第 34 卷 第 1 期 计 算 机 工 程 2008 年 1 月
Vol.34 No.1 Computer Engineering January 2008
·博士论文· 文章编号：1000—3428(2008)01—0010—04 文献标识码：A 中图分类号：TP393.08

分布式网络系统中的信任研究

王惠芳，朱智强，孙 磊
（信息工程大学电子技术学院，郑州 450004）

摘 要：目前在因特网中没有可靠的信任体系，如何在一个分布的、混乱的因特网环境中有效地表达和管理信任关系已成为一个研究热点。该文研究信任的定义、信任的逻辑模型，分析分布式网络系统中信任度评估模型、信任模型和信任管理，提出目前分布式网络系统中信任管理存在的问题。

关键词：信任度评估模型；信任模型；信任管理

Research on Trust of Distributed Network System

WANG Hui-fang, ZHU Zhi-qiang, SUN Lei
(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004)

【Abstract】 There is no reliable trust system in Internet nowdays. It attracts more and more attentions how to express and manage trust relations in a distributed and disordered network system. This paper analyzes the define of trust and logic model of trust firstly, furthermore researches especially on trust evaluation model, trust model, and trust management of distributed network systems. Also it discusses the problem of trust management in distributed network systems.

【Key words】 trust evaluation model; trust model; trust management

现实生活中有政府、法规保障下的信任体系，而网络社会中没有这种信任体系，如何在一个分布的、混乱的 Internet 环境中有效地表达和管理信任关系是一个重要的问题。在分布式网络系统中存在有许多信任，比如，对没有公开源码的程序需要有某种程度的信任；需要信任与在网络上通信的人；信任计算机硬件提供正确的运算结果等。本文着重分析分布式网络系统中通信实体(人、计算机、代理、进程等)之间的信任。

1 信任的定义

信任是人类生活中非常重要的方面。在日常生活中人们认为信任是理所当然的。每天做的大多数事情都用到信任，只是在大多数情况下没有意识到它的存在。信任是不容易定义的抽象东西，是生活中不可缺的。尽管如此，有一些学者还是尝试着定义信任，下面是给出的信任定义。

牛津英语字典定义信任如下：

(1)Noun: ‘confidence, strong belief, in the goodness, strength, reliability of something or somebody’, ‘ responsib ility’.

(2)Verb: have trust in——believe in the honesty and reliability of someone or something’,‘have confidence in’, ‘earnestly hope’.

一些学者从社会学、(社会)心理学和哲学的角度给出信任的定义[1]。Morton Deutsch基于心理学对信任进行的研究，在“合作和信任”中定义信任为：

(1)如果一个个体面临一个不明确的道路，这条道路可能会带来利益(Va^+)或伤害(Va^-)；

(2)他认识到Va^+ 或Va^- 的发生与其他人的行为有关；并且

(3)他认识到Va^-的强度要大于利益Va^+的强度。

如果他选择具有上述属性的道路，我说他作出信任选择；如果他不选择这条路，则他作出不信任的选择。

上述定义隐含着信任依赖于个体和在给定情况下对代价和利益的感知。值得注意的是在这个定义中，信任与感知到的危害连在一起。如果感知到的利益大于危害，则信任将不会在选择中占太大的比例。

1973 年，Deutsch 在《冲突的解决》一书中倾向于用信心定义信任。他定义信任为一个人相信他能从他人那得到他所想要的，而不是他害怕的信心。这种信心含有希望的意思。这是合理的，因为除非有希望获利，否则一个人不会作出信任选择。

2 信任的逻辑模型

除了关于信任的心理学、社会学方面的研究外，一些学者用形式化的逻辑方法对信任进行研究。Rangan 开发了一个基于模态逻辑的理论来形式化信任[2]。在这个模型中每一个agent维护一个它关于真实世界信念的集合：

一个 agent 信念的产生主要因为 agent 不知道分布式系统的整个状态。agent 的信念状态是基于它所在的本地状态，它认为它所在全局状态的信念。

在给定的全局状态下，有一些可能的状态(possible-world states)属于 agent 的信念集合。在 agent 认为的真实世界里的所有状态下，一个命题都为真，则 agent 相信这个命题。Rangan 将分布式系统定义为 Kripke structures, Kripke structures 是一个集合元组 $M = <W, I, J>$集合 W 表示所有可能的世界。函

基金项目：国家部委基金资助项目
作者简介：王惠芳(1972－)，女，副研究员、博士，主研方向：分布式系统安全；朱智强，研究员；孙 磊，博士
收稿日期：2007-02-10 **E-mail：**whf225@mail.china.com

数 I 将一个基本命题(s)映射到一个世界的集合，在这个集合中 s 为真。函数 J 将一个人映射到他在世界 W 中的关系。I 和 J 一起决定一个公式在世界 W 中的每一个世界里是否为真。Rangan 构造了一个形式理论来评估信念系统，如 agent(i)相信 agent(j)相信用逻辑语言写好的一些形式化公式(WFF)。agent 可以用推理规则对这些形式化公式进行操作和变换，并且可以用这些形式化公式与其它 agent 通信。Rangan 理论中有关信任的特点总结如下：

(1)agent 相信的命题在现实世界中可能不为真，但对于 agent 所在的所有可能状态和关系中，这个命题为真。

(2)对于全局状态，(s,t)称为一个 agent 可能的关系，当且仅当在全局状态 s 下，该 agent 能够考虑到全局状态 t 为一种可能的状态。

(3)如果 agent(j)发给 agent(i)的消息中含有一个 WFF 可能会触发 agent(i)的信念集合认可这个信念，则这个 WFF 与 agent(j)发来的其他消息是一致的。

(4)“在这个逻辑中，agent(i)认可的信念将不会与其它信念不一致”。Rangan一个有趣的例子：agent(j) 告诉agent(i) 一些WFF f。如果agent(j)再告诉agent(k) $\tilde{f}$，并且agent(k) 再告诉agent(i)，它们将与agent(i)的信念集合是一致的。否则，agent(i)相信agent(j)相信f和agent(j)对agent(k)撒谎(agent(k)相信$\tilde{f}$)。

(5)一个 agent 可能发送他不持有的信念，但只要这个信念与以前的消息一致，则将被别人认可。

Rangan 的信任理论可形式化地描述系统中的信任链并用定理证明是有效的，从而说明多种不同的应用系统。

Burrows 等提出了一个分析认证协议的逻辑(ban 逻辑)。它没有处理信任本身的特性，但可以逻辑地检验假设和协议的目的关于信任是否是一致的。

Gong, Needham and Yahalom 对 BAN 逻辑进行了改进后提出 GNY 逻辑。这个逻辑与 BAN 逻辑的区别之一是对信念和他们拥有的信息有不同程度的信任。

3 分布式网络系统中信任度评估模型

信任度评估模型主要涉及以下问题：(1)信任的表述和度量；(2)由经验推荐所引起的信任度推导和综合计算。代表性的信任度评估模型有 Beth 信任度评估模型和 Jøsang 信任度评估模型。这里分析 Beth 信任度评估模型。

文献[3]归结了 6 类不同类型的信任。这 6 类信任为：密钥生成，身份鉴别，保守秘密，相互不干扰，时钟同步和执行算法步骤。每一种类型的信任都有两种类型的信任：直接信任和推荐信任。直接信任是直接信任其他实体，而推荐信任是基于第三方的推荐信任另一个实体。在文献[3]中，Beth 等人对信任度评估模型引入了经验的概念来表达和度量信任关系，并给出了由经验推荐所引入的信任度推导和综合计算公式。

在 Beth 信任度评估模型中，经验被定义为对某个实体完成某项任务的情况记录，对应于任务的成败，经验被分为肯定经验和否定经验。若实体任务成功则对其的肯定经验记数增加，若实体任务失败则否定经验记数增加。模型中的经验可以由推荐获得，而推荐经验的可信度问题同样是信任问题。因此，Beth et.将信任分为直接信任和推荐信任，并分别给出评估模型。

3.1 直接信任评估模型

在文献[3]中，作者定义了直接信任为：P $\text{trust}_x^{\text{seq}}$ Q value V。如果P对Q关于信任类x的所有(包括直接的或推荐获得的)经验均为肯定经验，则P对Q存在直接信任关系。Seq是从P到Q的推荐路径上的中间人序列，不包括P和Q。V是一个信任关系值，它表示当Q被信任时，Q能成功完成的概率。这个值是取决于P对Q肯定经验的次数。Beth采用下面的公式描述直接信任度与肯定经验次数的关系：

$$Vz(p)=1-\alpha^{P}$$

其中，p 是 P 所获得的对于 Q 关于信任类 x 的肯定经验次数，α 则是对 Q 成功完成一次任务的可能性期望。该公式基于 Q 完成一次任务的可能性在[0,1]均匀分布这一假设。

3.2 推荐信任评估模型

Beth 定义推荐信任为“若 P 愿意接受 Q 提供的关于目标实体的经验，则 P 对 Q 存在推荐经验关系”。Beth 采用肯定经验与否定经验相结合的方法描述推荐信任度。推荐信任度与经验次数的关系采用如下公式描述：

$$V_r(p,n)=\begin{cases}1-\alpha^{P-n} & \text{if } p>n\\ 0 & \text{else}\end{cases}$$

其中，p,n 分别是 P 所获得的关于 Q 的肯定经验和否定经验。

3.3 信任关系的推导

在Beth信任度评估模型中，经验可以通过推荐获得，而对于同一个信任关系，多个不同的经验推荐者可能形成多条不同的推荐路径。这就需要有一个计算方法能够推导并综合所有推荐路径的经验信息，以获得一致的信任度。假设A对B的推荐信任度为V_1，B对C的直接信任度为V_2，B对D的推荐信任度为V_3，则A对C的直接信任度推导公式为

$$V_1 \odot V_2 = 1-(1-V_2)^{V_1}$$

A对D推荐信任度的推导公式简单的乘法：$V_1 \cdot V_3$，这表明推荐信任随着推荐的路径增长而降低。Beth模型还给出了推荐、直接信任度综合计算公式如下：

(1)推荐信任度综合计算公式

$$V_{com} = \frac{1}{n}\sum_{i=1}^{n} V_i$$

其中，V_i是由单个推荐路径而推导出的信任度，综合推荐信任度V_{com}是这些单个信任度的简单算术平均。

(2)直接信任度综合计算公式

设$P_i(i=1,2,\cdots,m)$是推荐路径上各不相同的最终推荐实体，$V_{i,*}$表示其最终推荐实体为P_i的各条推荐路径的信任度。

$$V_{com} = 1-\prod_{i=1}^{m} n_i\sqrt{\prod_{j=1}^{n_i}(1-V_{i,j})}$$

该公式考虑了同一个经验推荐者出现在不同推荐路径上的情况。相同的经验信息经过不同的路径被多次传递，产生不同的推导结果，该公式采用取推导值平均的方法得到一个唯一值。

Beth 模型对直接信任的定义比较严格，仅采用肯定经验对信任关系进行度量。但，其信任度综合计算采用简单的算术平均，无法很好地消除恶意推荐所带来的影响。

4 分布式网络系统中信任模型

在分布式网络系统中，本文认为分析系统中一个协议或框架的信任要求与分析所采用安全技术的完备性是同样重要的。有几种方法可以分析信任要求，可以按信任的一般概念找出系统中可信的实体；也可以先对信任分类，然后定义相互信任的方式[3]；也可以定义对其他实体的信任程度[4]。这些方法本文称为是对信任的建模。下面给出信任模型的定义。

定义 信任模型是一个系统中信任关系集合的模型。

4.1 用数字证书表达信任

一个证书是关于某个实体具有某种属性的签名申明。在数字证书中签名是发行者的密钥与证书数据计算得到的数据。传统来讲，证书的功能是将一个公钥与拥有者的名字绑定，称为身份证书。如果一个证书申明的是发行者本人的信息，则称为自签名证书。当然证书也可以有更多申明，如申明某个实体可以访问某个服务，这个证书称为授权证书。

在网络环境中通常用“证书”表达信任，通过使用证书来表达信任关系。一个证书是对某种信任的一个可靠申明。如身份证书就是发行者相信一个名字和一个公钥绑定的申明。证书形成一个链，信任就沿着这个链从一个实体传播到另一个实体。这些实体之间形成某种形式的信任关系。

在基于证书的分布式安全解决方案中，一个基本的信任要求是能够获得并且相信一个证书真正属于某个实体。这样才能保证加密的信息被真正的实体解密，或验证真正实体的数字签名。目前在网络中有两个主要的信任模型来解决这个信任要求，一个是 PGP 信任模型，另一个是 X.509 信任模型。

4.2 两种典型的信任模型分析

在数字化的网络世界信任关系是如何被管理的，目前网络中存在两种最典型的信任模型：PGP 和 X.509 信任模型。PGP 系统由于它的推荐机制而广泛用于个人邮件。X.509 尽管没有实质性的飞速发展，但目前在没有全球范围的 X.500 目录服务的情况下获得一些普及。

4.2.1 PGP 信任模型

PGP 信任模型是“Web of trust”。在 PGP 系统，一个用户生成一个(公钥，私钥)对，并与唯一的 ID 相连。ID 的形式为(Name, EmailAddress)。密钥存放在密钥库中。公钥记录包括 ID、公钥和创建的时间。公钥和私钥分别存放在公钥环和私钥环上。每个用户存储和管理这一对密钥环。

如果用户 A 得到用户 B 的正确的公钥，即用户 A 相信这个公钥确实是用户 B 的，则 A 对这个公钥签名，并传给用户 C。A 作为介绍者将 B 介绍给 C。A 签名过的密钥称为证书。每个用户必须设置 PGP 系统他信任哪些人作为介绍者，并且必须用自己的私钥对介绍者的公钥签名。另外，用户可以说明他对每个介绍者的信任程度，可指定为不知道、不信任、部分信任和完全信任 4 个信任级别。因此，PGP 中的信任大体分为直接信任和推荐信任 2 种，直接信任又分为 4 个等级。

每个用户存储他的信任信息于密钥环上，PGP 将给在密钥环上每个证书指定一个值，并且当这个值足够高时才使用该证书中的公钥。比如，一个用户可以要求一个公钥必须有 2 个完全可信的签名，才认为一个公钥有效。而另一个用户可能只要求一个完全可信的和 2 个部分可信的签名即可。值得注意的是，这里隐含着一个“安全策略”的概念，即在验证消息发送者的 ID 时，是否相信 ID 对应的公钥。密钥环和信任程度允许每个用户用这种有限的表达形式设计他自己的安全策略。因为 PGP 被设计为专为用户提供安全电子邮件的，所以用这种有限的表达形式描述安全策略的思想适合于 PGP，但不适合于更大范围的安全网络服务。另外要注意的是，A 对 B 的公钥进行签名不应解释为 A 担保 B 的可信性。更确切的解释应是 A 相信 B 的 ID 与这个公钥的绑定是正确的。另外，在 PGP 中信任是不可传递的，A 完全信任 B 作为介绍者，B 完全信任 C，不能推导出 A 对 C 的信任程度。

在 PGP 中，每个用户负责获取他所需的公钥证书，并指定对介绍者的信任程度。同时，每个用户必须创建他自己的密钥对，分发他的公钥。PGP 的用户可以信任任何他想信任的人，所有用户都是平等的。这种自由在用户之间形成了“Web of trust”，这种信任模型不存在官方的证书权威机构，对公钥进行签名的用户对使用这些公钥的用户来说是“可信的服务方”。

4.2.2 X.509 信任模型

X.509 信任模型是有严格层次的信任模型。X.509 认证框架试着解决 PGP 介绍者机制尝试解决的信任关系问题，即获取某个实体可信的公钥。像 PGP 一样，X.509 证书是签名绑定用户的 ID 和公钥，只是它比 PGP 证书包含更多的信息，如签名算法、有效期等。但它们的基本目的一样的，即仅绑定用户和密钥。

与PGP最大的差别在于，X.509 框架假设每个人都从官方证书机构(CA)获得证书。当用户A想与B进行安全通信，如果A与B的证书是同一个CA颁发的，则A可以用该CA的公钥验证B的证书。如果A与B的证书不是同一个CA颁发的，则需要一个证书链，即$cert_1$, $cert_2$,…,$cert_n$, $cert_i$ 是CA_i 的证书，当$1< i < n$ 时，CA_{i+1}的证书由CA_i颁发，且证书链的最后一个证书$cert_n$为B的证书。为了获得B的可信的公钥，A要验证这条链，则A必须知道CA_1的证书。

在 X.509 信任模型中信任是可传递的，A 完全信任 CA，CA 完全信任 B，则 A 完全信任 B。X.509 信任模型假设所有 CAs 组织成全球化的树状结构。所有的证书都由某个 CA 签发，并且这个 CA 能连到这个树上。在这个结构中所有 CA 有一个共同的祖先，即根 CA。

5 分布式网络系统信任管理

从上述可得出尽管信任管理(Trust Mangement, TM)这个概念是在 1996 年由 Blaze.等人在文献[5]中第 1 次引入，实际上，在基于 PGP 或 X.509 证书系统中隐含着信任管理的思想。

Blaze.等人给出的信任管理定义为：“用统一的方法说明和解释安全策略、凭证(credential)、以及对安全行为直接授权的关系”。基于这个定义，信任管理系统回答的问题是“凭证集合 C 能够证明请求 r 与本地安全策略 P 一致吗？”。也就是说，信任管理系统避开了“签名请求者是谁”这个问题。如果信任管理系统对问题“凭证集合 C 能够证明请求 r 与本地安全策略 P 一致吗？”的回答为“是”，则应用系统直接授权请求 r。也就是说信任管理系统判断的不是谁发出的签名请求，判断的是发出签名请求的实体是否有权获得请求的资源。另外如图 1 所示，信任管理系统与应用相对独立。当信任关系发生改变或添加新的信任关系时，不用更改应用程序，只需改变本地策略即可。

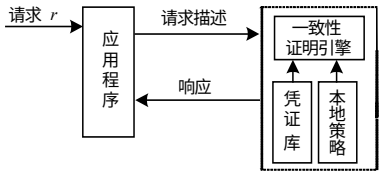

图 1 信任管理系统模型

从信任管理的模型可以看出，设计信任管理系统主要要解决 3 个方面的问题：(1)如何描述和表达安全策略和凭证；(2)设计策略一致性验证(Compliance-checking)算法；(3)信任管理系统与应用程序职能的划分。

5.1 信任管理系统的发展现状

目前的信任管理系统主要有：PolicyMaker, KeyNote,

REFEREE，SPKI/SDSI2.0。下面以 PolicyMaker 和 KeyNote 为例进行介绍。

PolicyMaker 是第 1 个信任管理系统，由 AT&T Research Laboratories 开发，最早由 Blaze 等在文献[5]中介绍。PolicyMaker 允许丰富的信任表达。PolicyMaker 直接在凭证中说明允许一个公钥做什么。传统的证书框架，如 PGP, X.509 并不在证书框架中绑定公钥的主人和访问权限，只绑定公钥和身份。访问权限与身份的绑定发生在证书框架以外，由 ACL 表决定。在 PolicyMaker 中直接将访问权限和公钥绑定。它的语法相对简单，断言的形式为

Source ASSERTS AuthorityStruct WHERE Filter

PolicyMaker 的凭证和安全策略都是可完全编程的，也就是说，可用任意的语言编写。策略和凭证一起称为断言(assertion)。简单来讲，assertion 可由(*f, s*)表达，*s* 表示授权者，*f* 是一个由可编程语言编写的代码，描述所授权力的性质和对象。在策略 assertion 中 *s* 为关键词 Policy，当应用程序调用 PolicyMaker 引擎时，要输入一个或多个策略 assertion，这些都是被本系统无条件信任的，它们形成了“ trust root ”。在凭证 assertion 中，*s* 为证书发行者的公钥。凭证必须由发行者签名，在凭证使用之前要验证签名。

PolicyMaker 的 assertion 可以由任何语言编写。由于凭证来自不同的发行者，可能有不信任者发行的凭证。为了断言语言能被 PolicyMaker 安全地解释，M.Blaze 等人引入了安全编程语言来编写断言。

PolicyMaker 不选择固定语言的目的是可以专注于一致性验证算法的设计、分析、和实现，这样当变化 Assertion 语言，或引入新的语言时，不需要重新做一致性验证算法。PolicyMaker 的一致性验证算法只能处理单调的断言，不支持否定断言(如某实体不能授予某种权利)。

PolicyMaker 本质上是一个推导引擎，即策略一致性验证算法，应用程序可以以连接库的方式调用它。它回答的问题是“ 一组凭证集合 *C* 是否能证明这个请求 *r* 与本地安全策略 P 一致 ”，*r* 中含有请求实施的行为。应用程序给 PolicyMaker 的输入参数(*r, C, p*)，它返回的是 yes 或 no 或实施行为需要满足的约束条件。另外，凭证和请求的签名验证由应用程序负责。这样设计的好处是应用开发者可以选择签名体制。

KeyNote 同样是由 AT&T Research Laboratories 开发的，它与 PolicyMaker 的设计原则是一样的，即用凭证直接授权代替认证加访问控制。但另增加了两个设计目标：标准化和更容易集成到应用程序。与 PolicyMaker 不同的是密码签名验证由 KeyNote 中信任管理引擎实现。另外，KeyNote 要求凭证和策略由特定的 assertion 语言编写。通过指定特定的 assertion 语言，KeyNote 提高了效率和互操作性，以及可以使书写良好的凭证和策略得到广泛使用。

应用程序输入给 KeyNote 的参数是：证书列表，策略，请求者的公钥和行为上下文(action enviroment)。行为上下文是属性/值的对应列表，包含与请求的行为和必要的信任判定有关的信息。行为上下文中的属性和属性值的指定必须精确地反映应用的安全要求。因此，选择哪些属性包含在 action-enviroment 中是集成 KeyNote 到应用程序的最关键工作。

KeyNote 返回的结果是应用程序定义的字符串，最简单的反回是授权“ authorized ”。

KeyNote 断言的形式类似于 e-mail 头，例子如下(详细内容见文献[6])：

KeyNote-Version: 1

Authorizer: rsa-pkcs-hex:”1023abcd” # 颁发者的公钥

Licensees: dsa-hex “986512a1” || #被授权者的 DSA 公钥

rsa-pkcs1-hex:”19abcd02” #被授权者的 RSA 公钥

Comment: Authorizer delegates read access to either of the Licensees

Conditions: ($file == “etc/passwd” && $access ==“read”) -> {return “ok2}

Signature: rsa-md5-pkcs1-hex:”f00f5673”

#颁发者签名

在 KeyNote 中，写在 Conditions 域的程序基本上是用于上下文中的变量测试。测试方法有串比较，数字操作和比较，和模式匹配操作。

KeyNote 与 PolicyMaker 的另一个差别在于，PolicyMaker 的一致性验证算法要求连同存储中间结果和断言之间通信的黑板“ blackboard ”一起，反复评估断言。而 KeyNote 采用深度优先搜索算法，试着至少找到一个满足的策略断言。

这两个系统都没有解决证书查找问题，如不负责查找缺少的证书，并且都不支持否定断言，如吊销的凭证。

5.2 信任管理系统的信任模型

TM 目标是分布式的授权和管理操作。它们不像传统的访问控制机制需要依赖于可信的计算基础(trusted computing base or TCB)，相反它们假设在开放网络中参与者都是不可信的，就像现实社会一样。分布化并不意味着无政府主义，如 PGP 的“ Web of trust ”。TM 提供建立本地关系和本地权威方法，这种关系是由参与者的个人和业务联系形成的。TM 不要求任何层次或固定的结构。所有的实体，对于它控制下的服务，有权进行授权。对于获得服务的实体来说，提供服务的实体就是权威。TM 用签名的证书实现访问权限的授权。一个公钥可以将它的一些权利授权给另一个公钥。这些证书形成一个复杂的网络，反映了对应私钥拥有者之间的信任关系。

6 结束语

目前提出的信任管理系统都还存在一些问题，每个系统的侧重点不同，问题也不同。这里不一一列举。但有一缺点是共同的，即确定的信任是一种静态的信任，不能随着时间和经验动态变化。实际上，信任会随着时间变化，如一个用户使用一个不了解的服务商提供的服务，如果这个服务商在一段时间内，提供了高质量的服务，用户对这个服务商的信任就会逐渐增加。有的信任管理系统加入了评估信任的简单算法，但由于信任的特性以及在不同的应用环境信任有不同的特性，因此不可能或者很难找到一种普遍适用的评估信任的算法。目前研究一定范围适用的信任评估算法并与信任管理相结合是信任管理研究的另一个热点。

参考文献

[1] Marsh S. Formalising Trust as a Computational Concept[D]. UK: University of Stirling, 1994.

[2] Rangan P V. An Axiomatic Basis of Trust in Distributed Systems[C]//Proceedings of the IEEE Symposium on Security and Privacy. [S. l.]: IEEE Press, 1988: 204-211.

[3] Yahalom R, Klein B, Beth T. Trust Relationships in Secure Systems——A Distributed Authentication Perspective[C]//Procee dingsof the IEEE Conference on Re-search in Security and Privacy. [S. l.]: IEEE Press, 1993.

（下转第 16 页）