云数据在数据挖掘中的隐私保护

陶林波[1] 沈建京[1] 游庆祥[2] 郭 佳[1]

(解放军信息工程大学理学院 郑州 450001)[1] (常州工学院 常州 213002)[2]

摘 要 云计算环境下用户数据的集中存储为数据挖掘提供了便利条件,同时也为用户的隐私保护带来了挑战。为了解决云数据在数据挖掘条件下的隐私保护问题,提出了云计算环境下的隐私保护模型。该模型以公有云为基础,增加了一个分类预处理模块,设定了分类标准,详细讨论了分类后数据的处理方法,并讨论了该模型下数据的检索、还原方法以及运行环境保护、数据的云端销毁等环节。最后对模型的复杂性及安全性进行理论的对比分析,证明了该模型在数据挖掘条件下对云数据隐私保护的有效性。

关键词 云计算,数据挖掘,核心数据,隐私保护

中图法分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.5.021

Privacy Protection under Cloud Computing against Data Mining

TAO Lin-bo[1] SHEN Jian-jing[1] YOU Qing-xiang[2] GUO Jia[1]

(School of Arts and Sciences, PLA Information Engineering University, Zhengzhou 450001, China)[1]

(Changzhou Institute of Technology, Changzhou 213002, China)[2]

Abstract Data storage under cloud environments is both convenient for data mining and challengeable for privacy protection. In order to solve the problem of privacy protection against data mining in cloud computing, a model was presented under the circumstance of public cloud. A classification and prepocessing module was added to the model. Classification criteria and processing methods of data were discussed in detail together with information retrieval, data reduction, file running protection and data destruction. The security effectiveness of the model was proved through theory analysis.

Keywords Cloud computing, Data mining, Core data, Privacy protection

云计算作为第三次IT革命,以高效率、低成本的使用模式给人们的生活带来了广泛而深远的影响。目前云计算并没有统一的定义,而是一个不断发展完善的发展方向,它可以被理解为以虚拟化技术为基础,以网络为载体,以用户为主体来提供基础架构、平台、软件等服务[1],是一种整合了大规模可扩展的计算、存储、数据、应用等分布式计算资源进行协同工作的超级计算服务模式[2],它把全球范围的软硬件资源看作一个巨大的资源池[3],对服务和资源进行共享与统一管理。这种模式的优势在于它能够更好地专注于为用户提高服务质量,降低用户的IT运营成本,当然在这种模式下,用户最关心的是数据的安全性[4]。而数据的安全性在于对数据的直接保护和对数据潜在信息挖掘的保护。数据挖掘本身需要一定规模的数据才能够实现有效挖掘,而云计算这种集中存储模式为数据挖掘提供了便利条件[5],特别是用户的消费状况、出行情况、工作情况、日常生活状况都会通过终端上传到云计算中心,汇集形成的大数据为云服务提供商有效地进行大数据挖掘、更好地发现用户使用模式、提高服务质量有着极大地帮助,但是这种挖掘在提供个性化服务的同时也极大地暴露了用户的个人信息、生活习惯、财务状况等敏感数据。一旦用户的隐私数据被非法获得或不经意泄露,将造成大范围扩散、恶意篡改,甚至恶意冒充利用,从而对用户的财产、名誉等造成极大的影响。本文将设计一种云数据在数据挖掘条件下的隐私保护模型。

1 云计算环境下的数据挖掘

传统意义上的数据挖掘以一定规模的数据集或者数据库为基础,采用数据挖掘算法进行有用模式的发现。云计算环境下的数据量是一种巨量的大数据(Big Data),它所涉及的数据规模大到无法用目前主流的数据挖掘软件在合理时间内撷取、处理、整理成为帮助用户决策的信息。大数据需要特殊的技术,包括大规模并行处理(MPP)数据库、分布式文件系统、分布式数据库、云计算平台、互联网和可扩展的存储系统等[6,7]。

目前常用的挖掘算法有分类法、回归分析、聚类、关联规则、神经网络、Web数据挖掘等[8,9]。这些方法多采用分布式的数据挖掘方式将分块的任务并行处理。相对于传统的数据挖掘发现的用户信息相对单一、信息之间的关联性不强等不足,云计算环境下的大数据挖掘[10]得到的信息更具有综合性与针对性,对用户私密数据的保护也更具挑战性。

到稿日期:2015-03-18 返修日期:2015-07-17

陶林波(1979-),男,博士生,工程师,主要研究方向为网络安全、大数据挖掘,E-mail:taotlb@126.com;**沈建京**(1961-),男,博士,教授,主要研究方向为人工智能、云计算;**游庆祥**(1974-),男,博士,讲师,主要研究方向为人工智能、图形处理;**郭 佳**(1985-),女,博士生,讲师,主要研究方向为云计算。

2 云计算环境下的隐私保护模型

2.1 数据的分类

不同数据泄露之后对用户造成的影响程度不同，根据这种影响，数据通常被划分为一般数据和核心数据，也有人将数据划分为公开数据、部分公开、受保护、隐私 4 类[11]，划分的标准主要以数据的重要程度来划分，第一种划分方法过于笼统，难以界定边界；第二种划分方法过于细化，容易出现重叠。本文以数据的共享范围和数据泄露后对数据所有者造成的影响程度两个指标为分类标准，将数据划分为以下 3 类。

(1)公开数据，主要来源于公开渠道，数据可以被任何人共享，泄露后基本不会对数据所有者造成影响。

(2)受限数据，只可以被数据所有者授权之后访问或者某一个小群体共享，泄露后对数据所有者造成一定的影响。

(3)核心数据，数据只能被数据所有者自己访问或指定访问者共享但不能被其修改，泄露后对数据所有者造成极大影响。

考虑到效率、成本和安全性几个方面，本文对这 3 种类型的数据采用不同的处理方法，如图 1 所示。

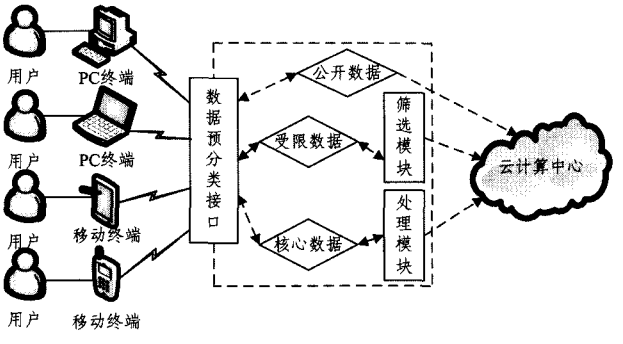

图 1 数据分类预处理

图 1 将现在的(用户，云端)模式进行了完善，中间增加了分类预处理模块，即(用户，数据分类预处理模块，云端)模式，云服务提供商为用户提供一个数据预分类选项，要求用户在将数据上传云端之前对数据的敏感程度做一个预分类，在用户做出选择并将数据上传的过程中，数据没有被直接上传到云计算中心，而是先经过分类预处理模块进行处理。

2.2 数据预处理

分类预处理模块中数据预分类过程相当于对数据增加一个属性，这个属性来源于敏感程度集(公开数据，受限数据，核心数据)，对应属性集 $P(p_1,p_2,p_3)$。满足 p_1 属性的公开数据直接存储在云端，降低了管理成本，通常此类数据占用的比例较大，可以直接利用现有云端资源管理。对于 p_2 属性的受限数据及 p_3 属性的核心数据，特别是 p_3 属性的核心数据，文献[12]采用公有云与私有云混合保存的混合云模式，这种方法的优点在于提高了核心数据的安全性，不足则是需要额外搭建与管理私有云环境。本文将这两类数据区分得更具体，并把它们放在公有云环境下讨论，即利用现有资源降低成本，同时保证安全性。对于核心数据来说，它是原始数据[13]，具有敏感性，不允许被挖掘或限制挖掘深度，受限数据则属于敏感数据，数据的敏感内容在后续数据挖掘中逐渐体现。

因此在云计算环境下，数据挖掘中隐私保护的重点在于保护原始敏感数据，同时要考虑到有些潜在的敏感数据与原始敏感数据有一定的关联性，避免被过度挖掘。p_2 属性的敏感数据挖掘过程可以描述为：设受限数据集为 D_{p_2}，核心数据集为 D_{p_3}，则有

$$D_{p_2} \cap D_{p_3} \neq \varnothing$$

或者是对集合 D_{p_2} 中包含的若干子项$(Val_1,Val_2,\cdots,Val_n)$，存在一些子项满足

$$Val_i \wedge Val_j \wedge \cdots \wedge Val_k \subseteq D_{p_3}$$

对集合 D_{p_3} 中的数据则设定为禁止挖掘，也就是取消或者尽量降低它与其他数据的关联性，即用现在的挖掘算法找不到或者难以找到这些数据的实质性关联。

对于具有属性 p_2 的数据要分两种情况讨论：1)在上传数据中敏感信息片段的组合直接构成核心数据；2)数据中只有小部分的敏感信息片段，组合之后不构成核心数据。不论哪种情况都要在分类预处理模块中先通过一个数据筛选模块，这个模块的作用在于对受限数据中隐含的敏感数据进行处理，既要方便服务商挖掘有效模式、提高服务质量，同时又要保证敏感信息不被过度泄露。该模块的具体结构原理如图 2 所示。

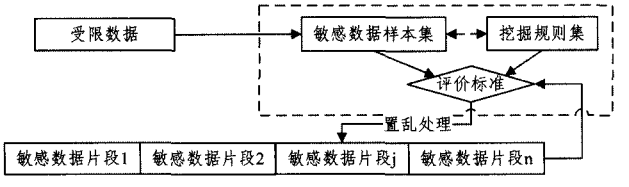

图 2 筛选模块原理

该模块作为分类预处理模块的一部分由云服务商提供，对用户来说是透明的，它由一个敏感数据样本集、挖掘规则集和一个评价标准 3 部分组成，分别记为 SS、RS、ES。对于一个 p_2 类型的数据样本，由于敏感信息相对较少或比较分散，因此用户将其设定为受限数据，但是当这类数据的规模达到一定程度后，在有效的数据挖掘算法下会很快找到很多核心数据，这对用户的隐私安全的威胁是极大的。比如数据中包含姓名、性别、住址、电话等信息，但信息相对零散，它们组合起来又很具体、很重要，因此利用筛选模块对这些信息进行分析处理，处理过程如下。

先将一个具有受 p_2 属性的数据文件 f_{p_2} 提交到敏感数据样本集 SS，SS 中存储一些常见的信息模式，比如姓名、电话、身份证号、银行卡号等信息模式，SS 对 f_{p_2} 进行扫描，经过模式比较，将文件中的敏感数据片段进行标注。接下来将标注的数据片段提交到挖掘规则集 RS 中，RS 提供了云计算环境下的常用挖掘算法，比如分类法、关联规则等挖掘算法。对于标注后的数据片段 RS，将它们按照内置算法进行挖掘，并将挖掘后的结果提交给评价标准 ES，ES 中内置了一个表格，如表 1 所列。

表 1 数据敏感程度评价表

算法 1	片段 1	片段 2	…	片段 m	阈值 t_1
算法 2	片段 1	片段 2	…	片段 m	阈值 t_2
…	片段 1	片段 2	…	片段 m	…
算法 n	片段 1	片段 2	…	片段 m	阈值 t_n

表中内置了 ES 提供的挖掘算法和每种算法设定的阈值，再将这些敏感信息片段提交给相应算法进行评定。上文提到了 f_{p_2} 文件有两种情况，可能从提交的 f_{p_2} 文件中直接挖掘出核心数据，也可能在 f_{p_2} 中没有挖掘出核心数据，但为后续的联合挖掘提供了挖掘样本，为了便于 ES 模块处理，采用以下方法统一处理。

ES 评价过程如下：

1)内置一个计数器 $count$，设初始值 $count=0$，if 有任意算法 $f_i \in ES$，存在

$$f_i(\sum_{j=1}^{h} \text{片段 } j) \geqslant t_i, 1 \leqslant h \leqslant m \quad (1)$$

2)则 $count=count+1$，随机抽取一个片段 j，加入一个字节的随机数作为干扰数据对其进行置乱处理，标注其已做处理标记，然后将处理后的片段 j 重新提交给步骤 1)，如果仍满足式(1)，则继续抽取片段进行置乱，已处理的片段跳过，直到满足步骤 3)。

3)f_i(片段 1)$\vee f_i$(片段 2)$\vee \cdots \vee f_i$(片段 m)$\leqslant t_i, 1 \leqslant i \leqslant n$。

4)判断是否有 $count=0$，如果有，则说明所有数据片段进行过算法扫描，没有发现核心数据，则在 m 个片段中随机抽取 $k(1 \leqslant k \leqslant \frac{m}{2})$ 个片段进行置乱。

5)将处理后的数据文件上传至云端，设计数器 $count=0$。

建立一个记录表 $T_1(ID, filename, f_m, p_2)$ 记录待处理文件信息，其中 ID 为用户云端登录用户名，f_m 为上传文件的编号，p_2 为数据分类属性。为了记录文件敏感信息片段修改的详细情况，建立表 $T_2(f_m, s_i, l_i, flag_i, k)$，并用 f_m 将 T_2 与 T_1 关联，s_i 为敏感数据片起始位置，l_i 为 s_i 对应长度，$flag_i$ 为是否置乱处理标志位，k 为干扰数据在数据片中的插入起始位置。这里 T_1 是一个一维向量表，T_2 则是由敏感信息片段数量决定的多维矩阵表。

对于具有属性 p_3 的数据，则直接经过处理模块将其进行安全处理，使其信息难以被有效挖掘。处理模块的处理过程如图 3 所示。

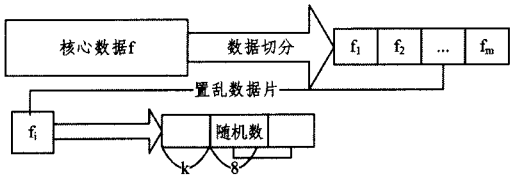

图 3　核心数据处理模块

处理模块先将要上传的核心数据按大小进行分块，形成一个分块序列，即对于一个上传文件 f，有 $f \subset D_{p_3}$，将 f 分解成数据片序列 $f_{s_1} f_{s_2} \cdots f_{s_m}$，建立一个记录表 $T_1(ID, f_m, p_3)$，其中 ID 为用户云端登录用户名，f_m 为上传文件编号，p_3 为数据分类属性。为了记录文件敏感信息片段修改的详细情况，建立表 $T_2(f_m, s_i, l_i)$，s_i 为文件 f 的片段编号，l_i 为片段 f_{s_i} 的长度。分片后对每个数据片段 f_{s_i} 加入一个字节的随机数作为干扰数据，使数据片被置乱，这样使得原始数据难以被直接利用，也使之难以被挖掘到与数据所有者的真实关联性。置乱之后的数据片记录表 T_2 扩展为 $T_2(f_m, s_i, l_i, k)$，k 为干扰数据在数据片中的插入起始位置，f_m 为 T_1 与 T_2 关联键值，其中 T_1 为一维向量表，T_2 为 m 决定的多维矩阵表。

2.3　数据的请求响应与还原

用户将数据上传后，如果要进行查询，则提交一个查询请求 q，q 实际上具有两个参数，即 $q(ID, filename)$，其中 ID 为用户云端登录用户名，$filename$ 为查询文件名。在结果返回给用户之前先要经过分类预处理的逆过程即一个还原模块，模块根据 $ID+filename$ 查询 $filename$ 的属性，当属性为 p_1 时，说明该数据为公共数据，允许被任意共享，也可以被数据挖掘工具挖掘，挖掘到的信息主要是公共信息，有助于服务商为用户提供更好的公共服务，所以将找到的文件直接提交用户，文件可以在云端打开也可以下载到本地打开。

当属性值为 p_2 时，还原模块根据 $filename$ 找到相应的 f_m，再通过 f_m 找到 T_2 表，将置乱过的数据做恢复处理后将原数据提交给用户，p_2 类型数据则不能在云端打开，可以下载到本地打开，p_2 类型数据的处理原则是允许限定挖掘，即挖掘的数据是经过处理的，也就是数据中的大部分类似于公共数据不做处理，这样会提高处理效率；但是小部分的敏感数据片段经过多次多个组合后，经过算法挖掘存在隐私泄露的风险。因此经过处理后这部分的挖掘结果就会与原始信息出现偏差，从而提高了抵抗风险的程度。

对于属性值为 p_3 的数据，同样还原模块根据 $filename$ 找到相应的 f_m，再通过 f_m 找到 T_2 表，将置乱过的数据做恢复处理后将原数据提交给用户。p_3 类型数据不允许在云端打开，可以下载到本地打开，但是云客户端会建立一个专用文件夹，对文件夹中的数据采用沙箱原理运行，核心数据在下载到本地之后在该模式下打开，本地磁盘上其他程序无法共享该数据，如果数据需要修改，则修改后上传至云端，本地运行数据将被销毁。这样设计的目的是不允许核心数据被挖掘或共享。

对于云端数据的销毁，公共数据直接删除，受限数据中的敏感数据片删除后至少进行一遍随机数填充，核心数据则要进行 3 遍以上的随机数填充。

3　模型的有效性评估

模型的有效性分析：当一个文件 f 在未经过分类预处理模块处理的情况下上传至云端，f 中的数据未分类都属于公共数据，即使存在 360 保险箱之类的安全措施，仍然与其他数据共用存储空间，面对数据挖掘仍然有隐私泄露的风险。下面对 p_1、p_2、p_3 3 类数据分别进行讨论。设 P' 为处理前的泄露风险概率，P'' 为处理后的泄露风险概率。对于 p_1 属性数据，由于分类预处理模块没有对其进行处理，则有

$$P'(p_1)=P''(p_1)$$

对于 p_2 属性数据，由于筛选模块对部分敏感数据片段进行了置乱处理，则有

$$P''(p_2) < \max(P(Val_1), P(Val_2), \cdots, P(Val_k)) \leqslant P'(p_2)$$

处理后至少有一个片段被置乱，因此处理后的风险泄露概率被降低。

对于 p_3 属性数据，由于处理模块对每个数据分块进行了置乱处理，假定数据单块风险泄露的风险为 0.5，则数据完全泄露的概率为

$$P''(p_3) < \prod_{i=1}^{m} (\frac{1}{2})^i \leqslant P'(p_3)$$

由以上分析可知，经过本文模型处理的数据在公有云中私密数据泄露的概率被有效地降低了。

结束语　本文设计了一种公有云环境下抵抗数据挖掘的隐私保护模型，模型详细讨论了数据的分类，并对不同类型的数据设计了对应的处理模块及还原方法和销毁手段，最后从风险泄露概率的角度进行了有效性评估。

参 考 文 献

[1] Chowdhury A, Tripathi P. Enhancing Cloud Computing Reliability Using Efficient Scheduling by Providing Reliability as a

Service[C]// 2014 International Conference on Parallel, Distributed and Grid Computing. 2014:99-104

[2] Wu Ji-yi, Ping Ling-di, Pan Xue-zeng, et al. Cloud Computing: Concept and Platform [J]. Telecommunications Science, 2009, 25(12):23-30(in Chinese)
吴吉义,平玲娣,潘雪增,等. 云计算:从概念到平台[J]. 电信科学,2009,25(12):23-30

[3] Lin Wei-wei, Qi De-yu. Survey of Resource Schedulingin Cloud Computing [J]. Computer Science, 2012, 39(10): 1-6(in Chinese)
林伟伟,齐德昱. 云计算资源调度研究综述[J]. 计算机科学,2012,39(10):1-6

[4] Hou Hong-feng, Wang Can, Wang Li-juan. Research on Information Security Issues of Cloud Computing [J]. Electronic Design Engineering, 2012, 20(22):60-62(in Chinese)
侯洪凤,王璨,王立娟. 云计算信息安全问题探讨[J]. 电子设计与工程,2012,20(22):60-62

[5] Marozzo F, Talia D, Trunfio P. A Cloud Framework for Parameter Sweeping Data Mining Applications[C]// 2011 Third IEEE International Conference on Cloud Computing Technology and Science. 2011:367-374

[6] Garcia T, Wang T. Analysis of Big Data Technologies and Methods-Query Large Web Public RDF Datasets on Amazon Cloud Using Hadoop and Open Source Parsers[C]// 2013 IEEE Seventh International Conference on Semantic Computing. 2013:244-251

[7] Yang Hang, Fong S. Countering the Concept-drift Problem in Big Data Using iOVFDT[C]// 2013 IEEE International Congress on Big Data. 2013:126-132

[8] Zhang Wei-dong, Zhang Kai, Dong Qing, et al. Calculation of Information Entropy in Data Mining with Decision Tree [J]. Computer Engineering, 2001, 27(3):71-73(in Chinese)
张维东,张凯,董青,等. 利用决策树进行数据挖掘中的信息熵计算[J]. 计算机工程,2001,27(3):71-73

[9] Zheng Xian. Status and Development of Web Mining[J]. Technology and Market, 2013, 20(3):64-65(in Chinese)
郑弦. Web 挖掘的现状和展望[J]. 技术与市场,2013,20(3):64-65

[10] Wang Zhen-qi, Li Hai-long. Research of massive Web log data mining based on cloud computing[C]// 2013 International Conference on Computational and Information Sciences. 2013:591-594

[11] Dev H, Sen T, Basak M, et al. An Approach to Protect the Privacy of Cloud Data from Data Mining Based Attacks[C]// 2012 SC Companion: High Performance Computing, Networking Storage and Analysis. 2012:1106-1115

[12] Chen Ke-you. The Research on Data Security and Privacy Issues in Hybrid Cloud Computing [D]. Nanchang: Jiangxi Normal University, 2013(in Chinese)
陈科有. 混合云计算数据安全与隐私保护问题研究[D]. 南昌:江西师范大学,2013

[13] LiLing-juan, Zheng Shao-fei. Analysis of Data MiningPrivacy Preserving Technology Based on Data Processing [J]. Computer Technology and Development, 2011, 21(3):94-97(in Chinese)
李玲娟,郑少飞. 基于数据处理的数据挖掘隐私保护技术分析[J]. 计算机技术与发展,2011,21(3):94-97

(上接第 86 页)

[10] Liang Bin, Hou Kan-kan, Shi Wen-chang, et al. A Static Vulnerabilities Detection Method Based on Security State Tracing and Checking[J]. Chinese Journal of Computers, 2009, 32(5):899-909(in Chinese)
梁彬,侯看看,石文昌,等. 一种基于安全状态跟踪检查的漏洞静态检测方法[J]. 计算机学报,2009,32(5):899-909

[11] Qin Xia-jun, Gan Shui-tao, Chen Zuo-ning. A Static Detection Technoogy of Software Code Secure Vulnerabiity Based on First-order Logic[J]. Scientia Sinica Informationis, 2014, 44:108-219(in Chinese)
秦晓军,甘水滔,陈左宁. 一种基于一阶逻辑的软件代码安全性缺陷静态检测技术[J]. 中国科学:信息科学,2014,44:108-219

[12] Zeng Fu-ping, Jin Hui-liang, LU Min-yan. Study on Software Defect Patterns[J]. Computer Science, 2011, 38(2):127-130(in Chinese)
曾福萍,靳慧亮,陆民燕. 软件缺陷模式的研究[J]. 计算机科学,2011,38(2):127-130

[13] Gong Yun-zhan, Yang Chao-hong, Jin Da-hai, et al. Software Defect Patterns and Testing[M]. Beijing: Science Press, 2011:21-22(in Chinese)
宫云战,杨朝红,金大海,等. 软件缺陷模式与测试[M]. 北京:科学出版社,2011:21-22

[14] Chen Z Q, Zhang Y, Chen Z R. A Categorization Framework for Common Vulnerabilities and Exposures[J]. Computer Journal Archive, 2010, 53(5):551-580

[15] Wu Shi-zhong, Guo Tao, Dong Guo-wei, et al. Software Vulnerability Analysis Technology[M]. Beijing: Science Press, 2014:3-6(in Chinese)
吴世忠,郭涛,董国伟,等. 软件漏洞分析技术[M]. 北京:科学出版社,2014:3-6

[16] Allen F E. Control Flow Analysis[J]. ACM SIGPLAN Notices, 1970, 5(7):1-19

[17] Ferrante J, Ottenstein K J, Warren J D. The Program Dependence Graph and Its Use in Optimization[J]. ACM Transactions on Programming Languages and Systems, 1987, 9(3):319-349

[18] Chen Yong-yan, Shu Hong-chun, Dai Wei. Function Vulnerability Detection Method Based on Parse Tree[J]. Computer Science, 2013, 40(8):119-123(in Chinese)
陈永艳,束洪春,戴伟. 基于语法解析树的函数漏洞发现方法[J]. 计算机科学,2013,40(8):119-123

[19] Howard M, LeBlanc D, Viega D J. 24 Deadly Sins of Software Security: programming flaws and how to fix them[M]. 董艳,包战,程文俊,译. 北京:清华大学出版社,2006

[20] Lv Lei, Liu Hong, Li Xin. Method of Building Control Dependence Sub-graph[J]. Computer Engineering, 2009, 35(15):50-52(in Chinese)
吕蕾,刘弘,李鑫. 一种建立控制依赖子图的方法[J]. 计算机工程,2009,35(15):50-52

[21] Zheng Bian-hong. Generating of Static Call Graph and Use Case Model[D]. Xi'an: Xidian University, 2007(in Chinese)
郑变红. 静态程序依赖图和用例模型的生成[D]. 西安:西安电子科技大学,2007

[22] Horwitz S, Reps T, Binkley D. Interprocedural Slicing Using Dependence Graphs[J]. ACM Transactions on Programming Languages and Systems, 1990, 12(1):26-60

[23] NIST[OL]. http://samate.nist.gov/SARD/view.php