单位代码： 10293 密 级：



硕 士 学 位 论 文



论文题目： 基于贝叶斯图卷积神经网络的城市道路

行驶时间分布估计模型研究及实现

1320047726

学号

姓名

导 师

学 科 专 业

研 究 方 向

申请学位类别

论文提交日期

张永超

柯昌博

工程硕士

Urban road based on Bayesian graph convolutional neural network

Research and implementation of travel time distribution estimation model

↑

Thesis Submitted to Nanjing University of Posts and Telecommunications for the Degree of

Master of Engineering



By

Yongchao Zhang

Supervisor: Prof. Changbo Ke

南京邮电大学学位论文原创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京邮电大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

本人学位论文及涉及相关资料若有不实，愿意承担一切相关的法律责任。

研究生学号：\_\_\_\_\_\_\_\_\_\_\_ 研究生签名：\_\_\_\_\_\_\_\_\_\_\_\_ 日期：\_\_\_\_\_\_\_\_\_\_\_\_

南京邮电大学学位论文使用授权声明

本人承诺所呈交的学位论文不涉及任何国家秘密，本人及导师为本论文的涉密责任并列第一责任人。

本人授权南京邮电大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档；允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索；可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。本文电子文档的内容和纸质论文的内容相一致。论文的公布（包括刊登）授权南京邮电大学研究生院办理。

非国家秘密类涉密学位论文在解密后适用本授权书。

研究生签名：\_\_\_\_\_\_\_\_\_\_\_\_ 导师签名：\_\_\_\_\_\_\_\_\_\_\_\_ 日期：\_\_\_\_\_\_\_\_\_\_\_\_\_

摘要

关键词：

Abstract

Key words:

目录

[专用术语注释表 V](#_Toc74851509)

[第一章 绪论 1](#_Toc74851510)

[1.1 研究背景及意义 1](#_Toc74851511)

[1.2 国内外研究现状 2](#_Toc74851512)

[1.2.1 基于风险的访问控制模型研究现状 2](#_Toc74851513)

[1.2.2 风险评估方法研究现状 2](#_Toc74851514)

[1.2.3 隐私策略生成方法研究现状 3](#_Toc74851515)

[1.3 研究内容 4](#_Toc74851516)

[1.4 论文组织结构 4](#_Toc74851517)

[第二章 雾计算下访问控制模型以及隐私策略相关研究 6](#_Toc74851518)

[2.1 雾计算体系结构 6](#_Toc74851519)

[2.2 访问控制模型 6](#_Toc74851520)

[2.2.1 基于任务的访问控制模型 7](#_Toc74851521)

[2.2.2 基于角色的访问控制模型 8](#_Toc74851522)

[2.2.3 基于属性的访问控制模型 9](#_Toc74851523)

[2.2.4 基于风险的访问控制模型 10](#_Toc74851524)

[2.3 风险评估方法 12](#_Toc74851525)

[2.4 隐私策略 13](#_Toc74851526)

[2.5 本章小结 14](#_Toc74851527)

[第三章 雾计算下基于风险评估的访问控制方法 15](#_Toc74851528)

[3.1 基于风险的XACML访问控制模型 15](#_Toc74851529)

[3.2 主体的风险评估 16](#_Toc74851530)

[3.2.1 基于访问次数的主体信誉值计算 16](#_Toc74851531)

[3.2.2 基于信誉值的主体威胁计算 17](#_Toc74851532)

[3.3 主体行为的风险评估 20](#_Toc74851533)

[3.3.1 基于访问矩阵的访问代价计算 20](#_Toc74851534)

[3.3.2 基于访问代价的期望威胁计算 22](#_Toc74851535)

[3.4 风险评估模型 23](#_Toc74851536)

[3.4.1 风险分数的量化 23](#_Toc74851537)

[3.4.2 风险评估计算模型和系统模型 24](#_Toc74851538)

[3.5 风险评估模型的验证 25](#_Toc74851539)

[3.5.1 风险分数的影响因素 26](#_Toc74851540)

[3.5.2 风险分数的准确性评估 28](#_Toc74851541)

[3.6 本章小结 29](#_Toc74851542)

[第四章 雾计算下基于风险的隐私策略自适应方法 31](#_Toc74851543)

[4.1 R2PA的概述 31](#_Toc74851544)

[4.2 文档分析 32](#_Toc74851545)

[4.2.1 基于API的签名和描述进行预处理 32](#_Toc74851546)

[4.2.2 基于预处理确定个人信息 34](#_Toc74851547)

[4.3 代码静态分析 35](#_Toc74851548)

[4.3.1 静态分析的概述 35](#_Toc74851549)

[4.3.2 条件分析 37](#_Toc74851550)

[4.3.3 信息保留分析和信息传输分析 38](#_Toc74851551)

[4.4 隐私策略自适应 39](#_Toc74851552)

[4.5 R2PA的验证 41](#_Toc74851553)

[4.5.1 文档分析和代码静态分析的正确性评估 42](#_Toc74851554)

[4.5.2 隐私策略的充分性评估 43](#_Toc74851555)

[4.6 本章小结 44](#_Toc74851556)

[第五章 原型系统及应用示范的设计与实现 45](#_Toc74851557)

[5.1 基于风险的XACML访问控制原型系统 45](#_Toc74851558)

[5.2 基于风险的XACML访问控制系统应用示范 46](#_Toc74851559)

[5.2.1 背景介绍 46](#_Toc74851560)

[5.2.2 需求分析 47](#_Toc74851561)

[5.2.3 详细设计 47](#_Toc74851562)

[5.2.4 具体实现 49](#_Toc74851563)

[5.3 基于风险的XACML访问控制系统评估 53](#_Toc74851564)

[5.3.1 实验设计 53](#_Toc74851565)

[5.3.2 实验结果及分析 53](#_Toc74851566)

[5.4 本章小结 54](#_Toc74851567)

[第六章 总结与展望 55](#_Toc74851568)

[参考文献 57](#_Toc74851569)

[附录1 攻读硕士学位期间撰写的论文 60](#_Toc74851570)

[附录2 攻读硕士学位期间申请的专利 61](#_Toc74851571)

[附录3 攻读硕士学位期间参加的科研项目 62](#_Toc74851572)

[致谢 63](#_Toc74851573)

# 专用术语注释表

**缩略词说明：**

|  |  |  |
| --- | --- | --- |
| XACML | eXtensible Access Control Markup Language | 可扩展访问控制标记语言 |
| PEP | Policy Enforcement Point | 策略执行节点 |
| PDP | Policy Decision Point | 策略决定节点 |
| PIP | Policy Information Point | 策略信息节点 |
| PAP | Policy Administration Point | 策略管理节点 |
| RAM | Risk Assessment Model | 风险评估模型 |
| RS | Risk Score | 风险分数 |
| ST | Subject Threat | 主体威胁 |
| SSU | Subject Security Uncertainty | 主体安全不确定性 |
| SSL | Subject Security Level | 主体安全等级 |
| SR | Subject Reputation | 主体信誉值 |
| D | Decay Rate | 衰退率 |
| ETO | Expectation Threat for Object | 对客体的期望威胁 |
|  | Probability | 主体损害客体的可能性 |
|  | Context-dependent | 上下文相关的 |
|  | Context-independent | 上下文无关的 |
| *Con* | Context | 上下文 |
| SC | Subject Context | 主体上下文 |
| EC | Environment Context | 环境上下文 |
| OC | Object Context | 客体上下文 |
| *sv* | sensitive vector | 敏感度向量 |
|  | Access Matrix | 访问矩阵 |
| APG | App Property Graph | 应用程序属性图 |
| AST | Abstract Syntax Tree | 抽象语法树 |
| MCG | Method Call Graph | 方法调用图 |
| ICFG | Inter-procedural Control Flow Graph | 程序间控制流图 |
| SDG | System Dependency Graph | 系统依赖图 |
| DRD | Dynamic Risk-based Decision | 基于风险的动态决策 |
| Fuzzy MLS | Fuzzy Multi-level Security | 模糊多级别安全 |
| R2PA | Risk-based Privacy Policy Adaptive | 基于风险的隐私策略自适应 |

# 绪论

* 1. 研究背景及意义

计算世界已变得非常广泛和复杂，未来即将进入一个新时代，万物都将互联。随着科技的快速发展，许多个人和组织开始为用户提供智能设备，如智能手机、智能家电、可穿戴嵌入式设备、传感器和执行器等。基础工作是通过大规模无线传感器网络和连接设备来完成的，这就被称为物联网（IOT）[1]。技术咨询组织Gartner强调，2020年底，连接设备的总数将超过200亿，这些设备用于不同的消费者和商业组织。此外，诺顿（Norton）安全组织预测，到2025年，设备的数量将超过210亿部[2]。根据思科对云的评估，2019年物联网产生的大数据有45%是在网络边缘存储、处理和分析的，数据流在国际数据中心（IDC）已经达到10.4ZB[3][4]。根据IDC预测，大数据的总量将会达到40ZB[5]。例如，波音787飞机每秒产生的数据可能超过5GB，无人驾驶汽车上的传感器和摄像头可能每秒产生1GB的数据[6]。据HIS预测，到2035年，世界上将会有5400万辆无人驾驶汽车[7]。

物联网应用程序可以使用云计算处理，但云计算很难实时处理来自边缘设备的数据，容易发生传输负载、网络延迟和用户隐私数据泄露等问题[8][9]。为了满足环境需求，数据将被迁移到雾计算环境中。雾计算具有网络、存储和计算等应用能力，是一种半虚拟化架构的分布式服务计算范式[10]。雾计算为了克服云计算距离用户远、延迟高的弊端，将雾节点分布在靠近用户的网络边缘，提供了有力的存储、计算和通信资源[11][12]。《雾计算市场的规模和影响》的研究中指出，2022年雾计算市场规模有望达到183亿美元，2018年至2022年的复合增长率将高达104.9%[13]，雾计算拥有广泛的市场潜力和应用范围。然而，雾计算兴起的同时也将带来数据安全的挑战，传统的访问控制很难抵御恶意窃取、入侵和破坏，这就需要采用更好的安全机制来加强雾计算系统的完整性、可用性和机密性。

在传统的访问控制系统中，例如基于角色的访问控制或基于属性的访问控制，当雾节点恶意访问用户隐私数据时，管理员手动修改访问控制策略。但是，雾计算是一种协同的计算环境，被盗的隐私数据可能会泄露给其它所有的雾节点，雾节点之间需要知道收集的个人信息。因此，雾节点的访问权限需要被自动修改，还需要自适应生成雾节点的隐私策略。首先，本文提出了雾计算下基于风险的XACML访问控制模型，在此模型基础上，研究了一个新的风险评估方法，以提高对雾节点风险评估的准确性，可以通过雾节点的风险分数自动修改访问权限；然后，研究了基于风险的隐私策略自适应方法，根据雾节点的风险分数，来提供终端用户或其它雾节点可阅读性的隐私策略；最后，实现了基于风险的XACML访问控制原型系统，同时基于该系统实现了上述两个方法，并使用车联网数据验证原型系统的可行性。课题研究成果可以有效保护雾计算下隐私敏感数据的安全，推进了风险评估和隐私策略自适应方法的应用，对以后研究新型访问控制系统具有重要的意义。

* 1. 国内外研究现状
     1. 基于风险的访问控制模型研究现状

Sepczuk M等人[14]提出了基于上下文数据、面向用户体验的新型认证管理模型，描述了其风险分析的过程，还提出了该机制的风险评估方法，可以支持多步骤认证过程。Jabal A A等人[15]提出了一个框架来分析访问控制策略，以确定是否满足风险要求，该框架收集由于网络环境或用户（即人和设备）触发的数据。Gupta S等人[16]提出了基于风险的多模态生物特征认证解决方案，使按需乘客和拼车服务更安全，更好地为乘客服务。Cao Y等人[17]提出了基于风险的网络物理空间访问控制框架，用于评估用户行为，确保授权用户执行的可疑行为能被正确处理。惠榛等人[18]提出了基于医疗大数据的风险访问控制模型，通过最大期望算法来识别医生不同的访问行为，可以根据风险值的大小自适应调整医生的访问权限。杨宏宇等人[19]根据事件的演算改进了基于属性的访问控制模型，提出了一种云计算下基于风险的动态自适应访问控制模型，并在该模型中构建了动态规则匹配。Choi D等人[20]提出了一个基于上下文感知的医疗大数据风险访问控制模型，根据用户的访问日志、权限文件和上下文信息来评估风险。Fugini M等人[21]提出了一种涉及物理资源的、基于环境风险的动态访问控制模型，以保证动态安全规则符合组织安全策略。

传统的访问控制判定结果是允许/拒绝的二值形式，没有考虑到用户的不确定性，无法适应动态的上下文环境。管理者需要人工制定访问控制策略，面对大量且复杂的物联网数据，制定策略的工作将极大地消耗人力资源。因此，本文针对以上问题，提出了雾计算下基于风险的访问控制模型，弥补了传统访问控制模型的不足。

* + 1. 风险评估方法研究现状

Atlam H F等人[22]提出了一种模糊推理系统与专家判断相结合的风险评估技术，以评估物联网系统中访问控制操作的安全风险。由于访问控制决策中总是存在固有的不确定性，这种不确定性会导致不可预测的风险，而这些风险需要明确的量化。因此，Fall D等人[23]通过分析不确定性提出了一种基于风险的访问控制模型。Khambhammettu H等人[24]描述了一个基于风险的访问控制系统背景下的风险评估模型，该系统通过确定与访问请求相关的安全风险，并将此类安全风险与操作需求以及情景条件进行权衡，从而做出授权决策。Santos D R等人[25]希望在云中使用基于风险的访问控制，并提出了一个基于XACML扩展的框架来执行访问控制策略。Xiao Quan[26]根据传统元模型提出了新型的风险指标体系，该体系通过敏感性分析识别出不同特征的技术创新风险。宛鑫[27]提出了一种基于商业银行的风险客户预警模型，能够准确识别和预测低风险和高风险的信贷客户。师伟[28]设计并实现了一个金融风险管理系统，该系统使用.net语言开发，使用多种评估方法来进行风险量化，可以制定不同的策略来满足用户不同的需求。孙存一等人[29]提出了一种银行信贷业与“机器学习+关联规则”相结合的风险评估算法，可以根据风险值的大小精准地识别出恶意访问的客户。Rubio-Medrano Carlos E等人[30]实现了一种基于用户信任值的动态自适应风险评估框架，可以有效防止攻击者破坏客体的属性来源，方便对每项策略进行风险评估。Sicari Sabrina等人[31]提出了一种端到端系统的风险评估方法，该方法以客观方式评估系统的静态和动态特征，保证了物联网平台组件的可靠性和稳健性。

现有的风险评估算法过于简单，考虑的风险因素较少，评估结论相对粗糙，而且不适合雾计算环境。针对以上情况，本文根据主体信誉值、客体安全属性和上下文进行风险评估，提高了评估结果的准确性，保证了用户隐私数据的安全。

* + 1. 隐私策略生成方法研究现状

Breaux T等人[32]提出了一个从HIPAA（Health Insurance Portability and Accountability Act）隐私规则中直接提取访问权利和责任的方法。Yu L等人[33]将隐私策略中描述的行为与代码中的实际行为进行比较，以检测出有问题的隐私策略。Costante E等人[34]使用了机器学习算法分析隐私策略的完整性。Zimmeck S等人[35]将众包和机器学习技术结合起来检查隐私策略中的基本策略条款。Miao D等人[36]提出了一种自动生成隐私策略的方法，但是它只能分析移动应用程序的源代码，由于缺乏源代码，无法生成正常应用程序的隐私策略。Rowan M等人[37]提出了一个Eclipse插件，它允许开发人员在开发过程中创建隐私策略，但是它不分析应用程序源代码，而且需要开发人员回答一系列的问题，如“你需要收集什么类型的数据”和“你是否与他人共享数据”。Rosen S等人[38]检测了隐私相关应用程序的行为，并向终端用户解释。然而，它是基于手动创建的知识库，该知识库将隐私相关API映射到适当的行为。

现有的隐私策略生成方法只能识别系统调用、方法名和类名，但不分析源代码，不能执行数据流分析，还不能执行信息保留分析和信息传输分析。针对以上情况，本文提出了隐私策略自适应方法，自动为雾节点构建正确和可读的隐私策略，对源代码进行静态分析，识别雾节点收集的个人信息和其相关的上下文信息。

* 1. 研究内容

通过对现有基于风险的访问控制模型、风险评估方法以及隐私策略生成方法三方面存在的问题和不足之处，本文有针对性地做了以下研究工作：

1. 针对现有访问控制模型无法满足雾计算环境多样化的访问控制条件，本文提出了基于风险的XACML访问控制模型，其包含了客体所有者，考虑了用户的隐私需求。在此模型基础上提出了一个新的风险评估方法，包含主体威胁（ST）和主体对客体的期望威胁（ETO），其中ST是主体的威胁度量，ETO是一种期望威胁，因为主体可能损害客体的安全属性。通过实验表明风险分数的精确性是高于DRD和Fuzzy MLS这两种方法的。
2. 在基于风险的XACML访问控制模型下，本文提出了一个可以为雾节点自适应生成隐私策略的新方法。该方法利用文档分析自动将API映射到个人信息，并静态分析每一个雾节点行为，提取调用敏感API的上下文信息，最终基于风险构建正确和可读的隐私策略。实验表明新方法构建的隐私策略比原始的隐私策略更加正确且易于理解。
3. 本文实现了基于风险的XACML访问控制原型系统，在此系统基础上实现了风险评估方法和隐私策略自适应方法，分别为雾节点提供风险评估和自适应生成隐私策略的功能，还添加了XACML节点管理的功能。最后根据车联网数据的应用示范测试整个原型系统的可行性和有效性。
   1. 论文组织结构
4. 绪论。本章首先介绍了雾计算数据环境的背景和面临的问题，然后介绍了基于风险的访问控制模型、风险评估方法和隐私策略生成方法三方面的国内外研究现状，最后概述了本文的研究内容和论文的组织结构。
5. 雾计算下访问控制模型以及隐私策略相关研究。本章首先介绍了雾计算体系结构，然后介绍了常见的访问控制模型，最后着重阐述了风险评估方法和隐私策略等相关知识。
6. 雾计算下基于风险评估的访问控制方法。本章首先介绍了基于风险的XACML访问控制模型，然后在此模型基础上提出了风险评估方法，根据主体信誉值、客体的安全属性和上下文来量化风险，最后通过对比实验验证了该方法生成的风险分数的准确性。
7. 雾计算下基于风险的隐私策略自适应方法。本章首先根据雾节点的风险分数提出了隐私策略自适应方法，然后进行文档分析，基于API签名和描述提取个人信息，之后进行代码静态分析，提取调用敏感API的上下文信息，最后根据简单英语的参考生成基于风险的隐私策略，并通过对比实验验证了该方法生成的隐私策略的可读性和正确性。
8. 原型系统及应用示范的设计与实现。本章首先介绍了基于风险的XACML访问控制原型系统的相关背景知识和系统结构，然后根据车联网数据的实际应用场景设计了基于风险的XACML访问控制系统的应用示范，最后验证了基于风险的XACML访问控制原型系统的有效性和可行性。
9. 总结与展望。本章首先总结了论文的主要研究工作，然后分析了现阶段研究工作存在的不足以及制定了下一步研究工作的计划与展望。

# 雾计算下访问控制模型以及隐私策略相关研究

* 1. 雾计算体系结构

雾计算的层次结构由以下三层组成，如图2.1所示。

边缘层：该层最靠近终端用户和物理环境，由各类物联网设备组成，例如智能手表、智能汽车和智能手机等。这些设备在地理位置上广泛分布，并具有一定的计算能力。它们负责感知物理对象或事件的特征数据，并将感知到的数据传输到上层进行处理和存储。

雾层：该层由许多位于网络边缘的雾节点组成，包括雾服务器、网关和基站等。雾节点广泛分布在边缘设备和云之间，它们具有计算、传输和临时存储数据的能力。

云层：该层由多个高性能服务器和存储设备组成，例如智慧城市、智慧工厂和智慧社区等。



图2.1 雾计算的体系结构

* 1. 访问控制模型

访问控制是指主体对客体的访问请求按照预定义的规则进行验证和授权。这些规则构成了一个授权策略，定义和实施这些规则的方法构成了一个访问控制模型。传统的访问控制主要有基于任务的访问控制、基于角色的访问控制和基于属性的访问控制。但由于雾计算环境的复杂，访问权限需要被自动修改，并且访问控制策略要自适应调整，传统的访问控制无法满足这些要求。因此，研究者们在传统的访问控制下提出了基于风险的访问控制，满足了雾计算的环境需求。

* + 1. 基于任务的访问控制模型

基于任务的访问控制模型（Task-Based Access Control，TBAC）根据任务来建立安全机制和实现安全模型，并实时提供了动态的权限管理。该模型不会出现主体被提前分配权限的情况，因为这样会导致系统拥有多余权限产生不安全的风险因素。模型中用户的访问权限由主体、客体和任务三者共同决定，如果访问控制中的任务被挂起或者终止时，用户的访问权限会消失，只有任务再次执行时，系统会重新分配用户的访问权限。基于任务的访问控制模型将访问控制中的各种关系与实际工作流相结合，因此具有极强的灵活性[39]。



图2.2 基于任务的访问控制模型

图2.2为基于任务的访问控制模型，一些重要的概念如下：

任务：访问控制中的逻辑单元，是与多个用户相关的可区分动作。

授权结构体：由一个或多个逻辑上联系在一起的授权步组成。

授权步：授权步是在访问过程中的最小单元，由受托人集和多个许可集组成。

许可集：受托集的成员被授予授权步时所拥有的访问权限。

受托人集：可被授予执行授权步的用户集合。

依赖：授权步之间或授权结构体之间的相互关系。

* + 1. 基于角色的访问控制模型

基于角色的访问控制模型（Role-Based Access Control，RBAC）是一个细粒度模型，可以根据系统中用户的活动和角色来规范用户对应用程序和资源的访问。RBAC根据主体在雾云计算环境中的职责和用户的角色对其进行授权。角色可能因主体（用户）不同而不同，这意味着在该模型中，主体的责任比主体本身更重要。

RBAC模型主要分为RBAC0、RBAC1、RBAC2和RBAC3四个模型。RBAC0包括用户、角色、权限、对象和操作等基本的元素，同时还增加了会话的元素，RBAC0模型如图2.3所示。在RBAC0模型的基础上，RBAC1增加了角色继承的概念，通过角色r1继承角色r2，那么r1就拥有了r2的所有权限，RBAC1模型如图2.4所示。RBAC2模型是基于RBAC0模型增加了约束的概念，解决了多个角色之间可能权限冲突的问题，RBAC2模型如图2.5所示。该约束根据生效时间的不同又可以分为两类，一类是在管理员为用户分配角色时，如果角色之间的权限存在冲突，则被定义为静态约束。另一类是在用户激活角色时发现角色之间的权限存在冲突，则被定义为动态约束。RBAC3模型是基于RBAC0模型对角色继承和角色约束进行了综合。

RBAC模型需要管理员对角色进行权限的分配，当角色的数量多到一定程度时，安全管理员对于角色的划分和授权工作将变得十分困难，还有它不考虑上下文信息（例如地址、时间和设备等）和用户动态的访问行为。



图2.3 RBAC0模型



图2.4 RBAC1模型



图2.5 RBAC2模型

* + 1. 基于属性的访问控制模型

基于属性的访问控制（Attribute-Based Access Control，ABAC）是根据访问者的主体属性和客体属性来决定是否授予用户访问权限，分离了策略管理和访问决策。由于主体和客体自带属性，管理员不再需要去分配，所以ABAC管理十分简单。ABAC会根据实际访问的情况改变对应策略，可以适应数据量大和动态性强的新型计算环境，为云计算、雾计算和物联网等新型计算提供了理想的访问控制模型[40]。

ABAC模型如图2.6所示，当策略执行点向策略判定点发送基于属性的访问请求时，首先策略判定点请求策略执行点查询相关的策略，还请求属性管理点查询属性信息，然后根据查询返回的结果综合判定，并将结果返回给策略执行点，最后策略执行点根据返回的访问控制结果决定是否允许主体的访问请求。



图2.6 ABAC模型框架

通过以上分析可以发现ABAC模型能较好地适应开发环境，但同样的问题是当数据规模量很大时，对属性的管理仍然是一项非常具有挑战性的工作。

* + 1. 基于风险的访问控制模型

基于风险的访问控制对每个访问请求都必须动态地分析和判定，不仅需要制定预定义的策略，还要考虑上下文信息，例如操作风险、用户需求和行为的好处[41][42]。在现在的研究中，主要有以下三种基于风险的访问控制模型。

1. 基于模糊逻辑的风险访问控制

研究者们提出了模糊逻辑方法来处理风险评估的不确定性，根据风险的真实度来计算结果，从而在访问控制中作出正确的请求决策。模糊逻辑方法运用了模糊集合理论来解决实际的问题，通过隶属度来表示，隶属度位于连续实区间[0, 1]中，而不是在集合{0, 1}中[43]。Leliott R[44]首次提出了模糊MLS模型（Fuzzy Multi-Level Security Model），可以基于模糊逻辑方法进行风险分析。Bell等人构建了一个可行的风险等级，如图2.7所示。该模型将未经授权而导致的预期损失价值的可能性定义为风险，根据风险判断是否授予用户访问权限。



图2.7 QRRAC的风险等级

1. 自适应风险访问控制

早在2004年，研究人员由于新型计算环境的动态特性提出了风险自适应访问控制（Risk Adaptive Access Control，RAdAC），具体的访问过程如图2.8所示。

RAdAC可以根据风险自适应作出访问决策，并基于风险动态地调整策略，具有很强的灵活性和自适应性[45]。因此，RAdAC决策基于以下访问因素。

①操作需要（Operational Need）：在RAdAC的访问过程中，操作需要在特殊的情况下比安全风险更加重要。

②安全风险（Security Risk）：安全风险是由用户的信任值、技术组件的鲁棒性、保护能力、访问客体的属性值、上下文环境以及历史访问记录组成的。

③环境因素（Situational Factors）：包括主体信息、客体属性或状态和应用程序的配置环境等上下文。

④访问控制策略（Access Control Policy）：RAdAC策略定义了风险因素的权重，并制定了与访问决策相关的策略。

⑤启发式（Heuristics）：启发式应被考虑进每个访问决策中，通过访问控制策略为其定义级别，根据三角分布指定的最大值、最小值和最可能值来量化风险，并使用蒙特卡洛模拟方法来衡量风险因素量化的不确定性。



图2.8 RAdAC抽象过程

1. 基于风险的XACML访问控制

随着云计算、雾计算和物联网等新型计算的出现，传统的访问控制无法保证新型计算中数据资源的安全，人工制定策略十分繁琐和复杂，并且风险没有考虑到用户的不确定性，使得TBAC、RBAC和ABAC很难适应新型计算环境的动态和异构特性，无法支持用户协作和信息共享。因此，在基于XACML的访问控制基础上，添加了风险评估方法和隐私策略自适应方法，保证了用户隐私数据的安全，有效地解决了上述问题。这部分的内容将在第三章和第四章进行详细的介绍。

* 1. 风险评估方法

基于风险的访问控制需要评估风险和权衡收益，使系统在风险评估之后采取正确的措施来降低风险。风险评估主要运用各种定性和定量的方法，为访问决策和制定策略提供理论依据。风险评估由以下三个重要部分组成：

1. 风险评估因素的确定

风险评估根据主体信息、上下文环境和访问请求的客体属性来量化风险，风险因素的不同，导致风险值不相同。风险因素主要分为6组，并为每组分配权重。第1组（请求者的特征）表示与请求访问资源的人或应用程序相关的风险，例如请求者的角色、组织中的排名、安全等级、访问级别和教育级别等因素；第2组（IT组件的特征）表示与组件相关的风险，例如机器类型、应用程序、网络、连接和身份验证等因素；第3组（情境因素）表示与请求情境相关的风险，例如请求者在特定任务中的角色和被请求资源的时间敏感性；第4组（环境因素）表示与请求环境有关的风险，例如请求者的地理位置；第5组（请求信息的特征）表示与资源本身相关的风险，例如分类级别、权限级别和加密级别；第6组（启发式）表示与以前类似请求相关的风险，例如恶意的违规和成功的传输。

1. 权重的计算

权重对应着风险因素的分配因子，通过制定不同比例的权重，使得在访问请求过程中量化的风险值不同。权重的估计与用户的历史访问记录有关，根据日志中的记录可以调整系统中各风险因素的权重，其结果解释了风险评估因素的不确定性。在分配权重的过程中，应当使用客观的方法对访问请求者进行风险评估，可以尽可能地采用多种对比方法，提高权重分配的合理性。

1. 评估方法的选择

风险评估首先分配各个风险因素的权重，并进行风险因素的量化，然后将通过权重的加权方法得到风险值，最终基于风险值判断是否授予用户访问权限。评估者可以根据不同的风险因素量化方式来选择不同的风险评估方法。风险评估包括定性评估方法、定量评估方法和综合评估方法，三种评估方法的比较如表2.1所示。

表2.1 风险评估方法的比较

|  |  |  |  |
| --- | --- | --- | --- |
|  | 优点 | 缺点 | 评估方法 |
| 定性评估方法 | 直观反映对象的深层含义，结果全面 | 主观性太强，对评估者要求过高 | 推导演绎理论分析、逻辑分析法、德尔菲法、风险综合评价法 |
| 定量评估方法 | 直观表示数据结果，结果严密和科学 | 复杂的事物会简单化、模糊化 | 熵权系数法、风险图法、因子分析法、聚类分析法 |
| 综合评估方法 | 定性与定量评估相结合，结果适用范围广 | 局限性较大 | 层次分析法、障碍树法、风险矩阵分析法、数据包络分析 |

* 1. 隐私策略

隐私策略是一份告知用户他们的信息将如何被收集、使用和披露的声明，未能提供正确的隐私策略可能会导致罚款。但是，编写隐私策略是一件复杂且容易出错的事。例如，隐私策略的作者可能不太了解应用程序的源代码，也不理解调用每个API函数的行为。此外，开发者可能不知道集成第三方库的内部结构，通常也不会提供源代码。现有的隐私策略无法解决这些问题，因为它们依赖于人工干预，无法为用户自适应生成正确和可阅读性的隐私策略。



图2.9 隐私策略样本（com.macropinch.swan）

现有的隐私策略可能包含五种信息：①联系方式及身份信息；②收集和使用的个人信息；③需要数据的原因；④信息将被披露的第三方；⑤用户权限。由于①、③和⑤无法通过分析应用程序识别，隐私策略自适应方法重点为②和④生成语句。例如，图2.9显示了com.macropinch.swan的隐私策略，包含了上面显示的个人身份信息，联系方式显示在底部，它们都在虚线矩形框中。这些信息属于①，并且隐私策略自适应方法无法生成。红色矩形框中的句子表示将要收集的信息，蓝色矩形框中的陈述表示信息如何被披露给第三方。这些信息分别属于②和④，由隐私策略自适应方法创建。

隐私策略的句子包含三个关键部分，包括执行者、行为和资源。其它部分都是可选的，例如条件、目的和目标。执行者是收集、使用和保留信息的实体。如图2.10所示，如果主语是“we”，该行为由应用程序执行；如果主语是第三方库，那么这些信息将会披露给第三方。行为是指执行者所做的事情，例如图2.10中的“collect”。资源是一个行为操作的客体，例如图2.10中的“your location information”。



图2.10 隐私策略中句子的结构

* 1. 本章小结

本章主要对雾计算下访问控制模型以及隐私策略的相关概念进行详细的介绍。首先，介绍了雾计算的三层体系结构，了解雾计算的相关背景知识；然后，介绍了三种传统的访问控制模型，在此基础上，着重阐述了基于风险的访问控制模型；最后，介绍了风险评估方法和隐私策略的构成及其句子结构。

# 雾计算下基于风险评估的访问控制方法

传统的访问控制很难适应动态的上下文环境，无法满足雾计算安全性和隐私性的要求。然而，基于风险的访问控制可以根据主体、客体和环境来计算风险分数，系统可以实时生成访问控制策略。因此，本章提出了雾计算下基于风险评估的访问控制方法，通过主体信誉值、客体安全属性和上下文来量化风险。对比试验结果表明，新方法提出的三类用户风险分数的准确性高于DRD和Fuzzy MLS。

* 1. 基于风险的XACML访问控制模型

在XACML访问控制框架中，主要包含PEP、PIP、PDP和PAP四个组件。

策略执行节点（PEP）：PEP拦截了对系统资源的访问请求，并将请求转发到策略决定节点（PDP），PDP可以作出访问决策。PEP执行从PDP接收到的访问决策，同时PEP履行了PDP在访问决策中可能包含的义务。XACML义务是从PDP到PEP的一条指令，说明了在授予访问之前和之后必须执行什么操作。

策略决定节点（PDP）：PDP评估访问请求，并根据描述访问请求元素（主体、客体和操作）的可用属性和和适用的访问控制策略作出访问决策。

策略信息节点（PIP）：PIP充当PDP根据ABAC策略评估访问请求所需属性的来源。PIP会提供访问请求所需的属性，还会收集与访问请求的主体、客体和环境有关的属性，在进行访问决策时通过上下文处理程序将属性提供给PDP。

策略管理节点（PAP）：PAP支持策略管理功能，包括添加、删除和修改访问策略，还存储由策略管理员定义的访问策略。

基于风险的XACML系统部署在云服务器中，客体所有者的隐私数据存储在云服务器中。在基于风险的XACML系统中，每当PDP收到来自PEP的访问请求时，PDP首先从PAP和PIP请求额外的信息，然后作出访问决定。基于风险的XACML参考模型如图3.1所示，PDP请求关于特定主体和客体的风险分数的信息，然后作出决策。当PEP接收到雾节点请求时，PAP和PIP匹配请求和资源，创建基本策略，然后验证请求是否被接受。在本文中，PDP必须检查主体的基本策略和风险分数，才能允许主体访问客体。一旦PEP接收到雾节点访问请求（步骤1），请求将被传输到PDP（步骤2），PDP请求PIP中的客体属性（步骤3-4），还会通过风险评估模型（RAM）计算风险分数。根据PDP提供的风险分数，PAP自动更新访问控制策略（步骤5-8）。最后，根据策略验证PDP是否允许主体的请求。PEP执行策略，并将日志传输到义务服务（步骤9-11）。



图3.1 基于风险的XACML参考模型

* 1. 主体的风险评估
     1. 基于访问次数的主体信誉值计算

定义1主体信誉值（SR）：主体信誉值定义为二元组<S, HR>，其中S是主体，HR是一段时间内主体请求并访问系统的日志。HR是一个二元组，是历史恶意访问的次数，是历史友好访问的次数。本文使用Beta分布来计算信誉值，公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (3.1) |

其中，是gamma函数。信誉值是由Beta分布的数学期望获得的，主体信誉值模型如图3.2所示。

|  |  |  |
| --- | --- | --- |
|  |  | (3.2) |

其中，和分别为当前恶意访问次数和友好访问次数，B是函数的衰减率，，。根据图3.3所示的贝叶斯概率模型，本文将雾节点分为三类。对于雾节点，如果友好访问次数明显大于恶意访问次数，那么计算得到的信誉值通常超过0.68。因此，这些雾节点被称为诚实的；反之，当雾节点恶意访问次数多于友好访问次数时，这些雾节点被称为恶意的；如果雾节点友好访问和恶意访问的累积次数稍有不同，其信誉值为0.5左右，则这些雾节点被称为自私的。

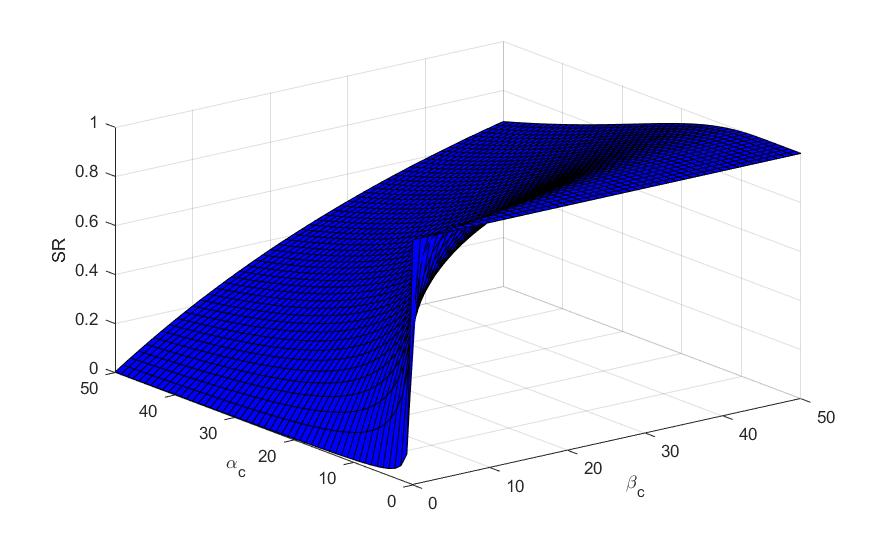


图3.2 主体信誉值模型

为了避免冷启动问题，系统会信任新的雾节点不会滥用资源，因此被分配为0，即，所以。也就是说，雾节点可以访问当前访问控制策略允许的资源。然而，当雾节点有恶意访问行为时，其信誉值会下降。如果用户的信誉值趋于0，根据风险评估模块，雾节点会变得更具有威胁性。

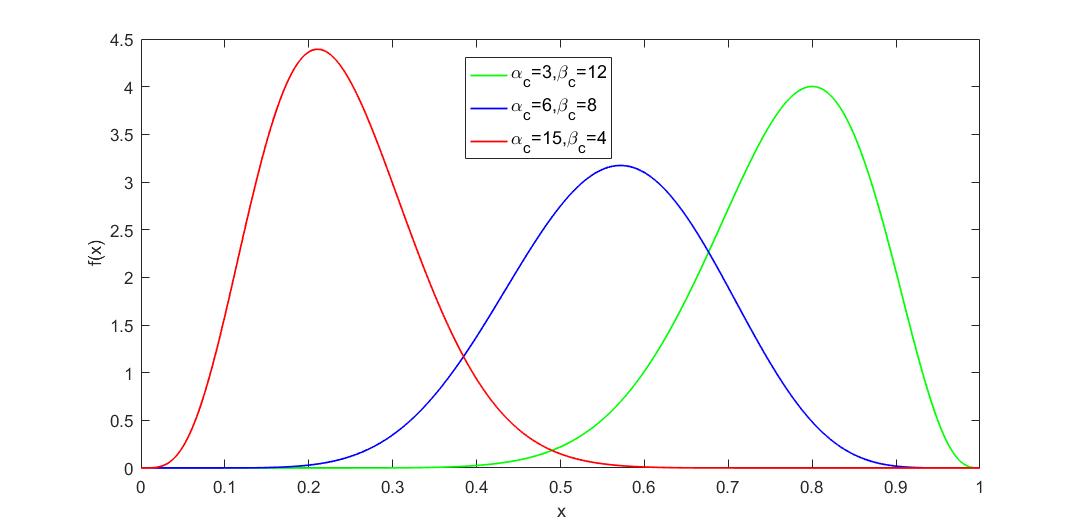


图3.3 贝叶斯概率模型

* + 1. 基于信誉值的主体威胁计算

定义2敏感度向量（*sv*）：敏感度向量是客体（用户隐私数据）的敏感程度。设为客体集合，敏感度可以描述为，是客体（）的敏感性程度。根据隐私敏感度的来源，本文将用户分为两类。有些用户对隐私数据很敏感，为满足用户的隐私需求，客体敏感度被设置为区间上[0.00, 1.00]的任意实数，0表示最弱的敏感度，1表示最强的敏感度。一旦用户在系统中设置了他们的隐私需求，用户可以在区间[0.00, 1.00]为客体敏感度配置一个具体的值。其他人对隐私数据不敏感，根据第一类用户的历史值，系统将计算出客体敏感度的平均值。客体敏感度分为5个等级：A++[0.91, 1.00]，非常敏感；A+[0.81, 0.90]，中度敏感；A[0.71, 0.80]，敏感；B+[0.50, 0.70]，轻微敏感；B[0.00, 0.49]，不敏感。一般情况下，当敏感度大于或等于0.50（），隐私数据为敏感隐私数据。

示例1：对于隐私数据级别，假设姓名：A，地址：A，{姓名，地址}：A+，{姓名，地址，手机号码}：A++，具体如图3.4所示。



图3.4 隐私敏感度的等级

对于敏感度，假设Bob对自己的隐私不敏感，而Alice、Mike和Joe在事件日志数据库中分别将A+赋值为0.9、0.8和0.85，此时计算敏感度的平均值，Bob的A+敏感度为0.85。

定义3主体安全不确定性（SSU）：SSU是在威胁值中对主体的安全级别权重，SSU的值是通过对主体信誉值的分析得出的。具体公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (3.3) |

其中，SR是一个雾节点的信誉值，D是衰减率，D的值为1-*sv*，即D=1-*sv*。主体安全不确定性和主体信誉值的关系如图3.5所示。随着信誉值的增加，那么安全级别的权重就会减小。也就是说，SSU会降低。

定义4主体威胁（ST）：由于考虑到SSL和SSU的主体属性，ST是敏感隐私数据的威胁值，具体公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (3.4) |

其中，SSL（主体安全级别）是雾节点历史风险分数平均值的倒数。假设SSL值区间为0-1。例如，SSL划分为5个级别，0-0.2，0.21-0.4，0.41-0.6，0.61-0.8，0.81-1。值越小，对SSL的威胁越大。

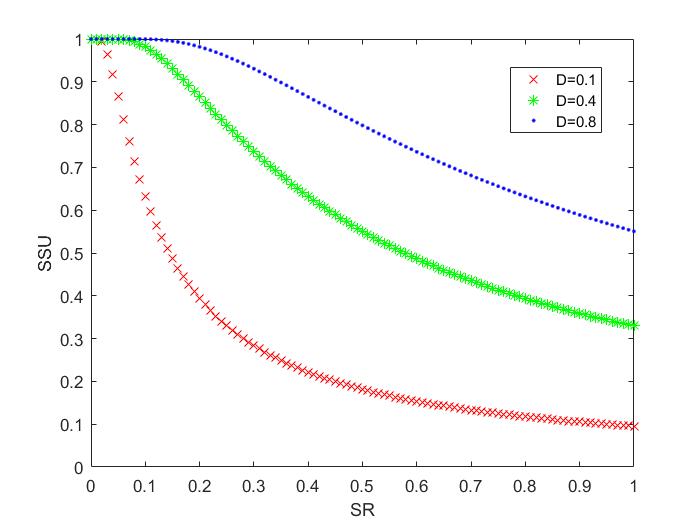


图3.5 SSU和SR的关系趋势

定义5上下文（*Con*）：上下文是一个三元组<SC, EC, OC>，其中，SC是主体上下文，例如雾节点的优先级，雾节点的权限等。EC是应用程序的配置环境，或者用户居住的自然环境，例如时间（白天或晚上），地址（在家或在外），网络状态（在线或离线）等。OC是客体属性或状态，例如，文件大于100M，访问行为是“下载”。

根据公式3.4可知，上下文*Con*的变化导致任何主体的安全级别SSL的变化，主体威胁ST将相应改变。SSL和ST之间的关系是成反比的。

示例2：这是一个主体威胁的示例。假设通过访问控制来保证学生管理系统的安全性，该学生通过内部网和因特网来查询信息。三个学生的查询记录如表3.1所示。

表3.1 三个学生的查询记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 主体 | 历史记录 | | 上下文 | 主体安全级别 |
| α | β | 位置 |
| Alice | 2 | 18 | 在学校 | 0.2 |
| Mike | 8 | 32 | 在学校 | 0.2 |
| Joe | 8 | 32 | 不在学校 | 0.6 |

假设B为0.1，D为0.2，即B=0.1，D=0.2。Alice主体威胁ST的计算方法如下。



同样，Mike与Alice不同，Mike主体威胁的值为0.174。



同样，Joe与Mike不同，Joe的主体威胁值为0.552。



根据上述情况，Mike的恶意行为次数大于Alice，所以Mike的ST值较大。Mike的上下文与Joe的上下文不同，这导致了ST的不同，尽管他们有着相同的行为。

* 1. 主体行为的风险评估
     1. 基于访问矩阵的访问代价计算

当经过身份验证的雾节点访问系统中的数据时，其行为可能会威胁敏感隐私数据的安全属性。隐私数据的安全属性包含可用性、完整性和机密性，即，其中是安全属性，A是可用性，I是完整性，C是机密性。威胁与上下文有关，例如，当因特网发生故障时，FTP会话很容易中断，威胁就会降低。

定义6访问矩阵（****）：当雾节点需要满足用户的功能需求时，雾节点访问用户的隐私数据。访问行为可能会改变隐私数据安全属性的值，因此，访问向量就是布尔矩阵的向量，它被定义为一个3行n列的布尔矩阵，。如果隐私数据的属性已被更改，布尔矩阵的向量值为1；如果属性未被更改，则向量值为0，即，其中。例如，在远程医疗过程中，医院的雾节点通过可穿戴设备采集生理数据，血压为70pa，心率为100bpm，温度为37℃。当雾节点修改血压和温度的值，并将心率的值传递给保险公司时，访问矩阵可以表示为。

定义7访问代价（Access Cost）：访问代价包含两类，一类是上下文无关的，另一类是上下文相关的。

1. 上下文无关的（****）

与上下文无关的访问代价是隐私敏感度和访问矩阵的一个函数，它们是正相关的。设访问代价矩阵为，由访问矩阵和隐私披露向量矩阵计算得出，即。具体表示为：

|  |  |  |
| --- | --- | --- |
|  |  | (3.5) |

示例3：当雾节点A请求私有数据时，包括姓名、地址、邮编、电话号码和信用卡号码，假设该行为的访问矩阵为。通过对用户需求的分析，可以得到隐私敏感度向量，即。然后，访问代价矩阵的计算如下：。

根据公式3.5，。因此，雾节点A请求隐私数据集的基于上下文无关（）的访问代价为4.1。

1. 上下文相关的（****）

上下文相关的访问代价是一个上下文和上下文无关的访问代价的函数，它们是正相关的。上下文相关的访问代价可以根据上下文无关的访问代价和上下文计算出来。本文分成三类来讨论，影响安全属性的访问代价计算如下：

|  |  |  |
| --- | --- | --- |
|  |  | (3.6) |

示例4：Alice的权限配置为view，请求雾计算中的数据对象以满足它的功能需求。具体上下文环境如表3.2所示。由示例3得到，，然后根据公式3.6计算值，即。

表3.2 上下文环境和上下文值

|  |  |
| --- | --- |
| 上下文环境 | 上下文值 |
| 记录太大 | *Con*1 = 1 |
| 事务会话已满 | *Con*2 = 0 |
| 数据未加密 | *Con*3 = 0 |
| 网络连接失败 | *Con*4 = 0 |

* + 1. 基于访问代价的期望威胁计算

定义8主体对客体的期望威胁（ETO）：设为主体损害客体的概率，是ETO的基数，，是风险容忍度，。因此，ETO的计算方法如下：

|  |  |  |
| --- | --- | --- |
|  |  | (3.7) |

ETO和的关系如图3.6所示，从中可以知道曲线趋势。假设主体的恶意行为越多，概率越高，也越高，因此ETO趋于1。假设功能需求对一个主体非常重要，较低的值将被配置。反之亦然，对于一个主体，假设隐私要求非常重要，那么将配置较高的值。

示例5：在示例4中，假设是0.2，是0.8，即，=0.8，然后，ETO的值可以通过公式3.7计算。即。

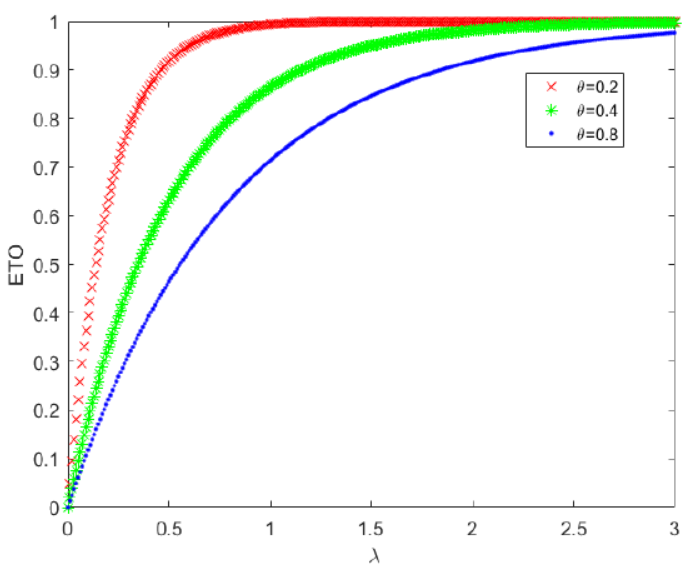


图3.6 主体对客体的预期威胁

* 1. 风险评估模型
     1. 风险分数的量化

定义9风险分数（RS）：主体的访问行为可能会损害客体的安全属性（如可用性、完整性和机密性）。风险分数为客体的期望威胁值，考虑主体威胁ST和主体对客体的威胁ETO，风险值可按公式3.8计算。

|  |  |  |
| --- | --- | --- |
|  |  | (3.8) |

其中，和是调整RS值的参数，，是恶意访问次数，是友好访问次数。当大于时，对于公式3.2，，此时主体一定是个恶意用户。因此，恶意用户的权限“允许”很快被修改为“拒绝”。通过调整的值，RS的值可以变大，具体如图3.7所示。当小于时，对于公式3.2，。尽管存在一些恶意行为，但该主体仍然可以被视为友好用户。通过调整的值，RS的值可以变大，具体如图3.7所示。

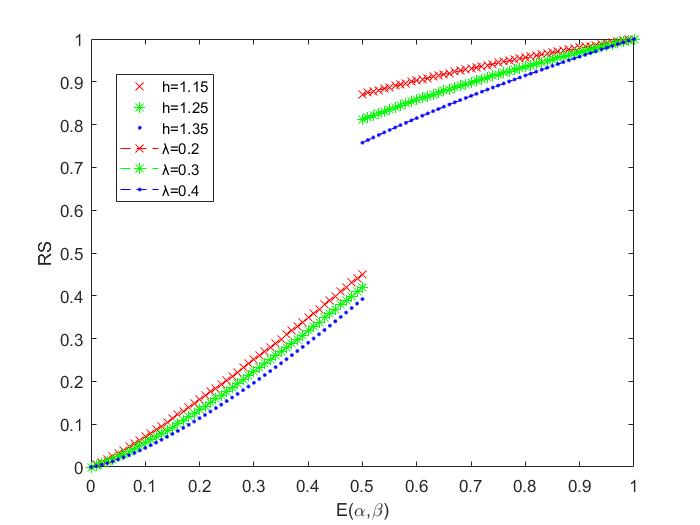


图3.7 分段函数调整RS值

定理1：当主体请求客体时，期望风险由主体的上下文和历史访问行为决定。

证明：根据公式3.8，，其中，。根据公式3.3，假设SR>0，。根据公式3.2，，因此，，ST是由决定的，即ST是由主体的历史访问行为决定的。

根据公式3.7，，其中，根据公式3.6，，其中，，因此，，因为是由客体的安全属性决定的。通常，客体的安全属性是一个常量，所以也是一个常数，，因此ETO是由概率和上下文Con决定的。

总而言之，RS是由上下文（）和主体的历史访问行为（或）决定的。

* + 1. 风险评估计算模型和系统模型

本文使用树形图描述了RAM计算模块，如图3.8所示。在左子树中，首先，根据和得到主体信誉值SR。然后，通过SR和D得到主体的安全级别权重SSU。最后，通过SSU和SSL计算主体威胁ST。在右子树中，首先，利用*sv*和得到上下文无关的访问代价。其次，分析了主体上下文、环境上下文和客体上下文，通过*Con*和得到了上下文相关的访问代价。然后，通过和得到了主体对客体的期望威胁值ETO。最后，通过ST和ETO计算风险分数RS。

RAM系统模型由请求处理器、RAM计算模块和数据库组成，其中RAM计算模块如图3.8所示。当主体（雾节点）向访问系统中的PDP请求隐私数据时，具体如图3.1所示，PDP将调用RAM系统模型并计算雾节点的风险分数。

首先，请求处理程序将雾节点ID传输给主体信誉值（SR），SR在日志事件中检索其友好访问次数和恶意访问次数，并计算SR的值。然后将SR值转化为主体安全不确定性（SSU），主体安全级别（SSL）是通过计算雾节点的历史风险分数的平均值的倒数获得的。ST是由SSU和SSL获取。

其次，请求处理器监控雾节点的计算行为，生成访问矩阵。同时，通过*sv*数据库中的对象id查询敏感度向量（*sv*），或者根据客体所有者的隐私偏好提供敏感度向量。上下文无关是通过*sv*和获得的，分析上下文并计算上下文相关（）。请求处理器分析雾节点的历史访问数据，得到损害隐私数据安全属性的概率（）。主体对客体的预期威胁（ETO）由和上下文相关（）的访问代价计算得出。

最后，通过ST和ETO得到风险分数（RS），RAM计算模块将RS发送给PDP。同时，RS将被恢复到日志数据库中。RAM系统模型如图3.9所示。



图3.8 RAM计算模型



图3.9 RAM系统模型

* 1. 风险评估模型的验证

系统RAM用Java实现并部署在模拟的雾环境中。在该雾计算中有100个用户，每个人可以分享或随机访问10种客体100次。用户的风险值由RAM计算，该客体分为四个安全级别，包括Top Secret，Secret，Confidential和Unclassified。客体敏感度的值配置在[0.5, 1]的区间内，客体敏感度的四个区间（[0.5, 0.6]，(0.6, 0.75]，(0.75, 0.9]，(0.9, 1.0]）分别映射为客体的四个安全级别（Unclassified, Confidential, Secret 和Top Secret）。

本文为用户随机分配安全级别，假设用户集合表示为，用户数量为k，即。其中是诚实的（Honest），是自私的（Selfish），是恶意的（Malicious）。系统RAM中的参数值，B=0.1，D=0.4，=0.3，=1.6，=0.4。三类用户访问行为（查看、修改、删除）的概率分别为0.33。

* + 1. 风险分数的影响因素

根据定理1中的风险评估模型分析，期望风险是由历史访问行为（或）和上下文决定（）的，其中是恶意访问次数，是友好访问次数。上下文（）随客体敏感度的变化而变化。因此，本文主要关注的是访问次数和客体敏感度如何影响风险分数。

1. 访问次数和风险分数之间的关系

在100次访问中，假设诚实主体的值在[0, 10]区间内，即。同样地，对于自私主体，，对于恶意主体，。在图3.10中，随着系统访问次数的增加，曲线最终趋于水平。绿色曲线在0.85左右，蓝色曲线在0.2和0.75之间，红色曲线在0.2左右。在(0, 100)的访问次数间隔内，绿色曲线保持在0.85左右，因为恶意主体的友好访问次数最少。根据公式，是最小的，由于，在三类主体中SR最小，即主体信誉值是最小的。假设SSL是不变的，由于，ST是最大的，根据公式2.8，RS也是最大的。因为和，绿色曲线相对平滑。相反，在访问次数间隔(0, 100)中，红色曲线保持在0.20左右，也相对平滑。然而，蓝色曲线在区间(0.2, 0.75)内上升和下降，由于自私主体恶意或友好访问次数的不确定性。对于自私主体，，的间隔可以被分成(10, 50)，50和(50, 60)。如果，那么；如果，那么；如果，那么。根据公式3.9，蓝色曲线在区间(0.2, 0.75)内上升和下降。

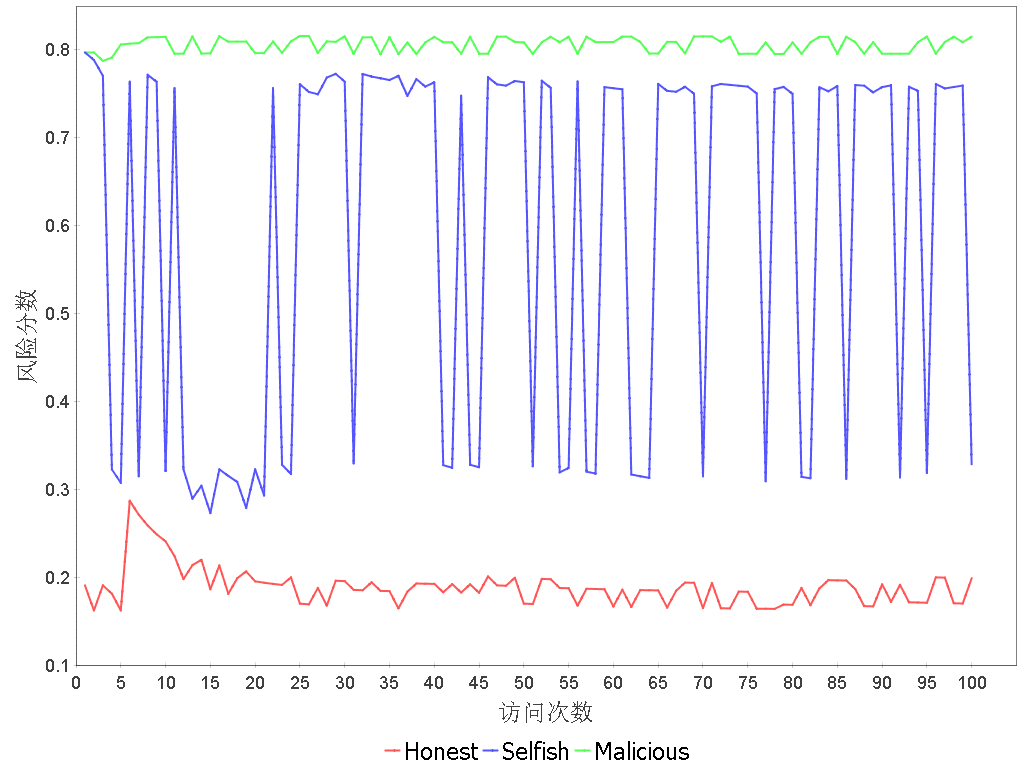


图3.10 访问次数和风险分数的关系

1. 客体敏感度和风险分数的关系

根据主体分类的定义，恶意主体的值是最大的，自私主体的值是中等的，诚实主体的值是最小的。因此，在图3.11中，恶意主体的风险分数（绿色曲线）是最大的，同样，诚实主体（红色曲线）的风险分数是最低的，并且自私主体的风险分数分布在绿色曲线和红色曲线之间。本文将客体敏感度区间(0.9, 1.0]映射为安全级别Top Secret，当客体敏感度为0.9时，风险分数上升至0.75左右。客体敏感度区间[0.5, 0.6]映射为安全级别Unclassfied。因此，在[0.5, 0.56]区间内，诚实主体的风险分数由0.8下降至0.26。

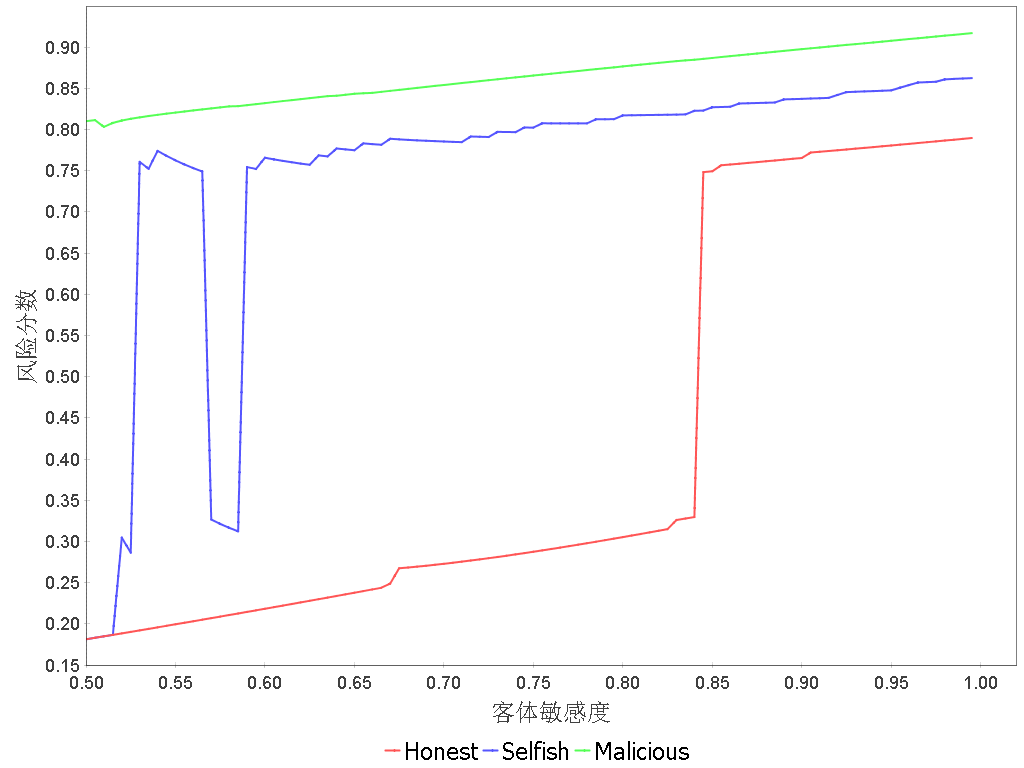


图3.11 客体敏感度和风险分数的关系

* + 1. 风险分数的准确性评估

本文模拟三类主体（诚实、自私和恶意）来访问RAM，诚实、自私和恶意分别有50个主体。在图3.12中，横轴上区间0-50代表50个诚实主体，区间51-100代表50个自私用户，区间101-150代表50个恶意主体。通过与DRD[46]（Dynamic Risk-based Decision）方法进行对比实验，分析了本文的方法RAM的优势。

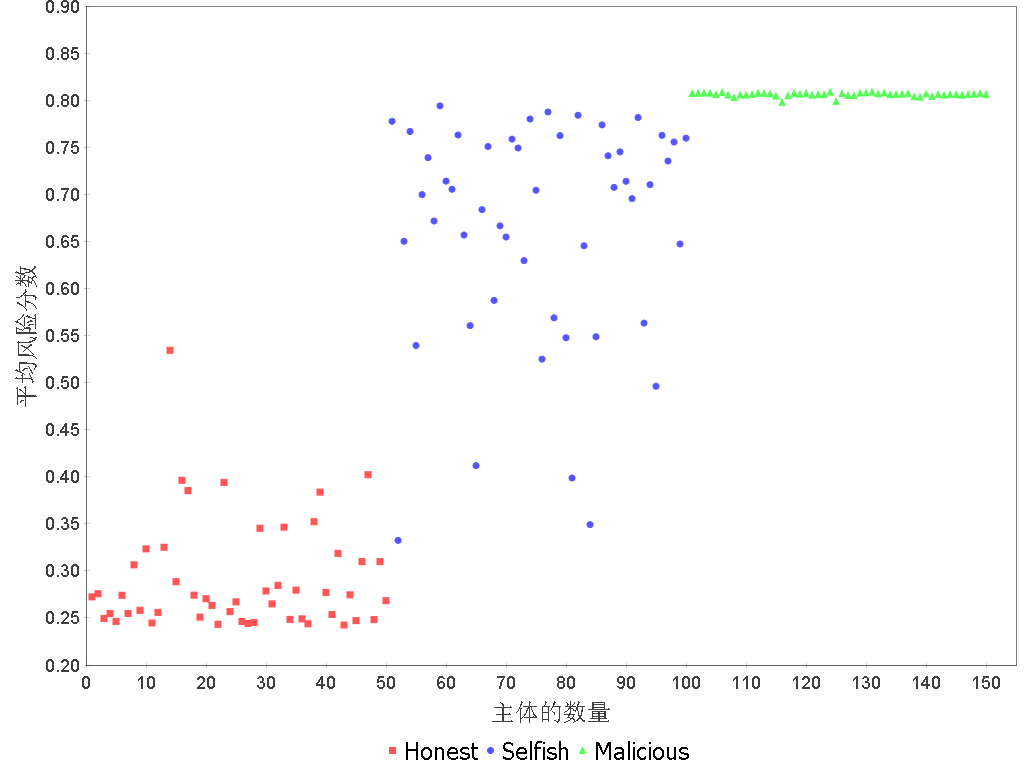


图3.12 RAM的平均风险分数

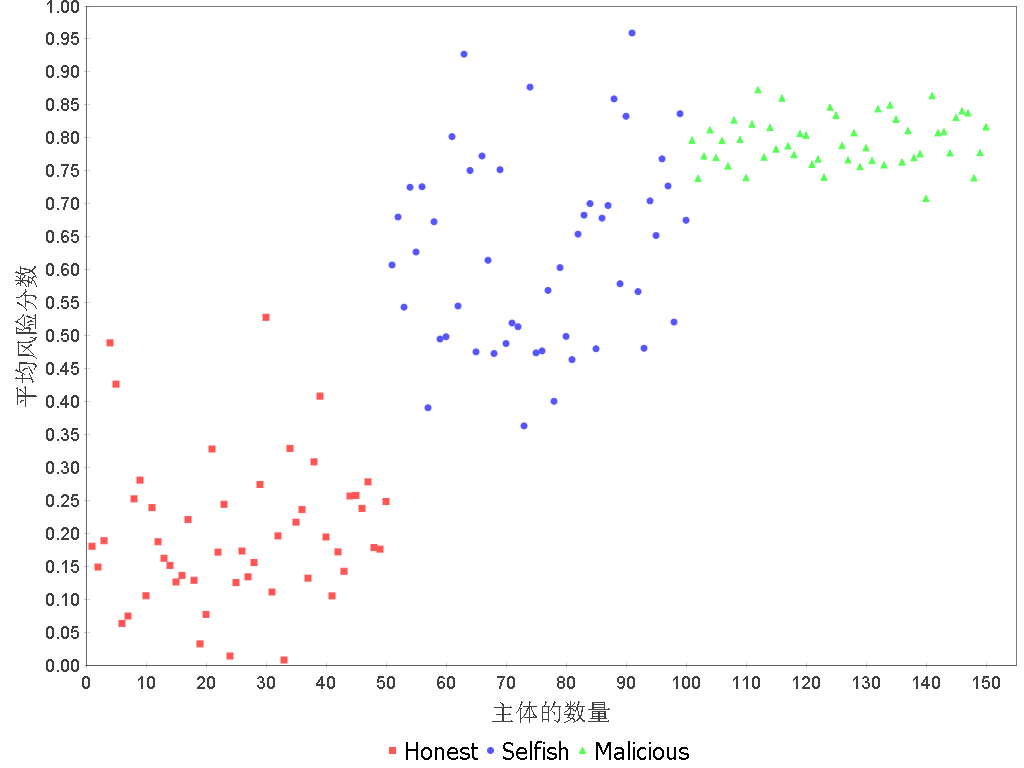


图3.13 DRD的平均风险分数

在图3.12中，大部分红点分布在区间(0.25, 0.35)内。然而，在图3.13中，红点分布区间在(0,0.55)内。对于诚实主体，红点的分布表明RAM风险分数的准确性高于DRD。在图3.12中，对于自私主体，蓝点分布在区间(0.35, 0.80)内。在图3.13中，大部分蓝点分布在区间(0.4, 0.85)内。然而，许多蓝点、红点和绿点重叠，这些结果将导致诚实、自私和恶意主体的风险分数相同。在图3.12中，对于恶意主体，绿点分布在0.82左右。然而，在图3.13中，绿点分布在(0.7, 0.85)之间，有部分恶意用户的风险分数偏低，导致恶意用户更容易泄露用户的隐私数据。

根据3.5.1章节分析了三类主体风险分数的区间分布，对于诚实主体，区间为(0, 0.40)；对于自私主体，区间为(0.40, 0.80)；对于恶意主体，区间为(0.80, 1.00)。本文通过风险分数的分布区间来计算准确性（A），即，其中是对应主体区间内的点数，是分类主体总的点数，A是准确性。在图3.14中，RAM的准确性高于DRD和Fuzzy MLS[47]（Fuzzy Multi-level Security）。对于诚实主体，RAM的准确性是96%，DRD的准确性是92%，Fuzzy MLS的准确性是80%。对于自私主体，RAM的准确性是94%，DRD的准确性是82%，Fuzzy MLS的准确性是72%。对于恶意主体，RAM的准确性是96%，DRD的准确性是44%，Fuzzy MLS的准确性是60%。

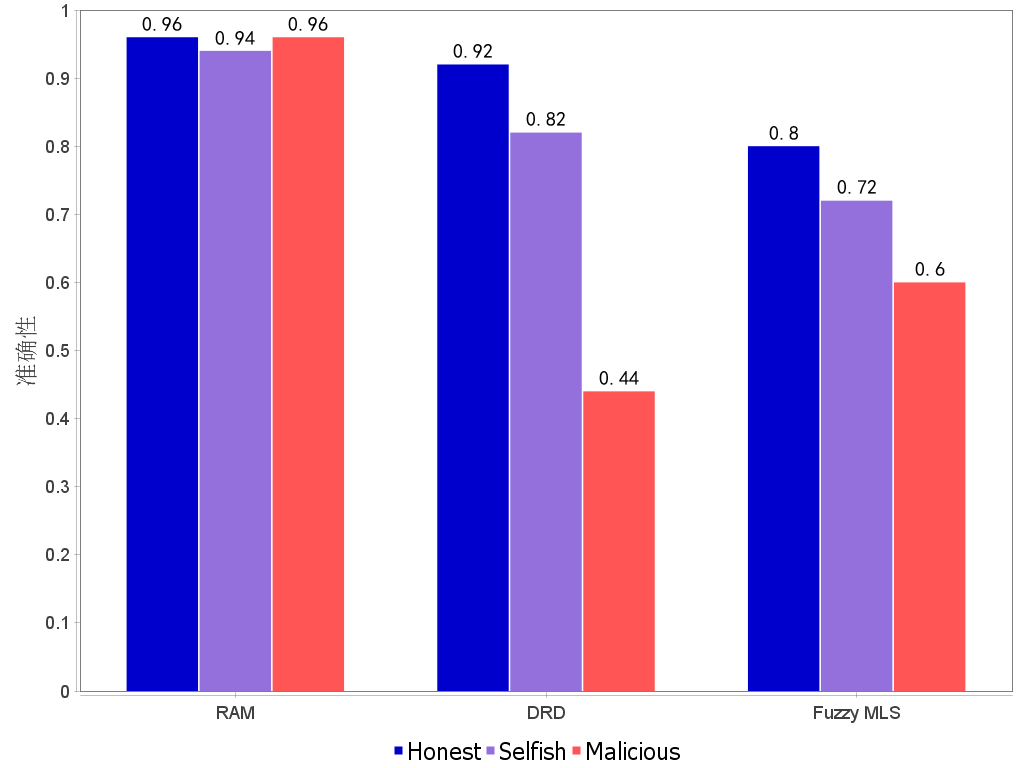


图3.14 RAM、DRD和Fuzzy MLS的准确性比较

* 1. 本章小结

本章首先介绍了基于风险的XACML访问控制模型，然后在此模型上介绍了风险评估方法，主要分为两部分，第一部分为主体的风险评估，根据主体信誉值计算主体威胁ST，第二部分为主体行为的风险评估，根据访问代价计算主体对客体的期望威胁ETO，之后将两部分威胁量化成风险分数，并提出风险评估计算模型和系统模型，最后通过实验验证访问次数和客体敏感度对风险分数的影响，并对风险分数作出准确性评估。

# 雾计算下基于风险的隐私策略自适应方法

隐私策略可以告知用户他们的信息将如何被收集、使用和披露，未能提供正确的隐私策略可能会导致雾节点中数据的安全威胁。雾计算是一种协同的计算环境，隐私数据可能会泄露给其它雾节点，每个雾节点有其自己的风险分数，根据风险分数为雾节点自适应生成隐私策略。因此，本章提出了雾计算下基于风险的隐私策略自适应方法，叫做R2PA。R2PA通过文档分析和代码静态分析为三类不同风险分数的用户自适应生成隐私策略。对比实验结果表明，新方法生成的隐私策略比原始隐私策略更正确，可读性更高。

* 1. R2PA的概述

R2PA根据雾节点的风险分数自适应构建正确和可读的描述，以方便生成雾节点中应用程序的隐私策略。R2PA由三个模块组成，如图4.1所示。

1. 文档分析模块

给定一个API及其对应API文档的描述，该模块利用信息提取技术[48]自动识别API使用的个人信息。该模块的输出被代码静态分析模块使用，来确定雾节点中部署的应用程序所收集的个人信息。

1. 代码静态分析模块

给定一个没有源代码的应用程序，并将选定的API映射到文档分析模块中的个人信息。该模块反汇编应用程序的APK文件，将其语句转换为中间代码（IR），并通过执行以下四个步骤检查IR来分析应用程序：①查找应用程序收集的个人信息；②查找第①项所述个人信息的收集者；③在调用选定的API之前，确定应用程序是否明确要求用户同意；④确定应用程序的信息保留策略。

1. 隐私策略自适应模块

通过代码静态分析识别雾节点中的应用程序收集的信息，该模块旨在自适应生成可阅读性的隐私策略。更准确地说，本文根据简单英语的参考定义了一个句子模板[49]，然后根据个人信息选择合适的动词来生成句子，并调用RAM自适应生成基于雾节点风险分数的隐私策略。



图4.1 R2PA的概述

* 1. 文档分析
     1. 基于API的签名和描述进行预处理

API描述提供了关于API行为的更多信息，例如API访问的信息。为了提取个人信息中的名词短语，本文根据描述找到该动词的主动词和宾语。但是，API描述可能无法提取准确的宾语。因此，本文利用了方法名和类名作为参考信息来确定需要提取的后置定语。应用程序的方法名通常遵循谷歌Java样式，用lowerCamelCase命名规则编写，这意味着除了第一个单词外，所有单词的首字母都用大写。类名可以是名词或名词短语，用UpperCamelCase命名规则编写，意味着所有单词都以大写字母开头。

敏感API是可以得到个人隐私信息的，Susi[50]提供了这样的函数列表，本文选择了79个API来获得以下信息：设备ID、IP地址、位置（包括纬度和经度）、国家名字、邮政编码、账户、电话号码、音频/视频、已安装的应用程序列表、访问的URL、浏览器的书签和cookies。通过将API的签名与其官方描述结合起来，并对它们进行预处理。API的签名包括API的类名、方法名、返回值的类型和参数类型。应用程序也可以通过内容提供商（例如content：//contacts）获取个人信息。PScout[51]列出了78个URI字符串，本文选择8个请求个人信息的URI字符串，包括联系人（3个URI字符串）、日程表、浏览器历史、短信（2个URI字符串）和通话记录。其它URI字符串将被忽略，因为它们的信息不会被R2PA考虑。PScout还列出了31种权限及其URI字段，从它们中选择了6种权限，这些权限包含160个URI字段。其它权限及其相关的URI字段将被忽略，因为它们与敏感的个人信息无关。

根据选择的敏感API进行预处理，预处理主要分为三个部分，首先从方法名中提取个人信息，然后从类名中提取个人信息，最后对API描述进行语法分析。

1. 从方法名中提取个人信息

首先删除方法名中的动词前缀，因为大部分信息检索函数的名称都是动词短语。方法名通常以动词前缀开头，如“get”或“read”，动词前缀后面紧跟着相应的个人信息，以小写字母为开头，后面所有单词都以大写字母开头。通过提取Susi函数列表中出现过一次以上的所有前缀，构造了一个包含178个动词的动词前缀列表。基于动词前缀列表，删除方法名中的动词前缀，并利用以大写字母开头的情况将剩余的名词短语分成不同的单词。根据NLTK[52]提供的停止词列表删除停止词，因为停止词是没有意义的，删除停止词可以使提取的名词短语更清楚。例如，方法名getAllVisitedUrls()，先删除动词前缀“get”，然后把剩下的字符串分成三个单词，包括“All”，“Visited”和“Urls”。最后“All”被删除，只保留“Visited Urls”。

1. 从类名中提取个人信息

类的完全限定名由包名和类名组成。例如，类名android.hardware.Camera，则包名为android.hardware，类名为Camera。目前，本文只提取类名，并将其转换为不同单词的列表。

1. API描述的语法分析

本文使用了Stanford Parser[53]，一个非常流行的语法分析工具[54][55]，可以构建句子的语法结构，识别敏感API描述的主要动词和其相关的宾语。Stanford Parser的输出为句子的短语结构树和依存句法信息。如图4.2和4.3所示，短语结构树具有层次结构，每一行代表输入句子的不同短语。词性标记也包含在短语结构树中，词性标记是指具有相似语法属性的一类词，这些单词在句子的语法结构中通常扮演相似的角色。常见的英语词性标记包括NN（名词）、VB（动词）、PP（介词）、NP（名词短语）和VP（动词短语）等。依存句法显示句子中单词之间的语法关系，虚拟节点“Root”有一个根关系，它指向句子的根单词，其它常见的语法关系还包括dobj（直接宾语）、nsubj（名词主语）和det（决定词）等。

****

图4.2 API描述的短语结构树

****

图4.3 API描述的依存句法信息

* + 1. 基于预处理确定个人信息

预处理步骤得到了API描述的语法信息，还有从方法名和类名中提取的个人信息，最后算法4.1基于预处理确定了API获得的个人信息。本文使用图4.3的API来解释算法4.1的执行过程。

在基于API签名和描述进行预处理的步骤中，首先通过Stanford Parser提供句子的短语结构树和依存句法信息（第2-3行），并使用依存句法信息识别根单词root（第4行）。例如，在图4.3中，动词“Return”被提取，根单词通常是一个句子的主要动词。然后通过ExtractObj函数提取根单词的直接宾语，帮助定位API描述中的个人信息（第5行）。在图4.3中，动词“Return”的直接宾语是名词“array”，此根单词的直接宾语只有一个单词。如果宾语被其它形容词或名词修饰，可以将它们放在直接宾语之前，以得到一个完整的短语。例如，对于名词短语“device ID”，如果ID是直接宾语，则提取名词“device”放在“ID”前面。最后算法4.1将方法名和类名中提取的个人信息作为参考，检查直接宾语是否包含足够的信息。如果方法名中存在个人信息，提取它并作为参考信息（第7-8行）。如果方法名是一个动词（不包含任何个人信息），则从类名中提取个人信息（第9-10行）。

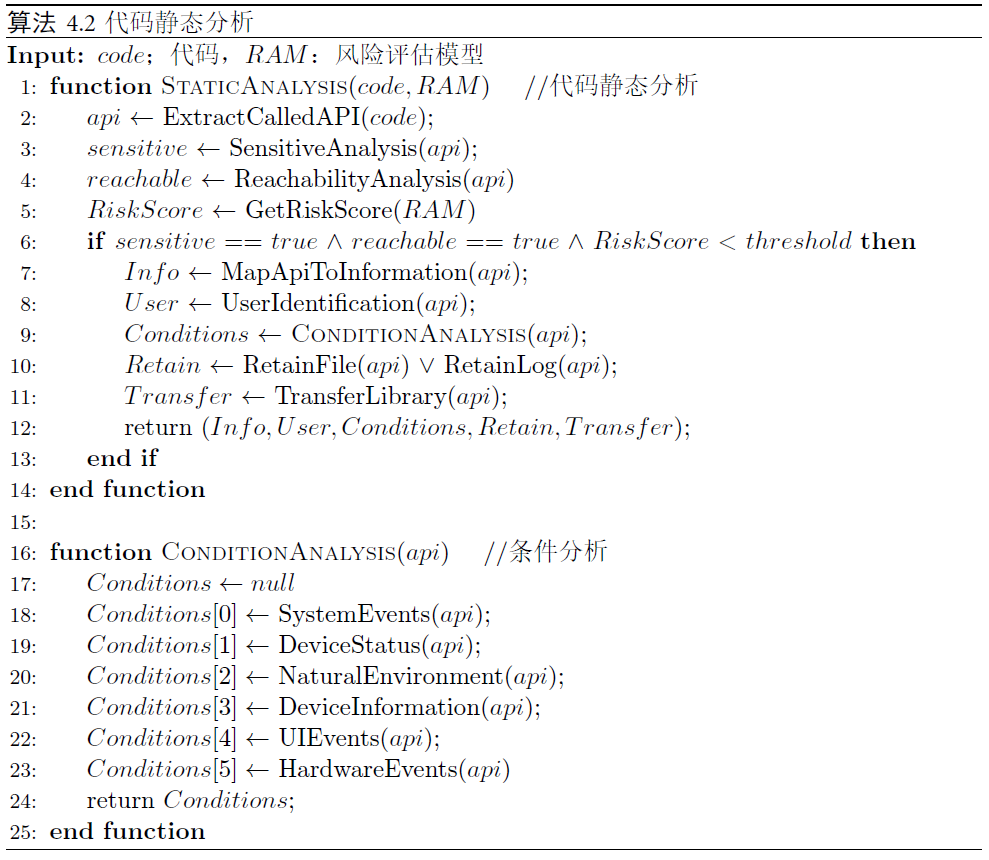
在基于预处理确定个人信息的步骤中，如果方法名和类名都不包含个人信息，该算法只返回直接宾语作为最终的个人信息（第26行）。在从方法名和类名中得到参考信息后，首先决定是否需要在直接宾语后添加额外的后置定语，可以通过计算直接宾语和参考信息之间的语义相似性来完成（第17行）。如果ESA[56]计算的语义相似值大于阈值，或者直接宾语包含参考信息，则直接返回直接宾语作为个人信息（第18-19行）。否则，直接宾语需要增加一些后置定语，使之更完整。在图4.3中，直接宾语为array，参考信息为Action Listeners，两者的语义相似度低于阈值。同时，直接宾语不包含参考信息，那么需要在list后面添加后置定语。函数FindInfo可以确定直接宾语的后置定语，FindInfo会检查短语结构树和依存句法信息（第21行）。如果直接宾语的子树包含“of...”/“from...”/“for...”/“that...”，则提取这些后置定语，并在直接宾语后面添加后置定语，以获取最终的个人信息（第22-23行）。在图4.3中，“list”后面添加了后置定语“of all the action listeners”。最后，“array of all the action listeners”作为API getActionListeners()访问的个人信息。



* 1. 代码静态分析
     1. 静态分析的概述

给定一个雾节点中部署的应用程序，静态分析模块首先识别被调用的敏感API，进行可达性分析，并调用RAM计算雾节点的风险分数，然后为每个API确定以下信息：①应用程序访问的个人信息；②应用程序中使用的第三方库及其访问的私人信息；③访问个人信息的条件；④是否保留个人信息；⑤是否传输个人信息。静态分析基于Vulhunter[57]，可以为每个应用程序构建应用程序属性图（APG），并将其存储在Neo4j图形数据库中以便进一步处理。APG是AST（抽象语法树）、MCG（方法调用图）、ICFG（过程间控制流图）和SDG（系统依赖图）的结合。

算法4.2描述了静态分析模块的基本过程。首先，识别敏感API/URI，并对这些敏感API进行可达性分析，以删除不可行的代码，还有调用RAM计算雾节点的风险分数，高风险的雾节点无法生成个人信息。然后，将选定的API/URI映射到个人信息，并确定私有信息是否被主机或第三方库访问。最后，对于可达的敏感API调用，提取了调用它们的条件，还进行信息保留分析和信息传输分析，以确定个人信息是否被保存在文件中或者是否被发送到其它主机。



算法4.2第2行ExtractCalled函数提取了代码中调用的API。第3行SensitiveAnalysis函数基于调用的API进一步提取敏感的API/URI。第4行ReachabilityAnalysis函数进行可达性分析，确定API/URI是否包含在不可行的代码中[58][59]。由于不可行的代码不会被执行，会导致生成的隐私策略无效。可达性分析是对MCG执行深度优先遍历，如果存在从入口点到调用敏感API/URI的方法的执行路径，则语句是可达的。第5行GetRiskScore函数会调用RAM计算雾节点的风险分数。第6行如果sensitive和reachable都为true，并且雾节点的风险分数小于阈值，则进行个人信息的提取。第7行MapApiToInformation函数通过文档分析模块将敏感的API/URI映射到个人信息。第8行UserIndentification函数判断第三方库是否收集个人信息，并记录API的类名。如果类名与第三方库的类名相同，则认为第三方库就是个人信息的使用者。第9-11行会在下面章节详细介绍，4.3.2介绍了如何提取敏感API的条件信息（第9行），第16-25行是具体的条件分析函数，4.3.3介绍了如何判断敏感API是否被保留或传输（第10-11行）。

* + 1. 条件分析

执行条件提供了调用敏感API的上下文信息，R2PA主要识别6种条件，包括系统事件、设备状态、自然环境需求、设备的具体信息、UI事件和硬件事件。

1. 系统事件

广播接收器可以注册意图来感知系统的状态变化。例如，在启动后，系统将广播android.intent.action.BOOT\_COMPLETED行为的意图。如果设备电量不足，android.intent.action.BATTERY\_LOW行为的意图将被广播。R2PA包含了54种不同的意图，应用程序接收到某些意图后调用敏感API，同时R2PA会记录这些意图的行为。

1. 设备状态

应用程序提供API来检查当前设备的状态。例如，开发人员可以调用PowerManager.isScreenOn()来确定设备是否处于交互状态。为了收集所有与设备状态相关的API，首先解析官方API文档，获得方法名前缀是“is”或返回值类型是布尔型的API。在此步骤中找到640个API，但是并非所有API都与设备状态相关，比如URL Util.isValidUrl（java.lang.String），最终检查并保留98个API。R2PA检查从入口点到敏感API路径上的所有分支语句，并根据数据依赖关系进行程序切片，记录语句中与设备状态相关的所有API。

1. 自然环境需求

应用程序可以检查当前的时间或地理位置。类似于设备状态，R2PA也会检查与时间（例如，Date.getHour()）和地理位置（例如，Location.getLatitude()）相关的API。如果API出现在入口点到敏感API的执行路径上，则记录该API。

1. 设备的具体信息

开发人员可以检查语言、屏幕大小和操作系统版本。例如，Locate.getDisplayLanguage()向用户返回适合显示的语言，Display.getSize()获取以像素为单位的显示大小，Display.getHeight()和Display.getWidth()返回显示的高度和宽度。

1. UI事件

对于应用程序来说，收集个人信息需要通知终端用户，比如弹出一个对话框。为了识别此类行为并将其写入隐私策略中，R2PA提取所有导致调用敏感API的UI回调。开发人员可以在布局文件中静态地定义回调方法（例如，通过android:onClick=“”属性）或在代码中动态注册事件监听器（例如，通过表4.1中setOnClickListener()）。表4.1介绍了回调函数和监听器注册函数的视图类。

表4.1 回调函数和监听器注册函数的视图类

|  |  |
| --- | --- |
| 回调函数 | 监听器注册函数 |
| onClick() | setOnClickListener() |
| onLongClick() | setOnLongClickListener() |
| onFocusChange() | setOnFocusChangeListener() |
| onKey() | setOnKeyListener() |
| onTouch() | setOnTouchListener() |
| onCreateContextMenu() | setOnCreateContextMenuListener() |

1. 硬件事件

用户可以按下BACK和HOME键来影响执行。例如，当用户按下HOME键时，调用当前活动的onPause()。当用户按下BACK键时，最后的活动onResume()将被执行。如果敏感API的入口点是onPause()或onResume()，系统将会记录它们。

* + 1. 信息保留分析和信息传输分析

1. 信息保留分析

一些个人信息不仅会被使用，还会通过互联网/短信发送或保存到文件/日志中，这些可以发送数据的指定API被称为接收器函数。此类行为应在隐私策略中加以识别和声明，为了捕获这样的行为，从敏感的API/URI到选定的接收器函数中基于数据依赖关系执行深度优先遍历。例如，在应用程序com.abukai.expenses.Uti（如图4.4所示）中，类com.abukai.expenses.Uti调用了getLongitude()和getLatitude()，并通过FileOutputStream的write函数将结果写入文件。然而，原始的隐私策略并没有提到记录用户的位置信息。



图4.4 com.abukai.expenses的信息保留

1. 信息传输分析

主机应用程序获取的个人信息可能会传输到第三方库[60]。例如，Ad库Admob提供AdRequest.Builder().setLocation()来传递位置数据给它。tool库可以获取个人信息（如地址）以方便识别问题。本文进行静态污染分析，通过检查从敏感API/URI到该数据用户的数据流，确定敏感信息是否会从主机应用传输到第三方库。

* 1. 隐私策略自适应

本文根据简单英语的参考生成句子，可以让读者更容易理解信息。简单英语的句子平均长度为15-20个单词，偏向于使用主动动词。因此，每个句子的结构定义为：[前置条件] 主语 动词 宾语 [目的][后置条件]。

1. 主语

主语是个人信息的收集者，如果是第三方库，则使用库的名称作为主语；否则，使用“we”作为句子的主语。

1. 动词

动词是一个句子的重要组成部分。在发现一款部署的应用程序收集个人信息的行为后，需要选择合适的动词来生成句子。当应用程序通过API获取信息时，对于大多数API，通常使用“access”、“use”或“check”作为主要动词。但是，如果程序访问外部存储，则需要进行数据流分析，以确定其实际的行为（即读或写）。如果应用程序使用文件对象来创建一个新的目录（例如，File.mkdir()），并将数据写入文件（例如，FileOutputStream.write()）或删除文件（例如，File.delete()），则使用“modify”作为主要动词。否则，使用“read”作为主要动词。

1. 宾语

宾语是收集的个人信息，通过调用文档分析模块可以得到。

1. 前置条件

前置条件是一个可选元素。如果在检查了4.3.2节中描述的四种情况（即系统事件、设备状态、自然环境需求和设备的具体信息）后，获得了敏感信息，则需要在句子前加一个前置条件（即“if...”）。例如，系统事件android.action.TIMEZONE\_CHANGED的描述为“The timezone has changed”。如果在接收该事件时使用敏感API/URI，需要使用“If the timezone has changed”作为前置条件。由于设备状态API的官方描述通常是“Returns true if...”的方式，提取这些前缀之后的子句，并将它们作为前置条件。例如，设备状态API PowerManager.isPowerSaveMode()的描述是“Returns true if the device is currently in power save mode”。如果应用程序在其分支语句中调用PowerManager.isPowerSaveMode()，则使用“If the device is currently in low power mode”作为前置条件。

1. 目的

目的是另一个可选元素，其描述了为什么应用程序访问个人信息[61]。目前，R2PA可以判断三种使用个人信息的目的，包括摄像头、位置和联系人。更准确地说，如果使用相机拍照（例如，在图4.5中调用Camera.takePicture()），则目的是“to take picture”。如果摄像机被用来录制视频（例如，调用MediaRecorder.setVideoSource()），目的变成了“to record video”。对于位置信息，如果被转移到广告的第三方库（例如，通过Admob的AdRequest.setLocation）,目的是“for target advertising”。否则，如果位置是关于社交网络相关的库（例如，Facebook SDK），则目的变成了“for location-based social networks”。对于联系信息，如果被SMS相关API使用（例如，SmsManager.sendTextMessage()），目的是“to send SMS”。如果联系信息是通过与电子邮件相关的意图发送的（例如，行为是ACTION\_SEND，意图为EXTRA\_EMAIL），目的是“to send email”。

1. 后置条件

后置条件是最后一个可选元素。该条件用来描述4.3.2节中的两种情况（即UI事件和硬件事件）。例如，如图4.5所示，当按下一个按钮（例如，OnClick()）后，API（例如，Camera.open()）会被触发，需要在句子后面加上“when you press \* button”。



图4.5 应用程序的代码片段

在为调用敏感API/URI的语句生成句子之后，需要增加一句话来描述该信息是否会被保留/传输。如果信息保留在文件或日志中，则在生成的句子后添加“This information will be written to file/log”。同样，如果信息是通过互联网或短信发送出去的，则添加“This information will be sent out through internet/SMS”。如果信息由主机应用程序获取并传输到第三方库，则添加句子“This information will be transferred to the third party library”。同样，如果信息是由第三方库收集并传输到主机应用程序，需添加句子“This information will be transferred to the host app”。

通过引用隐私策略模板，将生成的句子分为两个部分，包括个人身份信息和披露给第三方库的信息。隐私策略是基于雾节点的风险分数自适应生成的，主要为三类主体（雾节点）生成隐私策略，包括诚实主体、自私主体和恶意主体。对于诚实主体，风险分数的区间为(0, 0.40)；对于自私主体，风险分数的区间为(0.40, 0.80)；对于恶意主体，风险分数的区间为(0.80, 1.00)。风险分数越高，应用程序收集的个人信息越多，信息越敏感。图4.6展示了R2PA生成的示例隐私策略。



图4.6 R2PA生成的隐私策略

* 1. R2PA的验证

R2PA用Java实现并部署在模拟的雾环境中。本文从Google play下载了7181个应用程序，并且都提供了隐私策略，从中随机选择500个包含隐私策略的应用程序作为数据集，由于比较R2PA生成的隐私策略和原始的隐私策略需要时间耗费人工验证，这对于许多应用程序是不可能的。因此，在本文的实验中，选取了20个典型的应用程序，它们的方法名和类名都符合谷歌Java样式，而且API描述无法提取完整的宾语，这样R2PA生成的隐私策略更具有代表性。20个应用程序如表4.2所示。

表4.2 应用程序列表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 应用程序 | 序号 | 应用程序 |
| 1 | air.com.goodgamestudios.empirefourkingdoms | 11 | com.jb.gokeyboard.plugin.emoji |
| 2 | com.gentaycom.nanu | 12 | com.rapidcyber.intelliim.android |
| 3 | com.king.petrescuesaga | 13 | com.rovio.angrybirdsgo |
| 4 | com.lego.juniors.quest | 14 | com.sharkparty.slots |
| 5 | com.microsoft.offifice.offificehub | 15 | gpc.myweb.hinet.net.PopupWeb |
| 6 | jp.naver.SJLGBUBBLE | 16 | jp.konami.pesm |
| 7 | com.mobileiron | 17 | jp.naver.lineplay.android |
| 8 | air.com.sgn.cookiejam.gp | 18 | jp.naver.linetools |
| 9 | air.com.thomsonreuters.cnvne | 19 | org.asiministries.asievangelism |
| 10 | com.facebook.orca | 20 | si.modula.android.instantheartrate |

* + 1. 文档分析和代码静态分析的正确性评估

本文从三个方面评估R2PA的正确性，包括API映射到个人信息、可达性分析和条件分析。

1. API映射到个人信息

为评估API映射到个人信息的准确性，从Susi中随机选择150个函数。文档分析模块将API的签名和描述作为输入，然后手动验证输出。最终发现150个API中有142个是正确的结果，文档分析模块的准确度计算如下：



其中True Positive表示文档分析模块能够正确提取相关信息的API数量，Total表示本实验使用的API总数。

150个API中有8个映射错误，造成这些误差的原因有两个。第一个原因是Stanford Parser生成的短语结构树。例如，API Cursor.getColumnNames()的描述是“Returns a string array holding the names of all of the columns in the result set in the order in which they were listed in the result”。R2PA识别动词“Returns”的宾语为“a string array”，但忽略了“names of all of the columns”，这是因为在Stanford Parser的输出中“names of all of the columns”不在宾语“a string array”的子树中，并且算法4.1只搜索了宾语子树中的后置定语。第二个原因是语义相似度的阈值。例如，API Address.getAddressLine(int)的描述是“Returns a line of the address numbered by the given index...”。R2PA识别动词“Returns”的宾语为“line”，但忽略了“of the address”。这是因为宾语“line”和从方法名中提取的信息“Address Line”的语义相似度大于阈值，R2PA没有提取额外的信息。

1. 可达性分析

本文随机选择16个请求权限ACCESS\_FINE\_LOCATION的应用程序，并调用两个与位置相关的API（即getLatitude()和getLongitude()），并手动验证R2PA的结果。在两个应用程序中，所有调用这两个API的语句都可以从入口点到达。13个应用程序有可行和不可行的代码。例如，应用程序com.dooing.dooing在两个方法<com.dooing.dooing.NewTask：void m()>和<com.dooing.dooing.cg：void a()>中调用getLatitude()。但是，后一种方法不会被其它方法调用。在该应用程序中（即com.smartsol.congresoandroid），所有调用这两个API的语句都是不可行的。总而言之，通过可达性分析可以成功去除89个因不可行的代码而产生的错误句子。

1. 条件分析

本文将R2PA和AppContext[62]进行比较，指示它们处理表4.2所示的应用程序，实验只研究AppContext支持的四种条件。结果表明，AppContext找到了84个API及其对应的上下文信息，R2PA识别了这84个API中的83个。缺少的一个API是应用程序jp.konami.pesm中的<android.hardware.Camera：android.hardware.Camera open()>。通过手动检查该API，发现这个API在第三方库中使用。然而，应用程序不会执行代码，因为它没有请求清单文件中的CAMERA权限。

* + 1. 隐私策略的充分性评估

本文将使用表4.2中的应用程序对R2PA生成隐私策略的覆盖个人信息的范围与原始隐私策略进行比较。主要关注了10种个人信息，包括设备ID、位置、账户、摄像头、音频、已安装的应用程序列表、cookies、电话号码、通话记录和日程表，并会统计隐私策略中提到的第三方库的数量。

表4.3列出了手工验证的对比结果，其中N表示R2PA生成的隐私策略，O表示原始的隐私策略。结果表明R2PA生成的隐私策略包含45种个人信息，原始的隐私策略只包含了29种个人信息。与原始的隐私策略相比，R2PA能识别更多应用程序收集的个人信息。此外，R2PA生成的隐私策略列出了20个第三方库，而原始的隐私策略只列出了4个。例如，第20个应用程序的原始隐私策略只通知用户使用设备ID，而由R2PA生成的隐私策略将同时使用设备ID和相机。此外，第20个应用程序的原始隐私策略没有提到第三方库，而R2PA显示了该应用程序使用的5个第三方库。

根据表4.3发现原始隐私策略存在两类问题。首先，一些隐私策略是不完整的，因为它们没有包括应用程序访问的所有个人信息。例如，有四个应用（表4.3中的第1、6、10、17个应用）在代码中使用了应用列表，但没有一个应用在隐私策略中声明。其次，一些隐私策略包含了应用程序无法获取的个人信息。例如，第14款应用的隐私策略中包含了句子“we collect information regarding your device type, operating system and version, ..., the device’s location”。然而，应用程序不请求隐私策略中的位置相关权限，意味着应用不能访问位置信息。R2PA可以避免这些问题，因为它使用了代码级信息来生成隐私策略。

表4.3 R2PA生成的隐私策略和原始隐私策略的覆盖范围

|  |  |  |  |
| --- | --- | --- | --- |
| 个人信息 | | 应用序号（收集个人信息的应用） | 应用总和 |
| 设备ID | N  O | 2、3、6、7、8、10、14、15、17、20 | 10 |
| 4、9、10、13、14、15、17、19、20 | 9 |
| 位置 | N  O | 1、4、7、8、9、10、13、16 | 8 |
| 2、4、10、13、14、17、18、19 | 8 |
| 账户 | N  O | 5 | 1 |
| 5 | 1 |
| 摄像头 | N  O | 4、9、12、13、17、18、20 | 7 |
| 无 | 0 |
| 音频 | N  O | 19 | 1 |
| 无 | 0 |
| 应用列表 | N  O | 1、6、10、17 | 4 |
| 13 | 1 |
| Cookies | N  O | 3、6、8、9、13、16、17、18、19 | 9 |
| 3、7、8、9、13、14、16、17、20 | 9 |
| 电话号码 | N  O | 2、7 | 2 |
| 10 | 1 |
| 通话记录 | N  O | 2、7 | 2 |
| 无 | 0 |
| 日程表 | N  O | 12 | 1 |
| 无 | 0 |

* 1. 本章小结

本章首先在基于风险的XACML访问控制模型中提出了隐私策略自适应方法，然后分别详细介绍了方法中3个模块的执行过程，第一个模块为文档分析模块，通过算法4.1确定API收集的个人信息；第二个模块为代码静态分析模块，通过算法4.2进行条件分析、信息保留分析和信息传输分析；第三个模块为隐私策略自适应模块，通过简单英语的参考生成句子，并自适应生成基于风险的隐私策略。最后通过实验评估了文档分析模块和静态分析模块的正确性，还评估了隐私策略的充分性。

# 原型系统及应用示范的设计与实现

本章节设计并实现了基于风险的XACML访问控制原型系统，其中包括风险评估方法、隐私策略自适应方法和XACML节点管理等功能。首先简要介绍了该原型系统所需的实验环境和组成部分，然后展示了一个基于车联网数据的应用示范，主要流程包括背景介绍、需求分析、详细设计和具体实现，实现了基于风险的XACML访问控制系统和实际应用场景的结合，最后通过实验验证了原型系统的有效性和可行性。

* 1. 基于风险的XACML访问控制原型系统

基于风险的XACML访问控制系统开发环境为开源的Linux OS，开发模式采用B/S设计模式，选用了IntelliJ IDEA和Visual Studio Code等开发软件，使用了VUE、Mybatis、Spring、SpringBoot和SpringCloud等技术。



图5.1 基于风险的XACML访问控制系统结构

基于风险的XACML访问控制系统主要包括三个组成部分，第一个部分为风险评估方法，根据主体信誉值、客体的安全属性和上下文来量化风险分数；第二个部分为隐私策自适应方法，通过文档分析和代码静态代码分析自适应生成基于风险的隐私策略；第三个部分为XACML节点管理，可以删除或增加节点，并制定相应的访问控制策略。本文的研究重点为第一个部分和第二个部分，理论部分分别对应了第三章和第四章的内容。基于风险的XACML访问控制系统的结构如图5.1所示。

* 1. 基于风险的XACML访问控制系统应用示范
     1. 背景介绍

车联网（IoV）是以车为节点和信息源，利用互联网技术、全球定位技术、无线通讯技术、智能终端设备技术和信息网络平台服务等手段，使车与基础设施（V2I）、车与车（V2V）、车与人（V2P）之间实现实时互联，并将司机、车辆和道路等信息有机融合，实现“人—车—路”的和谐统一[63]。但是由于车联网数据量庞大，数据会实时更新变化，信息种类十分复杂，有些信息可能会涉及国家安全，例如公安网传输的数据等；有些数据可能涉及个人隐私信息，例如卡口系统检测的车辆轨迹信息等[64]。车联网数据平台需要不断地接入新的服务端、客户端、网络设备、存储设备和其它IT资源，当访问者数量多、处理数据量大时，访问者权限的管理将会变得十分繁琐，从而导致访问者权限难以正常维护，降低了车联网数据平台的安全性和可靠性[65]。本章基于车辆网的数据对原型系统进行评估，主要评估了风险评估方法和隐私策略自适应方法的正确性和可行性。

表5.1 车联网数据集元数据

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 数据类型 | 数据格式 |
| 1 | 出租等级 | string | 文本字符串格式 |
| 2 | 执照号 | string | 文本字符串格式 |
| 3 | 提供商编号 | string | 文本字符串格式 |
| 4 | 比率代码 | string | 文本字符串格式 |
| 5 | 上车时间 | string | 时间格式YYYY/MM/dd |
| 6 | 下车时间 | string | 时间格式YYYY/MM/dd |
| 7 | 乘客数量 | int | 普通整数格式 |
| 8 | 行驶时长 | int | 普通整数格式 |
| 9 | 行驶距离 | double | 普通浮点数格式 |
| 10 | 上车位置经度 | double | 普通浮点数格式 |
| 11 | 上车位置纬度 | double | 普通浮点数格式 |
| 12 | 下车位置经度 | double | 普通浮点数格式 |
| 13 | 下车位置纬度 | double | 普通浮点数格式 |

本章实验中，使用出租车行车位置记录来验证原型系统对车联网数据的隐私保护效果，这里的数据来源于真实的出租车详细行车位置数据[66]，包括出租等级、执照号、提供商编号、比率代码、上车时间、下车时间、乘客数量、行驶时长、行驶距离、上车位置经度、下车位置纬度、下车位置经度和下车位置纬度等属性信息，如表5.1所示。

* + 1. 需求分析

需求分析是实现基于风险的XACML访问控制系统的第一步，系统的实现都是围绕需求分析展开的，只有良好的需求分析才能实现原型系统的构建，该应用示范围绕车联网的现实背景需求展开，车联网节点需要访问用户的敏感信息。为了保护用户的个人信息，原型系统实现了风险评估方法和隐私策略自适应方法。由于系统基于XACML实现的，所以还实现了XACML节点管理功能。

从系统实现角度，系统主要通过风险评估方法和隐私策略自适应方法来实现用户的个人隐私信息保护需求。本文基于风险的XACML访问控制系统的具体操作主要由以下几个部分组成。

1. 提供用户注册和登录功能，保证了登录用户的个人信息安全。
2. 提供风险评估功能，通过主体信誉值、客体的安全属性和上下文量化风险，还可以计算主体威胁和主体对客体的期望威胁，通过列表展示了主体一段时间的风险分数变化。
3. 提供隐私策略自适应功能，通过文档分析和代码静态分析提取车联网收集的个人信息，并根据这些信息生成基于风险的隐私策略。
4. 提供XACML节点管理功能，包括删除节点、增加节点和访问控制策略管理三部分。
   * 1. 详细设计

该原型系统实现了基于风险的XACML访问控制，在访问控制的基础上，提出了风险评估方法和隐私策略自适应方法。系统采用了表示层、业务层和持久层三层结构进行设计和实现，如图5.2所示。



图5.2 系统设计

1. 客户端

普通用户是原型系统中的访问请求者，用户在浏览器上登录系统完成系统的认证安全环节之后发起访问请求，服务器进行风险评估，根据用户的风险分数判定是否允许用户访问资源。此外，服务器还会为用户自适应生成基于风险的隐私策略，帮助用户更加了解系统所收集的个人信息。管理员是原型系统中的管理者，可以进行策略的修改和制定，通过策略管理界面检索策略id和策略描述信息。此外，管理员还可以调整主体的安全级别、客体的敏感度和上下文值。

1. 表示层

表示层负责与用户交互，并展示业务层中的数据。本文在服务器部署VUE页面，通过VUE调用前端控制器，由前端控制器根据请求内容调用业务层的应用，同时负责业务层之间的联系。

1. 业务层

业务层包括决策层和应用层。决策层可以对访问请求进行风险判定，并将判定结果反馈给义务服务机制，判定结果信息再通过表示层显示给用户。应用层包含三种应用：风险评估、隐私策略自适应和XACML节点管理。

1. 持久层

持久层采用DAO（Data Access Object）和Mysql数据库进行实现，其存储了访问控制系统相关的数据信息，主要包括用户信息、资源信息、控制信息、策略信息和日志信息，相当于包含了PAP和PIP的功能。通过DAO接口，应用程序将底层的数据访问代码和上层的业务逻辑代码完全分离，使得各层保持相对独立，便于扩展和移植。

* + 1. 具体实现

根据前面章节的需求分析和详细设计等软件开发流程后，下一步是通过具体的操作页面来展示基于风险的XACML访问控制系统的实现，对本文所提出的相关理论研究的可行性和有效性进行展示，基于风险的XACML访问控制系统的具体实现如下：

1. 用户注册-登录模块

用户注册-登录模块是系统的使用入口，界面如图5.3所示，初次使用该原型系统的用户首先进行个人身份的注册，系统后台会自动维护用户信息，当用户再次使用该系统时，则输入用户名和密码进行身份验证。如果认证通过，用户可以使用该原型系统；如果认证不通过，系统则提示用户输入正确的用户名和密码。

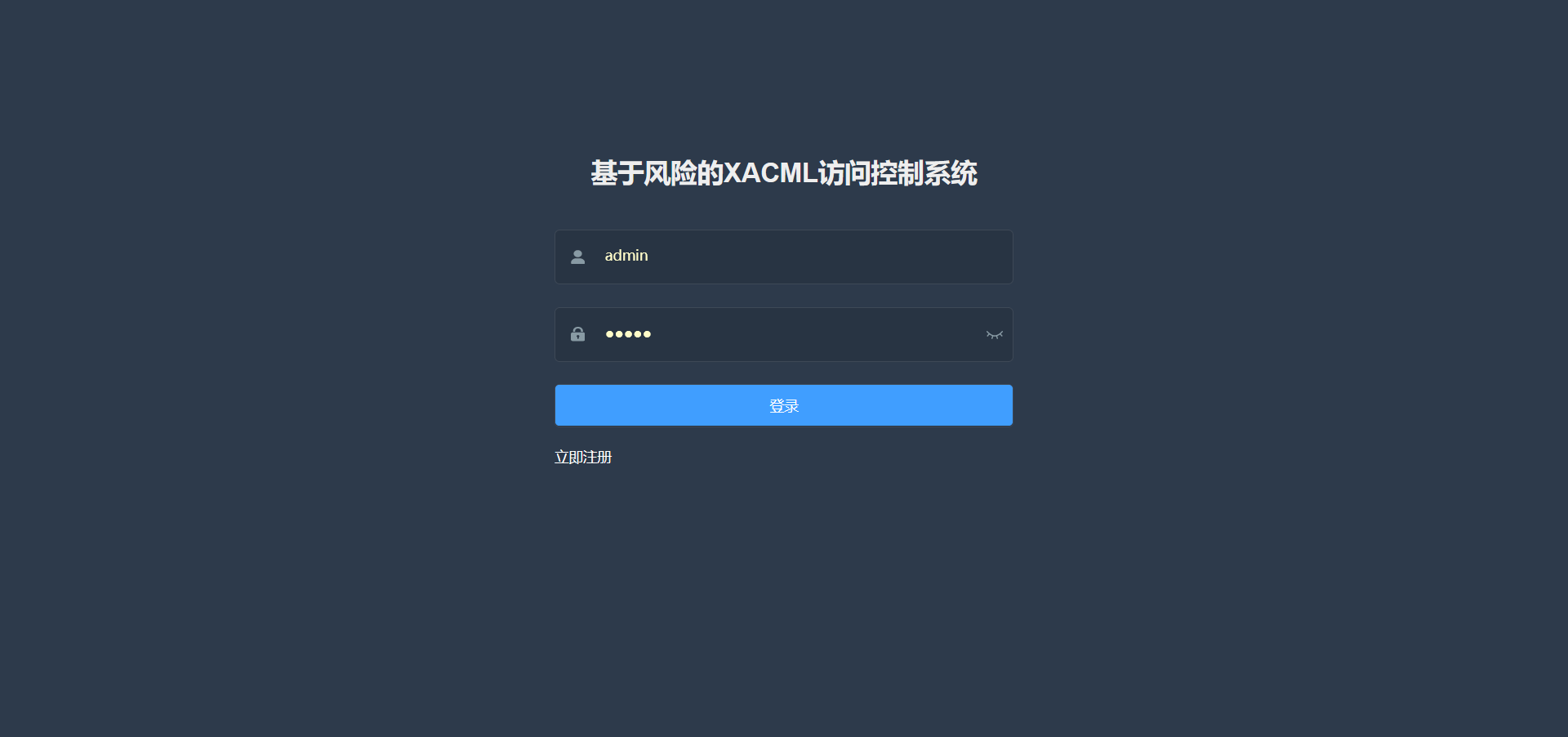


图5.3 登录和注册界面

1. 主体威胁模块

主体威胁ST是对主体的风险评估，首先根据用户的恶意访问次数和友好访问次数计算主体信誉值，然后通过客体的敏感度向量和主体信誉值计算主体安全不确定性，最后根据主体安全不确定性和主体安全级别计算出ST。所以ST的输入为用户恶意访问次数、用户友好访问次数、敏感度向量和主体安全级别，输出为ST。具体界面如图5.4所示。

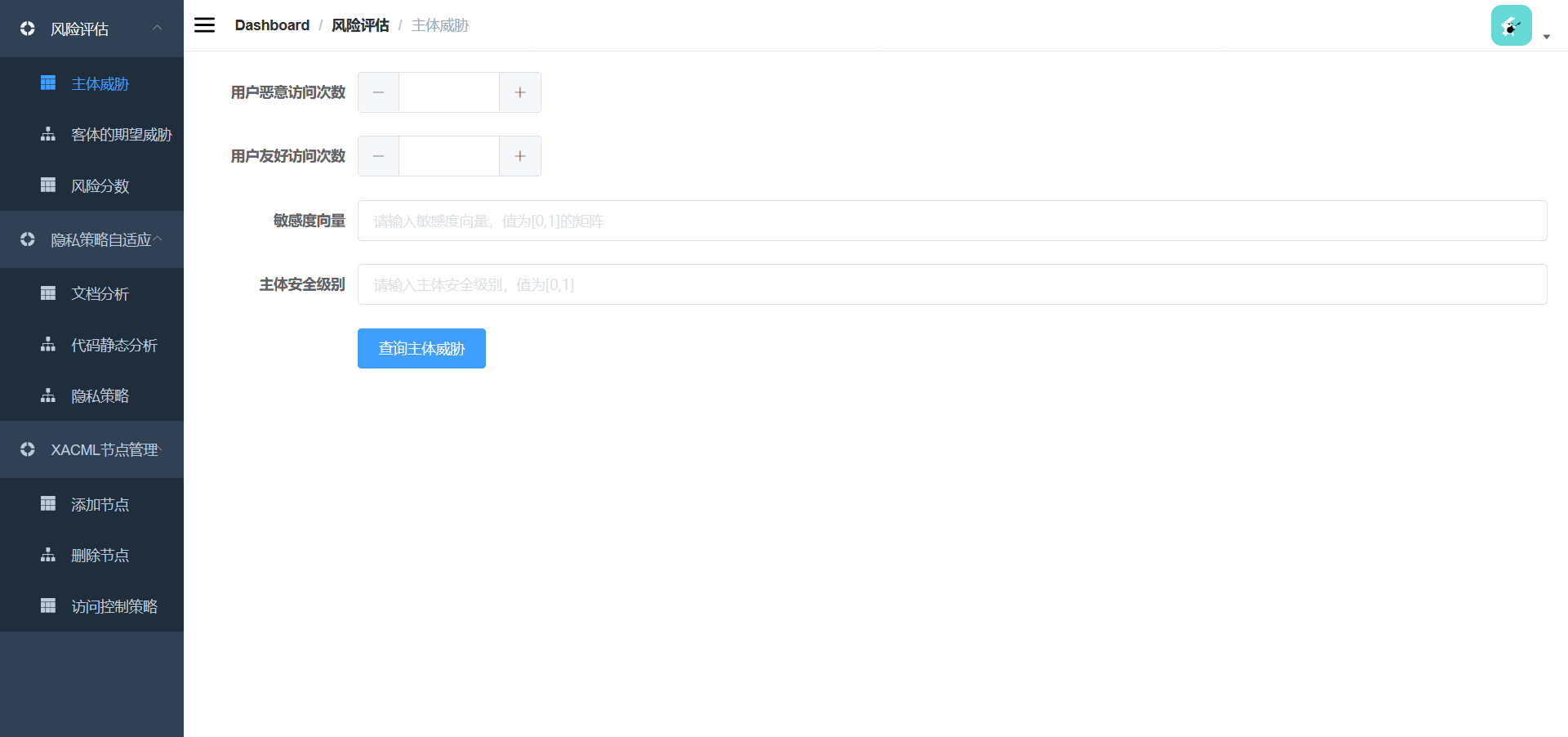


图5.4 主体威胁界面

1. 客体的期望威胁模块

主体对客体的期望威胁ETO是对主体行为的风险评估，首先基于访问矩阵和客体敏感度向量计算上下文无关的访问代价，然后根据上下文无关的访问代价和上下文值计算出上下文相关的访问代价，最后通过上下文相关的访问代价和主体损害客体的概率计算ETO。所以ETO的输入为访问矩阵、敏感度向量、上下文值和主体损害客体的概率，输出为ETO。具体界面如图5.5所示。

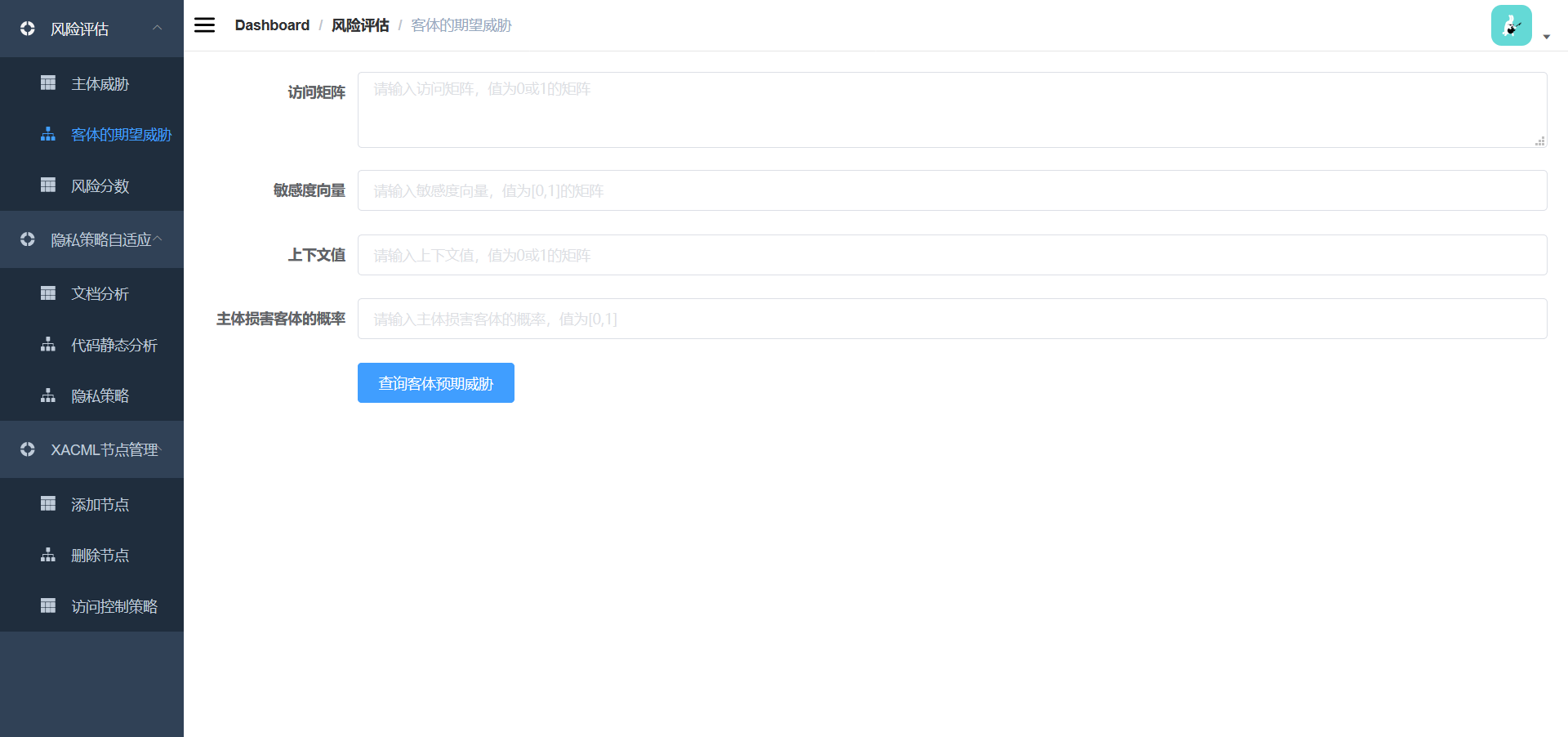


图5.5 客体的期望威胁界面

1. 风险分数模块

风险分数RS是风险评估最后量化的结果，通过ST和ETO两部分威胁计算出RS。风险分数界面如图5.6所示，通过列表展示了一个用户一段访问时间内风险分数的变化。

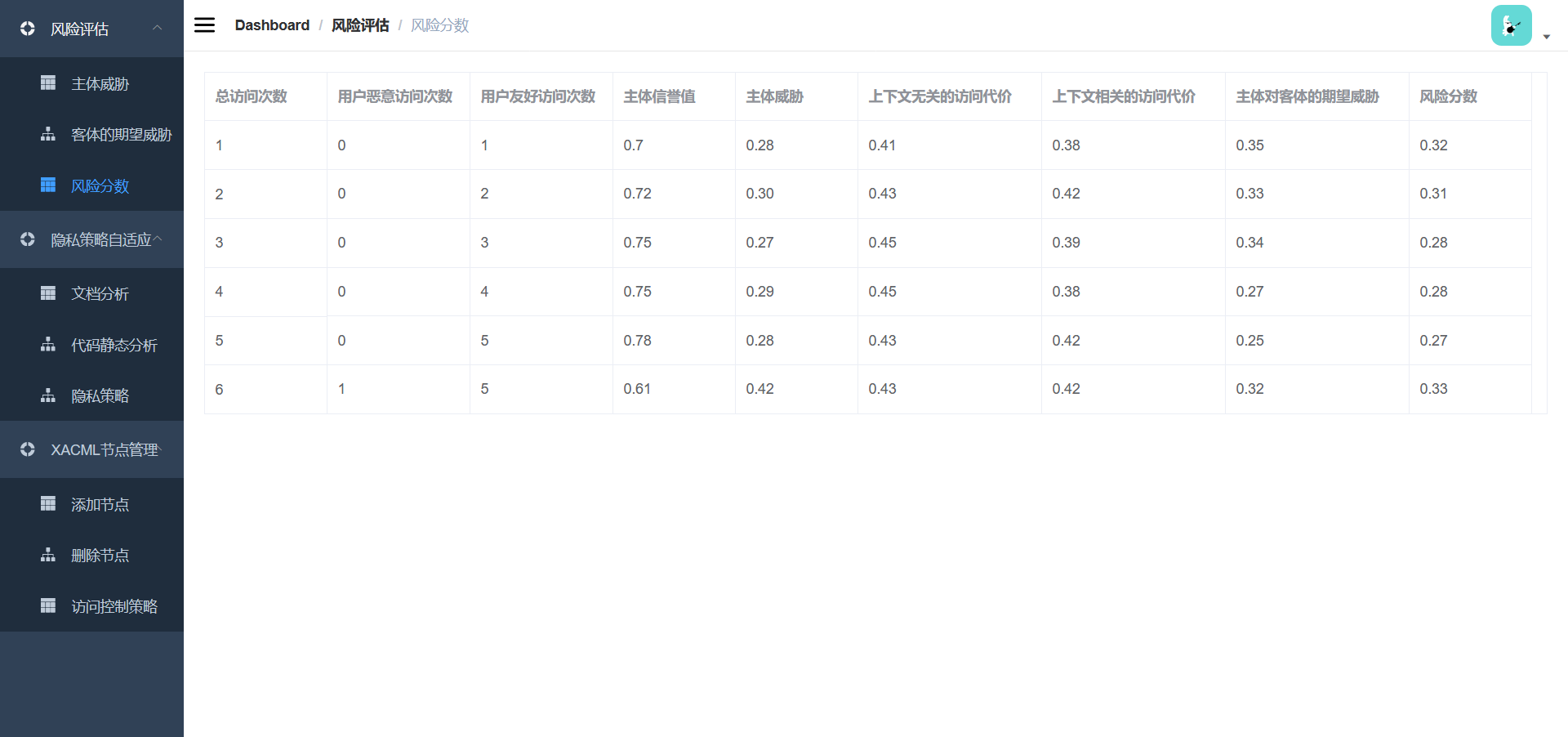


图5.6 风险分数界面

1. 文档分析模块

文档分析是R2PA中对API个人信息的提取，首先基于API的方法名和类名提取个人信息，然后通过Stanford Parser得到API描述的短语结构树和依存句法分析，最后根据算法4.1确定API收集的个人信息。文档分析的输入为API方法名、API类名和API描述，输出为收集的个人信息。具体界面如图5.7所示。

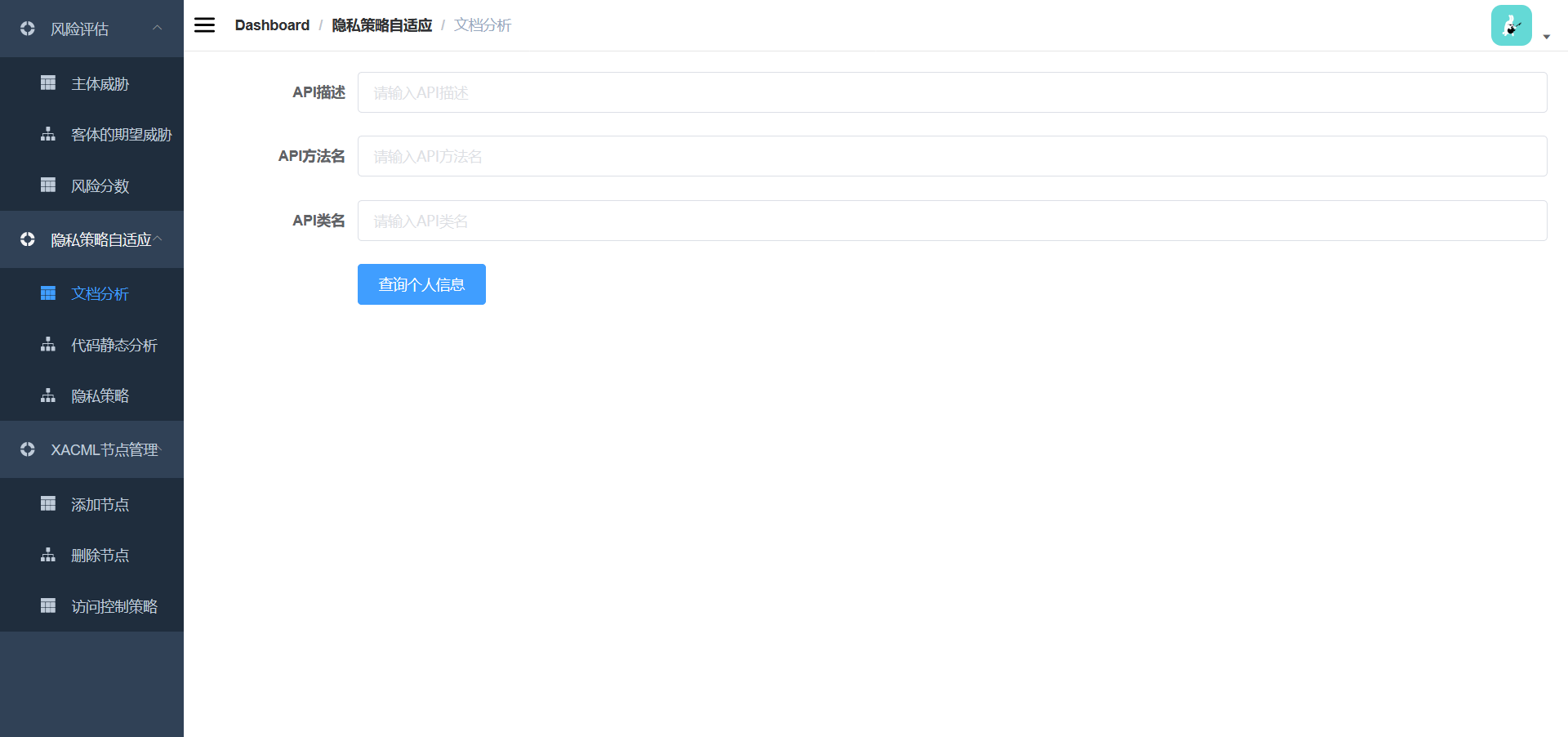


图5.7 文档分析界面

1. 代码静态分析模块

代码静态分析是R2PA中对调用敏感API上下文信息的提取，首先通过可达性分析识别出调用的敏感API，然后根据六种条件提取相关上下文信息，并进行信息保留分析和信息传输分析，最后根据算法4.2为用户生成可阅读性的隐私策略。代码静态分析的输入为应用程序的APK文件。具体界面如图5.8所示。

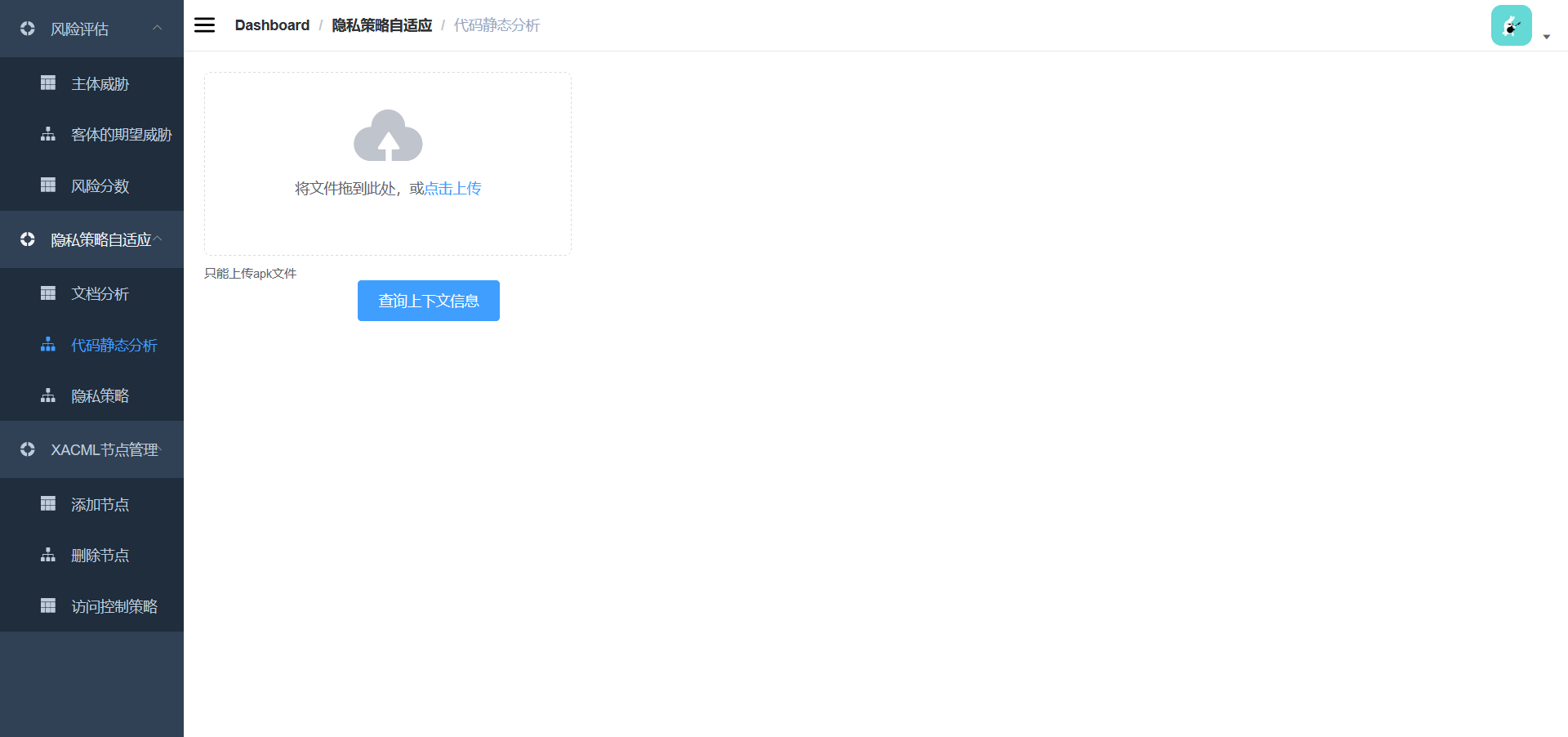


图5.8 代码静态分析界面

1. 隐私策略模块

隐私策略是R2PA中自适应生成隐私策略的部分，通过文档分析和代码静态分析提取到个人信息及其上下文信息，最后根据简单英语为用户自适应生成可阅读性的隐私策略。具体界面如图5.9所示。

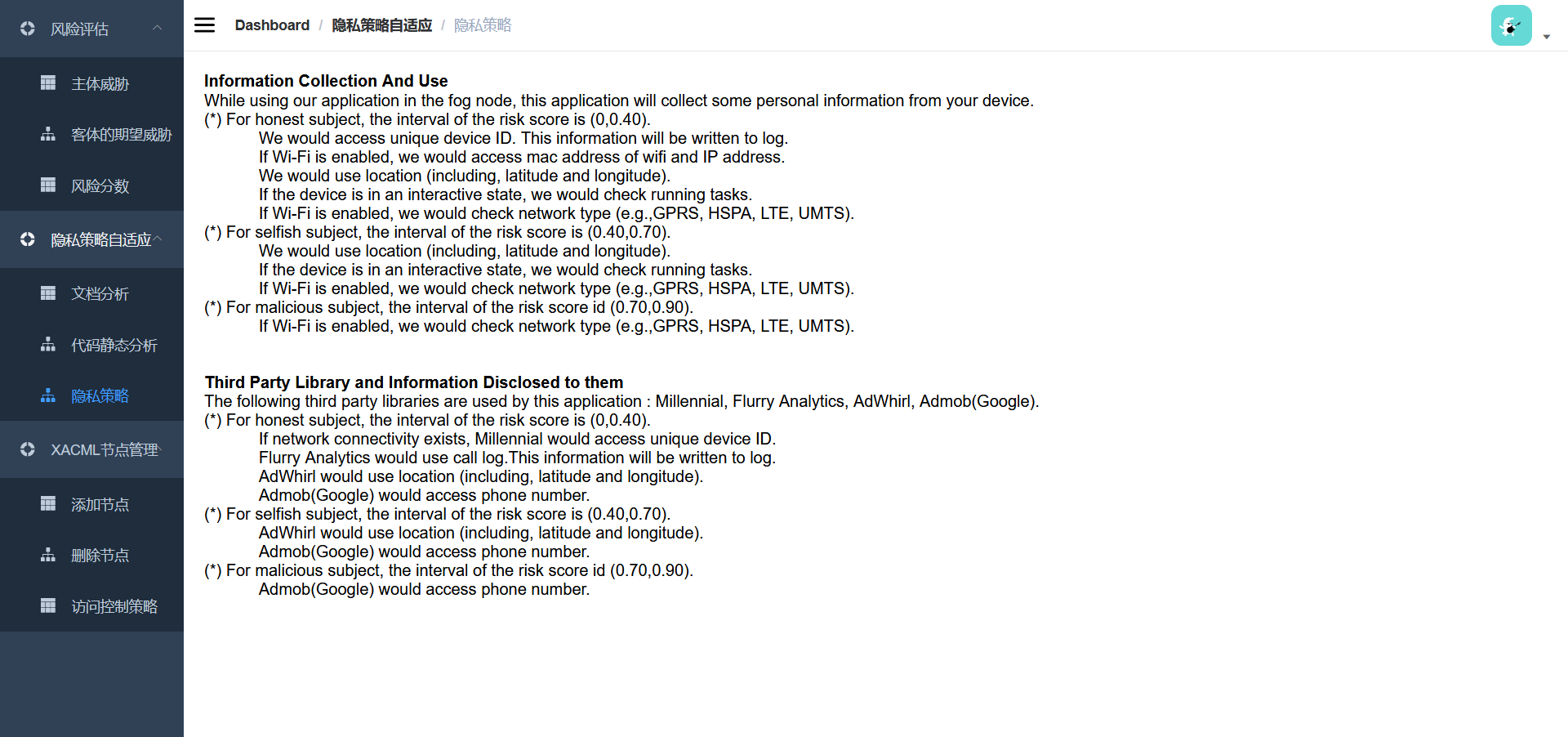


图5.9 隐私策略界面

* 1. 基于风险的XACML访问控制系统评估
     1. 实验设计

为了验证本原型系统的可行性，本文设计了车联网的访问控制场景进行实验。在此场景中将客体的敏感度向量设为[0, 1]，敏感度向量越高，表明客体的隐私程度越高。主体的安全级别也设为[0, 1]，主体的安全级别越高，表明主体的可信任程度越高。主体的行为包括读、写和执行，三类行为分别会对客体的安全属性（A、I和C）产生影响，如果有影响，值则为1，反之为0。实验准备如图5.10所示。

表5.2 实验准备

|  |  |  |
| --- | --- | --- |
|  | 个数 | 备注 |
| 主体 | 100 | 主体编号0-99，主体的安全级别随机分配，范围为[0,1]。 |
| 客体 | 2000 | 客体编号0-1999，客体的敏感度向量随机分配，范围为[0,1]。 |
| 行为 | 3 | 行为包括读、写和执行，对A、I和C产生影响。 |
| 规则 | 2000 | 规则编号0-1999分别对应客体0-1999，每个规则的相关请求为400个。 |

本文的风险评估是通过主体的信誉值、客体的安全属性和上下文计算的，风险分数由主体威胁ST和主体对客体的期望威胁ETO两部分组成。ST首先根据用户恶意访问次数和用户友好访问次数计算主体信誉值，然后根据主体信誉值和客体敏感度向量计算主体安全不确定性，最后通过主体的安全级别和主体安全不确定性计算出ST。ETO首先根据访问矩阵和客体敏感度向量计算上下文无关的代价，然后根据上下文无关的代价和上下文值计算上下文相关的访问代价，最后通过上下文相关的访问代价和主体损害客体的概率计算出ETO。

* + 1. 实验结果及分析

（1）访问请求允许率与主体安全级别的关系

本文以编号为555的规则为例，依次将对目标客体编号为555的客体发起的400个不同的访问请求传给PDP，400个访问请求中1-80的请求主体安全级别范围为(0, 0.2]，81-160的请求主体安全级别范围为(0.2, 0.4]，161-240的请求主体安全级别为(0.4, 0.6]，241-320的请求主体安全级别范围为(0.6, 0.8]，321-400的请求主体安全级别为(0.8, 1.0]，判定结果如表5.3所示。实验结果表明主体安全级别越高，主体的可信任程度就越高，从而导致访问请求允许率越高，所以访问请求允许率与主体安全级别成正比关系。

表5.3 访问请求允许率与主体安全级别的关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 请求编号 | 主体安全级别 | 系统风险阈值 | 允许访问个数 | 允许率 |
| 1-80 | (0, 0.2] | 0.8 | 18 | 22.5% |
| 81-160 | (0.2, 0.4] | 0.8 | 32 | 40..0% |
| 161-240 | (0.4, 0.6] | 0.8 | 50 | 62.5% |
| 241-320 | (0.6, 0.8] | 0.8 | 68 | 85.0% |
| 321-400 | (0.8, 1.0] | 0.8 | 78 | 97.5% |

（2）访问请求允许率与客体敏感度向量的关系

本文随机选取编号为555、666、777、888和999的规则为例，依次将对目标客体编号为555、666、777、888和999的客体发起访问请求，每个客体都有400个请求去访问，且请求的主体安全级别都是相同的，判定结果如表5.4所示。实验结果表明客体敏感度向量越高，客体的隐私程度越高，访问请求允许率越低，所以访问请求允许率与客体敏感度向量成反比。

表5.4 访问请求允许率与客体敏感度向量的关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 目标客体编号 | 客体敏感度向量 | 系统风险阈值 | 允许访问个数 | 允许率 |
| 555 | 0.5 | 0.8 | 380 | 95.00% |
| 666 | 0.6 | 0.8 | 340 | 85.00% |
| 777 | 0.7 | 0.8 | 285 | 71.25% |
| 888 | 0.8 | 0.8 | 205 | 51.25% |
| 999 | 0.9 | 0.8 | 125 | 31.25% |

* 1. 本章小结

本章首先介绍了基于风险的XACML访问控制原型系统结构图，然后介绍了基于风险的XACML访问控制系统应用示例，包括背景介绍、需求分析、详细设计和具体实现这四个步骤，最后基于车联网应用示例对原型系统进行评估，验证了原型系统的有效性和可行性。

# 总结与展望

本文分析了雾计算环境下的安全及隐私泄露等问题，从实际情况出发指出了解决该问题的必要性。通过分析国内外研究动态发现，目前用访问控制技术解决雾计算安全与隐私问题的研究已经有学者在做，但在方法上主要是基于角色的访问控制和基于属性的访问控制，在基于风险的访问控制方面仍然处于起步探索阶段。因此，本文提出了雾计算下基于风险的XACML访问控制模型，在此模型基础上，首先提出了一种新的风险评估方法，通过主体信誉值、客体安全属性和上下文量化雾节点的风险分数；然后提出了基于风险的隐私策略自适应方法，通过文档分析和代码静态分析为雾节点自适应生成正确和可读的隐私策略；最后设计并实现了基于风险的XACML访问控制原型系统，并基于车联网数据对该原型系统进行评估。主要内容总结如下：

1. 在基于风险的XACML访问控制系统的基础上，提出了风险评估方法，用来量化雾节点的风险分数。风险分数主要分为主体的风险评估和主体行为的风险评估两部分。主体的风险评估首先根据友好访问次数和恶意访问次数计算主体信誉值，然后通过主体信誉值和客体敏感度向量计算主体安全不确定性，最后根据主体安全不确定性和主体的安全级别计算主体威胁。主体行为的风险评估首先根据访问矩阵和客体敏感度向量计算上下文无关的访问代价，然后通过上下文无关的访问代价和上下文值计算上下文相关的访问代价，最后根据上下文相关的访问代价和主体损害客体的概率计算主体对客体的期望威胁。
2. 在基于风险的XACML访问控制系统的基础上，提出了隐私策略自适应方法，根据雾节点的风险分数自适应生成可阅读性的隐私策略。R2PA分为三个模块，第一个模块为文档分析模块，使用Stanford Parser得到API描述的句子结构树和依存句法信息，并通过算法4.1确定API收集的个人信息；第二个模块为代码静态分析模块，基于Vulhunter得到中间代码的应用程序属性图（APG），并对属性图进行深度优先遍历和可达性分析算法，通过算法4.2确定调用敏感API的上下文信息。第三个模块为隐私策略自适应模块，根据风险分数的大小为三类主体自适应生成正确和可读的隐私策略。
3. 本文设计并实现了基于风险的XACML访问控制原型系统。首先提出了原型系统的结构图，然后根据车联网数据实现了基于风险的XACML访问控制系统应用示范，包括背景介绍、需求分析、详细设计和具体实现等步骤，最后通过实验验证了该原型系统的有效性和可行性。

通过本论文的研究工作，为雾计算环境下的敏感数据提供了更细粒度的访问控制模型，该模型较传统访问控制模型，可以自适应调节雾节点的访问权限，并为雾节点自适应生成隐私策略，极大地降低了安全管理员授权管理操作和人工生成隐私策略的任务量。本论文尽管取得了上述研究成果，但是还存在以下不足之处：

1. 由于雾计算分布式的特点，不同的雾节点可能对授权管理有着不同的需求，在动态授权决策的改进过程中，还必须要考虑针对多种访问控制手段的兼容问题。

（2）本文提出了基于风险的隐私策略自适应方法，但是在信息保留和传输分析中还可以进行研究，根据分布式雾节点的风险分数大小，来判断雾节点之间是否可以传输隐私策略。

# 参考文献

1. Patwary A A N, Fu A, Naha R K, et al. Authentication, Access Control, Privacy, Threats and Trust Management Towards Securing Fog Computing Environments: A Review[J]. arXiv preprint arXiv:2003.00395, 2020.
2. Symanovich S. The future of IoT: 10 predictions about the internet of things[J]. cyber Security Blog, Norton by Symantec, Accessed, 2019: 02-17.
3. Xia H, Xiao F, Zhang S, et al. Trustworthiness inference framework in the social Internet of Things: A context-aware approach[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019: 838-846.
4. Networking C V. Cisco global cloud index: Forecast and methodology, 2015-2020. white paper[J]. Cisco Public, San Jose, 2016.
5. Gantz J, Reinsel D. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east[J]. IDC iView: IDC Analyze the future, 2012: 1-16.
6. Finnegan M. Boeing 787s to create half a terabyte of data per flight, says Virgin Atlantic[J]. Computerworld UK, 2013, 6.
7. Farjadian A B, Thomsen B, Annaswamy A M, et al. Resilient Flight Control: An Architecture for Human Supervision of Automation[J]. IEEE Transactions on Control Systems Technology, 2020. DOI: 10.1109/TCST.2019.2959542.
8. Mandelli S, Andrews M, Borst S, et al. Satisfying Network Slicing Constraints via 5G MAC Scheduling[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019: 2332-2340.
9. Zhang X, Zhu Q. D2D offloading for statistical QoS provisionings over 5G multimedia mobile wireless networks[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019: 82-90.
10. Firdhous M, Ghazali O, Hassan S. Fog computing:Will it be the future of cloud computing?[C]//The Third International Conference on Informatics & Applications,Kuala Terengganu, Malaysia. 2014:8-15.
11. 汪金苗, 王国威, 王梅, 等. 面向雾计算的隐私保护与访问控制方法[J]. 信息网络安全, 2019, 19(9): 41.
12. 曹咪, 徐雷, 陶冶. 雾计算认证与隐私保护研究综述[J]. 信息通信技术, 2018, 012(006):25-33.
13. OpenFog, Special Report - Size and Impact of Fog Computing Market (2017-2022)[EB/OL].[2018-11-10]. https://www.openfogconsortium.org/growth/.
14. Sepczuk M, Kotulski Z. A new risk-based authentication management model oriented on user's experience[J]. Computers & Security, 2018, 73: 17-33.
15. Jabal A A, Davari M, Bertino E, et al. Profact: A provenance-based analytics framework for access control policies[J]. IEEE Transactions on Services Computing, 2019.
16. Gupta S, Buriro A, Crispo B. Driverauth: A risk-based multi-modal biometric-based driver authentication scheme for ride-sharing platforms[J]. Computers & Security, 2019, 83: 122-139.
17. Cao Y, Huang Z, Yu Y, et al. A topology and risk-aware access control framework for cyber-physical space[J]. Frontiers of Computer Science, 2020, 14(4): 1-16.
18. 惠榛, 李昊, 张敏,等. 面向医疗大数据的风险自适应的访问控制模型[J]. 通信学报, 2015, 036(012):190-199.
19. 杨宏宇，宁宇光. 一种云平台动态风险访问控制模型[J]. 西安电子科技大学学报，2018,45（5）：80-88.
20. Choi D, Kim D, Park S. A framework for context sensitive risk-based access control in medical information systems[J]. Computational and mathematical methods in medicine, 2015, 34 (7): 1-9.
21. Fugini M, Teimourikia M, Hadjichristofi G. A web-based cooperative tool for risk management with adaptive security[J]. Future Generation Computer Systems, 2016, 54: 409-422.
22. Atlam H F, Alenezi A, Walters R J, et al. Developing an adaptive Risk-based access control model for the Internet of Things[C]//2017 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and ieee smart data (SmartData). IEEE, 2017: 655-661.
23. Fall D, Blanc G, Okuda T, et al. Toward quantified risk-adaptive access control for multi-tenant cloud computing[C]//The 6th Joint Workshop on Information Security. 2011: 1-14.
24. Khambhammettu H, Boulares S, Adi K, et al. A framework for risk assessment in access control systems[J]. computers & security, 2013, 39: 86-103.
25. Santos D R, Marinho R, Schmitt G R, et al. A framework and risk assessment approaches for risk-based access control in the cloud[J]. Journal of Network and Computer Applications, 2016, 74: 86-97.
26. Xiao Quan. A matter-element method for risk identification of technology innovation [J]. International Journal of Systems Assurance Engineering and Management, 2018, 9(3):716-728.
27. 宛鑫. 基于风险客户识别技术的预警建模及实现[D]. 中国科学院大学(工程管理与信息技术学院), 2015.
28. 师伟. 基于.NET 的金融风险管理系统的设计和实现[D]. 电子科技大学, 2015.
29. 孙存一, 张秋, 王彩霞. 大数据算法的精确制导:信贷客户识别以及特征要素分析[J]. 税务与经济, 2016(1):29-33.
30. Rubio-Medrano Carlos E., Zhao Ziming, Ahn Gail-Joon. Riskpol: A risk assessment framework for preventing attribute-forgery attacks to ABAC policies[C]. 3rd ACM Workshop on Attribute-Based Access Control, ABAC 2018, 2018:54-60.
31. Sicari Sabrina, Rizzardi Alessandra, Miorandi, Daniele. A risk assessment methodology for the Internet of Things [J]. Computer Communications, 2018,129:67-79.
32. Breaux T, Antón A. Analyzing regulatory rules for privacy and security requirements[J]. IEEE transactions on software engineering, 2008, 34(1): 5-20.
33. Yu L, Luo X, Liu X, et al. Can we trust the privacy policies of android apps?[C]//2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 2016: 538-549.
34. Costante E, Sun Y, Petković M, et al. A machine learning solution to assess privacy policy completeness: (short paper)[C]//Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society. 2012: 91-96.
35. Zimmeck S, Bellovin S M. Privee: An architecture for automatically analyzing web privacy policies[C]//23rd {USENIX} Security Symposium ({USENIX} Security 14). 2014: 1-16.
36. Miao D, Kagal L. Privacy informer: An automatic privacy description generator for mobile apps[J]. 2015.
37. Rowan M, Dehlinger J. Encouraging privacy by design concepts with privacy policy auto-generation in eclipse (page)[C]//Proceedings of the 2014 Workshop on Eclipse Technology eXchange. 2014: 9-14.
38. Rosen S, Qian Z, Mao Z M. Appprofiler: a flexible method of exposing privacy-related behavior in android applications to end users[C]//Proceedings of the third ACM conference on Data and application security and privacy. 2013: 221-232.
39. 王爽. 基于策略的访问控制在网络安全联动系统的研究与应用[D]. 华北电力大学, 2015.
40. 房梁, 殷丽华, 郭云川, 等. 基于属性的访问控制关键技术研究综述[J]. 计算机学报, 2017, 40(7): 1680-1698.
41. 解文冲, 杨英杰, 汪永伟, 等. 基于风险的访问控制操作需求计算方法研究[J]. 计算机工程与设计, 2013, 7.
42. 高志民, 王声远. 基于风险分析的应用系统访问控制模型[J]. 北京交通大学学报, 2011, 35(5): 21-25.
43. Běhounek L, Cintula P. From fuzzy logic to fuzzy mathematics: A methodological manifesto[J]. Fuzzy Sets and Systems, 2006, 157(5): 642-646.
44. Lelliott R . Fuzzy sets, natural language computations, and risk analysis[J]. Fuzzy Sets & Systems, 1988, 27(3):395-396.
45. McGraw R. Risk-adaptable access control (radac)[C]//Privilege (Access) Management Workshop. NIST–National Institute of Standards and Technology–Information Technology Laboratory. 2009, 25: 55-58.
46. Shaikh R A, Adi K, Logrippo L, et al. Dynamic risk-based decision methods for access control systems[J]. Computers & Security, 2012, 31(4): 447-464.
47. Cheng P C, Rohatgi P, Keser C, et al. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control[C]//2007 IEEE Symposium on Security and Privacy (SP'07). IEEE, 2007: 222-230.
48. Schütze H, Manning C D, Raghavan P. Introduction to information retrieval[M]. Cambridge: Cambridge University Press, 2008.
49. Cutts M. Oxford guide to plain English[J]. Refer, 2013, 29(3): 15.
50. Rasthofer S, Arzt S, Bodden E. A machine-learning approach for classifying and categorizing android sources and sinks[C]//NDSS. 2014, 14: 1125.
51. Au K W Y, Zhou Y F, Huang Z, et al. Pscout: analyzing the android permission specification[C]//Proceedings of the 2012 ACM conference on Computer and communications security. 2012: 217-228.
52. Bird S. NLTK: the natural language toolkit[C]//Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions. 2006: 69-72.
53. Cer D M, De Marneffe M C, Jurafsky D, et al. Parsing to Stanford Dependencies: Trade-offs between Speed and Accuracy[C]//LREC. 2010.
54. Cowie J, Lehnert W. Information extraction[J]. Communications of the ACM, 1996, 39(1): 80-91.
55. Xu Y, Kim M Y, Quinn K M, et al. Open information extraction with tree kernels[C]//Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013: 868-877.
56. Gabrilovich E, Markovitch S. Computing semantic relatedness using wikipedia-based explicit semantic analysis[C]//IJcAI. 2007, 7: 1606-1611.
57. Qian C, Luo X, Le Y, et al. Vulhunter: toward discovering vulnerabilities in android applications[J]. IEEE Micro, 2015, 35(1): 44-53.
58. Arlt S, Schäf M. Joogie: Infeasible code detection for java[C]//International Conference on Computer Aided Verification. Springer, Berlin, Heidelberg, 2012: 767-773.
59. Johnson R, Wang Z, Gagnon C, et al. Analysis of android applications' permissions[C]//2012 IEEE Sixth International Conference on Software Security and Reliability Companion. IEEE, 2012: 45-46.
60. Stevens R, Gibler C, Crussell J, et al. Investigating user privacy in android ad libraries[C]//Workshop on Mobile Security Technologies (MoST). Citeseer, 2012, 10.
61. Lin J, Amini S, Hong J I, et al. Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing[C]//Proceedings of the 2012 ACM conference on ubiquitous computing. 2012: 501-510.
62. Yang W, Xiao X, Andow B, et al. Appcontext: Differentiating malicious and benign mobile app behaviors using context[C]//2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. IEEE, 2015, 1: 303-313.
63. 孙笑宇. 大数据环境下车联网个性化信息服务模式研究[D]. 吉林大学, 2016.
64. 陆化普, 孙智源, 屈闻聪. 大数据及其在城市智能交通系统中的应用综述[J]. 交通运输系统工程与信息, 2015, 15(5): 45-52.
65. 张绍华, 潘蓉, 宗宇伟. 大数据治理与服务[M]. 上海科学技术出版社, 2016.
66. Dataj. 纽约市出租车行车位置记录[DB/OL]. http://dataju.cn/Dataju/ web/datasetInstanceDetail/76, 2018.3.

# 附录1 攻读硕士学位期间撰写的论文

[1] 柯昌博、吴嘉余、曹彦，面向云计算应用层演化的隐私保护方法研究，计算机工程与应用，已录用。

[2] 柯昌博、黄志球,、吴嘉余，面向大数据的隐私发布暴露检测方法，计算机科学，已录用。

# 附录2 攻读硕士学位期间申请的专利

[1] 柯昌博、汤梅、吴嘉余、肖甫，一种面向服务组合的隐私最小暴露方法，**201910084983.4，2019.1；**

**[2] 柯昌博、汤梅、吴嘉余、朱泽江、肖甫，一种针对Web协同的授权用户风险评估方法及系统，201910084984.9，2019.1。**

# 附录3 攻读硕士学位期间参加的科研项目

（1）国家自然科学基金，面向SaaS协同交互与演化的隐私暴露检测方法研究（61602262）。

# 致谢

三年的研究生学习生涯已接近尾声，值此论文完成之际，向所有关心、指导和帮助过我的老师、同学、朋友及家人表示真挚的感谢！

首先，我要感谢我的导师柯昌博教授，本篇论文的选题方向、研究点和思路，导师都进行了悉心指导。您三年来不仅在学术方面给了我很大的启发，还教会了我很多为人处世的道理。因此，我的学术和代码的水平得到了很大的提高，无论是从文献的收集还是论文的书写，我的每一点进步都凝聚着您付出的心血。

其次，我要感谢舍友和研究生生涯遇到的朋友，在校期间我和他们一起学习，一起生活，遇到困难同学总是尽力帮助我，同学的优秀和自律鼓舞着我前进，使我在研究生期间学到很多知识。

最重要的，我要感谢我的父母，他们一直以来都是我强有力的精神和物质支撑。父母在生活中对我无微不至的照顾，在我遇到困难时鼓励和安慰我，让我明白了很多做人和做事的道理。

最后，感谢在座的各位评审专家们，感谢您们的到来，感谢您们能够提出宝贵的建议和意见，谢谢您们的到来。