基于数据融合的全生命周期软件可信性定量评估方法

张卫祥 刘文红 吴 欣
(北京跟踪与通信技术研究所 北京 100094)

摘 要 提出了一种全生命周期的软件可信性定量评估方法。首先,建立全周期软件可信评估模型,对生命周期各阶段软件可信性进行逐层分解,分别设计定量或定性度量指标;然后,利用知识发现原理,获取软件可信特征树的权值分布;利用数据融合理论,对获得的多类型多量纲的可信度量数据进行分布式综合处理与推理;最后,给出软件可信性评估实例。工程实践表明,该方法能够有效保障软件评估过程的客观性和评估结果的准确度。

关键词 软件定量评估,生命周期,数据融合,可信性,软件工程
中图法分类号 TP311.5 **文献标识码** A

Quantitative Evaluation Across Software Development Life Cycle Based on Data Fusion

ZHANG Wei-xiang LIU Wen-hong WU Xin
(Beijing Institute of Tracking and Telecommunications Technology, Beijing 100094, China)

Abstract This paper brings out a method on quantitative software trustworthy evaluation across software development life cycle. First, builds a hierarchical assessment model with decomposition of software trustworthy on the various stages of software development life cycle, designs appropriate quantitative or qualitative metrics set; then, using of knowledge discovery in database techniques to obtain the weights of all software trustworthy characteristics; finally, using of data fusion theory to process and reason large volumes of multi-type measurement data. Engineering practice shows that it can effectively improve objectivity of the assessment process and the accuracy of the assessment results.

Keywords Quantitative evaluation, Software development life cycle (SDLC), Data fusion, Trustworthy, Software engineering

1 引言

随着信息技术的迅速发展,计算机软件的应用日益广泛,软件失效造成的后果也愈加严重,特别是在航空航天、金融保险、交通通信、工业控制等关系国计民生的重要领域,软件一旦失效将造成重大损失,因此对软件质量提出了更高的要求。虽然软件质量日益受到关注,但由于软件的特殊性,目前仍缺少有效的评估方法,对软件进行客观准确的评估特别是定量评估仍是软件工程领域的重要研究内容。

软件可信性是1990年代引入软件工程化领域的一个较新的概念,软件可信性研究得到国际上广泛的重视。简言之,可信性指的是系统在规定的时间与环境内可交付可信服务的能力,最初包含可用性、可靠性、防危性、保密性、完整性等特征,后来得到不断的延伸和扩充[1]。相比于其它概念,软件可信性更适用于航空航天等对软件安全可靠运行具有极高要求的应用领域中对质量进行刻画与评估。

软件评估的基本内容包括建立评估模型、获取度量数据、拟合评估结果等。在评估模型方面,较为通用的办法是将软件质量概念分解为若干层次,最低层次的软件质量概念再分解为量化指标[2,3]。Zhang等[4]提出了一种层次结构的软件可信性分解模型,能够较好地利用测试用例执行情况并得到软件可信评估结果,但该模型基于软件测试,原始测量数据限于测试阶段,软件可信性信息的采集与利用不够全面。

在软件度量与评估方法方面,模糊综合评判算法近年来以其操作简便实用而受到重视[5,6],但它在模糊计算及权值分配的客观性等方面存在缺陷,导致其在复杂系统的评估,尤其是同类事物的优劣比较时容易引起较大的偏差。杨善林等[7]提出一种基于证据理论的评估方法,但其在获取权值及度量值时均过多依赖专家打分而主观性较强,且未涉及全生命周期评估。

数据融合可广义地概括为把来自多源的数据,根据既定的规则,分析结合为一个全面的情报,并在此基础上为用户提供需求信息的过程[8]。本文给出一种软件可信性评估方法,基于全生命周期的软件可信性评估模型,在软件开发生命周期内各阶段采集可信性度量数据,并利用数据融合理论和知识发现方法对获得的多源多周期数据进行有效分析和整合,实施可信性定量评估,得到更为客观准确的评估结果。

2 全生命周期的软件可信性评估模型

2.1 评估模型结构

全生命周期的软件可信性评估模型(称之为QUEST Ⅱ模型)是一个分阶段分层次的结构模型,如图1所示。

张卫祥(1979-),男,硕士,工程师,主要研究方向为软件工程、软件测试与质量保证,E-mail:wxchung@gmail.com;刘文红(1968-),女,硕士,高级工程师,主要研究方向为软件工程;吴 欣(1975-),男,主要研究方向为软件测试。

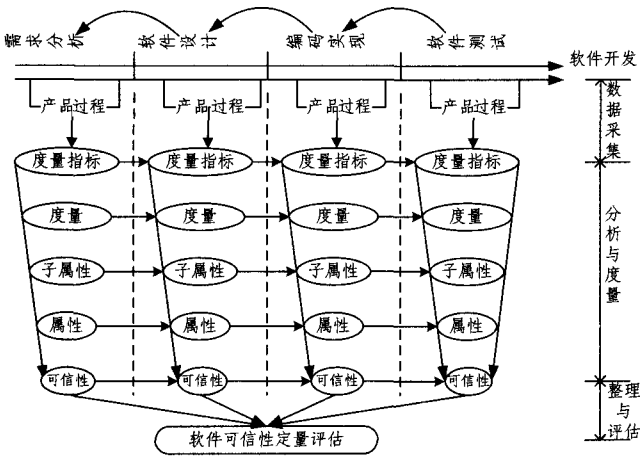

图 1　全周期软件可信性评估模型结构

全生命周期的软件可信性评估模型共有 5 层，首先把软件可信性(Trustworthy)分解为可用性、效能性、可靠性、安全性、实时性、可维护性、可生存性等 7 个可信属性(T-Attribute)；然后，把每个可信属性分解为若干个可信子属性(Sub-Attribute)；进一步，把每个可信子属性分解为若干个可信度量(T-Measurement)；最后，对每个可信度量设计若干个可信度量指标(T-Metric)以实施可信度量数据采集。可信属性、子属性、可信度量、度量指标等统称为可信特征(Trustworthy Characteristic，TC)。

该模型在软件需求分析阶段、软件设计阶段、编码实现阶段、软件测试阶段等软件开发生命周期各阶段分别进行可信度量数据采集，其可信属性、可信子属性的设计在各阶段一致，但可信度量、度量指标的设计在不同阶段不完全一致。

限于篇幅，不再详细描述模型分解结果。可根据待评估软件特点及软件评估要求对可信度量、度量指标进行设计和定制，对本评估方法不产生影响。

2.2　度量指标设计

根据 QUEST Ⅱ模型，度量指标为最低层可信特征，直接实施度量并获取可信度量数据。度量指标设计的好坏直接影响到评估结果质量的高低。

定量指标具有客观和直观的特点，尽量使用定量的度量指标会提高评估结果的准确性。但受限于软件自身及软件评估特有的主观性、应用背景和开发技术等诸多因素，有些指标还无法采用定量的方式进行评价，须采用专家评判的方式进行定性评估。在 QUEST Ⅱ模型中，既有定量度量指标，也有定性度量指标，如表 1 所列。

表 1　标准 QUEST Ⅱ模型中的可信特征数量

	可信属性	子属性	可信度量	度量指标	
				定性	定量
需求分析阶段	7	22	67	36	51
软件设计阶段	7	22	71	35	64
编码实现阶段	7	22	67	21	61
软件测试阶段	7	22	95	7	112

需根据定性或定量度量指标的特点，定义其各自度量方法，限于篇幅，不再详述。无论是定量指标还是定性指标，取得原始度量值后，在进行数据融合前还需进行数据预处理，以便于取得最终的定量评估结果，这将在下节进行介绍。

3　全生命周期的软件可信性定量评估方法

软件可信性评估模型具有复杂的层次结构，底层度量指标较多且具有多类型多量纲的特点，故需对获得的可信度量数据进行分析与整理，以得到较为客观的定量评估结果。

3.1　软件可信特征树

建立如图 2 所示的软件可信特征树(Trustworthy Characteristic Tree，TCTree)，软件可信特征树的各级节点是各层次的软件可信特征，其中，e_i 为可信属性，$e_{i,j}$ 为可信子属性，$e_{i,j,p}$ 为可信度量，$e_{i,j,p,q}$ 为度量指标。权值 w 表示同级指标间的相对重要程度，满足归一化。

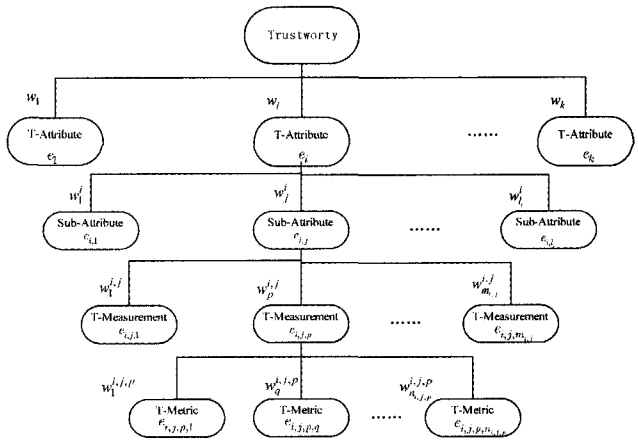

图 2　软件可信特征树

对软件开发生命周期的各阶段，都建立一个可信特征树。由于各阶段的可信度量、度量指标不完全一致，各阶段的可信特征树也不完全一样。

3.2　基于知识发现的权值获取

运用知识发现原理，利用经验数据库获取各级可信特征的权值[4]：

首先，利用经验数据库，根据软件性质、开发人员能力等因素对待评估软件进行分类，确定软件的技术复杂度和管理复杂度。借鉴模糊评判的思想，将技术复杂度和管理复杂度各分为 5 个档次。

然后，基于经验数据库，获取已知技术复杂度与管理复杂度的软件的不同层次软件可信特性的权值。可根据待评估软件的特点与实际需求，对获取的权值进行微调。

最终确定不同阶段软件可信特征树的各级权值 w。

3.3　基于效用统一的度量值预处理

对于定性指标而言，不同的评估专家对同样的一组评价等级会有不同的理解或者偏好，从而可能导致同样的好坏程度对应于不同的得分；即使对定量指标，由于不同指标的难易程度不一，也常会产生类似情况[7]。为此，基于效用统一的原则对原始度量值进行预处理，把定量或定性度量指标的原始度量数据转化为效用值。进行效用值转化还可一并解决软件评估过程中存在的多类型多量纲信息融合问题。

为了下面叙述方便，假设效用等级为 $G=\{G_i, i=1,2,\cdots,n\}$，可信度量指标 e 的度量值为 v_e，信息转化后对应的效用值为 μ_e。

3.3.1　定性指标度量值的转化

利用式(1)，对定性指标度量值进行信息转化：

$$\mu_e=\sum\lambda_i u(G_i) \tag{1}$$

式中，$u(G_i)$ 为效用等级 G_i 对于评估专家的效用值，λ_i 表示专家对效用等级 G_i 的置信度。

举个例子，假如某专家对定性指标 e 给出的评价信息为

$v_e=\{(average,0.3),(good,0.6),(excellent,0.1)\}$，而等级 *average*、*good*、*excellent* 对该专家的效用值分别为 0.6、0.8、0.9，则信息转化后的效用为：

$$\mu_e=0.3\times u(average)+0.6\times u(good)+0.1\times u(excellent)$$
$$=0.3\times0.6+0.6\times0.8+0.1\times0.9=0.85$$

3.3.2 定量指标度量值的转化

对每一个定量指标，需根据该指标的难易程度等实际情况，配置其效用曲线 $u(v)$。简单地，可以通过配置其在抽样点 $S=\{S_i\}$ 上的效用值实现，即为分段效用函数。

当获得定量指标 e 的效用曲线后，原始评估数据 v_e 的效用 $u(v_e)$ 可以通过式(2)计算。

$$u(v_e)=u(S_{i+1})-\frac{S_{i+1}-v_e}{S_{i+1}-S_i}(u(S_{i+1})-u(S_i)),\text{ if } v_e\in[S_i,S_{i+1}] \tag{2}$$

3.3.3 证据理论与 Dempster 合成规则

在软件可信性评估过程中，获取的度量数据等信息必然存在诸多的“不确定”或“不知道”，需要采用不确定性推理方法进行评估推理。

证据理论可以有效地处理不确定信息，是有效的数据融合方法之一，被广泛应用于处理多属性决策问题[9]。证据理论也被称为 Dempster-shafer 理论，或简称 D-S 证据理论。在证据理论中，用集合表示命题，命题的不确定性问题被转化为集合的不确定性问题。引入了信任函数，满足比概率论弱的公理，能够区分不确定和不知道的差异。可以将精确数据和具有不确定性特点的主观判断在统一的框架下进行相容建模。

设 Θ 为识别框架，Θ 的所有可能子集构成 Θ 的幂集 $P(\Theta)$。子集 $A\subseteq\Theta$ 称为命题。一个基本概率分配函数(Basic Probability Assignment，BPA)定义为映射 $M:P(\Theta)\to[0,1]$，满足 $m(\emptyset)=0$ 且 $\sum_{A\subseteq\Theta}m(A)=1$，又称为 *mass* 函数。

多个 *mass* 函数的正交和定义为(Dempster 合成规则)[10]：

$$m_{1,2,\cdots,n}(Z)=\begin{cases}m(\emptyset)=0, & Z=\emptyset\\ \dfrac{\sum_{\cap Z_i=Z}\left(\prod_{1\le i\le n}m_i(Z_i)\right)}{1-K}, & \emptyset\subset Z\subseteq\Theta\end{cases} \tag{3}$$

式中，$K=\sum_{\cap Z_i=\emptyset}m_i(Z_i)$ 为冲突因子，相当于正交运算过程中分配给空集的信任。为了保证 Dempster 合成规则应用的有效性(式(3)分母不为零)，假定作为证据的可信评估信息之间不完全冲突。

3.4 基于数据融合的定量评估算法

对定性或定量指标的度量值进行预处理后，即可对生命周期不同阶段的多层次的可信特征进行数据融合，一般有分布式计算或中心式计算两种方法。

分布式计算的主要思想是，首先在每一个给定的阶段，计算基于所有可信特征所获得的融合后验可信度分配，然后基于在所有阶段上所获得的融合后验可信度计算总的融合后验可信度。中心式计算的主要思想是，首先对每一个可信特征，基于不同阶段的度量数据计算融合后验可信度分配，然后基于这些融合后验可信度进一步计算总的融合后验可信度。

考虑到软件可信性评估模型设计和评估过程的实际情况，一般采用分布式计算方法，按照下面的算法实施定量评估。该算法具体步骤为：

Step1 建立统一的软件可信性评估等级 $H=\{H_s,s=0,1,2,\cdots,5\}=\{0,0.2,0.4,0.6,0.8,1.0\}$。对于软件生命周期任一阶段 C_x，利用式(4)把该阶段任一定性或定量度量指标 $e_{i,j,p,*}$(在不引起混淆时，简记为 e)的效用值 μ_e，在 H 上建立统一的置信度：

$$\beta_s=\frac{H_{s+1}-\mu_e}{H_{s+1}-H_s},\beta_{s+1}=1-\beta_s,\text{if } H_s\le\mu_e\le H_{s+1} \tag{4}$$

Step2 对每个度量指标，计算任意第 L 个专家给出的 *mass* 函数。其中 $\alpha\in(0,1]$ 为折扣系数，这里为 0.95，L 表示第 L 个专家：

对权重最大的关键指标 e_k，使用式(5)构造 *mass* 函数：

$$m^L(H_s|e_k)=\alpha\beta_s$$
$$m^L(H_\Theta|e_k)=1-\sum_s m^L(H_s|e_k) \tag{5}$$

对于非关键指标 e_t，设其权值为 w_t，对应关键指标的权重为 w_k，使用式(6)构造其 *mass* 函数。

$$m^L(H_s|e_t)=\frac{w_t}{w_k}\alpha\beta_s$$
$$m^L(H_\Theta|e_t)=1-\sum_s m^L(H_s|e_t) \tag{6}$$

Step3 利用 Dempster 合成规则(式(3))，对每个度量指标(可信特征树的叶子节点)e 的多个 *mass* 函数进行推理合成(即专家意见集结)，得到 e 的合成 *mass* 函数 $\overline{m}(H_s|e)$ 及 $\overline{m}(H_\Theta|e)$。

Step4 利用式(4)计算可信特征 $e_{i,j,p,*}$ 在 H 上的置信度 $\{\beta_s\}$，其中的效用值按下面公式计算：

$$\mu_{i,j,p,*}\triangleq\mu_{e_{i,j,p,*}}=\sum_s(H_s\times\overline{m}(H_s|e_{i,j,p,*})) \tag{7}$$

Step5 利用式(5)、式(6)计算可信特征 $e_{i,j,p,*}$ 的 *mass* 函数 $m(H_s|e_{i,j,p,*})$ 及 $m(H_\Theta|e_{i,j,p,*})$。

Step6 利用 Dempster 合成规则，对具有同一父结点 $e_{i,j,*}$ 的所有叶子指标 $e_{i,j,p,*}$ 的 *mass* 函数进行合成(向上层推理)，得到父结点 $e_{i,j,*}$ 的 *mass* 函数 $m(H_s|e_{i,j,*})$ 及 $m(H_\Theta|e_{i,j,*})$。

Step7 循环执行 Step4－Step6，继续向上层推理，直至待合成的可信特征为可信性(根节点)为止。

Step8 重复执行上述步骤，遍历生命周期所有阶段 C_*，直至所有待合成可信特征为可信性(根节点)，转入下一步。

Step9 再次利用 Dempster 合成规则，对每个可信属性 a_* 在生命周期不同阶段的 *mass* 函数进行合成(即多周期评估数据融合)，得到各可信属性 a_* 的全生命周期的综合 *mass* 函数 $M(H_s|a_*)$ 及 $M(H_\Theta|a_*)$。

Step10 再次利用 Dempster 合成规则，获得全生命周期的软件可信性的 *mass* 函数 $M(H_s)$ 及 $M(H_\Theta)$。

Step11 最后，使用式(8)获得全生命周期的软件可信性的定量评估结果：

$$T=\sum_s(H_s\times M(H_s)) \tag{8}$$

不失一般性，在上述算法中使用了 Dempster 合成规则。事实上，也可使用其它的证据合成规则(比如 Yager 合成规则[11]、孙全[12]等)，只需将步骤中的证据合成规则进行简单替换即可。

4 软件可信性评估实例

某数据处理软件(DPS)是工程关键软件，实时采集和接收多个来源的各种类型或格式的测量数据并进行实时分析与

处理，生成实时处理结果，其可信性高低具有重要影响。运用本文提出的评估方法，对 DPS 进行了全生命周期的可信定量评估。限于篇幅，这里只给出部分评估过程数据。

建立 DPS 的全生命周期的软件可信性评估模型，设计各级可信特征；根据前述基于知识发现的权值获取方法，DPS 的技术复杂度为 4、管理复杂度为 3，得到各层可信特征的权值分布。表 2 显示了需求分析阶段部分可信特征及其权值分布情况。

表 2　需求分析阶段 DPS 部分可信特征及其权值分布

属性	子属性	可信度量	度量指标
可用性 0.195	适合性 0.30	功能定义充分性 0.25	功能定义充分率 1.0
		功能定义完整性 0.25	功能定义完整率 1.0
		功能定义正确性 0.25	功能定义正确率 1.0
		数据元素定义 0.25	数据元素定义率 1.0
	准确性 0.25	数据处理精度定义 0.5	数据处理精度定义率 1.0
		计算准确性定义 0.5	计算准确性定义率 1.0
	互操作性 0.20	接口协议定义 0.2	接口方式定义率 1.0
		接口数据识别 0.2	接口数据识别率 1.0
		接口方式定义 0.2	接口方式匹配率 1.0
		接口需求文档化 0.2	接口需求文档化率 1.0
		接口需求可扩展 0.2	接口需求可扩展 1.0
	易操作性 0.15	操作一致性 0.4	操作一致性的定义 0.5 风格一致性的定义 0.5
		错误操作纠正的定义 0.2	错误操作纠正的定义 1.0
		默认值的定义 0.2	默认值定义率 1.0
		运行状态易监控性 0.2	运行状态易监控性 1.0
	依从性 0.10	依从性要求定义 1.0	依从性要求识别率 1.0
实时性 0.155	处理及时性 0.60	处理及时性定义 0.5	处理及时性定义率 1.0
		最坏情况下的处理时间 0.5	下限处理及时性定义率 1.0
	实时稳定性 0.40	处理时间最大抖动定义 1.0	处理时间抖动定义率 1.0
可靠性 0.120	成熟性 0.40	成熟性要求定义 0.6	成熟性要求定义 1.0
		失效后处理措施的定义 0.4	失效后处理措施的定义 1.0
	容错性 0.30	错误处理规则识别 1.0	错误处理规则识别率 1.0
	持续性 0.30	持续性要求定义 0.6	持续性要求定义 1.0
		应急计划的定义 0.4	应急计划的定义 1.0
安全性 0.175	安全保密性 0.40	权限控制定义 0.4	权限控制识别率 1.0
		关键事务识别 0.3	关键事务识别率 1.0
		资源安全性定义 0.3	资源安全性识别率 1.0
	完整性 0.30	关键数据识别 0.4	关键数据识别率 1.0
		完整性要求定义 0.6	完整性要求定义 1.0
	防危性 0.30	防危性要求定义 1.0	防危性要求定义 1.0
可生存性 0.100	易恢复性 0.50	易恢复性要求定义 0.4	易恢复性要求定义 1.0
		平均恢复时间定义 0.3	平均恢复时间定义 1.0
		最大恢复时间定义 0.3	最大恢复时间定义 1.0
	健壮性 0.50	临界负载条件识别 0.5	临界负载条件识别 1.0
		强度符合性定义 0.5	强度符合性定义 1.0
……			

利用本文基于数据融合的定量评估方法，对各定量或定性度量指标的度量值预处理后，得到其 *mass* 函数，如表 3 所列。评估专家由 EP1、EP2、EP3 等 3 人组成。根据算法反复进行数据融合推理，计算出 DPS 的可信性评估定量结果，如表 4 所列。

表 3　DPS 的部分度量指标的 *mass* 函数

E	EP1	EP2	EP3
e(1,1,1,1)		H4:0.05, H5:0.95	
e(1,1,2,1)		H4:0.05, H5:0.95	
e(1,1,3,1)		H4:0.43, H5:0.52	
e(1,1,4,1)		H4:0.36, H5:0.59	
e(1,2,1,1)		H3:0.08, H4:0.87	
e(1,2,2,1)		H4:0.12, H5:0.83	
e(1,3,1,1)		H4:0.12, H5:0.83	
e(1,3,2,1)		H4:0.00, H5:0.95	
e(1,3,3,1)		H4:0.19, H5:0.76	
e(1,3,4,1)		H4:0.00, H5:0.95	
e(1,3,5,1)	H4:0.95 H5:0.00	H3:0.24 H4:0.71	H4:0.95 H5:0.00
e(1,4,1,1)	H2:0.19 H3:0.76	H4:0.33 H5:0.62	H4:0.95 H5:0.00
e(1,4,1,2)	H3:0.38 H4:0.57	H4:0.52 H5:0.43	H3:0.19 H4:0.76
e(1,4,2,1)	H4:0.57 H5:0.38	H4:0.21 H5:0.74	H4:0.57 H5:0.38
e(1,4,3,1)		H4:0.00, H5:0.95	
e(1,4,4,1)	H3:0.10 H4:0.85	H4:0.33 H5:0.62	H3:0.19 H4:0.76

E	EP1	EP2	EP3
e(1,5,1,1)		H4:0.00, H5:0.95	
e(2,1,1,1)		H4:0.24, H5:0.71	
e(2,1,2,1)		H2:0.14, H3:0.81	
e(2,2,1,1)		H3:0.10, H4:0.86	
e(3,1,1,1)	H3:0.19 H4:0.76	H4:0.48 H5:0.47	H3:0.76 H4:0.19
e(3,1,2,1)	H3:0.38 H4:0.57	H3:0.24 H4:0.71	H4:0.38 H5:0.57
e(3,2,1,1)		H3:0.34, H4:0.61	
e(3,3,1,1)	H3:0.57 H4:0.38	H4:0.67 H5:0.29	H4:0.38 H5:0.57
e(3,3,2,1)	H4:0.48 H5:0.47	H4:0.62 H5:0.33	H2:0.47 H3:0.48
e(4,1,1,1)		H4:0.57, H5:0.38	
e(4,1,2,1)		H4:0.67, H5:0.29	
e(4,1,3,1)		H3:0.55, H4:0.40	
e(4,2,1,1)		H3:0.01, H4:0.94	
e(4,2,2,1)	H3:0.47 H4:0.48	H4:0.52 H5:0.43	H3:0.76 H4:0.19
e(4,3,1,1)	H3:0.95 H4:0.00	H3:0.43 H4:0.52	H3:0.95 H4:0.00
e(5,1,1,1)	H3:0.57 H4:0.38	H4:0.90 H5:0.05	H4:0.95 H5:0.00
e(5,1,2,1)	H3:0.57 H4:0.38	H4:0.57 H5:0.38	H3:0.76 H4:0.19
e(5,1,3,1)	H3:0.47 H4:0.48	H3:0.43 H4:0.52	H3:0.19 H4:0.76
e(5,2,1,1)	H2:0.29 H3:0.66	H4:0.52 H5:0.43	H3:0.57 H4:0.38
e(5,2,2,1)	H3:0.85 H4:0.10	H3:0.43 H4:0.52	H2:0.47 H3:0.48

（下转第 213 页）

元修改 Linux TCP 内核模块，使其在没有经过三次握手的过程下可以直接创建 TCP 连接。经仿真测试表明，增强的 SYN Flood 模型能更有效地防御高强度的 SYN Flood 攻击，并提供正常的 TCP 服务。基于 Linux 平台，增强的 SYN Proxy 模型实现简单，部署方便，较之目前现有的防御模型有更好的优越性。

参 考 文 献

[1] 一江水. TCP 协议三次握手过程分析[EB/OL]. http://www.cnblogs.com/rootq/articles/1377355.htm,2013-01-05

[2] 李蓬. DDoS 攻击原理及其防御机制的研究[J]. 通信技术，2010,43(4):96-98

[3] 胡鸿，袁津生，郭敏哲. 基于 TCP 缓存的 DDoS 攻击检测算法[J]. 计算机工程，2009,35(16):112-114

[4] 曾小荟，冷明，刘冬生，等. 一个新的 SYN Flood 攻击防御模型的研究[J]. 计算机工程与科学，2011,33(4):35-39

[5] 赵广利，江杨. Linux 平台下防御 SYN Flood 攻击策略的研究[J]. 计算机工程与设计，2009,30(10):2394-2397

[6] 徐图，何大可，邓子健. 分布式拒绝服务攻击特征分析与检测[J]. 计算机工程与应用，2007,43(29):146-149

[7] 王海花，杨斌. Linux TCP/IP 协议栈的设计及实现特点[J]. 云南民族大学学报：自然科学版，2007,16(1):73-76

[8] 赵国锋，邱作雨，张毅. 基于单片机的嵌入式 TCP/IP 协议栈的设计与实现[J]. 计算机技术与发展，2010,19(3):137-140

(上接第 183 页)

高，通信开销较小，能较好地完全量子组密钥的服务，对其大规模服务用户有一定的理论指导意义。后续研究中，将完善组播成员动态变化时密钥的更新环节，并且逐步将研究重点投入到存在中继节点的 QKD 组网环境中，从而提出一套功能更为完善、应用更为广泛的量子组密钥服务方案。

参 考 文 献

[1] Patrick P, John C, David K. Distributed Collaborative Key Agreement Protocols for Dynamic Peer Grou-ps[C]// Computer Science Technical Reports. 2002:02-015

[2] Yongdae K, Adrian P, Gene T. Group Key Agreement Efficient in Communication[C]// IEEE Transactions on Computers. 2003:19-57

[3] 张江，张萌，陈春晓，等. 高效的分布式组密钥协商机制[J]. 清华大学学报，2008,48(1):101-105

[4] 张玉臣，王亚弟，韩继红，等. 自组网环境下基于组合公钥的分布式密钥管理[J]. 计算机科学，2011,38(10):75-77

[5] 赵秀凤，徐秋亮，韦大伟. 群组密钥协商协议的安全性分析方法研究[J]. 计算机科学，2011,38(6):145-148

[6] 陈卫东，刘广伟，刘泽超，等. 分布式组播密钥管理协议中的组密钥生成算法研究[J]. 小型微型计算机系统，2010,31(7):1307-1310

[7] 刘成林，徐秋亮. 基于身份的多安全群组密钥协商协议[C]// 济南：第九届中国密码学学术会议论文集. 2006

[8] 赵龙泉. 基于密钥树的组密钥更新技术研究[D]. 郑州：解放军信息工程大学，2010

[9] 刘广伟. 安全组播中的组密钥管理协议研究[D]. 沈阳：东北大学，2009

(上接第 195 页)

表 4 DPS 的可信性评估结果

	需求分析	软件设计	编码实现	软件测试	多阶段融合后
可用性	0.909	0.933	0.901	0.839	**0.905**
实时性	0.885	0.876	0.888	0.839	**0.839**
可靠性	0.787	0.797	0.731	0.611	**0.792**
安全性	0.768	0.766	0.784	0.786	**0.798**
可生存性	0.679	0.732	0.760	0.644	**0.749**
效能性	0.758	0.744	0.766	0.795	**0.795**
可维护性	0.716	0.625	0.752	0.853	**0.714**
		可信性			0.799

作为关键软件，DPS 经过多次工程任务的考验并得到了用户好评，表 4 所列的评估结果较为符合该软件的实际情况，证明了所提全生命周期软件可信性评估方法的有效性。

结束语 软件质量度量与评估是一个重要而又困难的研究课题，是软件工程中迫切需要解决的一个难题。本文提出的全生命周期软件可信性评估模型综合采集生命周期各阶段的可信度量数据，设计的基于数据融合理论的定量评估算法能有效处理多阶段多类型多量纲的数据并进行合理推理，使用的基于知识发现的权值获取方法可以有效降低评估过程中的主观性。最后，工程实践证明该方法能够给出较为准确的定量评估结果。

下一步将就如何改进可信度量指标的设计，以更全面有效地采集软件全生命周期各阶段的度量数据进行更加深入的研究。

参 考 文 献

[1] 刘克，单志广，王戟，等. 可信软件基础研究重大研究计划综述[J]. 中国科学基金，2008,22(3):145-151

[2] McCall J. The Automated Meaz of Software Quality[C]// 5th COMMPSAC. 1981

[3] ISO/IEC 9126 Information Technology—Software Product Evaluation—Quality Characteristics and Guidelines for Their Use. First Ed, Dec. 1991

[4] Zhang Wei-xiang, Liu Wen-hong, Du Hui-sen. A Software Quantitative Assessment Method Based on Software Testing[C]// Lecture Notes in Artificial Intelligence (LNAI) 7390. Springer, 2012:300-307

[5] 王胜芝，鲜明，王雪松，等. 软件质量综合评价方法研究[J]. 计算机工程与设计，2002,23(4):16-18

[6] 董剑利，时宁国. 基于软件质量评估的模糊综合评判算法研究与改进[J]. 计算机工程与科学，2007,29(1):66-68

[7] 杨善林，丁帅，褚伟. 一种基于效用和证据理论的可信软件评估方法[J]. 计算机研究与进展，2009,46(7):1152-1159

[8] 康耀红. 数据融合理论与应用[M]. 西安：西安电子科技大学出版社，1997

[9] 李烨，蔡云泽，尹汝泼，等. 基于证据理论的多类分类支持向量机集成[J]. 计算机研究与发展，2008,45(4):571-578

[10] Shafer G. A Mathematical Theory of Evidence[M]. Princeton U P, Princetion, 1976

[11] Yager R R. On the D-S framework and new combination rules [J]. Information Sciences, 1987,41(2):93-138

[12] 孙全，叶秀清，顾伟康. 一种新的基于证据理论的合成公式[J]. 电子学报，2000,28(8):117-119