ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2000/00(00)-0000-00
doi: 10.3778/j.issn.1673-9418.

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-89056056

Android 应用隐私条例与敏感行为一致性检测*

王靖瑜[1+],徐明昆[1],王浩宇[2+],徐国爱[3]

1. 北京邮电大学 网络技术研究院(计算机科学与技术), 北京市 100876
2. 北京邮电大学 计算机学院, 北京市 100876
3. 北京邮电大学 网络空间安全学院, 北京市 100876

Automated Detection of the Inconsistence between App Behavior and Privacy Policy of Android Apps*

Wang Jingyu[1+], Xu Mingkun[1],Wang Haoyu[2+],Xu Guoai[3]

1.Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
2.School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
3.School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
Corresponding author : E-mail: haoyuwang@bupt.edu.cn

Wang JY. Automated Detection of the Inconsistence between App Behavior and Privacy Policy of Android Apps. Journal of Frontiers of Computer Science and Technology, 2000, 0(0): 1-000.

Abstract: Mobile apps frequently request access to sensitive information. Google recommended that developers should publish privacy policy document when uploading an app, that is, developers need to declare how the privacy information is used, collected, or shared, with the aim of making the user aware of how the privacy information is used for better protection users' privacy. Recent work showed that there is a huge gap between the app behavior and the privacy policy, thus many research studies focus on detecting the inconsistence between app behavior and the privacy policy. However, most of them only focus on static analysis and use white-list to identify third-party libraries, which is inaccurate and incomplete.

The National Natural Science Foundation of China Project under Grant No.61401038 and Grant No.61702045 (国家自然科学基金); the National High Technology Research and Development Program of China under Grant No.2015AA017202 (国家高技术研究发展计划项目); the Guangdong Provincial Science and Technology Department Frontier and Key Technology Innovation Project under Grant No.2016B010110002 (广东省科学技术厅前沿与关键技术创新项目);the State Grid Corporation of China Key Technology Innovation Project under Grant No.SGRIXTKJ[2017]265 (国家电网公司科技项目); the BUPT Youth Research and Innovation Program under Grant No.2017RC40 (北京邮电大学青年科研创新计划专项);Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia under Grant No.ITSM200601 (北京邮电大学智能通信软件与多媒体北京市重点实验室项目).
Received 2000-00, Accepted 2000-00.

An automated detection tool is proposed to check whether the app privacy document is consistent with the app behavior. First, an improved natural language approach is used to extract the declared sensitive behavior in the privacy policy. Then, use both static analysis and dynamic analysis to analyze the sensitive behavior of mobile app. Besides, a clustering-based approach is used to identify third-party library used in the app, which is more accurate than the traditional white-list based approach. Finally, conduct the consistence detection with the statement of privacy policy and the analysis of the privacy permission in code. Based on the experiment of 455 apps, the tool can accurately extract 94.75% of the privacy information in the privacy policy statement. Experiment results show that for roughly 50% of the apps, there exist inconsistence between app behavior and privacy policy.
Key words: privacy policy; permission; static analysis; dynamic analysis; the third party; mobile application

摘 要: 移动应用会频繁使用敏感信息，因此，Google 建议开发者在上传应用时发布隐私条例文档，即开发者需要声明隐私信息如何被使用、收集、或分享，其目的是使用户了解隐私信息如何被使用，从而更好的保护用户隐私。近期，许多研究表明应用的敏感行为和隐私条例存在巨大差异。尽管很多工作关注于隐私条例与应用行为的一致性分析，然而现有工作均使用静态分析和白名单分析第三方库的方法，导致隐私条例的一致性检测结果的不准确和不完整。

本文提出一种自动化检测应用隐私条例文档是否与应用行为相一致的工具。首先，使用一种改进的自然语言处理的方法提取隐私条例文档中的隐私信息和应用敏感行为；然后，使用静态分析和动态分析相结合的方法分析应用实际的隐私行为，同时区别于传统的白名单对照方式，使用了基于聚类的第三方库的检测方法提高了检查的准确性，最后将文本中声明的隐私信息行为和代码中分析出的隐私权限进行一致性校验。实验对 455 个应用进行分析，工具对隐私条例中隐私信息提取的准确率为 94.75%，大约有 50%的应用存在着应用行为和隐私条例文档不一致的问题。
关键词: 隐私条例:权限;静态分析;动态分析;第三方库;移动应用

文献标志码：A **中图分类号：**TP309

1 引言

在移动智能终端和多样的移动应用给用户带来便利的同时，移动平台上各种新的安全和隐私问题也日益凸显。当前的移动平台（尤其 Android 平台）上广泛存在权限滥用的问题。很多应用经常申请不必要的敏感权限，使用户隐私面临被泄露的风险。用户很多时候不了解应用是否需要使用权限以及为什么使用权限，很难对应用的权限进行管理。

Google 建议开发者在上传应用时发布隐私条例文档[1]，即开发者需要声明用户相关的隐私信息如何被使用、收集、或分享，其目的是使用户了解隐私信息如何被使用，从而更好的保护用户隐私。尽管应用开发者为了向用户说明软件潜在的隐私风险而发布相关的隐私条例，但是用户很难直观的判断文档的正确性，当应用收集或分享了隐私条例中声明之外的信息时，用户对此并不知情。相关研究表明[2,3,4]，很多应用行为与其隐私条例的不一致性存在滥用权限和隐私泄露的风险。Android 应用中大量使用第三方库，因此隐私条例中应该包含第三方库隐私信息的行为。然而，相关研究工作表明很多应用开发者对第三方库的行为并不了解[5]。一方面，第三方库一般不会提供源代码，并且数据的收集也非透明化。另一方面，很多第三方库存在越权行为[6]，仅通过第三方库的文档很难了解这些行为。此外，很多应用（包括一些恶意应用）会故意隐藏其敏感行为，通过编写虚假的隐私条例使用户相信该应用的安全性。美国对隐私信息的收集要求很严格，不准确的隐私条例会造成罚款，例如，FTC 将会对未获得父母同意收集孩子个人信息的应用罚款$800000[7]。因此，检测移动应用的隐私条例是否与应用行为相一致对于保护用户隐私至关重要。

近年来，有不少相关研究关注于隐私条例分析，然而准确进行隐私条例与应用行为的一致性分析还存在以下主要挑战：

1. 隐私条例文档中句子形式复杂多样，因而使用

非人工的手段分析和提取有用信息很困难的。若要提取出关键成分,需要使用自然语言处理方法生成句子间词语的依赖关系和层次关系，从中找出信息特征（主谓宾定状补）。同时，还要解决句子中关联词组、连词和状语成分等的提取问题，因为这些成分无法直接从句子的依赖关系和层次关系获得,需要再次进行分析。在之前的研究工作中[8]，没有对状语、连词和词组进行分析，会造成提取的信息不完整的现象。例如“we collect information about your device ID, phone number.”,若没有分析状语和词组，“about”修饰的“device ID”和“phone number”将无法提取出来，造成假阴性，使得实验结果中隐私条例不完整的比例增大。

2. 隐私条例中包含第三方库隐私信息的行为，之前研究工作通常使用白名单方法来检测第三方库。然而白名单方法覆盖率不全，现有工作标记的白名单远远少于可用的第三方库数量，并且不能应对代码混淆问题。研究表明，超过 50%的第三方库都存在不同程度的代码混淆[9]。不能准确检测第三方库会导致应用行为分析的不准确，从而导致一致性分析出现漏报的问题。
3. 现有的隐私条例一致性分析工作均使用静态分析来检测应用的敏感行为，然而静态分析在处理 Java 反射和动态加载时存在局限性，导致应用行为分析的不准确[10]。

为了解决这些挑战，本文首先使用一种改进的自然语言处理的方法对应用开发者编写的隐私条例文档进行分析，准确提取开发者声明的应用敏感行为。本文基于 Standford Parser 设计了提取特征的算法，并增加了连词、修饰宾语的状语和关联词组成分的分析，相对于之前研究工作，提取的信息更加完善，通过将提取出的信息进行归类，用于一致性分析。其次，使用静态分析和动态分析相结合的方法分析应用实际的隐私行为，提高了应用行为分析的准确性。此外，区别于传统的白名单对照方式，使用基于聚类的第三方库的检测方法提高了第三方库检测的准确性。通过在 455 个应用上进行实验表明，工具在隐私条例中提取隐私信息的准确率达到 94.75%，并检测出有 49.7%的应用存在隐私条例一致性的问题。

2 研究背景及相关工作

2.1 隐私条例

隐私条例包含的信息类型主要包括可体现用户身份的信息、应用程序运行时可能会使用或分享的个人信息、信息使用的目的性等[11]。例如，图 1 为应用 com.appsbar.JosephWeather 的隐私条例示例，示例中展示了应用中声明使用信息的方式。

Joseph Weather has created this ***privacy statement*** in order demonstrate our firm commitment to privacy. The following discloses the information gathering and dissemination practices for this web site.
声明隐私条例

This site ***uses*** your IP address to help diagnose problems with our server and to administer our site.
收集的信息

Since we ***do not collect*** any personal information on this site, we do ***not share*** any personal information that you do share with us should you do so with any third parties nor do we use any personal information for any purposes beyond responding to your contact.
不收集的信息

图 1 com.appsbar.JosephWeather 的隐私条例

2.2 问题定义

本文的研究目标为自动识别应用隐私条例与应用行为的不一致性，主要存在三种问题：

（1）不完整的隐私条例：良好的隐私条例应该包含应用使用的所有的隐私数据及其使用方式，否则为侵犯用户隐私。图 2 为 air.mwe.cookingcuteheartcupcakes 的应用隐私条例声明，在隐私条例中没有声明使用位置信息，但在实际行为中，在代码中调用 getLastKnownLocation()获取位置信息。

隐私条例：
We use third-party services to collect Non-personal and non-identifiable information (“Non-personal Information”). Non-personal Information which is being gathered consists of technical information and behavioral information, as detailed below:

Technical information:
• Type of operation system (e.g. Windows, Linux, etc.)
• Type of Browser (e.g. Internet Explorer, Firefox, Chrome, Safari, etc.)
• Screen resolution (e.g. 800×600, 1024×768, etc.)
• Language (e.g. English)
• Android or iOS Vendor (e.g. Galaxy, HTC, iPhone, etc.)
• Flash version

敏感行为：
android.location.LocationManager getLastKnownLocation()

图 2 应用 air.mwe.cookingcuteheartcupcake 的隐私条例描述和应用敏感行为

（2）不正确的隐私条例：不正确的隐私条例是指应用在隐私条例中声明不会收集、使用或分享某些隐私信息，但实际进行了这种行为。例如，图 3 中，应用 com.macropinch.hydra.android 的隐私条例声明不会使用位置信息“We will not collect personal information, including your geographic location information, names…”,然而却在代码中发现它调用了<android.location.LocationManager getLastKnownLocation()> 获取用户的位置信息。

隐私条例声明:
We will not collect personal information, including your ***geographic location information***, names, physical addresses…
敏感行为：
android.location.LocationManager getLastKnownLocation()

图 3 应用 com.macropinch.hydra.android 的隐私条例描述和应用敏感行为

（3）不一致的隐私条例：当应用声明不会使用某些隐私信息，但在第三方库中使用了这些隐私信息时，就产生了不一致问题。例如，图 4 中，应用 com.bestringtonesapps.cuteringtones 在隐私条例中声明“We do not collect information such as your name, address, phone number…”，但其使用的第三方库 Fabric 使用了 READ_PHONE_ STATE 权限。

隐私条例声明：
We do not collect information such as your name, address, ***phone number***…
敏感行为：
android.telephony.TelephonyManager listen()
第三方库： Fabric

图 4 com.bestringtonesapps.cuteringtones 的隐私条例描述及其使用的第三方库

2.3 相关工作

Rocky[4]等人使用拓扑图将关注隐私信息进行层次关联，将声明不确切的信息和没有声明的信息进行违规的程度判定，但是，应用行为分析过程中，静态分析过程中没有考虑内容提供商使用的隐私信息同时没有考虑到应用运行时加载和 Java 反射过程中的应用行为，且没有分析使用的第三方库。

Yu 等人[5,8]研究应用隐私条例文档及应用行为的不一致性问题，并为每个应用自动化生成相匹配的隐私条例文档。研究中使用 Standford Parser 获取隐私条例声明的信息，并使用静态分析获取应用行为。然而，在隐私条例分析时，他们没有进行连词和状语成分分析，会导致缺失部分文档中声明的隐私信息，降低检测的准确率。此外，在分析应用所使用的第三方库时，使用了白名单的方式，无法应对代码混淆以及覆盖率不高。此外，只使用静态分析的方式可能无法应对动态加载及 Java 反射机制，造成应用行为分析不准确，影响检测结果。

Primal 等人[12]对 Google Play 中的针对于 13 岁以下的青少年或儿童的应用进行隐私权限的分析，使用监控网络流量和运行时截图的方式获取其使用的敏感信息和交互行为，但却使用人工的方式将其与隐私条例进行匹配，过多的人工干预，使得准确率和效率都大幅下降。

本文使用自然语言处理工具来分析隐私条例信息，在相关研究工作[8]的基础上，增加了状语、连词和词组成分的分析，使得提取出的隐私条例信息更完善。此外，除了使用静态分析的方式检测应用行为，本文还使用动态分析的方法获取动态加载和 Java 反射产生的应用行为，同时使用基于聚类的第三方库检测工具检测应用中第三方库使用的隐私行为获取应用行为信息，解决白名单方式第三方库检测不全和无法获取第三方库实际应用行为的问题。最后，使用显示语义分析的方法将隐私条例信息和应用行为进行一致性判定。

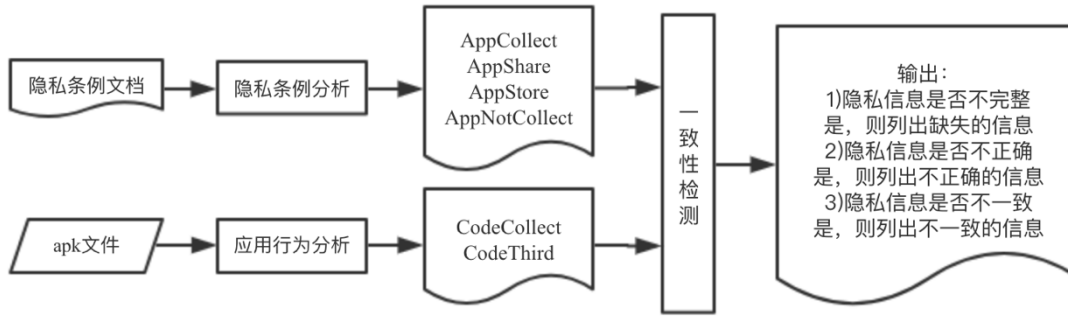

图 5 系统总体流程图

3 研究方法

3.1 总体介绍

如图 5 所示，对于每个应用，首先提取应用对应的隐私条例中信息，包括信息的类型以及信息的使用行为，然后对应用进行静态分析和动态测试获得应用实际的敏感行为，最后将二者分析的结果进行一致性检测，判断隐私条例是否存在不完整、不正确或者不一致的现象。

3.2 隐私条例分析

3.2.1 句子结构

图 6 一个常见的隐私条例的句子结构

如图 6 所示，隐私条例中的句子主要包含三个关键部分，包括执行者，动作和资源，其它的成分如条件（在什么条件下使用该信息）、目的（使用该信息的目的）、时间（什么时候使用）等是可选成分。

执行者：是收集、保存或分享隐私信息的实体，即应用开发者或第三方库开发者。执行者是动作的发出者，通常是主语。

动作：执行者的行为，例如：收集、保存或分享，如图 6 的“collect”。

资源：是动作的执行对象，即隐私信息的部分，在图 6 中为“your device information”。

3.2.2 动词分类

本文随机从谷歌应用商店中挑选出 50 个隐私条例文档，对其进行动词（行为）特征提取，并根据[13]中的结果，将隐私条例中常见的动词分为 3 类：

（1）Collect 动词：用于描述应用使用过程中对信息的收集和使用的行为，例如：collect、use 等。

（2）Store 动词：用于描述应用使用过程中对隐私信息进行保存的行为，例如：retain、store、save 等。

（3）Share 动词：用于描述应用使用过程中把隐私信息分享给第三方的行为，例如：share、disclose 等。

基于以上三种行为，我们将隐私条例中分析的结果分为 AppCollect，AppStore，AppShare 三种，分别代表中代码中收集、保存、分享的以及第三方库使用的隐私信息。同时，文档中可能存在否定句的成分，所以相反的，否定句的结果分为 AppNotCollect，AppNotStore，AppNotShare 三种，分别对应代码中不会收集、保存、分享的隐私信息。

3.2.3 隐私条例分析方法

隐私条例分析的目的是提取出隐私条例文档中声明的隐私信息及其行为，通过对隐私条例的主谓宾、状语、连词、词组和否定特征等分析，筛选出的信息进行归类。图 7 为该隐私条例分析方法。

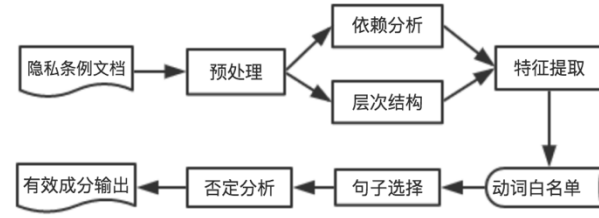

图 7 隐私条例分析

（1）预处理

该工具需要从文本中提取出隐私条例内容，然后将其切分为句子的形式。我们使用了 Beautiful Soup[14]将文本转化成 html 的格式，隐私条例文档中存在着大量的特殊符号，可能会在分析时产生错

误，所以需要将“：”、“；”、“.”等非 ASCII 符号替换成“.”加换行的形式，保证在使用 Standford Parser 进行分析时不会产生句子分割错误的现象。同时，我们将文本中的所有字符都转换为小写字母。

（2）语义分析

预处理后我们可以得到需要分析的隐私条例文档的内容。为了完成(4)，我们使用 Standford Parser[15]将文档中的例句“we share your information with the third party to improve service”转化成层次结构和依赖关系关系。

层次结构：图 8 为例句的层次结构。层次结构中的句子被拆分成短语，每一层代表一个短语，每一个单词和短语都分别有一个词性标签，常见的标签有：名词（NN）、动词（VB）、形容词（ADJ）、副词（ADV）、人称代词（PRP）等。图 8 中，动词为“share”，名词词组是“your information”。

```
(ROOT
  (S
    (NP (PRP we))
    (VP (VBP share)
      (NP (PRP$ your) (NN information))
      (PP (IN with)
        (NP (DT the) (JJ third) (NN party)))
      (S
        (VP (TO to)
          (VP (VB improve)
            (NP (NN service))))))
    (. .)))
```

图 8 层次结构分析

依赖关系：图 9 为例句的依赖关系图。依赖关系描述了句子中每个单词间的相关关系，常见的关系有：sbj 代表主语，root 代表代表依赖关系的根节点、nsubjpass 代表被动语态中的主语，cc 或 conj 代表并列关系（and，or 等）。图 9 中句子的根节点“share”，主语是“we”，宾语是“information”[16]。

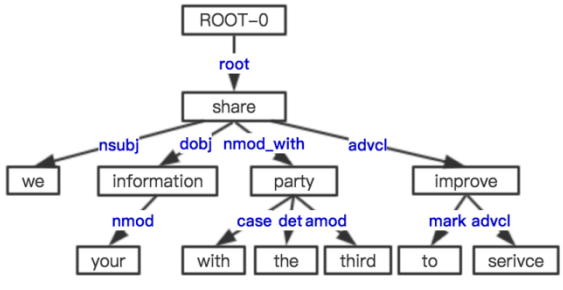

图 9 依赖关系分析

（3）特征提取

由于隐私条例文档中的内容包含多种信息，如应用介绍、联系方式等与本文无关的信息，同时隐私条例的编写没有固定的格式，所以句子冗余信息过多。因此，我们提取主要特征来简化句子结构。首先，利用依赖关系提取句子模式为“主语—谓语—宾语”的部分。我们对 50 个不同的隐私条例文档做了 TF-IDF 分析进行谓语特征的提取，然后筛选出最常见的动词 39 个动词入动词的白名单中，用于筛选隐私条例中的动作 注意这不是隐私行为同意改为动作，例如：collect，share，disclose，keep 等，进而可以剔除一些不相关的隐私行为（make，have 等）。其次，根据有效动词从依赖关系中提取其主语和宾语成分，构成主谓宾最短的句子模式；若句子的结构为主语—be allowed / accessed to+动词—宾语模式或者主语—be able to+动词—宾语模式，则根据依赖关系提取“xcomp”或“ccomp”成分作为实际的隐私信息行为，例如“we are allowed to collect your phone number”。为了过滤掉特征中的无意义词语，本文使用停用词列表[17]去除特征中的无用词,例如：your，our 等；对于宾语去掉于隐私信息分析无用的词，例如 service 等。

为了提高隐私信息提取的准确性，本文还进行状语、连词和词组成分的提取。

首先，部分隐私条例的隐私信息会出现在状语中，忽略状语的分析可能使得提取出的信息不完整。例如，facebook 的隐私条例其中一句是“we collect information about the purchase or transaction.”，只提取主谓宾的成分后获得的信息是“we collect information”，“information”是一个无意义的信息，真正修饰“information”的成分为 about 后的状语内容。为了解决这类问题，我们提取出宾语子树成分中含有“about”，“of”，“from”，“including”，“such as”等词为子树根节点的名词短语部分，放入资源列表中。

其次，相关研究工作[8]没有考虑连词成分的提取，导致信息提取不全。例如，应用 LeadBolt 的隐私条例其中一句为“We may also collect and store information locally on your device using mechanisms such as local storage identifiers.”，只提取主谓宾成分

表 1　6 种句子模式

编号	特征名	语义模型	例句
S1	主动语态	执行者+行为+资源	We will collect your phone number.
S2	被动语态	资源+行为	Your personal information will be used.
S3	被动 allowed/able 描述	执行者+be allowed / able to+行为+资源	We are allowed to store your location information.
S4	目的描述	执行者+行为+资源+to+行为+资源	We will use GPS to get your location.
S5	状语描述	执行者+行为+资源+介词+资源	We will collect your information about device Id.
S6	连词描述	执行者+行为+连词+行为+资源 或者 执行者+行为+资源+连词+资源	We will collect and share your address. Or We will collect your device information and location information.

后获得的信息是“we collect information”，无法提取出其连词成分 store，而 store 也是本文关注的隐私行为，因此可能会在一致性检测中造成不完整的问题发生。此外，名词并列情况如“And also you may choose to submit an e-mail address and password for account recover process(Log-in) and later communication.”，若不进行连词成分提取，将忽略信息“password”。因此，本文还根据句子依赖关系“cc”，“conj_”，提取出谓语和宾语的连词成分添加到资源列表和行为列表中。最后，有些常见的隐私信息并不是单一词语的形式，例如“device Id”，“phone number”等，若只根据依赖关系，只能提取出一个词“Id”，”number”，会造成一致性检测匹配失败造成错误提示不完整的隐私条例的问题发生。例如，应用“com.facebook.katana”中其中一条隐私条例为“you may choose to submit account information in order to connect your WhatsApp, LINE, Facebook”，若不做词组成分的分析，那本句提取出的宾语是 information，是停用词列表中词，那么本句将无隐私信息提出，造成假阴性；而使用词组成分分析，可以提取出“account information”，不会造成假阴性。因此，本文根据句子依赖关系“nn”，“compound”，“amod”对宾语列表中的宾语进行词组拼接解决以上问题。

（4）句子选择

本文中将提取出的句子模式分为 6 种模式，在隐私条例文档中，这些模式的句子将被提取出来，其余的将被过滤掉，不做进一步分析。表 1 中为 6 种句子模式的语义模型和例句。

S1 和 S2 模式的匹配：从句子的依赖关系中提取出根节点，判断其是否存在于动词筛选列表中，若是，则继续分析其相关的成分，否则，找到与根节点相关的连词成分和主语后，过滤掉该该动词。

S3 模式的匹配：从句子的依赖关系中提取出根节点，判断其是否是“allowed”或者“able”，若是，则找到其补语“xcomp”或者“ccomp”关系，判断该动词是否存在于动词筛选列表中，若是，继续分析相关成分，否则过滤掉该动词。

S4 模式的匹配：从句子的依赖关系中提取出根节点，然后找到其“xcomp”关系的引导词，判断两个动词是否在动词筛选列表中，若是，则继续分析相关成分，否则过滤掉该动词。

S5 模式的匹配：该模式主要用于提取出的资源的子树中的其它资源，当获取资源列表后，通过层次结构关系，判断其子树的根节点中是否包含“about”等介词，若有，则提取出子树中的名词和名词短语部分，添加到资源列表中。

S6 模式的匹配：用于句子中包含多个并列行为和资源的情况，从句子的依赖关系中提取出根节点，判断其是否存在于动词筛选列表中，然后检查是否有连词关系“conj_”,若有则进入筛选阶段，然后将其加入行为列表或资源列表中。

（5）否定分析

本文从两个方面判断句子是否具有否定含义：

a.资源为 nothing，例如“Nothing will be used”。

b.句子中根节点被否定含义词修饰，例如：“we will not/hardly collect information”，使用了否定词[17]列表来确定该句子是否具有否定含义。

```
'astrology.jyothishadeepthi.tamil.demo'---android.accounts.AccountManager.getAccountsByType
'astrology.jyothishadeepthi.tamil.demo'---type :com.google
'astrology.jyothishadeepthi.tamil.demo'---java.lang.Throwable
        at com.android.reverse.apimonitor.AccountManagerHook$2.descParam(AccountManagerHook.java:41)
        at com.android.reverse.apimonitor.AbstractBehaviorHookCallBack.beforeHookedMethod(AbstractBehaviorHookCallBack.java:13)
        at com.android.reverse.hook.MethodHookCallBack.beforeHookedMethod(MethodHookCallBack.java:12)
        at de.robv.android.xposed.XposedBridge.handleHookedMethod(XposedBridge.java:611)
        at android.accounts.AccountManager.getAccountsByType(Native Method)
        at astrology.jyothishadeepthi.tamil.demo.FlashMainActivity.getAccountNames(FlashMainActivity.java:1995)
        at astrology.jyothishadeepthi.tamil.demo.FlashMainActivity.login(FlashMainActivity.java:1138)
        at astrology.jyothishadeepthi.tamil.demo.FlashMainActivity.datevalidation(FlashMainActivity.java:1291)
        at astrology.jyothishadeepthi.tamil.demo.FlashMainActivity$1.run(FlashMainActivity.java:229)
        at java.lang.Thread.run(Thread.java:841)
```

图 10 astrology.jyothishadeepthi.tamil.demo.apk 的部分调用栈

3.3 应用行为分析

本文首先使用静态分析的方式获取敏感行为，然后动态分析的方式获取应用使用动态加载和 Java 反射使用的敏感行为，最后使用第三方库检测工具分析第三方库的行为。

3.3.1 静态分析

（1）反编译

在进行静态分析之前，首先应先获得该应用的代码，但通常情况下，应用的开发者不会公开其源代码，所以，本文使用 apktool[18]将输入 apk 文件反编译，获得应用的 smali 代码 AndroidManifest.xml，文件和一些其他的文件。

（2）权限提取

应用可以通过两种方式获取隐私权限。一种是调用敏感 API，例如调用 getCellLocation()来获取设备的位置信息；另一种是通过内容提供商获取应用使用的敏感信息，如调用 android.content.ContentResolver.query()，参数为 content://com.android.calendar 来获取日历信息。

本文中，为了确定应用中使用的隐私权限，我们从 Pscout[19][20]选取 89 个包含隐私信息的敏感 API，这些隐私信息为：device ID，subscriber ID，sim serial number，location，account，calendar，phone number，camera，audio，device version，message，log。这些隐私信息为隐私条例中的常见信息，也是用户最关注的信息。然后，从 smali 代码中与这些敏感 API 的特征（包名，方法名）进行对比，如果 smali 代码中存在一致的特征，则认为调用了该敏感 API，记录下该 API 使用的权限。同时，本文还从[19]中挑选了 12 个涉及隐私信息的 URI，若应用调用了这些 URI，我们根据该 URI 记录下其使用的权限信息。提取敏感 API 和 URI 的权限的同时，根据 Google Java Style[21]的驼峰式函数命名方式，可以将实际使用的权限信息和隐私信息进行映射，例如：getDeviceId()需要声明 READ_PHONE_STATE 权限来获取 device ID 信息，因此可以将 READ_PHONE_STATE 和 device ID 进行关联，同理，Pscout 将 content://contacts 映射为 android.permision.READ_CONTACTS，因此可以将其与“contact”关联。

3.3.2 动态分析

Android 应用中动态加载和 Java 反射机制使用的很普遍，恶意软件可以使用 Java 反射机制获取 Android 的隐藏 API 并获取到系统内部才能够调用的 API 和权限[22]，对用户安全造成极大威胁。而只使用静态分析的方式并不能检测出动态加载和 Java 反射过程中使用到的权限信息，因此，为了完整的获取应用使用的隐私权限，本文使用动态分析和静态分析的方式获取权限。

本文使用 Xposed 框架和 Droidbot[23][24]对应用进行动态测试。Droidbot 是一种基于应用 UI 界面的自动化测试工具，可以触发应用运行时的所有可能行为，包括 apk 在运行时的截图（用户点击行为、弹框等）、调用敏感 API 及其使用的权限、应用请求方式以及请求的 URL 等。此外，本文在 Droidbot 中添加应用运行时的调用栈，判断被调用的 API 的位置是第三方库还是应用本身。图 10 为应用 astrology.jyothishadeepthi.tamil.demo.apk 的部分调

用栈。将动态分析的结果和第三方库分析出的包名和权限比对，可筛选出程序中非第三方库产生的敏感行为，进一步与静态分析的结果相结合，可以获得应用中所有使用的敏感行为信息。

3.3.3 第三方库分析

Android 应用中经常使用第三方库，包括广告库、社交网络库、开发工具库等。但实际上，开发者对第三方库所使用的权限信息甚至功能并没有完整的了解。相关研究均使用白名单方法检测第三方库，然而白名单方法存在覆盖不完整和不能处理代码混淆等缺点。本文使用 Libradar[9]来替代白名单方式进行第三方库的分析。LibRadar 是一种基于聚类的第三方库检测方法，能够快速准确检测应用中使用的第三方库、类别、包名及其使用的权限等信息。此外，本文使用 Libradar 检测第三方库的实际应用行为而不使用第三方库声明的隐私条例的原因是：(1)第三方库声明的隐私条例也可能会出现与实际应用使用的隐私信息不一致，直接进行隐私条例的比较获得的结果不准确，(2)第三方库没有声明隐私条例的，将无法进行隐私条例不一致性的分析。

3.4 一致性检测

一致性检测目的是检测应用声明的隐私条例文档和应用实际的敏感行为是否存在不完整、不正确和不一致的问题。图 11 为一致性检测的流程，检测分为三方面：

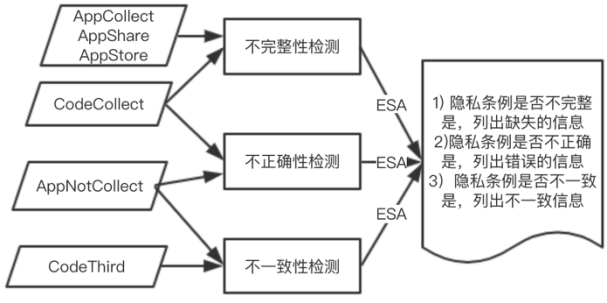

图 11 一致性检测模块流程图

(1)不完整性检测：将应用分析中应用本身的行为与隐私条例分析的使用的信息进行对比。

(2)不正确性检测：将应用分析中应用本身的行为与隐私条例分析的不使用的信息进行对比。

(3)不一致性检测：将应用分析中应用使用的第三方库与隐私条例分析的不使用的信息进行对比。

完成隐私条例分析和应用行为分析后，可以获得代码中(codeCollect,CodeThird)和隐私条例文档中声明的信息(AppCollect/AppNotCollect, AppShare,AppStore)作为一致性检测的对象。

3.4.1 不完整隐私条例检测

如果代码中检测出的隐私权限对应的隐私信息未在隐私条例文档中声明，则判定为不完整的隐私条例。确定应用行为分析中使用的权限(CodeCollect)是否出现在隐私条例文档的声明中(AppCollect, AppShare, AppStore)。若没有，则记录下该权限和信息内容。

匹配代表代码中的权限和隐私条例文档中的信息指的是相同的信息。但是代码分析中提取出的信息是根据使用的方法(getDeviceId())或权限映射(ACCESS_FINE_LOCATION 对应 location)[25]。二者的表达方式通常存在着很大的差异，本文使用 ESA[26]来比较两个文本间语义的相似程度，ESA 通过比较与词相关的维基文档的权重向量来计算相似度，每个维基概念都是由出现在这个文章中的词向量来表示，向量的矢量是通过 TFIDF 模型得出的权值，这些权值表明了词和概念之间联系的紧密度。因此，当两个文本间的相似度值到达一个阈值时，则认为这两个文本所指的是同一事物。本文中阈值设定为 0.5，在该阈值下通过人工确认，可以将大多数描述不同的同类信息匹配成功。

但是在进行代码中权限信息和隐私条例中信息匹配过程中，需要先确定隐私条例中声明的信息具体对应的权限，因此需要将该信息与权限映射的所有关键词（例如：ACCESS_FINE_LOCATION 映射关键词之一为 location）频繁匹配，通过 ESA 计算找到隐私条例声明信息的对应权限，使得效率大幅降低。因此本文对匹配模型优化，将进行相似度计算的信息的结果存储到权限对照表中，例如，格式为：phone number：READ_PHONE_STATE。当再次遇到该信息时，首先查找权限对照表中是否存在该信息，若存在直接提取出权限，若不存在再进行 ESA 计算文本间相似度获取相应权限。通过这种方法，可以将隐私条例中的信息与其对应的权限一一匹配。

然后，将代码中的权限与隐私条例进行比较，若代码中的权限不存在于隐私条例中，则记录下该权限，并判定为不完整；否则，则判定为一致，不存在不完整的问题。

3.4.2 不正确隐私条例检测

如果隐私条例中声明不会使用某隐私信息，但是却在代码中检测出来，则判定为不正确的隐私条例。将隐私条例中声明不会使用的信息通过 ESA 相似度计算或查找权限对照表的方式找到其对应的权限，然后遍历除第三方库外代码中使用的权限，若在代码中的权限中找到了该权限，则判定为不正确；否则，则判定为一致，不存在不正确的问题。

3.4.3 不一致隐私条例检测

如果隐私条例中声明不会使用某隐私信息，但是却在第三方库中检测出来，则判定为不一致的隐私条例。将 3.4.2 中隐私条例文档中的信息经过 ESA 计算后获取的权限的结果与代码中第三方库使用的信息进行匹配，如果第三方库使用了隐私条例中声明的不会使用的权限，则判定为不一致；否则，则判定为一致，不存在不一致的问题。

4 实验结果

实验对 Google Play 中所有类别的排名前 60 的应用共 2940 个进行了隐私条例的链接的爬取，其中 1765（60%）个应用存在隐私条例的链接，本文随机从中选取 455 个有隐私条例声明的应用作为本次实验的数据集。

4.1 隐私条例分析

为了验证隐私条例分析的准确率，我们从 120 个隐私条例文档中随机选取选取了 400 个句子进行分析，其中肯定句 300 个，否定句 100 个，总词数 8296 个。然后获取隐私条例分析的结果，进行人工检查，相关数据见表 2。

表 2 特征提取 400 个句子分析结果

种类	*句子数量*	*正确数量*	*正确率*
肯定句	300	282	94%
否定句	100	97	97%
总计	400	379	94.75%

以上结果表明，隐私条例的模块的句子分析平均的准确率为 94.75%。

(1)肯定句：有 18 个肯定句未能准确提取信息，其原因是 Standford Parser 未分析出词组形式的连词，例如，在“We may use log information as specifically indicated in the Mobile App, as well as to assess...” 例句中，连词是“as well as”，但被分析为“multi-word expression”，不是连词特征，无法提取出之后的行为及资源信息。

(2)否定句：有 3 个否定句未能准确提取信息，未提取出否定形式的隐私行为的原因时否定信息没有出现在主语谓语或者修饰谓语的副词中，而使用了合成形容词，“This non-personally-identifiable information may be shared with third-parties to provide more relevant services and advertisements to users.”例句中存在着否定含义的名词（宾语）“non-personally-identifiable”，在分析中没有将其归类到否定句中。

为了验证状语、连词和合并词成分提取的有效性，人工选取了 15 个隐私条例文档进行（1）提取状语、连词和词组成分（2）只提取状语和词组（3）只提取连词和词组（4）不提取状语、连词和词组的对比分析，图 12 为实验结果。

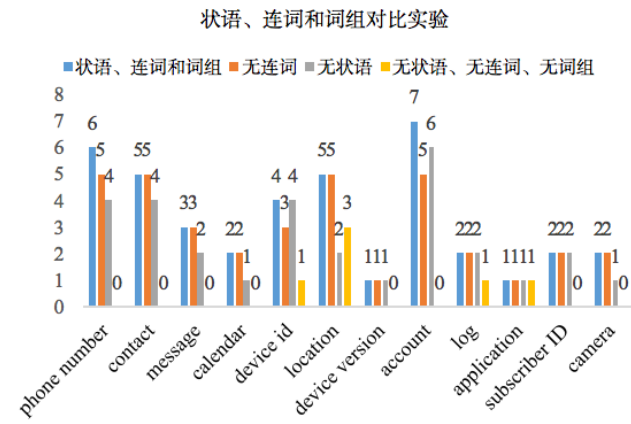

图 12 状语、连词和词组对比实验

数据表明 15 个隐私条例文档声明了 40 个隐私信息，缺失连词分析时可以提取出 36（90%）个信息，缺失状语分析时可以提取出 30 个（75%）隐私信息，缺失状语、连词和合并词分析只能提取出 6 个（15%）隐私信息。实验结果表明，在不分析状语、连词和词组的情况下[10]，只能提取出15%声明的信息，缺失了大部分的信息，而使用状

语、连词和词组分析后使得准确率大幅度提升。

4.2 应用行为分析

本文采用了动态分析和静态分析相结合的方式对 455 个应用行为进行分析，图 13 是静态分析方法检测出的隐私信息，使用静态分析共检测出 1158 个敏感信息，其中使用次数最多的两种信息分别是 location 和 deviceID。

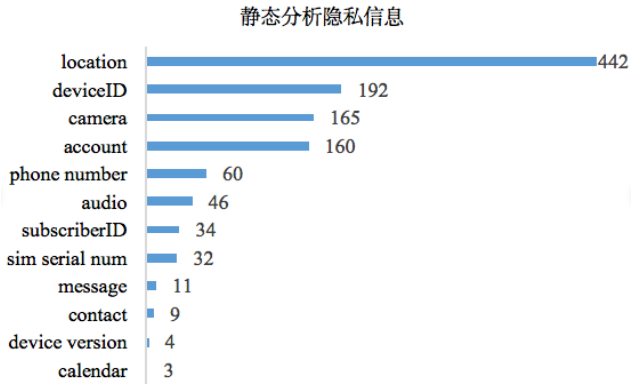

图 13 静态分析隐私信息

在动态分析中，使用 Droidbot 检测应用前 40 秒的动态行为，实验中共检测出 70 个动态运行时应用的权限信息。图 14 为动态分析使用的隐私信息（包括应用本身和第三方库使用的）。

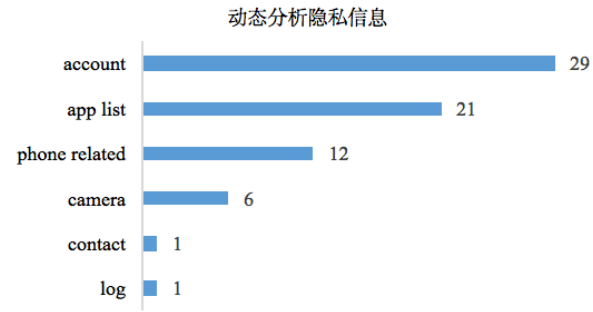

图 14 动态分析隐私信息

4.3 第三方库分析

本文对 455 个应用使用了第三方库检测工具 LibRadar 替代白名单的方式检测 apk 文件中，使用的隐私信息和权限的分析，同时，对分析结果进行统计。实验结果表明在 455 个应用中，有 430 个应用（94.5%）使用至少一个以上的第三方库。实验中共检测出第三方库的类型有 11 种（其中一种没有归类，使用“Unknown”替代），455 个应用中每一类被使用的具体数量如图 15 所示。如果使用白名单方式，只检测最常见的三类(Development Aid, Advertisement, Social Network)第三方库，只能检测出 1576 个（59.1%）第三方库，无法分析其他类别及混淆的第三方库。

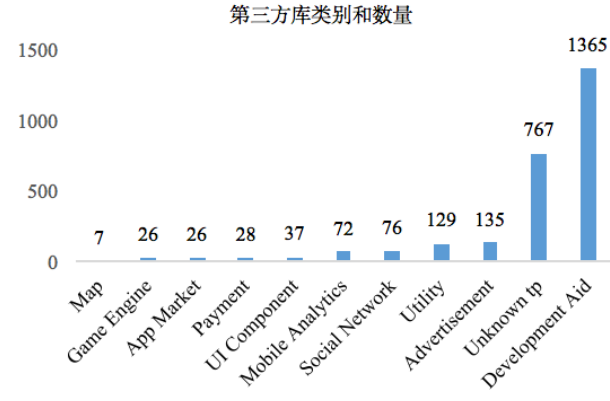

图 15 第三方库类别和数量

4.4 一致性检测

4.4.1 不完整隐私条例信息

该工具对 455 个应用进行了不完整隐私条例的检测，共检测出 224 个不完整的隐私条例，图 16 中列出了缺失的权限及其对应的有不完整的隐私条例的问题的应用个数，通过人工方式从中挑选出 150 个进行准确性判断，发现出现了 12 例误判现象。

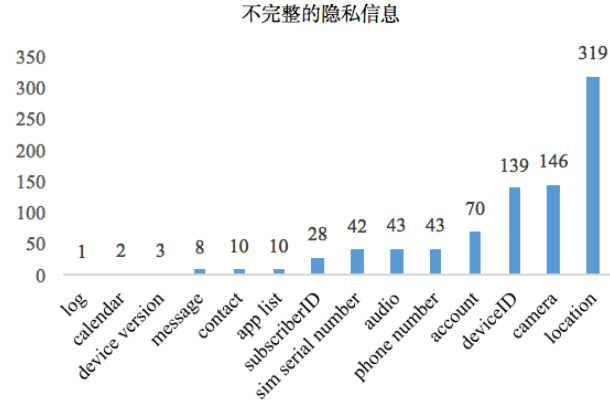

图 16 不完整的信息

假阴性：实验中的第一种假阴性是由于 Standford Parser 中依赖结构分析错误导致，例如：“We use the email address you upload…”，错误的将 email 的定位为 address 的宾语，将 address 定位为动词；第二种情况是因为动词白名单中缺少该动词导致，如“we ask for your personal information such as….”，“ask for”未在白名单中，导致声明的隐私信息未被提取出来。

假阳性：实验中未发现该类型错误。

4.4.2 不正确隐私条例信息

本文中共检测出 4 个不正确的隐私条例，经过人工的校验，发现这些隐私条例都是不一致的隐私条例，没有出现误判的情况。表 3 中列举了实验中不正确的隐私条例对应的应用及信息。

表 3 不正确应用及隐私信息

应用名称	不正确隐私信息
aws.apps.networkInfoIi	device ID
com.bestringtonesapps.cuteringtones	device ID,phone number
imoblife.toolbox.full	phone number
com.macropinch.hydra.android	location

4.4.3 不一致隐私条例信息

本文共检测出 2 个不一致的隐私条例，应用为 com.bestringtonesapps.cuteringtones 和 com.nimblebit.vegas 使用的权限为 READ_PHONE_STATE，使用该权限的第三方库分别是 Fabric 和 Javax，这些不一致的应用中，经过人工的校验，发现这些隐私条例都是不一致的隐私条例，没有出现误判的情况。

4.4.4 结果分析

本文从 4 个方面论述 Android 应用隐私条例与敏感行为一致性检测方法的准确性，首先，使用隐私条例分析检验 400 个隐私条例达到 94.75%的准确率，其次，使用动态分析的方式检测出 70 个动态加载或 Java 反射触发的隐私行为，然后，对 455 个应用中使用的第三方库进行分析，实验结果表明 94.5%的应用使用了第三方库，最后，通过对 455 个应用的一致性检测发现其中有 224 个不完整的隐私条例，4 个不正确的隐私条例和 2 个不一致的隐私条例，经人工抽样检测，准确率达到了 92.3%。

5 讨论

本文使用 Standford Parser 处理复杂的隐私条例文档，提取主谓宾、状语、连词、词组等特征筛选隐私信息及其行为，然后将其与应用分析结果进行比对，分析隐私条例中是否存在不完整、不正确和不一致三种问题。但是，为了提高分析的准确性隐私条例分析方法可以从 2 个方面进一步进行优化：

（1）将隐私条例中复杂的条件状语进行分析：比如“if you don’t agree”，“without your permission”等，这些条件可能会影响句子的实际含义。

（2）将隐私条例中的从句进行分析：实验中，大量假阳性和假阴性的例子是由 Standford Parser 无法分析无引导词的定语从句或宾语从句造成的，比如“Information gathered through cookies and Web server logs may include the date…”，分析时会将“gathered”当成 root 节点，导致依赖关系错误，所以应增加对从句中有效成分的分析。

6 总结

本文提出一种对 Android 应用隐私条例敏感行为一致性的检测方式，能够检测隐私条例中存在的三种问题：不完整、不准确和不一致性问题。本文在隐私条例分析时增加了对状语、连词和词组成分的分析，提高了隐私信息提取的准确性。此外，在静态分析的基础上，使用动态分析和第三方库检测工具分别获得应用使用的动态行为和第三方库使用的权限，使得提取出的应用行为更完善。其中，隐私条例分析能达到 94.75%的准确率，代码分析较传统的方式进行优化并增加动态分析，使得结果更加准确和完整，共检测出 70 个应用使用的隐私信息使用动态加载和 Java 反射。在第三方库检测中，检测到 94.5%应用使用 11 种类型的第三方库。针对 455 个流行应用的分析表明，49.7%的应用的隐私条例存在不一致问题。

References：

[1] “Play 管理中心帮助”, https://support.google.com/googleplay/android-developer/answer/113469?hl=zh-Hans.

[2] Path social networking app settles ftc charges it deceived consumers and improperly collected personal information from users’ mobile address books.https://goo.gl/Z31BAU, 2013.

[3] “Mobile malware evolution 2016”, https://securelist.com/analysis/kaspersky-security-bulletin/77681/mobile-malware-evolution-2016/

[4] Slavin R, Wang X, Hosseini M B, et al. Toward a frame-

work for detecting privacy policy violations in android application code[C] //Proceedings of the 38th International Conference on Software Engineering. ACM, 2016: 25-36.

[5] Yu L, Zhang T, Luo X, et al. Autoppg: Towards automatic generation of privacy policy for android applications[C]//Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices. ACM, 2015: 39-50.

[6] Liu B, Liu B, Jin H, et al. Efficient privilege de-escalation for ad libraries in mobile apps[C]//Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services. ACM, 2015: 89-103.

[7] “Path Social Networking App Settles FTC Charges it Deceived Consumers and Improperly Collected Personal Information from Users' Mobile Address Books”, https://www.ftc.gov/news-events/press-releases/2013/02/path-social-networking-app-settles-ftc-charges-it-deceived.

[8] Yu L, Luo X, Liu X, et al. Can We Trust the Privacy Policies of Android Apps?[C]//Dependable Systems and Networks (DSN), 2016 46th Annual IEEE/IFIP International Conference on. IEEE, 2016: 538-549.

[9] Ma Z, Wang H, Guo Y, et al. Libradar: Fast and accurate detection of third-party libraries in android apps[C]//Proceedings of the 38th International Conference on Software Engineering Companion. ACM, 2016: 653-656.

[10] Wang H, Guo Y, Tang Z, et al. Reevaluating Android permission gaps with static and dynamic analysis[C]//Global Communications Conference (GLOBECOM), 2015 IEEE. IEEE, 2015: 1-6.

[11] “Privacy Policy”, https://www.ftc.gov/site-information/privacy-policy.

[12] Reyes I, Wiesekera P, Razaghpanah A, et al. " Is Our Children's Apps Learning?" Automatically Detecting COPPA Violations[J]. 2017.

[13] T. Breaux, H. Hibshi, and A. Rao, “Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements,” *Requirements Engineering*, vol. 19, no. 3, 2014.

[14] Richardson L. Beautiful soup[J]. Crummy: The Site, 2013.

[15] D. Cer, M. Marneffe, D. Jurafsky, and C. Manning, “Parsing to stanford dependencies: Trade-offs between speed and accuracy,” in *Proc. LREC*, 2010.

[16] Stanford Parser, “Stanford typed dependencies manual,” http://nlp. stanford.edu/software/dependencies manual.pdf.

[17] “Negative vocabulary word list,” http://goo.gl/qX7UtK.

[18] Winsniewski R. Android–apktool: A tool for reverse engineering android apk files[J]. 2012.

[19] 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. ACM, 2012: 217-228.

[20] S. Rasthofer, S. Arzt, and E. Bodden. A machine-learning approach for classifying and categorizing android sources and sinks. In Proc. NDSS, 2014.

[21] “Google Java style ”, https://google.github.io/styleguide/Javaguide.html

[22] Shao Y, Ott J, Chen Q A, et al. Kratos: Discovering inconsistent security policy enforcement in the android framework[C]//Proc. of ISOC NDSS. 2016.

[23] Zweng J. Xposed Module NFC AID Re-Routinghttps[J]. github. com/johnzweng/XposedModifyAidRouting, 2013.

[24] lynnlyc. 2016. DroidBot: Automatic testing of apps in DroidBox. https://github.com/lynnlyc/droidbot. (2016). Accessed: 2016-03-10.

[25] “Manifest permission“，https://developer.android.com/reference/android/Manifest.permission.html

[26] Gabrilovich E, Markovitch S. Computing semantic relatedness using wikipedia-based explicit semantic analysis[C]//IJcAI. 2007, 7: 1606-1611.

Jingyu Wang was born in 1993. She is pursing the Master degree at Beijing University of Posts and Telecommunications. Her research interests include mobile security.

王靖瑜(1993–), 女, 黑龙江省牡丹江人, 目前在北京邮电大学攻读硕士学位, 专业为计算机科学与技术，主要研究领域为移动安全。

Mingkun Xu was born in 1963. He received the Master degree in Software Engineer fromXi'an Jiaotong University in1986. He is a senior engineer at Beijing University of Posts and Telecommunications. His research interests include cloud computing..

徐明昆（1963–），男，1986 年从西安交通大学获得软件工程硕士学位，目前为北京邮电大学高级工程师，主要研究领域为云计算。

Haoyu Wang was born in 1991. He received the Ph.D degree in Computer Science from Peking University in 2016. He is a lecturer at Beijing University of Posts and Telecommunications. His research interests include mobile security, software engineering. etc.

王浩宇（1991–），男，河南省周口人，2016 年从北京大学获得博士学位，目前为北京邮电大学讲师，主要研究领域为移动安全，软件工程。

Guoai Xu was born in 1972. He received the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications in 2002. He is a professor at Beijing University of Posts and Telecommunications. His research interests include software security, mobile security, etc.

徐国爱（1972–），男，江西鄱阳人，2002 年从北京邮电大学获得博士学位，目前为北京邮电大学教授，主要研究领域为软件安全，移动安全。