_{ij}=\frac{S_{ij}}{S_{ij}+P\times F_{ij}+C},P<-1,C>1 \tag{8}$$

忽略式(4)，并定义了节点享受服务时所扣除的信任值后的公式，即

$$T_{ij}^{(k)}=T_{ij}-\sum\omega,0<\omega<0.1 \tag{9}$$

式中，ω 指享受一次服务时所需支付的信任值，ω 的倍数(倍数大于 1)可抵消节点提供一次良好的服务所获得的真实评价，相互抵消的两部分不再存储。

缺点：(1)需要额外的存储空间存储被扣除的信任值；

(2)节点享受服务后，对其信任值的计算会存在一定的误差。

方案 3　采用奖励机制的信任模型

此方案的本质是对积极提供良好服务的节点提供更快的下载速度或更高享受服务优先级等有差别的服务。例如，文献[2]根据信任值大小分配相应的带宽和 TTL，但没有降低信任值的因素，不能解决本文提到的问题；文献[11]把所需要下载的文件分为 5 个等级，节点的信任值越高，可下载等级越高的文件。如果节点在一个时间段中没有提供服务，则它的信任值会有相应程度的降低，该文献奖励机制的本质是：节点的信任值越高，权限越大，可获取的文件资源越多。本方案采用文献[11]的模型设计，无需另行设计。

TMTS 较上述 3 种方案最大的优点是：遏制恶意节点积累一定的信任值，于近期内进行恶意的攻击，并且节点的信任值更高。

4　仿真及结果分析

本文仿真基于文件共享 P2P 网络的查询周期模型[15,16]，采用 Java 代码实现。仿真环境设置如表 1 所列。

表 1　仿真环境设置

描述			缺省值
网络	拓扑构造		Power Law
	节点数目		500 个
	共享文件数目		5000 个
	搭便车节点比例		20%
	恶意节点比例		20%
	节点度		3
	TTL		4
	时间窗口大小		5 个 Cycle
节点	正常节点	活动状态	100%
		进行响应	有匹配进行响应
		转发请求	100%
		文件提供	95%可信
		请求文件	随机
	搭便车节点	活动状态	100%
		进行响应	不响应
		转发请求	不转发
		文件提供	不提供
		请求文件	随机
	恶意节点	活动状态	100%
		进行响应	100%
		转发请求	100%
		文件提供	100%不可信
		请求文件	随机

规定节点初始信任值为 0，信任值达到 0.5 才能享受服务。对于新节点，式(3)中 C 值较小，需要提供数次良好的服务，信任值就可以达到 0.5。信任值达到 0.5 后，C 值较大，节点的信任值以较慢速度增长。本文为了降低实验的复杂性，对信任值达到 0.5 的节点，定义为高信任值节点。

本文对 TMTS 与不考虑时间窗口的信任模型中的搭便车节点和正常节点的信任值做了对比，如图 1 所示。

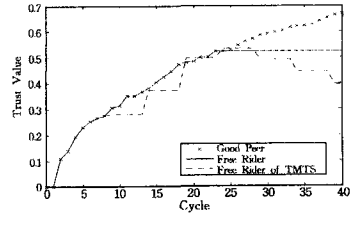

图 1　节点信任值变化情况

正常节点信任值达到0.5后，信任值增长速度有所降低；不考虑时间窗口的信任模型，节点在第22个周期信任值达到0.5后，不再提供服务，而转变成搭便车节点，其信任值一直处于稳定状态；TMTS注重近期的服务，在第24个周期才达到0.5，如果此时转变为搭便车节点，其信任值处于缓慢下降状态，节点需要经常提供良好的服务，其信任值才维持在0.5左右。

为了证明TMTS能够有效遏制大量高信任值节点向搭便车节点的转变，下面对TMTS与EigenTrust信任模型中搭便车节点的数量做了对比。搭便车节点初始值为100个(这部分搭便车节点向非高信任值节点转变)，如图2所示。

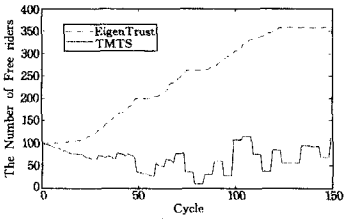

图2 搭便车节点数量变化情况

如图2所示，EigenTrust信任模型中搭便车节点的数量以较快的速度增长，在第120个周期时，搭便车节点的数量不再增长，此时系统只剩下极少部分诚实的节点维持系统的运行，即EigenTrust系统预设的可信节点。在TMTS的开始阶段，搭便车节点的数量呈缓慢降低趋势，因为真正的搭便车节点信任值逐渐降低，并且高信任值节点转变成搭便车节点的数量也较少。此后，搭便车节点的数量处于相对稳定的波动状态，即搭便车节点的数量在50上下以较小的范围波动，有效遏制了高信任值节点大量向搭便车节点的转变。但此时为什么还有少量的搭便车节点存在？为此，对TMTS中的部分节点进行了分析，发现存在以下的有趣现象，如图3所示。

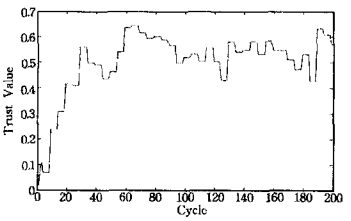

图3 高信任节点信任值的波动

图3显示部分高信任节点的信任值在0.5处上下波动。如果高信任节点发现信任值低于0.5，就提供良好的服务，以获取一定的信任值。当信任值达到一定的值后，不再提供服务。如此周而复始，即高信任值节点处于与搭便车节点的相对转化状态。高信任值节点为避免自身带宽及硬件资源被大量或长期占用，短暂地处于搭便车节点状态，能够有效地缓解网络负载极不均衡的问题。文献[12]指出，至今多数对等网络软件开发者依然容忍搭便车现象存在，并提出了3个需要考虑的方面。因此，系统可以有少量的搭便车节点存在。

为了突出TMTS信任模型能够有效遏制恶意节点于近期内进行恶意攻击方面的优势，提出了3种降低搭便车节点的方案与其作对比，实验结果如图4所示。

TSTM与其它3种方案在同一周期时，信任值偏低，因为TSTM采用时间窗口机制，信任值变化不如其它3种方案灵敏，并且对数字越大的时间窗口给予的权重值越大，较依赖近期的服务情况；遏制了节点具有较高的信任值时，立即进行恶意的攻击，并且节点的信任值更加可信。然而，在第180个周期时，TSTM在遏制节点进行恶意攻击方面逐渐变弱。因此，当时间窗口的数字较大时，逐步加大近期的时间窗口的权重值，并比较数字较小的时间窗口对应的权重值是否高于系统设置的最低权重值。如果低于指定的权重值，不再存储数字较小的时间窗口的相关信息。

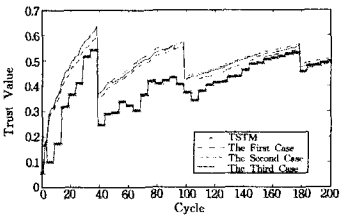

图4 在恶意攻击情况下节点信任值的变化

结束语 鉴于现有信任模型不能有效遏制高信誉节点向搭便车节点的转变，本文在此基础上定义了时间窗口，即更注重近期的服务，对近期的时间窗口给予的权重值较大，迫使节点需要经常提供良好的服务，信任值才能维持在较高的水平。同时提出了其它解决方案，但这些解决方案在抵御恶意攻击方面显得较差。仿真实验证明，本方案可有效降低信任模型中搭便车节点的大量存在，使其数量保持在相对较低的状态，并且节点的信任值更高。

参 考 文 献

[1] Kamvar S D, Schlosser M T, Garcia-Molina H. The EigenTrust Algorithm for Reputation Management in P2P Networks[C]// Proceedings of the 12th International Conference on World Wide Web. Budapest, Hungary: ACM Press, 2003: 640-651

[2] Kamvar S D, Schlosser M T, Garcia-Molina H. Incentives for Combatting Freeriding on P2P Networks[C]// European Conference on Parallel Processing. Lecture Notes in Computer Science. Berlin, Germany: Springer-verlag, 2003: 1273-1279

[3] Yuhua L, Yuling L, Wei C, et al. The Research Based on Trust Value Against Vulnerability in P2P Networks[C]// Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology. USA: IEEE, 2009: 85-89

[4] Xiong L, Liu L. PeerTrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(7): 843-857

[5] Leitao G, Shoubao Y, Jing W, et al. Trust Model Based on Similarity Measure of Vectors in P2P Networks[C]// Grid and Cooperative Computing-GCC. Lecture Notes in Computer Science. Berlin, Germany: Springer-verlag, 2005: 836-847

[6] Srivatsa M, Xiong L, Liu L. Trustguard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks[C]// Proceedings of the 14th International World Wide Web. New York, USA: ACM Press, 2005: 422-431

[7] Zhou Run-fang, Hwang Kai. PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2007,

18(4):460-473

[8] Spitz S, Tuchelmann Y. A Trust Model Considering the Aspects of Time[C]// Proceedings of the Second International Conference on Computer and Electrical Engineering(ICCEE). USA: IEEE Computer Soc, 2009:550-554

[9] Srivatsa M, Liu L. Vulnerabilities and security issues in structured overlay networks: A quantitative analysis[C]// Proceedings of the 20th Annual Computer Security Applications Conference(ACSAC). USA: IEEE Computer Society, 2004:252-261

[10] 李勇军,代亚非. 对等网络信任机制研究[J]. 计算机学报,2010,33:390-405

[11] Yuh-Min T, Fu-Gui C. A free-rider aware reputation system for peer-to-peer file-sharing networks[J]. Expert Systems with Applications, 2011, 38(3):2432-2440

[12] 余一娇,金海. 对等网络中的搭便车行为分析与抑制机制综述[J]. 计算机学报,2008,31(1):1-15

[13] 常俊胜,王怀民,尹刚. DyTrust:一种 P2P 系统中基于时间帧的动态信任模型[J]. 计算机学报,2006,29(8):1301-1307

[14] Karakaya M, Korpeoglu I O. Free riding in Peer-to-Peer Networks[J]. IEEE Internet Computing, 2009, 13(2):92-98

[15] Query Cycle Simulator [EB/OL]. http://p2p.stanford.edu/www/qcsim.htm

[16] Schlosser M, Condie T, Kamvar S. Simulating a File-Sharing P2P Network[C]// Proceedings of the 1st Workshop on Semantics in Grid and P2P Networks. USA: Stanford Inforlab Publication Server, 2003

[17] 胡建理,吴泉源,周斌. P2P 环境下基于信誉的信任模型研究[J]. 计算机科学,2009,36(9):1-6

(上接第 78 页)

四方的安全电子商务支付协议满足金钱原子性、商品原子性以及确认发送原子性。

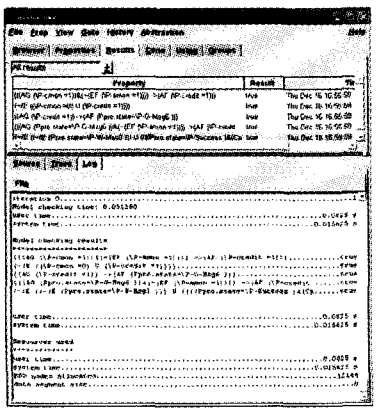

图 6 基于四方的安全电子商务支付协议实验检测结果

本次实验利用符号模型检测工具 SMV 对基于四方的安全电子商务支付协议进行原子性安全检测,证明了协议具备电子支付原子性安全需求,存在很大的实际意义。基于四方的安全电子商务支付协议首次实现传统商品在电子商务支付活动中的原子性。而据调查显示,人们在网络购物活动中,支付后收不到商品、假货的存在以及退换货的不便是影响网民网上购物热情的重要因素。而基于四方的安全电子商务支付协议解决了电子商务支付安全的原子性问题,彻底改变了客户在电子商务退换货活动中的被动地位,让退换货的主动权掌握在客户手中,这将对电子商务的普及和发展起着巨大的作用。

结束语 根据基于四方的安全电子商务支付协议模型,利用 SMV 工具从协议的原子性方面对协议进行了验证,证明了基于四方的安全电子商务支付协议符合电子商务的原子性要求,确保了电子商务活动中电子支付的金钱原子性、商品原子性以及确认发送原子性,保证了电子支付的安全,很好地满足了传统实物商品的原子性要求,这将对电子商务的发展起到一定的推动作用。

参 考 文 献

[1] 刘义春,张焕国,王丽娜. 电子支付协议的原子性研究综述[J]. 计算机科学,2005,32(2):93-96,113

[2] Tygae J D. Atomicity in electronic commerce[C]// Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing. May 1996:8-26

[3] Cox B, Tygar J D. NetBill security and transaction protocol[C]// Proceedings of the 1st USENIX Workshop on Electronic Commerce. July 1995:77-88

[4] 甘早斌,肖仕成. 基于四方的安全电子商务支付协议研究[J]. 计算机科学,2011(10)

[5] 薛锐,冯登国. 安全协议的形式化分析技术与方法[J]. 计算机学报,2006,29(1):1-20

[6] Lu Shi-yong, Smolka S A. Model Checking the Secure Electronic Transaction (SET) protocol[C]// Proceedings of the 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. 1999:358-365

[7] Lu Si-mei, Zhang Jian-lin, Luo Li-ming. The automatic verification and improvement of SET protocol model with SMV[C]// Proceedings of the International Symposium on Information Engineering and Electronic Commerce(IEEC '09). Ternopil, May 2009:433-436

[8] Talukder K H, Harada K. Modeling and verification of some communication protocols[C]// Proceedings of the 8th International Conference, Advanced Communication Technology (ICACT 2006). Phoenix Park, February 2006:2193-2198

[9] McMillan L. Symbolic Model Checking[M]. Pittsburgh, USAL: Kluwer Academic Publisher, 1993

[10] Liu Xia, Huang Qi, Chen Yong. Model Checking of Wireless Transaction Protocol[C]// Proceedings of 2009 World Congress on Computer Science and Information Engineering. 2009: 620-623

[11] Ning Ning, Zhang Jun, Gao Xiang-yang. Formal Verification of SDG via Symbolic Model Checking[C]// Proceedings of Second International Conference on Intelligent Computation Technology and Automation(ICICTA '09). October 2009:521-524