为了宣传信息安全知识，培养大学生的创新意识、团队合作精神，教育部高等学校网络空间安全专业教学指导委员会自2008年开始举办全国大学生信息安全竞赛。经过15年的发展，竞赛得到国内网络安全行业的充分认可，已成为国内高校最具影响力的学科竞赛，竞赛培育、选拔了一大批优秀网络安全作品，为国内网络安全领域输送了大批网络安全人才。竞赛将进一步加强与行业企业联系，进一步对接网络安全技术发展前沿、聚焦信创产业产品、技术需求，改革创新赛制，激发大学生创新活力，为营造网络安全教育、技术、产业融合发展的良性生态作出积极贡献。

——教育部高等学校网络空间安全专业教学指导委员会秘书长、
北京电子科技学院副院长　封化民

安卓应用隐私合规检测方法研究

王申奥，王亚龙，王乾旭，贺紫怡，李　晖
（西安电子科技大学 网络与信息安全学院，陕西 西安 710071）

摘　要： 近年来，移动应用超范围收集用户隐私信息，强制索取敏感权限等现象屡见不鲜。业界现有的隐私合规检测产品因缺乏对隐私政策的分析从而产生较高的误报率和漏报率。针对国内现行合规要求，设计并实现了一套大规模的半自动化合规检测框架。通过对现有应用市场中1 941款应用进行实证评估，检测到52款典型违法违规移动应用。实验结果表明，该方法实用性强，拓展性高，具有广泛的应用前景。

关键词： 隐私合规；权限滥用；自然语言处理；动静态程序分析
中图分类号： TP311.5　　**文献标识码：** A　　**DOI：** 10.19358/j.issn.2097-1788.2023.01.001
引用格式： 王申奥，王亚龙，王乾旭，等. 安卓应用隐私合规检测方法研究[J].网络安全与数据治理，2023，42(1)：4-14.

Research on detection of Android application privacy compliance

Wang Shenao，Wang Yalong，Wang Qianxu，He Ziyi，Li Hui
（School of Cipher Engineering，Xidian University，Xi′an 710071，China）

Abstract： In recent years，it is common for mobile applications to collect user privacy information in excess of the scope and abuse sensitive permissions. The existing privacy compliance detection products in the industry lack the analysis of privacy policies，resulting in high false positive and false negative. This study designs and implements a large-scale semi-automated compliance detection framework to address the current compliance requirements in China. The system extracts permission phrases through automated analysis of privacy policies and identifies sensitive permission calls through hybrid program analysis，ultimately achieving consistent compliance detection of privacy policies and permission calls. The empirical evaluation of 1 941 applications in the existing application market detects 52 typical illegal and non-compliant mobile applications. The experimental results show that the method is practical and highly scalable，and has a wide application prospect.

Key words： privacy compliance；permission abuse；natural language processing；dynamic and static program analysis

0 引言

近年来，移动应用超范围收集用户隐私信息，强制索取敏感权限等现象屡见不鲜。为了保护用户的个人隐私信息，监管部门要求企业或组织在隐私政策以简洁易读的方式告知用户他们如何收集、存储和管理用户的个人信息。然而，根据武汉大学2021年的相关调查[1]显示，77.8%的用户在安装App时“很少或从未”阅读过隐私协议，69.69%的用户

会忽略 App 隐私协议的更新提示。尽管一些服务提供商已经提高了其隐私政策的可理解性和可读性,但这些政策仍然篇幅太长,难以阅读[2-3]。此外,2021 年国家计算机网络应急技术处理协调中心和中国网络空间安全协会共同发布的《App 违法违规收集使用个人信息监测分析报告》[4]中也显示,超范围收集用户隐私信息,违反用户"知情同意"原则的违法违规应用在各主流应用市场仍然广泛存在。

近来,隐私合规分析的相关工作在国外颇受关注,逐渐被应用到大规模网站隐私合规性分析、移动应用隐私泄露检测等领域。移动应用的隐私合规分析主要包括隐私政策文本分析与程序分析两个部分。静态程序分析执行效率高,然而由于缺乏运行时路径信息,静态分析往往会产生一定程度的误报。动态污点分析通常是利用插装和代码重写为污点数据创建污点标记,优点是准确率更高,但插装和代码重写往往带来更大的性能开销。隐私合规研究往往是在程序分析的基础上结合隐私政策文本进行合规性检查。隐私政策文本分析作为国外新兴的研究热点,已经陆续建立起丰富的隐私政策语料库。然而在中文领域,隐私政策命名实体识别的研究仍然缺乏,中文隐私政策的公开语料库也仍处于空白。这些问题制约了国内隐私政策与程序分析相结合的自动化合规检测技术的发展。

为了解决上述问题,本文通过人工注释构建危险权限术语词典,提出利用双向最大匹配算法实现基于词典的隐私政策自动标注,从而构建中文隐私政策权限词实体识别语料库。在此基础上,本文为隐私政策语料构建预训练字嵌入,通过双向长短期记忆神经-条件随机场(Bi-directional Long Short-Term Memory-Conditional Random Field,BiLSTM-CRF)架构实现最优标签序列预测,从而完成权限词实体识别任务。在应用程序动静态混合分析部分,基于 Androguard 实现交叉引用并对程序实际调用的危险权限进行静态分析。通过隐私政策声明权限集与实际调用权限集的一致性分析,实现了对超范围收集敏感信息行为的检测。此外,依托 Frida 动态插桩与 Hook 技术,对敏感应用编程接口(Application Programming Interface,API)进行重载,记录函数调用堆栈、调用频次、关键参数等行为日志信息,针对同意隐私政策前收集、静默状态下频繁访问敏感信息实现运行时状态监测。

1 相关工作

自 2018 年欧盟出台《通用数据保护条例》(General Data Protection Regulation,GDPR)以来,国内外学者针对安卓隐私泄露与合规检测等问题进行了大量研究。关于安卓隐私合规检测的工作最早可以追溯到对隐私泄露的研究,但二者的侧重又有所不同。安卓隐私泄露的相关研究关注于数据流的动静态混合污点分析,目标是确定安卓隐私泄露的潜在路径,而从技术上讲,如果这些数据流在隐私政策中被披露,它们就不属于"泄露"行为。因此,相比单纯基于数据流的动静态污点分析,合规检测更关注于隐私声明自动化分析与安卓隐私数据泄露的交叉领域。目前的相关工作重点关注于提取隐私政策中声明的权限信息并与实际权限调用进行一致性分析,下面分类介绍这两个方面的研究进展。

1.1 动静态混合污点分析

总体来说,程序分析方法主要包括静态分析和动态分析两种。其中静态分析主要是通过对 APK 文件进行反编译,分析从源点(Source)到汇点(Sink)的潜在数据流路径,从而检测隐私泄露的方法。典型的静态污点分析工具有 FlowDroid[5]和 IccTA[6]等。FlowDroid[5]拓展了 Soot 框架,提出了按需追踪的反向别名分析方法,从而支持上下文敏感、流敏感、域敏感、对象敏感,对于直接赋值,函数调用和别名传播三种污点传播方式都能实现高精度分析。IccTA 在 FlowDroid 的基础上为每个组件创建虚函数,并建立组件间连接,解决了组件间通信导致的隐私泄露问题。但是由于缺乏运行时路径信息,以上静态分析方法都很难正确地对分支语句(判断和控制)进行处理,可能衍生出一些实际场景中根本不可能被执行的程序路径,从而产生一定的误报。

动态分析则是通过模拟应用的实际运行情况,获取应用运行时的行为数据,从而检测数据能否从污染源点传播到污染汇点。典型的动态分析方法包括 TaintDroid[7]和 TaintEraser[8],它们通常是利用插装和代码重写为污点数据创建污点标记,根据指令类型和指令操作数设计相应的传播逻辑,在此基础上进行污点标记的存储与追踪。然而,插装和代码重写往往带来更大的性能开销,在纯动态的情况下,原始程序的每条指令通常需要 6~8 次额外的污点追踪指令来传播污点标签。

目前安卓隐私泄露的最新研究更多地集中在

轻量级动静态混合分析。混合分析中的静态部分旨在预优化追踪逻辑,缩减追踪范围,从而减小动态分析的性能开销。FSAFlow[9]通过修改 FlowDroid[5]框架实现目标路径与关键分支信息的搜索,选择违反预定义的隐私策略的潜在路径,利用有限状态自动机进行编码,将少量状态管理代码插装到程序相应位置,防止运行时信息泄露。

1.2 隐私声明与权限调用一致性分析

相比于动静态混合污点分析的隐私泄露研究,隐私声明与权限调用的一致性分析更关注于隐私声明的自动化解析,从而为应用行为动静态隐私泄露分析提供判断依据。

早期研究[10-11]大多是利用 AndroidManifest.xml 文件或者应用描述信息来识别开发者对于应用权限调用的相关声明,一些经典的工作如 WHYPER[10]和 AutoCog[11]利用自然语言处理技术从应用描述中提取权限信息并与应用实际调用的权限进行对比,从而识别隐私违规行为。然而,应用描述信息字符有限,不能够完整地表达开发者对于权限调用的实际声明。此外,AndroidManifest.xml 文件中列举的权限信息也不能够直接反映开发者的权限声明和用户所知情的授权范围,因此其研究局限性较强。

最近的一些工作[12-17]更关注于隐私政策与应用程序关联分析的合规检测方法。Slavin[12]等人构建了最早的隐私声明与权限调用的一致性分析框架,通过对 50 个移动应用隐私政策的人工注释构建了隐私策略术语词典,使用基于描述逻辑的形式化语言构建权限术语本体,在此基础上构建了隐私策略本体短语与 API 方法的映射,从而识别弱违规和强违规行为。这项工作的局限性在于没有区分弱违规的程度,用户隐私的范围也只限于通过安卓权限获取的敏感信息,而忽略了用户的动态输入的隐私信息(如手机号码、身份证 ID 等)。Wang[13]等人随后在这项工作的基础上做了改进,他们考虑了用户通过 UI 界面输入的隐私信息,通过构建 UI 控件树从控件标签中获取语义,从而识别应用收集的敏感信息类型,完成隐私政策声明与敏感信息收集行为的一致性检测。

另外一些工作如 Polisis[14]和 PolicyLint[15]则是将深度学习应用到隐私政策的自动化分析任务中。Polisis[14]通过训练特定于隐私声明领域的词嵌入并构建 CNN 多层次分类模型,以 88.4%的准确率实现了隐私政策的结构化的解析。PolicyLint[15]在这项工作的基础上考虑数据对象和实体的否定,从语义层面对谷歌应用市场中 11 430 个应用程序的隐私声明中存在的矛盾进行分析,实现了否定词敏感的隐私政策结构化解析。Andow 等人提出的 PoliCheck[16]则是在二者的基础上将隐私政策解析进一步细化,通过区分敏感信息收集的第一方主体与第三方 SDK,对数据流路径末端汇点的目的域名进行匹配,实现了实体敏感的隐私合规检测。Nguyen[17]等人着重关注于用户同意隐私政策前的违规收集行为,通过与应用无 UI 交互状态下的网络流量审计,检测程序中存在的违规敏感信息传输。

然而,目前隐私政策自动化解析的相关研究还主要集中在英文领域。隐私政策文本分析作为国外新兴的研究热点,已经陆续建立起丰富的隐私政策语料库。Usable Privacy Policy 项目提供了十余个常用的英文隐私政策语料库,如 OPP-115 Corpus[18]、APP-350 Corpus[19]、Opt-out Choice[20]等。基于这些语料数据训练的自然语言处理模型能够从隐私政策中自动提取隐私信息相关的结构化数据。但目前国内尚未建立相关中文隐私政策语料库,中文隐私政策文本自动分析的相关研究也比较缺乏。

2 隐私政策自动化解析

为了实现隐私政策与应用程序的联合分析,需要实现隐私政策的自动化信息抽取,得到其中权限声明相关的结构化数据。在隐私政策解析部分,本文将权限信息的结构化抽取作为命名实体识别任务,构建了中文隐私政策权限词实体识别语料库,并基于 BiLSTM-CRF 完成模型训练,方案架构如图 1 所示。

2.1 中文隐私政策语料库构建

为了完成隐私政策中危险权限词的命名实体识别任务,需要构建人工标记的隐私语料数据集用于模型训练。因此,在本部分,本文首先获取隐私政策原始语料数据,原始语料的收集主要考虑了以下两个方面:

(1)样本量:为了防止数据集过小造成的过拟合现象,同时综合考虑模型的准确度和人工标注的时间成本,最终选定了用于人工标记的隐私政策语料 93 篇,待拓展隐私政策语料 965 篇。

(2)抽样方法:为了使样本分布尽可能与所有在架应用分布保持一致,本文按照以下两条规则进行抽样。一是样本空间热门应用(high-profile)和冷门应用(long-tail)比例适当(即样本要符合在架应用下载量的幂律分布规律),结合小米应用市场的应用

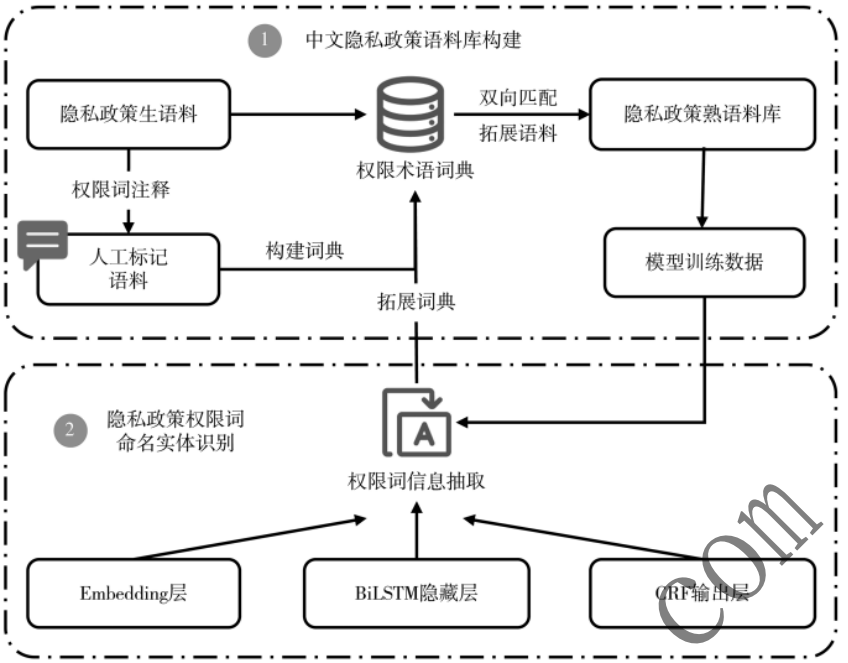

图 1 隐私政策自动化解析方案架构

下载量，最终选取头部应用 231 款，冷门应用 827款；二是样本应用类型覆盖范围尽可能广泛，最终选取包括金融理财、社交聊天、图书阅读等在内的 14 类常见类型应用。

(3)收集方法：应用隐私政策的获取方式有两种，一是提取应用内嵌在安装包内的隐私政策，但这种方法需要对应用进行反编译，时间成本高昂；二是通过提取开发者上传到应用市场的隐私政策，通常这些隐私政策存储在服务器上，因此可以利用 Selenium 自动请求隐私政策链接，进一步调用 Beautiful Soup 库进行格式化解析，去除多余标签，从而完成隐私政策提取。

在隐私政策的自动化解析中，应重点关注危险权限词的实体识别任务。因此在本小节，本文以 Android 定义的危险权限组归纳出九类实体类型，如表1 所示。

命名实体识别广义上属于序列标记任务的范畴，因此其语料标注规范也遵循序列标记的通用规范，常见的序列标注格式有三种：BIO 格式、BMES 格式以及 BIOES 格式。三者都是字符粒度级别的标注格式，各标签含义如表 2 所示。

在此选择 BIOES 格式进行标注(B 代表实体词的开始；I 代表实体词中间位置的字符；O 代表非实体字符；E 表示实体词的结束；S 代表单个字符组成的实体)。相比于其他两种标注格式，BIOES 格式提供了更丰富的信息，便于在模型预测时提取实体。

表 1 九类危险权限术语词类型

实体标签	实体名称	实体描述
CALENDAR	日历权限组	读/写/访问日历等
CAMERA	相机权限组	拍摄照片，访问摄像头等
CONTACTS	联系人权限组	读/写通讯录，账户列表等
LOCATION	位置权限组	获取粗略/精确位置等
MICROPHONE	麦克风权限组	访问麦克风及音频录制等
PHONE	手机状态权限组	获取设备状态，通话记录等
SENSORS	传感器权限组	访问设备传感器等
SMS	短信权限组	读/写短信等
STORAGE	存储权限组	读/写存储空间等

表 2 常用序列标注规范中的标签含义

标签	含义	说明
B	Begin	表示字符在实体块的开始位置
I	Inside	表示字符在实体块内部
M	Middle	表示字符在实体块内部，与 I 相同含义
E	End	表示字符在实体块的结束位置
S	Single	表示单个字符组成的实体块
O	Outside	表示字符不属于任何类型的实体块

在中文隐私政策危险权限词实体识别任务中，共安排三位注释员分别对 93 篇隐私政策原始语料进行注释。为了解决人工标注文本存在的歧义，评估标注序列的共现频率，结合注释规则确定有效标注：当两

位注释员同时为某个词序列添加标记后,才将该注释作为实体标记,引入第三位注释员进行判断。对93篇隐私政策标注并去重后共获得危险权限词注释964份,但由于标注歧义等情况,最终经过注释规则与人工审核确定了有效危险权限词标记677个。

通过人工标注获取的熟语料注释准确性高,不容易出现注释歧义,但缺点是人工注释费时费力,由于泛化能力的需求,实际场景中的命名实体识别任务很难仅靠人工标注的语料数据进行模型训练。解决小样本命名实体识别语料缺乏的方法之一是通过双向最大匹配算法(Bidirectional Maximal Matching, BMM)进行语料拓展。

双向最大匹配算法是一种典型的基于词典的序列标注方法,它通过比较正向最大匹配(Forward Maximal Matching, FMM)和反向最大匹配(Reverse Maximal Matching, RMM)的标注结果得到正确的标注序列。

正向匹配算法从文本串起始处正向扫描,取出子串与字典进行匹配。后向匹配算法同理,差别在于后向匹配算法从结尾处对字符进行切分匹配。双向匹配算法正是在二者结果的基础上进行综合。如果前向匹配和后向匹配的切分结果词数不同,则返回词数较少的;如果词数相同且切分结果相同,则返回任意一个结果;如果词数相同但分词结果不同,则返回单字较少的分词结果。

本文将人工标记的93篇语料和BMM拓展的965篇语料进行合并,得到最终的1 058篇隐私政策语料数据,作为隐私政策权限词实体识别训练语料。

2.2 隐私政策权限词命名实体识别

在该小节中,本文利用先前获得的标记语料构建数据集,在此基础上训练特定于隐私词领域的双向长短期记忆神经-条件随机场(Privacy Specific Bi-directional Long Short-Term Memory-Conditional Random Field, PRI-BiLSTM-CRF)模型,从而完成中文隐私政策权限词实体识别任务。

首先,对数据集进行划分。按照18:1:1的比例切分得到训练集、验证集、测试集。得到训练、验证、测试集的语句数量分别为103 553、6 543、6 305,分别加载数据集并进行预处理。预处理过程首先是对数据集进行分句,之后对标注格式进行检查,最后利用word2vec预训练100维字向量特征。为了充分地利用隐私政策中危险权限词的长度特性,本文额外构建20维的词长度特征。根据语句分词结果,将单字成词的特征标记为0,词首长度特征为1,词间长度特征为2,词尾长度特征为3,按照以上规则为每个字构造词长度id,通过Embedding层将词长度id处理为20维的向量特征,作为字向量特征的补充,提供更丰富的信息。之后将100位字向量特征与20维词长度特征拼接,最终得到120维向量输入模型,在此基础上进行训练与更新。

接下来对模型架构进行介绍,该模型由Embedding层(字向量嵌入层)、BiLSTM隐藏层、Linear层(线性层)与CRF输出层构成。模型结构如图2所示。

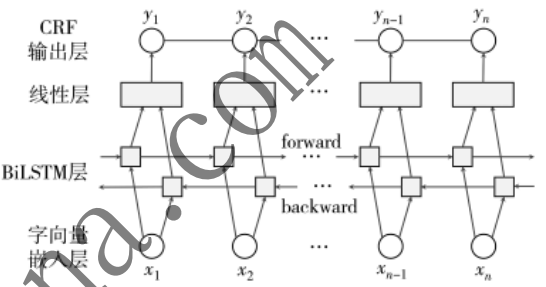

图2 PRI-BILSTM-CRF模型架构

在Embedding层,首先加载预训练的100维字向量并将词长度id转化为20维词长度特征,然后拼接得到120维向量输入。BiLSTM层则是以Embedding层的120维向量特征作为输出,通过前向和后向传播得到各自的隐状态,最后将前向LSTM(128维)和后向LSTM(128维)的隐状态拼接(256维)送入Linear层获得发射分数(标签向量)。而CRF层接收发射矩阵作为输入,通过计算标签的转移分数生成标签序列的概率分布,使用Viterbi算法解码得到最优路径。

在模型评估方法上,命名实体识别模型的评估方式通常分为两种:一是通用的基于字符标签进行直接评估,二是考虑实体边界和实体类型的评测。前者是字符级别的评估,主要方法是将所有测试样本的真实标签列表和预测标签列表进行比对,直接计算Precision、Recall和F1值。然而这种评估方式应用于序列标注时效果并不理想。在序列标注中,单个错误的字符标签的实际影响范围涉及整个实体,因此序列标注任务更多采用基于实体级别的评估方法,只有当实体边界和实体类别同时被标记正确,才能认为实体识别正确。Sang等人提出的CoNLL-2003[21]基准是一种典型的实体级别的命名实体识别模型评估方法,其中Precision被定义为模

型预测正确的命名实体占所有命名实体的百分比，Recall 是模型预测正确的命名实体占语料库中所有命名实体的百分比，只有当实体预测标签与真实标签完全匹配时才认为预测正确。利用 CoNLL－2003 方法对模型训练结果进行评估。根据验证集上的模型评估结果，保存最优模型。

通过构建中文隐私政策语料库并训练隐私政策权限词命名实体识别模型，本文实现了隐私政策的自动化信息抽取，得到其中权限声明相关的结构化数据。

3 应用程序动静态混合分析

在本节中，通过应用程序动静态混合分析来识别危险权限调用情况，完成隐私政策文本与程序实际权限调用的一致性检测。其中，在程序静态分析部分，通过反编译与源码分析实现基于 Androguard 框架的危险权限调用检测。在动态分析部分，利用基于 Frida 的 Hook 技术，实时获取应用使用的敏感权限并返回函数调用堆栈日志等。应用程序动静态混合分析整体方案如图 3 所示。

3.1 构建 API－PERMISSION 映射

根据 APK 文件精确识别出应用实际使用的危险权限，是 APP 合规检测的重点以及难点所在。通过构建权限映射关系，分析程序代码中的敏感 API，来完成危险权限调用识别。

根据安卓权限机制，所有危险权限都应该在程序中动态申请，因此在危险权限获取和使用的过程中，通常需要依赖某些特定的敏感 API 来实现。此外，Intent 对象的某些特定 ACTION 动作需要申请相关权限，通过 Uri 访问 Content Provider 时，特定 Uri 对象也意味着应用使用了相关权限。因此，敏感 API、Intent ACTION 和 Content Provider Uri 都能够关联应用的危险权限调用行为。无论是静态分析还是动态分析，核心思路都是捕获程序的敏感 API、Intent ACTION 或 Content Provider Uri，实现对危险权限调用及敏感信息收集的检测。

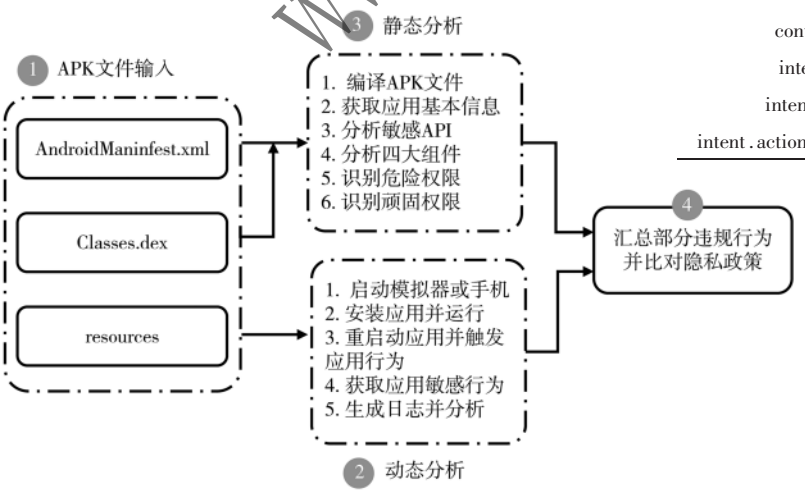

图 3 应用程序动静态混合分析方案架构

为了能够识别应用程序的实际权限调用情况，首先的工作是构建全面、精准的 API－PERMISSION 映射关系。在以往的工作中，PScout[22]利用 Soot 框架执行静态分析并构建函数调用图，在图上执行向后可达性分析，识别出可达权限的所有 API 调用，并重复可达性分析直到权限检查数量收敛，最终构建出 Android API 16 到 23 版本的权限映射结果；Axplorer[23]研究安卓框架的内部结构，生成高精度调用图，并通过控制流切片进行分析，最终生成 Android API 16 到 25 版本的权限映射结果。然而 PScout 和 Axplorer 的工作还存在一定的局限性，由于研究年限较早等原因，所涉及的 API 版本也只局限在 16 到 25 版本，因此，在 PScout 和 Axplorer 工作的基础上，本文对 API 26 到 32 版本中危险权限相关的 API 的映射关系进行了补充。其中部分映射如表 3 所示。

表 3 部分 API－PERMISSION 映射示例

敏感 API/字符串	对应危险权限
API:Camera.open	CAMERA
API:sendTextMessage	SEND_SMS
API:startRecord	RECORD_AUDIO
content://com.android.calendar	READ_CALENDAR
content://com.android.contacts	READ_CONTACTS
content://sms/inbox	READ_SMS
intent.action.CALL	CALL_PHONE
intent.action.SENDTO	SEND_SMS
intent.action.DATA_SMS_RECEIVED	RECEIVE_SMS

对于 Intent 和 Content Provider 而言，危险权限通常与特定的 ACTION 或 Uri 相关联。Intent 对象启动 Activity 时，某些特定 ACTION 动作需要申请相关权限，例如 android.intent.action.CALL 需要使用 CALL_PHONE 权限。访问 Content Provider 时，需要将指定 Content Provider 的 Uri 对象传递给 Content－Resolver 类，然后通过 Uri 定位资源，

进而完成增删查改等操作,因此特定 Uri 对象的访问也意味着应用需要提前申请并获取相关敏感权限。

在对 1 900 余款 App 进行测试和人工审核后,获取了 9 组 24 种危险权限的共计 9 200 余种 API-PERMISSION 映射关系。

3.2 基于 Androguard 的静态分析

基于 Androguard 的静态分析模块建立在 API-PERMISSION 映射库的基础上。该模块具有两大功能:一是识别应用实际调用的危险权限;二是识别应用程序的顽固权限申请行为。

为了识别应用实际调用的危险权限,采用以下流程进行静态分析:首先基于 Androguard 访问 APK文件中的所有信息,并使用 AnalyzeAPK()函数分析DEX文件,得到源码中存在的所有类与字符串;接下来根据构建好的权限映射关系,遍历得到的所有类和字符串,如果存在敏感 API 或特定 Intent.ACTION/Content Provider Uri 对应字符串,则根据映射关系库输出对应危险权限。通过上述过程识别出源码中存在的危险权限后,将识别出的危险权限集与隐私政策中声明的权限集进行一致性分析,判断应用存在的违规权限调用及超范围隐私信息收集行为。

应用程序的顽固权限申请行为检测主要包括两个步骤:第一步是根据函数之间的交叉引用,建立程序的函数调用图(Call Graph);第二步是根据函数调用图,对权限申请点进行动态回溯,查找可达的程序退出点,确定是否存在从权限申请函数到终止应用函数的路径。

首先,根据交叉引用建立函数关系图,遍历程序中所有方法,将这些方法作为有向图中的节点,添加到函数调用图中。然后查找每个函数节点的交叉引用,并通过图中的有向边表示函数之间的调用关系。图 4 展示了一个典型的函数调用图示例,其中,浅灰色表示 Activity 内部节点的相互调用,而深灰色则表示内部节点对外部节点的调用。

为确定权限申请的相关入口点是否会在一定条件下(即用户拒绝权限申请时)导致应用运行终止,可采用可达性分析方法,对以此类入口点为根节点的函数调用图进行遍历,以确定是否存在从权限申请函数到终止应用函数的路径。表 4 中展示了一些常见的权限申请入口点与程序终止出口点。

表 4 部分权限申请入口点与程序终止出口点示例

权限申请入口点	程序终止出口点
content.Context.checkPermission	app.Activity.finish
app.Activity.onRequestPermissionsResult	Java.lang.System.exit
content.Context.checkSelfPermission	os.Prosess.killProcess
app.Fragment.onRequestPermissionsResult	app.Activity.finishAffinity
Context.checkCallingorSelfPermission	Activity.moveTaskToBack

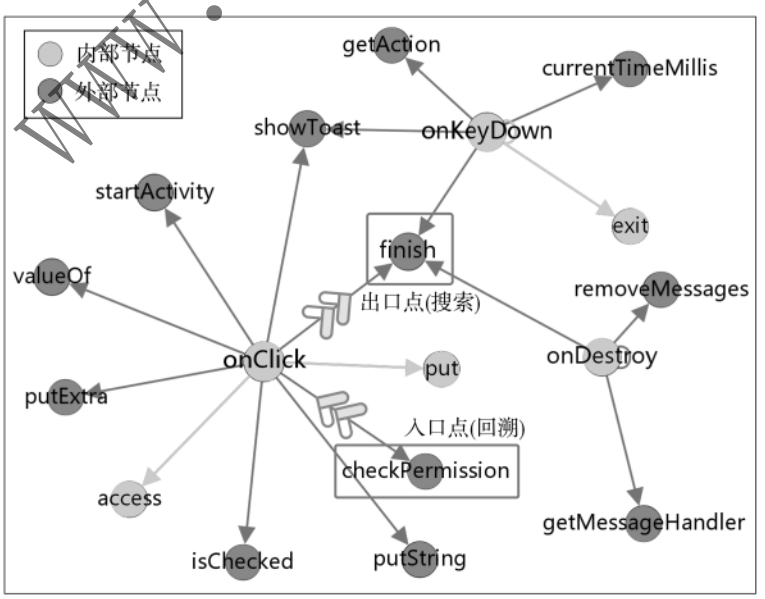

图 4 静态分析得到的函数调用图

当应用存在顽固权限申请行为时，权限申请函数和应用终止函数通常为并列关系。checkSelfPermission 检查程序权限申请状态，该函数作为顽固索权的入口点，finish 函数在未获得授权时结束程序，作为顽固索权终止点，二者在函数调用图中都是通过 onClick 调用，基于这种位置关系，可对入口点进行有界回溯，搜索函数关系图中是否存在对应的程序终止点。

3.3 基于 Frida 的动态分析

Frida 是一款基于 Python 和 JavaScript 的轻量级可编程调试框架，它具有细粒度的流程控制、代码级的可定制体系并且可以不断进行动态调试。动态分析基于 Frida 框架自动化 Hook App 所有敏感 API 的调用行为，利用 Python 实现远程过程调用(RPC)，通过 JavaScript 脚本生成 Native 函数实现敏感 API 拦截与重载，记录待检测样本运行时获取隐私信息的行为，动态分析方案架构如图 5 所示。

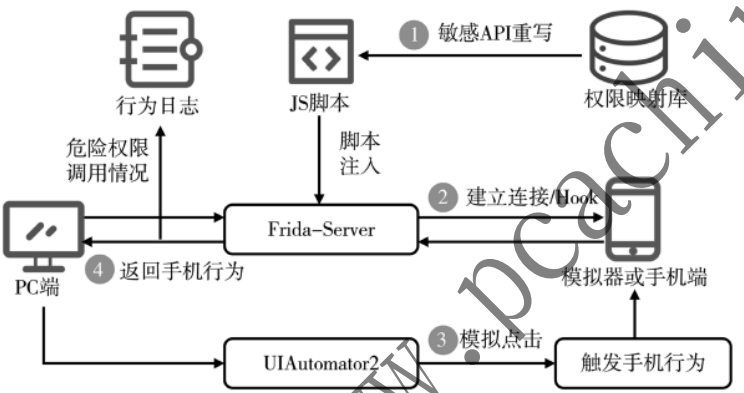

图 5 Frida 动态插桩分析方案架构

Frida 框架分为客户端和服务端，服务端中运行着 Frida-Server 后台进程和 Hook 目标进程。客户端 RPC 远程过程调用连接到服务端上的 Frida-Server，当 Frida-Server 接收到 Hook 指令时，Frida-Server 会向目标进程中注入 JavaScript 脚本及其执行引擎环境。通过 Frida 动态插桩技术，可以将 JavaScript 脚本插入到平台原生 App 的内存空间中，实现对程序逻辑的跟踪、监控甚至修改，因此 Frida 框架可以实现从 Java 层到 Native 层的函数 Hook。执行 Hook 操作时需要找到 Hook 位置，即类名加方法名，通过将 JavaScript 注入到黑盒进程中，Frida 框架能够 Hook 目标程序中几乎任何方法。在找到 Hook 位置后，需要在目标进程中触发对应的方法。Frida 框架通过修改 Dalvik 虚拟机的 accessFlags 标志，将需要 Hook 的 Java 方法注册为 Native 方法，并修改函数的入口为自定义的内容。当系统执行到待 Hook 的目标方法时，虚拟机优先执行 Native 方法，此时 Frida 拥有了程序控制权，可以优先执行注入代码，实现对程序逻辑的跟踪、监控与修改。

JavaScript 脚本是注入目标进程并 Hook 敏感 API 的关键。在 JavaScript 脚本中，首先通过 java.use() 函数找到目标类并实例化一个对象，调用目标类成功后，使用 overload() 函数对敏感 API 进行拦截并重载，在完成重载后，可以实时获得敏感 API 的参数，并可以根据需要对参数内容进行修改。

当手机端进行触发敏感行为的操作时，返回权限调用时间、权限调用类型、关键参数内容以及函数调用堆栈等信息。通过动态分析，可实现以下两种隐私违规行为的检测：

(1)用户同意隐私政策前的违规权限调用与信息收集行为。通过模拟用户操作到达隐私政策控件所在的 Activity，测试在点击同意的控件之前，应用是否存在收集 IMEI、设备 MAC 地址、通讯录和短信等敏感信息的行为。

(2)程序静默状态下的频繁收集行为。App 在静默状态或后台运行时，如果存在权限调用与信息收集行为，也会实时触发 Hook。

4 实验与分析

4.1 测试方案

本文对隐私政策权限词实体识别模型性能以及隐私合规检测整体方案在实际场景下的检测能力进行了测试，具体测试方法如下：

(1)模型测试部分，采用考虑实体边界和实体类别的 CoNLL-2003[21] 方法进行命名实体识别性能评估，评估指标包括 Precision、Recall 和 F1-score，每 100 次训练对模型进行评估，累计 1 000 次训练 F1-score 无优化后停止迭代。

(2) 检测能力测试部分，采取实证评估方法对来自小米应用市场与 360 手机助手的 1 941 款应用进行隐私合规检测。

4.2 测试环境与系统架构

本文实现的隐私合规检测系统包括 PC 端与手

机模拟器端两个部分。PC 端在Windows 10 21H1 64 位系统上进行测试,处理器为 Inter Core i5-11300H 3.10 GHz,内存为 Micron Technology 16 GB,软件环境包括 Python 3.10,Androguard 3.3.5,Uiautomator2,PyTorch 1.11.0 等,手机模拟器采用逍遥模拟器 7.6.1 版本,机型为 OPPO RENO PRO,系统版本为 Android 7.1.2。隐私合规检测系统整体架构如图 6 所示,自下而上包括隐私政策语料库、隐私政策解析模块、应用程序分析模块、隐私合规项检测模块。

4.3 实验结果分析

模型测试部分,每 100 次训练正向传播和反向传播完成后,计算模型在验证集上的损失函数,同时评估危险权限词实体识别 F1 值,实验结果如图 7 所示。

图 7 中,方框坐标轴及折线代表训练过程中验证集上的损失,三角坐标轴及折线代表训练过程中验证集上的 F1 值,在第 1 800 次训练时获得最小损失和最大 F1 值,此后 1 000 次训练中 F1 值无优化,模型停止迭代。

在测试集上对最优模型在 9 类权限词实体识别任务的 Precision、Recall 和 F1-score 进行评估,测试结果如表 5 所示。

最终,在 9 类危险权限词实体识别任务上的平均 Precision 为 91.09%,平均 Recall 为 93.21%,平均 F1-score 为 91.92%,结果说明模型在实体识别任务上有较优的性能。

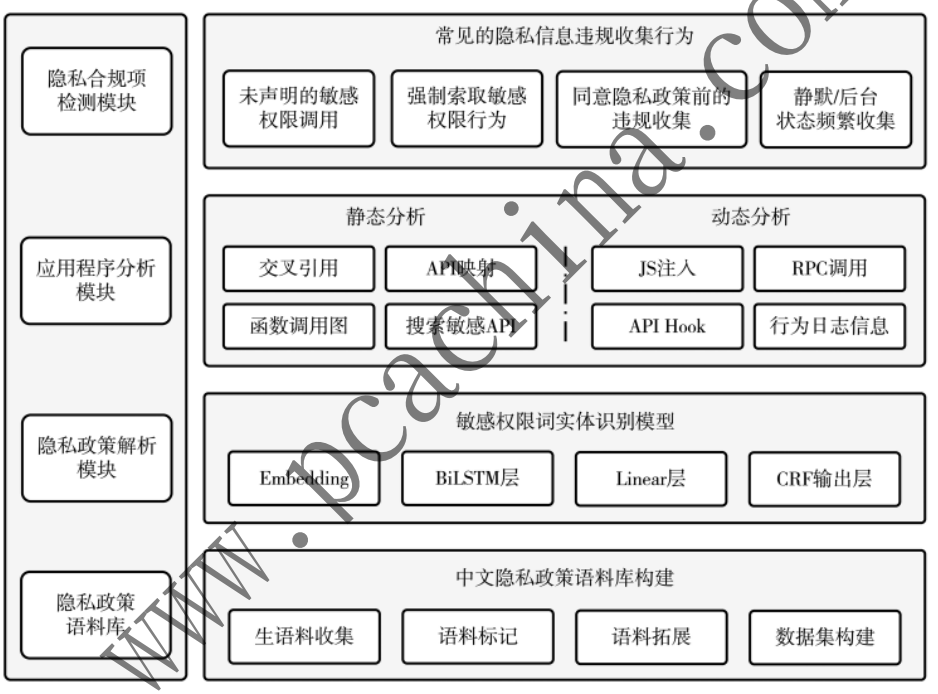

图 6 系统架构图

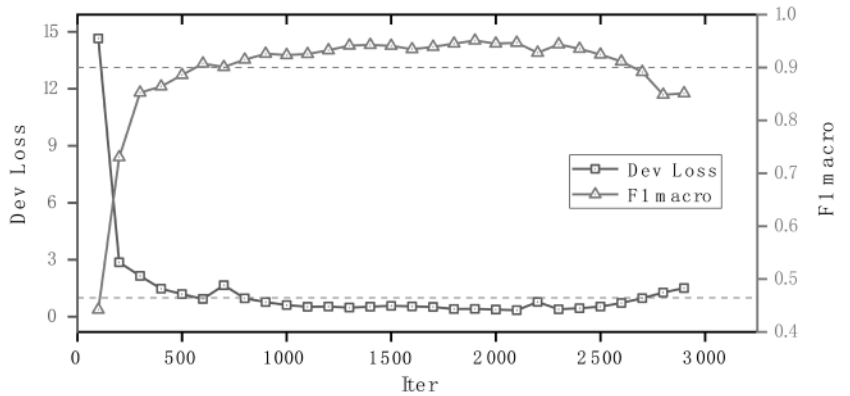

图 7 模型在验证集上的损失与 F1 值

表 5 最优模型在 9 类权限词实体识别下的测试结果

(%)

实体标签	Precision	Recall	F1-score
CALENDAR	87.50	92.11	89.74
CAMERA	90.20	92.00	91.09
CONTACTS	97.94	95.96	96.94
LOCATION	96.98	96.40	96.69
MICROPHONE	86.27	84.62	85.44
PHONE	97.29	92.16	94.66
SENSORS	75.00	100.00	85.71
SMS	96.00	100.00	97.96
STORAGE	92.70	85.63	89.02
AVERAGE	91.09	93.21	91.92

在检测能力测试部分,利用该系统对小米应用市场和 360 手机助手的 1 941 款应用进行合规检测,共计发现 52 款 App 存在隐私合规问题,包括 32 款应用未声明全部隐私权限,即存在隐私政策声明与权限调用不一致,5 款应用存在用户同意隐私政策前收集敏感信息,3 款应用无隐私政策声明,1 款应用的隐私政策存在冗余声明,1 款应用存在强制索取权限等。

5 结论

随着 App 违法违规收集使用个人信息专项治理工作的不断深入,监管部门、企业开发者以及个人用户对于 App 隐私合规检测的需求都不断增加。现有隐私合规检测产品由于忽略对隐私政策的评估,因而产生较高的误报率和漏报率。

该研究从实际需求出发,构建了安卓应用隐私政策与权限调用一致性合规检测系统,主要包括四个创新点:(1)构建首个中文隐私政策语料库并训练危险权限词实体识别模型,以平均 91.92% 的 F1 值完成了隐私政策危险权限词信息抽取;(2)提出轻量级、低开销的安卓应用权限调用动静态混合分析方案,通过对程序中的敏感 API 进行定位或重载,实现了程序实际调动危险权限的动静态检测;(3)构建半自动化的安卓应用隐私声明与权限调用一致性合规检测框架,通过将隐私声明权限集与程序实际调用的危险权限集进行一致性分析,识别未声明的权限调用与敏感信息收集行为;(4)针将强制索取权限、用户同意隐私政策前收集敏感信息、后台或静默状态下频繁收集敏感信息等子功能进行集成,实现对于主流违规行为的检测。

该方案设计还存在一定的改进空间:(1)本作品对于“隐私”的关注范畴仅限于应用程序通过危险权限获取到的敏感信息,但在实际应用场景中,仍存在部分隐私信息是用户通过应用的 UI 界面提供的,例如手机号码、身份证号码、搜索偏好等,对于这类数据的合规保护需要更多结合 UI 控件及数据流的污点分析方法进行研究;(2)在隐私政策解析工作中重点关注危险权限术语的实体识别,对于语义理解与句法分析未能展开深入探讨与研究。

综上所述,该研究在现阶段下很好地解决了国内 App 隐私合规检测所面临的技术挑战,有效弥补了隐私政策与程序分析两结合在一致性合规检测领域的空缺。同时,该研究构建的中文隐私政策语料库也能够为日后的相关研究提供基准数据集,具有重要现实意义和广阔应用前景,对 App 隐私合规的长效治理工作起到积极的促进作用。

参考文献

[1] 光明日报与武汉大学联合调研组.保障安全的“权利书”,还是窃取信息的“任意门”——App 隐私协议现状调查[N/OL].(2021-08-19).https://epaper.gmw.cn/gmrb/html/2021-08/19/nw.D110000gmrb_20210819_1-07.htm.

[2] GLUCK J, SCHAUB F, FRIEDMAN A, et al. How short is too short? Implications of length and framing on the effectiveness of privacy notices[C]//Twelfth Symposium on Usable Privacy and Security (SOUPS 2016), 2016: 321-340.

[3] MCDONALD A M, CRANOR L F. The cost of reading privacy policies[J]. A Journal of Law and Policy for the Information Society, 2008, 4(3): 543-568.

[4] 国家计算机网络应急技术处理协调中心. App 违法违规收集使用个人信息监测分析报告[EB/OL]. [2022-12-29]. https://www.cert.org.cn/publish/main/upload/File/APP%20abusing%20report.pdf.

[5] ARZT S, RASTHOFER S, FRITZ C, et al. FlowDroid: precise context, flow, field, object-sensitive and life-cycle-aware taint analysis for android apps[J]. ACM Sigplan Notices, 2014, 49(6): 259-269.

[6] LI L, BARTEL A, BISSYANDÉ T F, et al. IccTA: detecting inter-component privacy leaks in Android apps[C]// 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. IEEE, 2015: 280-291.

[7] ENCK W, GILBERT P, HAN S, et al. TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones[J]. ACM Transactions on Computer Systems(TOCS), 2014, 32(2): 1-29.

[8] ZHU D, JUNG J, SONG D, et al. TaintEraser: protecting sensitive data leaks using application-level taint tracking[J]. ACM Sigops Operating Systems Review, 2011, 45(1): 142-154.

[9] YANG Z, YUAN Z, JIN S, et al. FSAFlow: lightweight and fast dynamic path tracking and control for privacy protection on Android using hybrid analysis with state-reduction strategy[C]//Proceedings of the 43rd IEEE Symposium on Security and Privacy(SP). IEEE, San Francisco, 2022: 23-25.

[10] PANDITA R, XIAO X, YANG W, et al. WHYPER: towards automating risk assessment of mobile applications[C]//22nd USENIX Security Symposium, 2013: 527-542.

[11] QU Z, RASTOGI V, ZHANG X, et al. AutoCog: measuring the description-to-permission fidelity in Android applications[C]//Proceedings of the 2014 ACM Sigsac Conference on Computer and Communications Security, 2014: 1354-1365.

[12] SLAVIN R, WANG X, HOSSEINI M B, et al. Toward a framework for detecting privacy policy violations in Android application code[C]//Proceedings of the 38th International Conference on Software Engineering, 2016: 25-36.

[13] WANG X, QIN X, HOSSEINI M B, et al. Guileak: tracing privacy policy claims on user input data for Android applications[C]//Proceedings of the 40th International Conference on Software Engineering, 2018: 37-47.

[14] HARKOUS H, FAWAZ K, LEBRET R, et al. Polisis: automated analysis and presentation of privacy policies using deep learning[C]//27th USENIX Security Symposium, 2018: 531-548.

[15] ANDOW B, MAHMUD S Y, WANG W, et al. PolicyLint: investigating internal privacy policy contradictions on Google Play[C]//28th USENIX Security Symposium, 2019: 585-602.

[16] ANDOW B, MAHMUD S Y, WHITAKER J, et al. Actions speak louder than words: entity-sensitive privacy policy and data flow analysis with PoliCheck[C]//29th USENIX Security Symposium, 2020: 985-1002.

[17] NGUYEN T T, BACKES M, MARNAU N, et al. Share first, ask later(or never?) studying violations of GDPR's explicit consent in Android apps[C]//30th USENIX Security Symposium, 2021: 3667-3684.

[18] WILSON S, SCHAUB F, DARA A A, et al. The creation and analysis of a website privacy policy corpus[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016: 1330-1340.

[19] ZIMMECK S, STORY P, SMULLEN D, et al. Maps: scaling privacy compliance analysis to a million apps[C]//Proc. Priv. Enhancing Tech., 2019: 66.

[20] SATHYENDRA K M, WILSON S, SCHAUB F, et al. Identifying the provision of choices in privacy policy text[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017: 2774-2779.

[21] SANG E F, DE MEULDER F. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition[J]. arXiv preprint arXiv: cs/0306050, 2003.

[22] 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.

[23] BACKES M, BUGIEL S, DERR E, et al. On demystifying the android application framework: re-visiting Android permission specification analysis[C]//25th USENIX Security Symposium, 2016: 1101-1118.

(**收稿日期**:2022-12-29)

作者简介:

王申奥(2001-),**通信作者**,**男**,**本科**,**主要研究方向**:**新兴软件安全**、**数据安全**。E-mail:shenaowang@foxmail.com。

王亚龙(2001-),**男**,**本科**,**主要研究方向**:**软件安全**、**密码学**。

李晖(1968-),**通信作者**,**男**,**博士**,**教授**,**主要研究方向**:**密码信息安全**、**信息论与编码理论**。

版权声明

凡《网络安全与数据治理》录用的文章，如作者没有关于汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权等版权的特殊声明，即视作该文章署名作者同意将该文章的汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权授予本刊，本刊有权授权本刊合作数据库、合作媒体等合作伙伴使用。同时，本刊支付的稿酬已包含上述使用的费用，特此声明。

《网络安全与数据治理》编辑部