基于信任的P2P拓扑进化机制

胡建理[1,2] 吴泉源[1] 周 斌[1]

(国防科学技术大学计算机学院 长沙 410073)[1] (广州军区广州总医院信息科 广州 510010)[2]

摘 要 现有的非结构Peer-to-Peer(P2P)系统缺乏对拓扑公平性的考虑,并且不能对某些节点的恶意行为进行有效的抑制。其主要原因在于构造的拓扑对节点信任度的不敏感性,忽略了P2P网络中各节点的异构性。据此,首先给出了基于反馈可信度的节点全局信任度计算模型,然后在此基础上提出了一种针对非结构化P2P网络的自适应拓扑进化机制。利用该机制,可使高可信节点占据拓扑的有利位置,低可信节点处于不利位置,从而体现拓扑的公平性。该机制同时能够对节点的恶意行为进行有效的抑制,并具有激励性质,鼓励节点提供更好的服务,以获得更高的响应率。分析和仿真结果表明,该机制较之现有机制,在拓扑的有效性和激励性上有较大的提高。

关键词 对等网络,自适应拓扑,信任,激励机制

中图法分类号 TP393 **文献标识码** A

Effective Trust-based Topology Evolution Mechanism for P2P Networks

HU Jian-li[1,2] WU Quan-yuan[1] ZHOU Bin[1]

(Institute of Networks & Information Security, School of Computer, National University of Defense Technology, Changsha 410073, China)[1]

(Information Department of Guangzhou General Hospital under Guangzhou Area Command, Guangzhou 510010, China)[2]

Abstract Current unstructured peer-to-peer (P2P) systems lack fair topology structures, and take no consideration for malicious behaviors of peers. The main reason is that the topology is not sensitive to peer's trust, and cannot accommodate heterogeneity of peers over the network. Thus, a feedback credibility based global trust model was presented. Then, based on the trust model, an adaptive topology evolution mechanism for unstructured P2P networks was proposed. Through this mechanism, trusted peers can migrate to the centric position, while untrusted peers to the edge of the topology, guaranteeing fairness during topology evolution. On the other hand, the mechanism can effectively counter the malicious behaviors of peers, and also has the incentive functionality, which incents peers to provide more high-quality services in order to get more return on services. Analysis and simulations show that, compared with the current topologies, the resulting topology mechanism is more effective and robust in combating the selfish or malicious behaviors of peers.

Keywords P2P, Topology evolution, Trust, Incentive mechanism

虽然有关P2P的应用日益广泛[1],但仍然缺乏有效的P2P拓扑构造机制来保证网络的良性发展[2]。在P2P网络中,存在大量的不合作节点,严重影响了系统的可用性。例如,系统中Free-riding节点只使用其它节点提供的资源,而不共享自己的资源;恶意节点提供大量不可靠甚至欺诈性的服务等。因此,如何提高系统的可用性,激励理性自私的参与节点积极提供高质量的服务,惩罚不良行为节点,是保证自组织系统健康发展的内在需求。

P2P网络的可用性与P2P网络拓扑有着紧密的关系,而网络的拓扑进化与节点的异构性密切相关。这种异构性具体表现在节点的最大连接数、计算能力及诚实程度等属性,它们在很大程度上都可归结于节点在信任度上的异构。目前主流的分布式P2P网络[3-5]在设计拓扑进化机制时都没有考虑节点的异构性,节点在拓扑上的地位是相同的,欠缺公平性,激励效果不足。因此本文借鉴社会网络中信任关系的概念,根据节点的交互历史,提出一种基于节点信任度的P2P拓扑进化机制(简称为TTEM),使合作的节点在P2P拓扑中占据有利位置,能更有效地获得服务。而Free-riding节点及恶意节点被排斥到网络的边缘,处于不利的拓扑位置,难以获得高质量的服务。分析与仿真实验表明,TTEM机制较现有的拓扑进化机制,在有效性与激励效果上有明显的提高。

本文第1节介绍了相关工作;第2节给出了一种基于反馈可信度的节点信任度计算模型;第3节具体阐述了基于节点信任度的拓扑进化机制;第4节给出了拓扑进化机制的仿真实验结果及分析;最后对本文进行了总结。

到稿日期:2009-02-20 返修日期:2009-04-22 本文受国家973重点基础研究发展规划项目基金(2005CB321800),国家863高技术研究发展计划项目基金(2007AA010301),国家杰出青年科学基金(60625203)和国家自然科学基金(60873204)资助。

胡建理 博士,工程师,CCF会员,主要研究方向为分布式计算、信息安全等,E-mail:lxman82@gmail.com;**吴泉源** 教授,博士生导师,主要研究方向为人工智能、Web服务和信息安全等;**周 斌** 副教授,主要研究方向为分布式对象技术。

1 相关工作

关于自适应 P2P 拓扑的相关工作主要有以下几个方面：

(1)基于节点处理能力的拓扑进化。B. F. Cooper[6] 提出了一种拓扑机制，使节点可以自组织为相对有效的网络来解决节点的过载问题。按照该机制，节点之间的连接分为两类：搜索连接(发送搜索消息)和索引连接(发送索引信息)。当节点处于过载状态时，按照邻居节点发送消息的数量依概率中断连接。

(2)基于节点物理位置的拓扑进化。Y. Liu 等人在文献[7,8]中提出了自适应的非结构化拓扑进化机制，它通过选择物理位置更近的节点作为邻居来解决 P2P 拓扑与底层物理网络之间的匹配问题，从而提高 P2P 网络的性能。

(3)基于节点信任度的自适应拓扑进化。T. Condie[9] 提出的自适应拓扑进化机制(简称 APT)，其基本思想是每次交易完成之后，节点计算与其交易节点的信任度，并与其邻居节点的信任度进行比较，从而判定是否用当前交易节点取代信任度较低的邻居节点，并发起与交易节点的连接请求。收到连接请求的节点同样会依据其邻居节点的信任度来决定是否接受该连接。

值得一提的是，(1)与(2)中所涉及的拓扑进化机制并没考虑 Free-riding 和恶意节点的问题，而(3)中的 APT 机制则能够在一定的程度上抑制 P2P 网络中的 Free-riding 现象与恶意节点问题。

TTEM 与 APT 的不同之处在于：

(1)对于节点信任度的计算，APT 给出的局部信任度计算模型过于粗糙。本文提出的一种基于反馈可信度的节点全局信任度计算模型，能更有效地识别与扼制更为广泛的恶意节点的攻击。

(2)APT 在进行拓扑进化时，仅考虑到了当前与之交易的节点与邻居节点的信任度比较。而在 TTEM 中，不仅考虑到了当前交易节点，同时还考虑到了历史上与之交易的节点，即从全部交易节点中选出信任度最高的节点与邻居节点进行比较。

2 节点信任度模型

首先给出满意度评价函数与局部信任度的定义，然后给出反馈可信度的定义，最后引出全局信任度的定义。

定义 1(满意度评价函数) 节点交互之后彼此提交满意度的评价，将节点 i 对节点 j 交互满意度的评价定义为 Map 函数 $f(i,j)$：

$$f(i,j)=\begin{cases}1, & totaly\ satisfactory\\ 0, & tataly\ unsatisfactory\\ e(\in(0,1)), & \text{else}\end{cases} \tag{1}$$

采用概率可能性的方法来区分节点提供的不同服务的质量，1 表示节点 i 对节点 j 完全满意，0 表示节点 i 对节点 j 完全不满意，值越大表示满意度越高。

定义 2(局部信任度) 即归一化的局部满意度。在时间区间 t(t 视具体的应用而定，如 6 个月)内，假设节点 i 和节点 j 之间交互的次数为 m，则直接信任评价可定义为：

$$D_{ij}=\begin{cases}\dfrac{\sum_{k=1}^{m}f(i,j)}{m}, & m\neq 0\\ 0, & m=0\end{cases} \tag{2}$$

D_{ij} 是节点 i 根据直接交易历史对节点 j 作出的信任评价，也即节点 i 对节点 j 提供的反馈。当 $m=0$ 时，表示节点 i 与节点 j 之间没有交互历史。将节点 i 对节点 j 的局部信任度设定为 0。

定义 3(反馈可信度) 用来刻画反馈节点(服务消费者)对评价主体(服务提供者)提供的反馈信息真实准确程度。一般来说，反馈可信度与以下几种因素相关：

(1)节点间的交互频繁程度。一般来说，交易越多，则节点间的反馈可信度越高；(2)节点间的评分行为的相似程度。节点 i 与参考节点 j 的评分相似性越高，则说明 i 与 j 对网络中其它节点的看法越一致。

引入交易密度因子 $TNum_{ij}$ 来描述节点 i 与 j 交易的频繁程度，并定义交易密度因子为：

$$TNum_{ij}=\frac{m}{n}*\beta^{\frac{1}{m}}\ \beta\in(0,1)\cap m\neq 0 \tag{3}$$

其中，m 表示节点 i,j 之间交易的次数，n 表示节点 i 与所有其它节点交易的总次数。当 $m=0$ 时，令 $TNum_{ij}=0$；β 为交易密度调节常数，引入该常数是为了更合理地描述节点间交易频繁程度的实际状况，使之更准确地反映节点交易密度的差异。

将描述节点间评分行为的相似性的量记为 $TSim_{ij}$，用来表征节点行为的一致性。设节点 i 与节点 j 的公共交互节点集合记为 $CSet(i,j)$，那么节点 i 和节点 j 对公共交互节点评价差异 $TDif_{ij}$ 可定义为：

$$TDif_{ij}=\frac{\sum_{k\in CSet(i,j)}|D_{ik}-D_{jk}|}{|CSet(i,j)|} \tag{4}$$

设节点 i 对节点 j 容忍的最大评价偏差为 θ，则可以将 $TSim_{ij}$ 定义为：

$$TSim_{ij}=\begin{cases}TSim_{ij}+\dfrac{(1-TSim_{ij})}{2}*\left(1-\dfrac{TDif_{ij}}{\theta}\right), & TDif_{ij}<\theta\\ TSim_{ij}-\dfrac{TSim_{ij}}{2}*\left(1-\dfrac{\theta}{TDif_{ij}}\right), & \text{else}\end{cases} \tag{5}$$

综合上述两种因素，可定义反馈可信度 Cr_{ij} 为：

$$Cr_{ij}=TNum_{ij}*TSim_{ij} \tag{6}$$

因此，由以上分析可以看出，反馈节点交易次数越多，评分行为一致性越强，则其反馈可信度也越高。

定义 4 称矩阵 $R=(R_{ij})$ 为反馈品质矩阵，其元素 $R_{ij}=D_{ij}*Cr_{ij}$。与一般网络信任关系矩阵(D_{ij})不同，反馈品质矩阵不仅考虑了各节点提供的局部信任评价信息，而且考虑了节点本身的反馈可信度，这两种信息的聚合很好地刻画了反馈信息的实际信任状况。

定义 5 网络 N 中对任意节点 i 的全局信任度为 T_i，其定义为：

$$T_i=\sum_{j\in K}D_{ji}*Cr_{ji}*T_j \tag{7}$$

其中，K 为与 i 曾经交互过并对 i 提供过反馈评价的节点集合。

3 基于信任度的拓扑进化机制

节点的信任度反映了节点在未来进行合作的可能性。故从直觉上来讲，根据节点的信任度调整拓扑，可以使互相合作的节点彼此保持连接，而将不合作的节点排斥到网络边缘，从

而实现 P2P 拓扑对合作节点的激励。

在介绍拓扑调整机制之前，首先定义后文中所使用的符号。

$N(i)$：节点 i 的邻居节点集合；

$M(i)$：节点 i 有其信任记录的节点集合；

$TCN_{\min}(i)$：节点 i 可以接受的邻居节点最低信任度门限值，即如果邻居节点的信任度低于该值，i 会断开与该邻居的连接；

$TCR_{\min}(i)$：节点 i 设定的接受其它节点连接请求时其它节点所具有的最低信任度门限值，如果低于该值，则拒绝该节点的连接请求。

$Fv(i)=\{j \mid T_j \geqslant TCR_{\min}(i), j \in M(i)-N(i)\}$：节点 i 愿意但尚未与其建立连接的节点集合；

$SNS(i)=\{j \mid T_j \geqslant TCN_{\min}(i), j \in N(i)\}$：节点 i 的信任度不低于其可接受的最低门限值的邻居集合；

$Fv(i)_{\max}=\{j \mid j \in Fv(i), \forall k \in Fv(i) \text{ and } k \neq j\ T_j > T_k\}$：$Fv(i)$中信任度最大的节点；

$N(i)_{\min}=\{j \mid j \in N(i), \forall k \in N(i) \text{ and } k \neq j\ T_k > T_j\}$：$N(i)$中信任度最小的节点；

$\tau^i_{\min}$：节点 i 维护的最小连接数；

$\tau^i_{\max}$：节点 i 维护的最大连接数。

3.1 发送连接请求

任意节点 i 可以每隔一定的时间进行拓扑更新，当拓扑更新时机到达时，它按如下步骤进行：

第一，当满足条件 $Fv(i)\neq\varnothing \cap |N(i)|<\tau^i_{\max}$ 时，节点 i 会向 $Fv(i)_{\max}$ 发送连接请求，该请求仅表明节点 i 有与节点 $Fv(i)_{\max}$ 连接的意愿，连接成功与否需要经过协商。此时节点 $Fv(i)_{\max}$ 会根据自身的情况决定是否接受连接，当连接协商成功后，$i.negotiation(Fv(i)_{\max})$ 返回 *true*，表明双方都愿意建立连接。建立连接后，节点 i 将 $Fv(i)_{\max}$ 从 $Fv(i)$ 中清除。

第二，当满足条件 $Fv(i)\neq\varnothing \cap |N(i)|=\tau^i_{\max}$ 时，其过程同第一，只是节点 i 的连接数已达到上限，因此断开与节点 $N(i)_{\min}$ 的连接。

在 $Fv(i)=\varnothing \cap |SNS(i)|<\tau^i_{\min}$ 的情况下，则向网络中的某一随机节点发送请求。如果连接请求被接受且 $N(i)>\tau^i_{\max}$，则断开与 $N(i)_{\min}$ 的连接。节点 i 将尽力与信任度不低于 $TCN_{\min}(i)$ 的 $\tau^i_{\min}$ 个邻居保持连接，但当它尝试了许多次且没有节点愿意接受它的连接请求时，只好放弃。

节点 i 作为连接请求者对应的拓扑进化算法为

```
Procedure connect_demander(i) {
  if (Fv(i)≠∅∩|N(i)|<τ^i_max)
      if (i. negotiation(Fv(i)_max)=true) {
          addConncetion(Fv(i)_max);
          remove Fv(i)_max from Fv(i);}
  if (Fv(i)≠∅∩|N(i)|=τ^i_max)
      if (i. negotiation(Fv(i)_max)=true) {
          addConncetion(Fv(i)_max);
          remove(N(i)_min);
          remove Fv(i)_max from Fv(i);}
  if (Fv(i)=∅∩|SNS(i)|<τ^i_min) {
       negotiate with a random peer, and establish connection with it;
      if (N(i)>τ^i_max)
          remove(N(i)_min);}}
```

3.2 接受连接请求

节点 j 作为连接接受者，当满足条件 $T_i > TCR_{\min}(j) \cap |N(i)|<\tau^j_{\max}$ 或 $T_i > TCR_{\min}(j) \cap |N(i)|=\tau^j_{\max}$ 时，经协商与节点 i 建立连接，并在节点 j 的连接数达到上限时，断开与 $N(j)_{\min}$ 的连接，其对应的算法如下：

```
Procedure connect_receiver(i) {
  if (T_i>TCR_min(j)∩|N(i)|<τ^j_max)
      if (j. negotiation(i)=true)
          addConncetion(i);
  if (T_i>TCR_min(j)∩|N(i)|=τ^j_max)
      if (j. negotiation(i)=true) {
          addConncetion(i);
      remove(N(j)_min);}}
```

为完成以上操作，节点 i 需要维护两个表，即邻居节点列表(如图 1(a)所示)与有交易记录节点列表(如图 1(b)所示)。

ID_{i1}	T_{i1}
ID_{i2}	T_{i2}
⋮	⋮
ID_{ik}	T_{ik}

(a)

ID'_{i1}	T'_{i1}
ID'_{i2}	T'_{i2}
⋮	⋮
ID'_{im}	T'_{im}

(b)

图 1 Peer i 的数据结构

图 1 中，$ID_{i1},\cdots,ID_{ik}$ 与 $ID_{i1}',\cdots,ID_{im}'$ 分别为邻居节点列表与有交易记录节点标识序列，而 $T_{i1},\cdots,T_{ik}$ 与 $T_{i1}',\cdots,T_{im}'$ 分别为这些节点对应的信任度序列。

由上述拓扑进化算法可知，设节点 i 的平均最大连接数为 k，则在最坏的情况下，该算法维护网络拓扑通信开销为 $O(k)$。由于与节点 i 的连接数有限，因此拓扑维护的通信开销不大。根据节点所维护的数据结构，易知其存储开销为 $O(k\times(p+q))+O(m\times(p+q))\approx O(k)+O(m)$，其中 p,q 分别为存储节点标识与信任度所占的字节数，m 为节点 i 存储的历史交易条目数。此外，通过周期性地删除过时的信息和低信任度节点的交易记录来节省存储空间。

4 实验及分析

4.1 实验设置

我们模拟的节点包括 3 种不同类型：1)良好行为节点，提供良好的服务和诚实推荐，记为 N 类节点；2)Free-riding 节点，不提供服务和推荐，记为 F 类节点；3)恶意节点，提供低质量的服务和不诚实推荐，记为 V 类节点。在实验中，3 种类型的节点的比例为 13:5:2，表 1 为模拟实验的参数设置。实验的仿真硬件平台配置为 AMD Athlon™ 64 X2 Dual 1.9GHz，1GMB 内存；仿真软件基于 Java 实现。

表 1 仿真参数设置

参数	描述	缺省值
N	社群节点总数	1000
$\tau_{\min}$	最小邻居连接数	3
$\tau_{\max}$	最大邻居连接数	8
$TCN_{\min}$	断开邻居节点的最低信任度门限值	0.3
$TCR_{\min}$	接受连接请求的信任度最小门限值	0.5
β	交易密度调节常数	0.5
θ	节点间所能容忍的最大评价偏差	0.1
P_{res}	响应服务请求的可能性	1

4.2 仿真实验

4.2.1 对 Free-riding 节点的抑制效果仿真

实验的目的是检验拓扑进化机制能否有效地将 F 类节点排斥到网络边缘。排斥到边缘意味着与良好行为节点之间的拓扑距离加大，因而处于边缘的节点难以获得良好行为节点提供的服务。实验对 TTEM 和 APT 机制进行了对比测试。这里，采用平均最短路径长度(the shortest path length, SPL)的方法来衡量节点在拓扑中位置的改变，即对任意节点 i，平均考虑 i 到网络中其余节点的最短路径长度：

$$spl_i=\frac{1}{|N\backslash i|}\sum_{j\in N\backslash i}ShortestPath(i,j) \tag{8}$$

其中，N 表示 N 类节点集合，$|N\backslash i|$ 表示除节点 i 以外剩余节点的集合。

如图 2 所示，对于 TTEM 机制，在 40 个查询周期之后，F 类节点到 N 类节点的平均最短路径长度趋于一个较大值(相对于 APT，TTEM 有更多节点从网络中断开)；N 类节点到 N 类节点的平均最短路径长度趋于一个常数，约为 2.8。而对于 APT 机制，在 40 个查询周期之后，F 类节点到 N 类节点的平均最短路径长度约为 4.82；N 类节点到 N 类节点的平均最短路径长度则趋于 4.2。显然，APT 机制并不能有效地将 F 类节点排斥到网络边缘，其主要原因在于，APT 机制仅根据节点当前邻居的局部反馈进行拓扑调整，忽略了在整个交易历史中节点的信任度，进而导致 F 类节点并不能被很好地区分，并被排斥到网络边缘。而且，APT 下的 N 类节点到 N 类节点的平均最短路径长度也大于 TTEM 机制下的平均长度，这说明 TTEM 比 APT 具有更好的聚集效果，能使 N 类节点快速聚合到一个社区，从而获得更高的搜索效率。

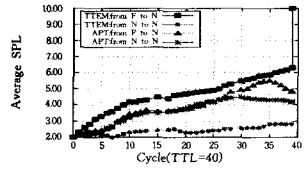

图 2 F 类与 N 类节点的平均路径长度比较

4.2.2 对恶意节点的抑制效果仿真

实验的目的是检验拓扑进化机制能否有效地将 V(恶意)类节点排斥到网络边缘。将恶意节点排斥到网络边缘能够有效地减少其恶意行为产生的影响。这主要是因为处于边缘的节点与良好行为节点之间的拓扑距离较大，这可以通过设置较小的 TTL(time-to-live)值，使得良好行为节点发出的服务请求难以到达恶意节点，从而有效地避免恶意节点的影响。实验对 TTEM 和 APT 机制进行了对比测试，并采用式(8)的方法度量节点在拓扑中的位置变化。

如图 3 所示，对于 TTEM 机制而言，在 55 个周期执行之后，V 类节点到 N 类节点的平均最短路径长度趋于较大值，因而该机制可以有效地将恶意节点排斥到网络边缘。此外，还考察了 TTEM 机制下的 N 类到 N 类节点的平均最短路径，其长度趋近于 2.6。由此推出，可以通过减小 TTL 的设置，避免恶意节点接受到查询，从而减少其恶意的影响。在 APT 机制下，V 类节点到 N 类节点的平均最短路径长度趋于 5.2，因而该机制不能有效地将恶意节点排斥到网络边缘。N 类节点到 N 类节点的平均最短路径长度趋于 4，远大于 TTEM 机制下的平均长度。显然，这将导致搜索效率的降低，因为较大的最短路径长度通常意味着较高的通信开销。

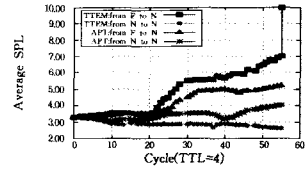

图 3 V 类与 N 类节点的平均路径长度比较

4.2.3 网络的有效性仿真

网络的有效性描述合作节点如何有效获取可信的资源。本文使用有效响应率来衡量网络的有效性。有效响应率(the effective response rate, ERR)：令 V^r_{good} 为某一周期发起请求且收到响应的合作节点集合。如果节点 i 发起请求，得到的 r_i (>0)个响应中有 r_i^a 是由合作节点给出的，则有：

$$ERR_i=r_i^a/r_i$$

由此可得：$ERR=\sum_{i\in V^r_{good}}ERR_i/|V^r_{good}|$。

图 4 比较了在网络中 F 类节点与 N 类节点的比例为 1∶4 时TTEM 和 APT 的有效响应率。由于采用连接信任的原因，APT 的有效响应率在初始阶段迅速增长。但随着仿真周期的增大，在 TTEM 中恶意节点被迅速识别出来并被排斥到网络的边缘。因此，TTEM 的有效响应率在第 110 个周期超过 APT 的有效响应率，并在第 290 个周期之后稳定于 1。而当仿真进行到 800 个周期时，APT 对应的 ERR 值约为 0.84。

由图 4 可知，TTEM 机制显然比 APT 机制更具优势。它可将表现好的正常节点更有效地聚集在一起，使得这些节点更容易获得较高的回报率。而 APT 机制仅根据节点的局部信任度对拓扑进行调整，不能准确地反映节点在系统中的真实信任状况，因此系统无论对 N 类，还是 F 类，或 V 类节点准确评价与拓扑调整都需要更长的周期，使可信节点的请求得不到及时响应。

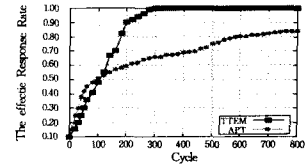

图 4 有效响应率比较

结束语 本文提出了基于反馈可信度的节点全局信任度的定义，并在此基础上给出了一个基于信任度的自适应 P2P 拓扑进化机制 TTEM。TTEM 机制可以针对节点的全局表现进行拓扑调整，体现了拓扑的公平性。它同时能够对节点的恶意行为进行有效的抑制，并具有激励性质，鼓励节点提供更好的服务，以获得更高的响应率。分析和仿真表明，TTEM 机制在有效性和激励效果等方面比现有机制有较大的提高。

参 考 文 献

[1] Bawa M, Cooper B F, Crespo A, et al. Peer-to-Peer research at Stanford[J]. ACM SIGMOD Record, 2003, 32(3): 23-28

[2] Dou W. The research on trust - aware P2P topologies and con - structing technologies[D]. Changsha: National University of Defense Technology, 2003

[3] Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for Internet applications[R]TR-819. MIT, 2001

(下转第 166 页)

1)在服务器端安装并配置 Windows 2003 Server 系统自带的 CA 认证组件，即数字证书服务器；

2)在客户端从浏览器访问数字证书服务器，获取并安装 CA 根证书；

3)在服务器端 IIS 服务管理器中配置 SSL(secure sockets layer，安全套接字层)通讯协议；

4)在客户端从浏览器访问数字证书服务器，申请并安装数字证书。

由于本方案采用可信硬件模块 USB Key 来保存数字证书，因此，步骤 4)在实际操作中是通过配置 USB Key 来申请并保存数字证书的。

5.2 USB Key 应用

原型系统使用北京飞天诚信公司的 USB Key 产品 ePass1000ND。ePass1000ND 采用 8 位处理器，提供 8kB 安全数据存储空间，无需驱动。每个 ePass1000ND 硬件都有一个 64 位序列号，作为全局唯一标识[7]。

使用 ePass1000ND 访问 CA 服务器 SSL 加密站点，申请并保存数字证书与秘密密钥。对系统中各实体的身份认证均通过验证 USB Key 中保存的数字证书来完成。在 USB Key 的应用中，通过以下设置加强 USB Key 应用的安全性：

1)设置 USB Key 只允许单进程访问，确保不被跟踪；

2)对 USB Key 访问设置 PIN 码(个人识别码)保护；没有 PIN 码与管理员密码无法访问某些文件；

3)设置 USB Key 的 PIN 码的最大可重试次数。当 PIN 码连续输入错误达到最大可重试次数时，USB Key 自动锁死，可防范穷举攻击。

5.3 MS CryptoAPI 编程接口应用

目前业界实际使用的 PKI 编程标准主要有两种：RSA Lab 的 PKCS # 11 和 Microsoft 的 CryptoAPI。MS CryptoAPI 是 Win32 应用程序的通用加密接口，在本方案中被用来实现 USB Key 证书存储、RSA 密钥对生成及对消息传输的数字签名。MS CryptoAPI 的体系结构如图 4 所示。

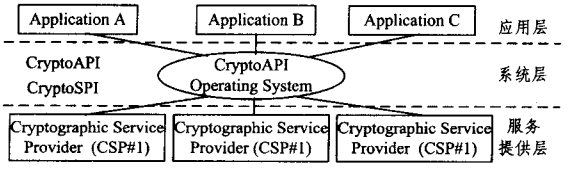

图 4 MS CryptoAPI 的体系结构

MS CryptoAPI 环境中实际提供加密服务的模块是 CSP (Cryptographic Service Provider，加密服务提供者)，应用程序不直接与 CSP 打交道，而是通过 CryptoAPI 来协调应用程序与 CSP 之间的交互。由于 ePass1000ND 已经实现了一个 PROV_RSA_FULL 类型的 CSP，因此本系统在 USB Key 应用中只需编程调用 CryptoAPI 来建立与 ePass1000ND 之间的通讯。

结束语 基于信息隐藏的外包数据库版权保护系统构建在 PKI 基础之上，借助 PKI 机制较为容易地实现了身份认证、密钥分发、可信时戳等必需的安全性要求和功能。水印技术与加密、数字签名等技术的结合弥补了水印在主动安全保护方面的不足。设计的数据库水印协议满足了数据库数据容量大、更新频繁的要求，具有抵抗多种水印攻击的能力。可信硬件模块 USB Key 的引入，增强了系统各实体间安全交互的能力。

该系统的主要不足在于：系统的整体安全性建立在第三方可信计算平台基础上，存在一定的安全失效风险。

参 考 文 献

[1] Hacigumus H, Iyer B, Mehrotra S. Providing Database as a Service[C]// ICDE 2002. San Jose, California, USA, 2002

[2] 朱 勤，于守健，乐嘉锦，等. 外包数据库系统安全机制研究[J]. 计算机科学，2007，34(2)：152-156，195

[3] Rosenblatt W, Trippe W, Mooney S. Digital Rights Management: Business and Technology[M]. New York: M &T Books, 2002

[4] Garnett N. Digital Right's Management, Copyright, and Napster [J]. ACM SIGecom Exchanges, 2001, 2(2): 1-5

[5] 俞银燕，汤帜. 数字版权保护技术研究综述[J]. 计算机学报，2005，28(12)：1957-1967

[6] 朱勤，刘良旭，乐嘉锦. 一种基于 m 序列的关系数据库鲁棒水印算法[J]. 小型微型计算机系统，2008，29(8)：1486-1490

[7] Epass1000ND[EB/OL]. [2007-12-02]. http://www.ftsafe.com.cn/products/viewproduct.php? p=epass1knd, 2008

(上接第 98 页)

[4] Ratnasamy S, et al. A Scalable Content - Addressable Network [C]// Proceeding of ACM SIGCOMM. NewYork: ACM Press, 2001: 161-172

[5] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems[C]// IFIP/ACM International Conference on Distributed Systems Platforms. Kluwer Academic Press, 2001: 329-350

[6] Cooper B F, Garcia-Molina H. Ad hoc, self-supervising peer-to-peer search networks[R]. Stanford University, 2003

[7] Liu Y, Zhuang Z H, Xiao L, et al. AOTO: Adaptive overlay topology optimization in unstructured P2P systems[C]// Proc. of the IEEE GLOBECOM 2003. San Francisco, 2003

[8] Xiao L, Liu Y, Ni L M. Improving unstructured peer - to - peer systems by adaptive connection establishment[J]. IEEE Trans. on Computers, 2005, 54(9): 1091-1103

[9] Condie T, Kamvar S D, Garcia-Molina H. Adaptive peer-to-peer topologies[C]// Lambrix P, Duma C. eds. Proc. of the 4th Int'l Conf. on Peer-to-Peer Computing. New York: IEEE Press, 2004: 53-62