doi:10.3772/j.issn.1002-0470.2016.07.004

基于信任网络的软件可信评价方法①

阎 林②*,**,*** 张建标③*,**,*** 张 艾****

(*北京工业大学计算机学院 北京 100124)

(**可信计算北京市重点实验室 北京 100124)

(***信息安全等级保护关键技术国家工程实验室 北京 100124)

(****北京工业大学-都柏林国际学院 北京 100124)

摘 要 研究了可信计算领域中的软件可信评价问题,提出了一种基于信任网络的软件可信度量方法,该方法通过信任网络模型将用户划分成不同的用户域,采用可信度量票据的方法保障设备的可信性;在设备可信的基础上建立可信风险树模型,分别从可信和风险两方面对软件进行可信度量,计算异常行为的风险概率;采用多台设备协同工作的方式,通过滑动窗口模型从用户域范围和管理域范围对软件进行可信评价。最后通过实验验证了该方法的正确性和可信性。

关键词 可信计算,可信度量,信任网络,信任域,可信软件

0 引 言

随着计算机与网络技术的发展,软件在信息社会中发挥着日益重要的作用[1,2]。随着软件在金融、军事及经济等敏感领域应用的不断深化,对软件可信性需求也愈加急迫[3,4]。然而目前可信软件构造技术、可信性度量与评价方法严重缺乏,使得软件产品在推出时就含有已知或未知的缺陷,对软件的安全运行构成了不同程度的威胁[5-7]。如何在软件开发和运行中保证软件的高可信性,已成为软件理论和技术研究的重要方向[8,9]。

可信计算组织(Trusted Computing Group,TCG)从实体行为角度对可信进行了定义:"如果一个实体的行为,总是以预期的方式,达到预期的目标,则称其为可信的[10]"。一个可信的软件应该是在任何运行环境中,软件的行为及结果可以预期,运行时的行为状态可以监控[11-13]。由于运行环境的差异,单一的运行环境不能为软件的可信性给予充分的评价,只有多种运行环境共同对软件进行可信评价,才能充分发现软件的威胁[14,15]。本文基于信任网络理论和可信连接架构(trusted connection architecture,TCA),结合静态可信度量和动态可信度量技术,提出了基于信任网络的软件可信评价方法。

1 基于票据的信任网络模型

将信任网络模型作为网络管理模型,将可信连接架构(TCA)作为设备之间通信的可信性保障,基于信任网络的软件可信评价模型如图 1 所示。

管理框架分为三层:管理域管理者,用户域管理者和终端设备。其中,管理域管理者是全局信任关系的可信根,负责维护信任网络的可信性。当管理域管理者和用户域管理者进行通信时,结合管理域可信第三方建立 TCA 模型,并通过用户域票据选择可信用户域;用户域管理者是局部信任关系的可信根,负责维护当前用户域的可信性。当用户域管理

① 北京市委组织部优秀人才培养计划(Q0007016201501)资助项目。

② 男,1986 年生,博士生;研究方向:可信计算;E-mail: chriszt@126.com

③ 通讯作者,E-mail: zjb@bjut.edu.cn

(收稿日期:2016-05-31)

者和终端设备进行通信时,结合用户域可信第三方建立 TCA 模型,并通过用户身份票据和平台配置票据选择可信终端;终端设备是用户的操作平台,也是软件运行的终端环境。用户域管理者采集软件在可信终端上运行的可信度量值,从用户域角度对软件进行局部可信评价。管理域管理者采集可信用户域的局部可信评价,从管理域角度对软件进行全局可信评价。

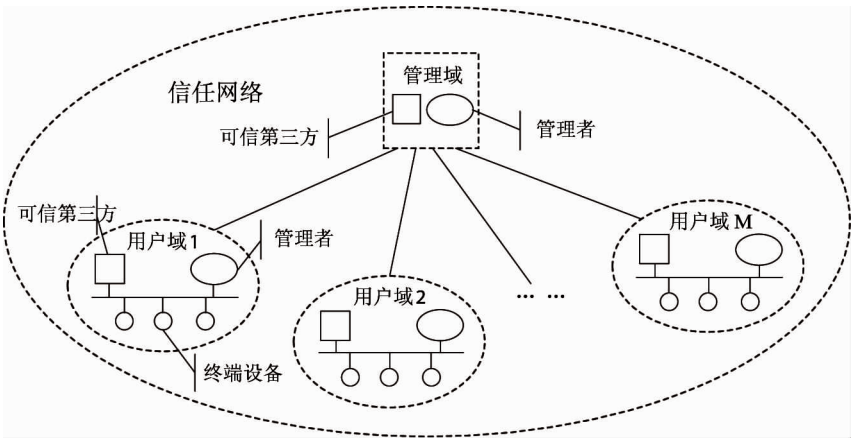


图 1 基于信任网络的软件可信评价模型的逻辑结构

定义 1 用户域票据:设用户身份属性和平台配置属性分别用 $U_{i,j}=\{u_{i,j}^1, u_{i,j}^2,\cdots,u_{i,j}^\alpha\}$ 和 $P_{i,j}=\{p_{i,j}^1, p_{i,j}^2,\cdots, p_{i,j}^\beta\}$ 表示,用户域票据的计算方法为

$$ticket_{UD_i}=sig_{MTTP}(SK_{MTTP},\Phi_{MTTP}(\sum_{j=1}^{n}U_{i,j},\sum_{j=1}^{n+1}P_{i,j})) \tag{1}$$

其中,$\Phi_{MTTP}(x)$ 表示管理域可信第三方提供的票据生成函数。首先通过所有的用户身份属性和平台配置属性生成用户域票据,之后管理域可信第三方的私钥 SK_{MTTP} 对该票据进行数字签名,生成的 $ticket_{UD_i}$ 为最终的用户域票据。注意,票据与证书的区别是:票据中不存在公钥。例如,两组相同的数据会生成两个相同的票据,但不会生成两个相同的证书。

定义 2 用户身份票据:设 $U_{i,j}=\{u_{i,j}^1, u_{i,j}^2,\cdots, u_{i,j}^\alpha\}$ 表示需要向用户域管理者申请用户身份票据时提供的 α 项用户身份属性。当用户域管理者收到之后,生成用户身份票据的方法:

$$Uticket_j=sig_{UTTP_i}(SK_{UTTP_i},\Phi_{UTTP_i}(ticket_{UD_i},U_{i,j})) \tag{2}$$

其中,$\Phi_{UTTP_i}(x)$ 表示索引号为 i 的用户域可信第三方提供的票据生成函数;$sig_{UTTP_i}(x)$ 表示索引号为 i 的用户域可信第三方提供的数字签名函数;SK_{UTTP_i} 表示索引号为 i 的用户域可信第三方进行数字签名操作的私钥;$Uticket_j$ 表示索引号为 j 的终端设备得到的用户身份票据。

定义 3 平台配置票据:设 $P_{i,j}=\{p_{i,j}^1, p_{i,j}^2,\cdots, p_{i,j}^\beta\}$ 表示需要向用户域管理者申请平台配置票据时提供的 β 项平台配置属性。当用户域管理者收到之后,生成平台配置票据的方法为

$$Pticket_j=sig_{UTTP_i}(SK_{UTTP_i},\Phi_{UTTP_i}(ticket_{UD_i},P_{i,j})) \tag{3}$$

定义 4 可信终端:如果终端设备使用者用户身份票据和终端设备的平台配置票据同预期的结果相同,则称该终端设备为可信终端。

定义 5 可信用户域:如果用户域管理者的用户域票据同预期的结果相同,则称该用户域为可信用户域。

2 软件可信度量方法

2.1 基于票据的静态可信度量

通过可信第三方对软件和用户域票据进行数字签名,使数字签名在具有不可否认性的同时具备标

识用户域来源的功能。将数字签名附加在软件的指定位置得到发行的软件。具体方式为

$$\begin{cases} s_{UTTP_i} = sig_{UTTP_i}(SK_{UTTP_i}, \Phi_{UTTP_i}(ticket_{UD_i}, app_{raw})) \\ app_{rel} = app_{raw} \parallel s_{UTTP_i} \end{cases} \tag{4}$$

其中 app_{raw} 代表软件的原始二进制文件，app_{rel} 表示发行的软件，s_{UTTP_i} 表示索引号为 i 的用户域对软件的数字签名。当软件在运行过程中出现问题时，可通过 s_{UTTP_i} 发现该软件的来源。当软件的完整性被破坏时，可直接标识为不可信，阻止软件的运行；反之，只能表示软件未被篡改，允许软件运行，但不能确保其可信性。

2.2　基于可信风险树的动态可信度量

为了保障软件运行行为的可信性，需要对软件的行为进行实时监控，定位软件的漏洞等安全隐患。在软件运行时通过建立可信风险树模型实施动态可信度量，从可信性和风险性两个方面进行可信度量。可信风险树模型如图 2 所示，可信风险树的构造如图 3 所示。

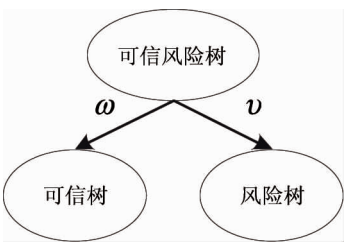


图 2　可信风险树模型

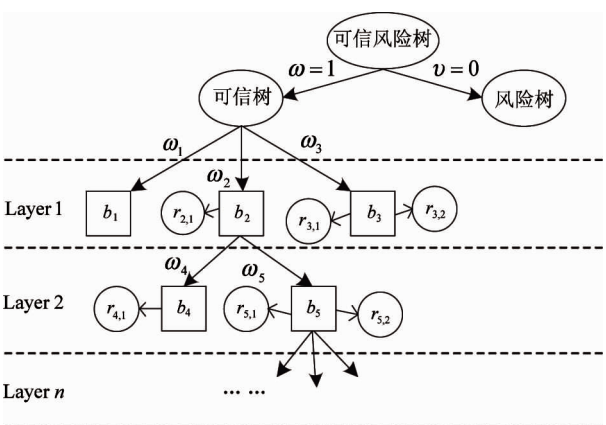


图 3　可信风险树的构造

从图中可以看出，可信风险树由可信树和风险树构成。可信树描述软件的预期行为，风险树描述软件的异常行为。ω 和 υ 分别表示可信风险树的信任权重和风险权重，始终满足

$$\omega + \upsilon \equiv 1 \tag{5}$$

初始状态为软件可信，即 $\omega = 1$，$\upsilon = 0$，此时建立可信树，风险树为空集。随着可信度量的进行，当遇见异常行为时更新 ω 和 υ 的权重值。当技术资料完善时，可直接得出软件的预期行为和依赖资源；如果缺乏相关资料，可通过逆向工程技术分析出软件的预期行为和依赖资源。通过对预期行为和依赖资源的分析，排除软件内部的调用序列，筛选出软件对操作系统的调用序列，通过系统调用序列和依赖资源建立分层的可信树。分层的原则为：当遇见过程转移指令时，则进入下一层。

定义 6　设预期行为集合为 $B = \{b_1, b_2, \cdots, b_i, \cdots, b_n\}$，每一个元素 b_i 是一个四元组：

$$b_i = \langle bid, bcount, rcount, bscore \rangle$$

其中 bid 表示行为的标识符，$bcount$ 表示该行为的子行为数量，$rcount$ 表示该行为直接调用资源的数量，$bscore$ 为该行为节点实际得到的分数。

定义 7　设资源集合为 $R = \{r_1, r_2, \cdots, r_j, \cdots\}$，每一个元素 r_j 是一个三元组：

$$r_j = \langle rid, rtype, rcontent \rangle$$

其中，rid 表示资源的标识符；$rtype$ 表示资源的类型，包括文件、图片和音频等；$rcontent$ 表示资源的内容。

当可信风险树构造完成之后，通过父节点的权重与子节点数量的比值为可信树中所有的行为节点分配权重：

$$\omega_i = \omega_{Parent(i)} / \sum Brother(i) \tag{6}$$

用 b_i^* 表示用户实际产生的一个行为，行为节点的预期总分数用 $btotal$ 表示，包括子行为的数量、资源的数量和自身行为三部分的总和，$btotal = bcount + rcount + self$。其中 $self$ 表示该行为节点运行的情况，当全部运行成功时为 1，否则为 0。当其中某些项无法完成时，则扣除相应的分数，将实际得到的分数保存在 b_i^* 的 $bscore$ 属性中，作为该行为节点的实际得分。

$$\begin{cases} \tau = \prod_{\text{风险路径}} \omega_i \cdot b_i^* \langle bscore \rangle \\ \omega = \omega - \tau \\ \upsilon = \upsilon + \tau \end{cases} \tag{7}$$

用户域管理者每隔一段时间需要从终端设备中选择出可信终端,采集可信终端对软件的可信度量值,并从用户域角度做局部可信评价。在局部可信评价的基础上,由管理域通过每个可信用户域的局部可信评价从管理域角度做全局可信评价。因此,首先需要定义可信终端向用户域管理者发送的可信度量值的概念。

定义 8 可信度量值:设可信终端对软件的可信度量值为四元组:

$$mv = \langle tid, ts, \omega, RTree \rangle \tag{8}$$

其中 *tid* 表示终端设备的标识符, *ts* 表示时间戳, ω 表示信任权重,风险树表示为 *RTree*。

3 基于信任网络的可信评价方法

3.1 滑动窗口模型

当服务端长时间接收每个客户端发送的 *item* 之后,可以通过这些数据对软件进行可信评价。这里论述一种滑动窗口模型,保障可信评价是长时间数据的表现。利用滑动窗口大小来体现可信评价的时间和空间特性,可以保障数据的规模性和可扩展性,根据窗口的时间戳来保证近期数据的重要性和远期数据的衰减性。同时,随着窗口的移动与更新防止单次的分数过低或者过高的情况,滑动窗口模型如图 4 所示。

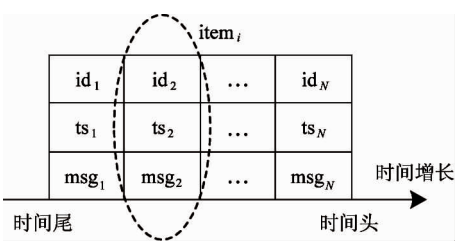


图 4 滑动窗口模型

窗口的大小始终保持为 *N*, 当 *item* 数量很多时,只保留窗口内的 *N* 项评价记录,这样可以保证 *item* 的可扩展性。即使在较少 *item* 是高分数或低分数的情况下,由于总的可信评价是按照全部 *N* 次计算的,所以即使每个 *item* 获得很高或很低的分数时,由于实际发送的次数远比 *N* 小,所以并不能在用户域的范围内获得很高或很低的评价值。随着窗口内 *item* 的更新,很高或很低的 *item* 也会被移除滑动窗口。可信评价的基本思想是:当有新的 *item* 到来时,通过窗口的右移,把时间最长的左边 *item* 移除,新的 *item* 移入窗口的最右边,保证 *item* 的可扩展性。

3.2 可信评价方法

在可信评价时,通过滑动窗口模型对可信度量值的集合建模。图 4 中, *item* 对应 *mv*;msg_i 对应信任权重 ω。可信评价的思路是:越近期的可信度量值所占的比重越大,每项 *mv* 在总的评价中所占的比例与该条目的时间成正比,可信评价用下式计算:

$$eva = \sum_{i=1}^{N} \left(1 - \frac{ts_{now} - ts_i}{\sum_{j=1}^{N} (ts_j - ts_{j-1})} \right) \cdot \omega_i \tag{9}$$

其中 ts_{now} 表示当前的时间戳, *eva* 表示在 ts_{now} 时刻对软件的可信评价。通过 *RTree* 计算风险树中每个节点风险概率,计算的方法是:单一异常行为节点的总数与所有异常节点总数的比值,通过下式计算:

$$p_{b_j}^{risk} = \frac{\sum_{j=1}^{m} b_j}{\sum_{i=1}^{M} b_i} \tag{10}$$

通过上式可以得到软件中异常行为出错的风险概率,对每个节点的风险概率进行降序排列就能找到风险概率高的节点。

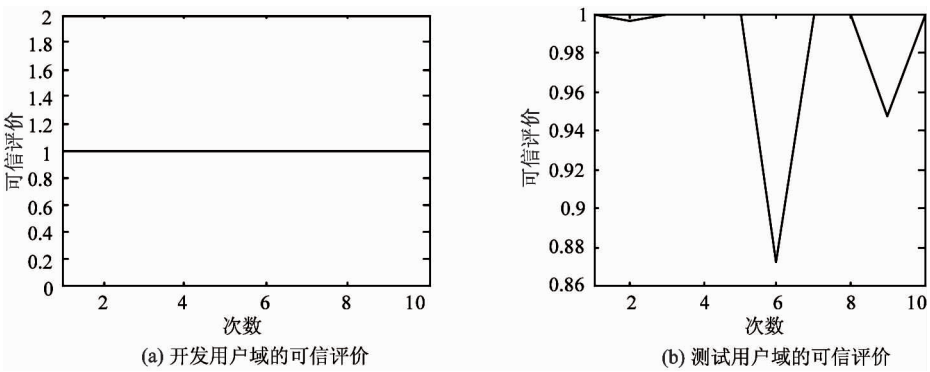
4 实验与分析

4.1 软件可信评价

为了说明本文方法的有效性,实验以为某单位开发的一种移动办公客户端软件为实验对象做可信评价。信任网络由一个管理域和两个用户域组成,两个用户域分别是开发用户域和测试用户域。开发用户域负责软件的设计和开发工作,包括 10 台相同的终端设备,采用处理器为 Intel Core i7-2670QM CPU 2.20GHz、内存为 8GB 的计算机,操作系统为 Windows 7 x64,内核版本为 NT 6.1;测试用户域负责软件的测试工作,包括 50 台软硬件不同的终端设备。信任网络中服务器的处理器均采用 Xeon E7-4809 v2,内存为 32GB 的计算机,操作系统为 Red

Hat Enterprise Linux。当模型对软件进行 10 次局部可信评价时,得到的可信评价值如图 5 所示。

从图中可以看出,开发用户域对软件的可信评价全部是 1,这是因为软件是由开发用户域设计,当所有异常情况解决完成后将软件发布给测试用户域;测试用户域在可信评价时都出现异常情况,这是因为测试用户域中终端设备运行环境的差异性造成的,软件还不能兼容测试用户域中所有的终端设备。



(a) 开发用户域的可信评价　(b) 测试用户域的可信评价

图 5　两个用户域 10 次的可信评价

通过计算测试用户域中异常行为的风险概率,选取测试用户域中风险概率最高的 3 个异常行为作分析,如表 1 所示。

表 1　异常行为的风险概率

序号	异常行为	风险概率
1	CreateProcess(“manage-bde. exe”,…)	0. 132
2	PsSetCreateProcessNotifyRoutineEx(…)	0. 128
3	CreateKey(…)	0. 119

分析:序号 1 中的 manage-bde. exe 是 BitLocker 的控制台管理程序,该技术在 Windows Vista 以及之后的操作系统中被引入,如果终端设备的操作系统低于 Vista 版本,则无法找到该程序;序号 2 中的 PsSetCreateProcessNotifyRoutineEx 是内核层中用户捕获创建进程的行为,也是在 Vista 以及之后的操作系统中被引入;序号 3 中的 CreateKey 行为用户注册表操作没有操作系统版本的限制,但由于 64 位操作系统使用 WOW64(Windows-on-Windows 64-bit)技术来保障 32 位软件的兼容性,注册表的位置发生了变化,因此无法操作指定位置的注册表项。由此可以看出,通过对异常行为的风险概率的分析,可以快速准确地定位异常行为的发生原因,测试用户域将表 1发送给开发用户域,可使软件的完善工作更具有针对性。

4.2　可信评价方法比较

将本文的方法与文献[16]的基于支持向量机方法和文献[17]的基于信任领域和评价可信度量的信任模型(trust model based on the trust area and evaluation credibility,TMEC)方法进行比较。在运行环境差异性方面,通过多台软硬件异构的终端设备共同评价软件的可信性、兼容性和稳定性,尽可能多地发现软件的漏洞和安全隐患;在运行环境的可信性保障方面,通过基于票据的可信选择机制筛选出可信终端和可信用户域,在此基础上对软件进行可信度量和可信评价;在网络环境的可信性方面,通过引入可信第三方建立可信连接架构(TCA)模型,确保终端设备和用户域管理者的通信以及用户域和管理域通信的可信性;在软件可信性方面,结合静态可信度量和动态可信度量技术,通过基于票据的数字签名机制确保软件运行前的完整性,通过可信风险树模型,监控软件运行时的系统调用序列和资源的依赖关系。本文方法和其他方法的比较结果如表 2 所示。

表 2 可信性评价方法的比较

评价方法	运行环境的差异性	运行环境的可信性保障	网络环境的可信性保障	软件可信性	
				运行前	运行时
支持向量机	×	×	×	×	√
TMEC 方法	√	×	×	×	√
信任网络方法	√	√	√	√	√

注："√"表示支持，"×"表示不支持。

5 结 论

本文提出了基于信任网络的软件可信评价方法。该方法首先通过基于票据的可信选择，确保终端设备和用户域的可信性，在此基础上实现对软件的可信度量和可信评价；然后通过基于票据的数字签名和基于可信风险树的可信度量，将静态可信度量和动态可信度量相结合，确保软件运行前完整性的同时，度量运行时的可信性和风险性；最后通过多台软硬件异构的终端设备共同对软件进行可信度量和可信评价，及时准确地发现软件的缺陷和安全隐患。

本文提出基于信任网络的软件可信评价方法尤其适用于评价自主研发专用软件的场景，当软件在小范围可信评价通过之后，可将软件向大范围推广。本文方法已经成功应用于为某单位开发的一种移动办公客户端软件，在软件的可信保障上取得了较好的效果。下一步的工作是对可信评价方法进一步规范化，扩大适用范围，进而从开发、测试、完善等多角度建立一种可信软件的通用设计开发方法。

参考文献

[1] 沈昌祥，张焕国，冯登国等. 信息安全综述. 中国科学 E 辑：信息科学，2007，37(2)：129-150

[2] 沈昌祥，张焕国，王怀民等. 可信计算的研究与发展。中国科学：信息科学，2010，40(2)：139-166

[3] 沈昌祥. 网络空间安全战略思考与启示. 金融电子化，2014，(6)：11-13

[4] 沈昌祥. 信息安全导论. 北京：电子工业出版社，2009. 163-165

[5] Knight J, Randell B. Fundamentals of dependable computing for software engineers. Leiden: CRC Press, 2012. 11-15

[6] Wang Q, Lu Y, Xu Z J, et al. Software reliability model for component interaction mode. *Journal of Electronics (China)*, 2011, 28(4): 632-642

[7] Yang N H, Yu H Q, Qian Z L, et al. Modeling and quantitatively predicting software security based on stochastic Petri nets. *Mathematical and Computer Modeling*, 2012, 55(1): 102-112

[8] Tyagi K, Sharma A. An adaptive neuro fuzzy model for estimating the reliability of component-based software systems. *Applied Computing and Informatics*, 2014, 10(s1): 38-51

[9] Chen X H, Liu J, Liu Z M. Requirements monitoring for Internet-ware: an interaction based approach. *Science China: Information Science*, 2013, 56(8): 1-15

[10] Trusted Computing Group. TPM main specification, version 1. 2. 2003. http://www. trusted-computinggroup. org

[11] 丁卫涛，徐开勇. 基于软件行为的可信评价研究. 计算机科学，2016，43(1)：202-225

[12] 刘玉玲，杜瑞忠，冯建磊等. 基于软件行为的检查点风险评估信任模型. 西安电子科技大学学报(自然科学版)，2012，39(1)：179-184

[13] 田俊峰，张亚姣. 基于马尔可夫的检查点可信评估方法. 通信学报，2015，36(1)：230-236

[14] 韩强. 基于行为的软件可信性度量理论与关键技术研究：[博士学位论文]. 北京：北京邮电大学计算机学院. 2013. 18-20

[15] 吴昊，毋国庆. 程序的动态完整性：模型和方法. 计算机研究与发展. 2012, 49(9): 1874-1882

[16] 丁卫涛，徐开勇. 基于软件行为的可信评价研究. 计算机科学，2016, 43(1)：202-225

[17] Shen G H, Huang Z Q, Xie B, et al. Survey on software trustworthiness evaluation: Standards, models and tools. *Journal of Software*, 2016, 27(4): 955-968

A scheme for software trustworthiness evaluation based on trusted networks

Yan Lin[*,**,***], Zhang Jianbiao[*,**,***], Zhang Ai[****]

([*] College of Computer Science, Beijing University of Technology, Beijing 100124)

([**] Beijing Key Laboratory of Trusted Computing, Beijing 100124)

([***] National Engineering Laboratory for Critical Technologies of Information Security Classified Protection, Beijing 100124)

([****] Beijing-Dublin International College, Beijing University of Technology, Beijing 100124)

Abstract

The software reliability evaluation in the field of trusted computing was studied, and a scheme for software trustworthiness evaluation based on trusted networks was presented. The scheme divides users into different user domains through the trusted network model, and then adopts the trusted selection mechanism based on tickets to protect the trustworthiness of devices, and builds the model of the trusted risk tree from trusted terminals. Two aspects of trustworthiness and riskiness are considered to measure software, and the risk probability is calculated from abnormal behaviors. Multiple terminals and the sliding window model are adopted to evaluate the software trustworthiness in the range of the user domain and management domain. The experiments proved the correctness and the credibility of the method.

Key words: trusted computing, trusted measurement, trusted network, trusted domain, trusted software