多维度的安卓应用相似度分析

张希远[1,2] 张 刚[3] 沈立炜[1,2] 彭 鑫[1,2] 赵文耘[1,2]
(复旦大学软件学院 上海 201203)[1] (上海市数据科学重点实验室(复旦大学) 上海 201203)[2]
(上海理工大学光电信息与计算机工程学院 上海 200093)[3]

摘 要 基于安卓的智能设备的普及和移动互联网的发展带来了安卓应用的繁荣,但同时也带来了移动应用的开发、维护、安全等方面的问题。采取了多种技术,提取了安卓应用的功能描述、权限声明及源代码,并基于这些信息对1173个安卓应用进行了统计分析、相似度计算、聚类以及交叉对比;利用多个维度的安卓应用特征相似度分析,初步得到了安卓应用多个维度的相关规律,其可辅助不同的安卓应用的开发和管理任务,如权限过度声明检测、重打包检测、应用描述完善、领域内的公共类库的发现和提取等,从而帮助改善安卓市场的生态并提高安卓应用的开发效率。

关键词 安卓应用,功能描述,权限,源代码,相似度

中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.3.037

Similarity Analysis of Multi-dimension Features of Android Applications

ZHANG Xi-yuan[1,2] ZHANG Gang[3] SHEN Li-wei[1,2] PENG Xin[1,2] ZHAO Wen-yun[1,2]
(School of Software, Fudan University, Shanghai 201203, China)[1]
(Shanghai Key Laboratory of Data Science, Fudan University, Shanghai 201203, China)[2]
(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)[3]

Abstract The popularity of Android smart device and the development of mobile Internet bring prosperity to the Android application. But it also brings problem of development, maintenance and security about mobile application. This paper adopted a variety of techniques and extracted the features of Android applications including functional description, permission and source code, and performed statistical analysis, similarity calculation, clustering and cross-comparison on the information of 1173 Android applications. Through similarity analysis on three dimensions of features, we obtained some related regular pattern, which can assist kinds of development and management tasks on Android application such as excessive permission detection, re-packed detection, application description improvement, class library discovery and extraction of certain domain. Thereby it can help improve the ecology of Android market and improve the development efficiency of Android application.

Keywords Android applications, Functional description, Permission, Source code, Similarity

1 绪论

智能手机的安卓操作系统(Android)自2007年发布以来迅速占据了移动终端平台的主导地位[1],其遥遥领先的市场占有率也激发了安卓应用的蓬勃发展。安卓的开放性给第三方开发商提供了一个十分自由宽松的环境,任何开发人员都可以自主开发安卓手机应用并发布到网络上,但是这也带来了一系列的问题。首先,不同应用市场对应用的分类、描述等存在很大差异,某些应用的描述并不能真正说明应用的功能和提示用户所需的权限;更有一些人员为了盈利对一些应用稍作修改,在嵌入广告或者恶意代码后重新打包,将其上传到了安卓应用市场中[2]。其次,自由的开发环境也导致一些恶意软件的产生,这些恶意软件会做出盗取用户隐私、恶意扣费等行为,从而侵害用户的权益,存在着巨大的安全隐患[3]。此外,虽然安卓系统已经试图通过权限设置来保证系统的安全性,并且规定应用必须在安装时声明所需要使用的系统权限才能执行相应权限范围内的操作,但是一些开发人员为了贪图方便而过度声明权限,这样可能会导致一些安全隐患。

考虑到安卓应用的描述、权限声明、代码等特征可以看作是从不同的侧面对同一应用的描述,它们之间可能存在某些内在关联。例如,文献[4]针对文本描述和权限之间的对应关系进行了研究,使用关系模型有助于发现权限过度声明的情

到稿日期:2015-01-26 返修日期:2015-06-04 本文受国家"863"高技术研究发展计划项目(2013AA01A605),国家自然科学基金:安卓应用开发中模式驱动的代码推荐与完成技术研究(61402113)资助。

张希远(1990-),女,硕士生,主要研究方向为软件工程,E-mail:zxiyuan@outlook.com;张 刚(1976-),男,博士,高级工程师,CCF会员,主要研究方向为软件开发方法和软件维护;沈立炜(1982-),男,博士,副教授,主要研究方向为软件产品线与自适应系统;彭 鑫(1979-),男,博士,副教授,博士生导师,CCF高级会员,主要研究方向为自适应软件、软件维护与演化、软件产品线等;赵文耘(1964-),男,博士,教授,博士生导师,CCF高级会员,主要研究方向为软件工程、软件开发工具及其环境、企业应用集成(EAI)。

况。本文采取了经验研究的方式，从应用市场上收集各类应用及其相关信息，包括应用的功能描述、源文件等，并解析 Androidmanifest. xml 文件获取的权限信息。通过自然语言处理、相似度计算、代码克隆检测等技术分别对功能描述、权限信息和源文件进行分析，获得不同应用之间功能描述、权限、代码 3 个维度的相似度关系。通过不同维度相似度之间的交叉比较，从中找出安卓应用关于功能、权限及代码的相关规律，从而为开发和进一步的研究提供帮助。本文的主要工作与贡献包括：

1)采取了经验研究的方式，对 1173 个安卓应用从功能描述、权限信息和源代码 3 个维度进行了交叉对比研究。

2)发现应用的功能描述和权限之间存在一定的关联。尽管应用的功能相似其隐含权限也应该相似，但是在实际的安卓市场中，功能相似的应用之间其权限并不一定相似；功能相似的某类应用权限声明局限于某一范围中，因此可以通过分析某个应用声明的权限是否在其所处类别的通常声明的权限范围内，来研究应用权限声明的合理性与安全性，寻找可能存在安全隐患的应用。

3)发现功能描述相似与代码相似之间不存在必然联系，功能相似的应用代码不一定相似，功能不相似的代码也可能调用相同的第三方库；通过模块相似度比较可以找出被经常使用但尚未被规范化的代码块，这些模块可规范为第三方库以简化安卓应用开发。

本文第 2 节介绍了整体研究方案以及功能描述、权限、代码 3 个维度的相似度分析方法；第 3 节对 3 个维度进行交叉分析和经验研究；第 4 节介绍安卓系统与安卓应用的相关研究；最后对全文进行总结与展望。

2 相似度分析方法

本节将详细说明多维度安卓应用相似度分析的各个步骤，包括信息采集、各维度相似度计算方法。

2.1 总体方案设计

本文利用自然语言处理、相似度计算、代码克隆检测等技术，从功能描述、权限、代码 3 个维度对安卓应用进行相似度分析。整体方案设计如图 1 所示，主要包括以下模块：

1)爬虫模块。从开源安卓应用市场 F-Droid[5] 采集安卓应用的功能描述信息及源代码，并从 Android 官方网站上收集最新版本的系统权限列表[6]存储到数据库中。

2)功能描述预处理模块。采集的功能描述使用的是自然语言描述，因而需要对功能描述进行分词、词干化处理、消除停止词与噪音词等自然语言处理，以利于功能描述相似度计算。

3)权限解析预处理模块。对源代码中 AndroidManifest. xml 文件进行解析，获取应用对应的权限，将其以二进制字符串形式存放在数据库中，以方便比较。

4)功能描述相似度计算模块。通过基于 WordNet[7] 的单词相似度 Wup 算法[8] 与带有权重的功能描述相似度计算方法计算应用之间的功能描述相似度。

5)权限相似度计算模块。利用 Jaccard 相似度系数[9] 计算应用之间的权限相似度。

6)代码相似度计算模块。利用 CCFinder 工具[10] 找出不同应用之间相似的代码段，并计算应用之间整体代码相似度和应用中的各功能模块代码相似度。

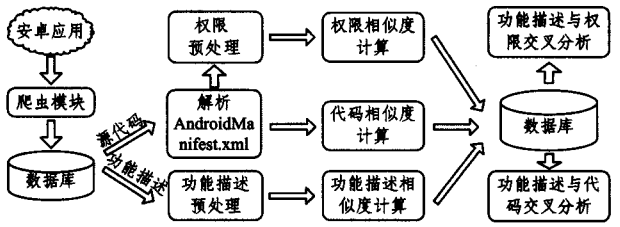

图 1 本文总体方案设计

本文设计了相应的实验，通过实际比对的方式对功能描述、权限以及代码 3 个维度的相似度进行了交叉分析和研究。

2.2 应用及信息采集

为了从多维度进行安卓应用相似度分析，需要收集各类应用及相关信息作为分析研究的基础数据。目前安卓应用市场比较丰富，除 Google 官方安卓应用市场 Google Play[11] 外，更有安卓市场、豌豆荚、应用汇、Apktop 等应用市场。不同应用市场提供的应用信息存在较大差异；通过观察发现，存在这样几项公共的信息类别：功能描述、Apk 或源文件以及权限信息。功能描述是对该应用主要功能的概括与简介，是决定用户下载与否的首要观察信息。Apk 或源文件是用户下载的信息，Apk 是安卓应用安装文件的统一后缀名，用户通过下载 Apk 文件将应用安装到自己的手机上，一些开源应用市场则提供了源文件。权限信息存储在 AndroidManifest. xml 文件中，AndroidManifest. xml 文件需要从 Apk 文件或源文件中提取。在分析比较了各类安卓应用市场后，为了方便对代码相似度进行研究，本文选择了开源安卓应用市场 F-Droid 中的应用及其信息作为数据来源，并且选择应用数量最多的英语语言应用作为采集对象。

数据采集通过爬虫模块从对应开源应用市场采集。爬虫模块将收集到的应用名称、功能描述信息、下载链接等相关信息存入应用采集数据库中，根据下载链接批量下载对应源代码压缩文件，并解压至相应路径下。

2.3 数据预处理

通过爬虫模块采集到的原数据需要经过一系列的预处理过程才能作为计算相似度的基础数据，主要包括对功能描述信息和权限信息的预处理。

2.3.1 功能描述预处理

功能描述信息有介绍应用功能、提示用户等作用。本文通过自然语言处理技术对功能描述进行预处理，以保证功能描述相似度计算的准确性。其主要过程分为以下步骤：分词、词形还原、去掉停止词以及排查自定义词汇，下面将分别说明。

计算两段文字的相似性主要通过相同词汇的出现对比或潜在语义检索等方式进行计算，这些处理方式都需要对文字中每个单词进行相关操作。因此预处理模块首先对功能描述进行特殊符号处理，利用正则表达式将非英文字母字符用空格替代，通过这种方式排除非英文字符，之后按照空格进行分词，将功能描述段落转换成词汇组形式。

其次，为了准确表达应用功能，方便应用之间的关联比较，需要对分词后的每个单词进行词形还原，如与游戏应用相关的 play、playing、plays 等词都会被还原成 play，在保证语义不变的同时进行统一化处理，以方便应用之间进行横向比较。

再次，描述预处理模块还需要处理停止词。英语语法中

存在大量代词、量词等停止词，如 it、that、a、an 等，这些词汇与表述应用功能不存在实际关联，会影响相似度计算的准确性。因此，预处理模块参照停止词汇表去除功能描述中的停止词，从而使功能描述完全由名词、动词等体现应用功能的单词组进行表示。

最后，针对应用名称等开发人员自定义的单词而言，其对于应用功能的表示不具有普遍性，也需要进行排除。描述预处理模块通过使用自然语言处理工具 WordNet 语义网进行排除。WordNet 是由 Princeton 大学的心理学家、语言学家和计算机工程师联合设计的一种基于认知语言学的英语词典，并且该词典也在不停扩展新增词汇，将应用描述中的单词与 WordNet 中的词汇进行匹配，若某个单词没有被收录在 WordNet 中，则认为其是自定义单词，通过这种方式能有效排除不利于相似度计算的自定义词汇。

经过上述一系列预处理工作，能够获得充分代表应用功能的词汇组，如表 1 所列。

表 1　应用功能描述预处理

应用	功能描述	功能描述(预处理后)
0x benchmark	Puzzle game, Port of the 2048 game by Gabriele Cirulli. It's playable without network connection.	puzzle game port game playable network connection
A Time Tracker	Time Tracker, Easily start/stop time tracking for any tasks. Offers summary report view and export.	time tracker easily start stop time track task offer summary report view export
Addi	Math calculation environment, Addi is a mathematical computing environment like Matlab and Octave, but made to work on Android devices., The goal is for 100% compliance with how Octave works including a compatibility mode that makes the behavior more similar to Matlab (already very similar).	math calculation environment mathematical compute environment octave make work android device goal compliance octave work include compatibility mode make behavior similar

2.3.2　权限信息预处理

为了确保系统的安全性，安卓系统通过一系列权限来限制应用对某些敏感资源的访问，所有涉及到系统中可能带来安全问题的敏感操作都必须通过权限声明，即在 AndroidManifest. xml 文件 uses-permission 标签中列出，并且允许第三方开发商自定义权限来进行安全保护，涉及到未声明权限的操作将被禁止。这些权限声明会在安装应用时呈现给用户，让用户对应用进行授权。本文根据安卓官方网站收集最新版本的系统权限列表，当前安卓提供的系统权限总数为 151 个。权限预处理模块提取 AndroidManifest. xml 文件中 uses-permission 标签中声明的权限与权限列表进行匹配，得到有关权限列表的二进制字符串，0 表示未声明权限，1 表示该权限被声明，使用 151 位的二进制字符串可以统一、高效地表示安卓应用声明的权限列表，方便横向比较。

2.4　多维度相似度计算

本文通过计算功能描述相似度、权限相似度以及代码相似度 3 个维度对安卓应用进行交叉分析，从而找出其潜在规律。本节将详细介绍 3 个维度的相似度计算方式。

2.4.1　功能描述相似度

功能描述相似度表征应用之间功能的相似性。由于在自然语言中存在许多一词多义、同义词、近义词等情况，因此忽略语义单纯对功能描述中的单词进行匹配和比较很难准确评估两个应用之间的功能相似性。

为了确保功能描述的相似度能够准确表现应用之间的功能相似关系，本文利用 WordNet 与 ws4j(Wordnet Similarity for Java)中的 Wup 算法来进行单词之间的语义相似度计算。WordNet 不仅是一个电子英语词典，更是一个覆盖范围宽广的英语词汇语义网。名词、动词、形容词和副词各自被组织成一个同义词的网络，每个同义词集合都代表一个基本的语义概念，并且这些集合之间也由各种关系连接，比如上/下位关系、整体局部关系等。Wup 算法通过利用两个单词的词义在 WordNet 分类学上的深度与它们词义的最长公共子序列在分类学上的深度进行计算，得出两个单词之间的相关性系数。该系数为 0 到 1 之间的小数，系数值越高代表两个词之间的相关性越高，反之则表示相关性越低；该系数能够有效表示同义词之间的近似关系，如 fight 和 struggle 都有“战斗”的语义，单纯使用单词匹配将无法识别这两个词的相关性，而使用 Wup 算法对两个词的所有语义进行计算可得到相关性系数最大值为 1，表示这两个词代表着相同的语义。此外，一些同类别词汇、上/下位词、整体局部关系词汇通过 Wup 算法也能得到较高的相关性系数。

为了计算应用与应用之间功能描述的整体相似度，需要综合计算组成功能描述的单词间的相关性系数。为了提升不同应用所有单词之间相似度系数的计算效率，相同单词相似度仅计算 1 次，并记录在缓存列表中，应用中每个单词出现的次数作为其权重。对于应用功能描述来说，出现次数越多的单词往往越能表示其主体功能。为了排除少数单词对整体功能的干扰，算法考虑将权重排名靠前的词纳入计算。

功能描述相似度计算公式如下：

$$\theta=\frac{\sum\limits_{i\in A}\max\limits_{m\in B}\alpha_{i,m}\cdot w_i+\sum\limits_{j\in B}\max\limits_{n\in A}\alpha_{n,j}\cdot w_j'}{N_A+N_B} \tag{1}$$

其中，$\alpha_{i,m}$表示应用 A 中第 i 个单词相对于应用 B 中第 m 个单词的相关度；$\max\limits_{m\in B}\alpha_{i,m}$表示 A 中第 i 个单词对应 B 中所有单词相关度的最大值；相应地，$\max\limits_{n\in A}\alpha_{j,n}$表示 B 中第 j 个单词对应 A 中所有单词相关度的最大值；w_i 表示 A 中第 i 个单词的权重即其在 A 中出现的次数；w_j'表示 B 中第 j 个单词的权重即其在 B 中出现的次数。N_A、N_B 分别表示 A、B 中所有单词的权重之和，即应用 A、B 中纳入计算的单词数量。$\sum\limits_{i\in A}\max\limits_{m\in B}\alpha_{i,m}\cdot w_i$ 表示应用 A 对应用 B 相关度的最大可能性，应用 B 对应表示亦然，而 θ 则表示 A、B 应用功能描述之间的相关性即相似度。某个单词与对应应用的单词相关度取最大值而非将所有对应单词相关度相加是考虑到应用描述可能表示应用中多种功能，或是多个单词形成了统一的功能，如将 A 中某个词与 B 中所有词的相似度都考虑进来则弱化了相同功能在相似度值中的体现，使得相似度值无法准确代表应用之间的相似性。

表 2 列出了应用 24h Analog Clock 与不同应用利用两种方法计算所得的功能描述的相似度值。24h Analog Clock 应用的主要功能为显示当前时间，若用户允许访问网络与具体地理位置，则可根据从网络取得的信息显示当地时间，并提供当地日出与日落时间。表中第二列为与 24h Analog Clock 进

行相似度比较的应用名称；第三列为利用本文所述方法计算所得功能描述相似度值；第四列则是通过单词匹配的方式对预处理后的功能描述信息计算相似度 Jaccard 系数，即两个应用的交集单词数量除以并集单词数量。

表 2　功能描述相似度值计算方式的对比

应用 A	应用 B	本文功能描述相似度	Jaccard 系数相似度
24h Analog Clock	Color Clock	100%	8.7%
	Minecraft Clock	100%	11.5%
	RvClock	100%	11.5%
	Jelly Clock	87.5%	7.4%
	My Location Widget	63.6%	3.3%
	Wi-Fi Widget	63.6%	8.0%
	A Time Tracker	53.8%	3.3%
	Clock Live Wallpaper	53.8%	3.6%
	ChessWatch	52.2%	6.7%
	Halachic Prayer Times	50.0%	4.0%
	Simple Alarm Clock	47.4%	3.3%
	…	…	…
	Yahtzee	0.0%	4.5%
	HUD	0.0%	11.6%
	File Explorer	0.0%	6.3%

从表 2 中可以看出，通过上文描述的相似度计算方式能有效找出与 24h Analog Clock 相关的应用，排名靠前的多数为时钟显示相关应用，并且由于突出了应用对应功能在相似度值计算中的重要性，其区分度远远大于使用 Jaccard 系数所得的相似度，其中 My Location Widget 和 Wi-Fi Widget 与 24h Analog Clock 相似度较高是由于三者均为手机桌面部件，并且 24h Analog Clock 与网络、地理位置高度相关。另一方面，Yahtzee 是一种骰子游戏，HUD 是利用手机为汽车导航的应用，File Explorer 是文件管理软件，这三者与 24h Analog Clock 相关度较低，通过本文方法得到它们的功能描述相似度为 0，而仅通过寻找相同单词的方式即计算 Jaccard 系数则无法与其他应用进行区分。

2.4.2　权限相似度

通过权限预处理模块，每个应用的权限声明由一个 151 位二进制位组成的字符串表示。因此权限相似度非常适合利用 Jaccard 系数进行计算。Jaccard 系数又称为 Jaccard 相似性系数，是用来比较样本集中的相似性和差异性的一个概率。Jaccard 系数等于样本集交集与样本集并集的比值。

权限相似度 Jaccard 系数计算公式如下：

$$J=\frac{|A\cap B|}{|A\cup B|} \tag{2}$$

其中，分子为 A、B 应用的权限字符串的各位分别做与操作得到的 1 的数量，分母为 A、B 应用的权限字符串的各位分别做或操作得到的 1 的数量。权限交集代表两个应用共同使用的权限集合，并集代表两个应用涉及到的权限总和。将计算所得的权限相似度 Jaccard 系数与交、并集权限对应位数均存储到数据库中，以供后续研究分析。

2.4.3　代码相似度

代码是程序功能的实际体现，利用代码相似度分析安卓应用能够为重打包检测、代码重用等多种研究提供基础支持。

基于安卓系统的应用开发主要利用 Android SDK，使用 Java 语言作为编程语言，由于开发者对变量名、语句执行顺序等具有不同的命名与使用方式，因此不适合通过简单匹配计算代码相似度。

本文基于 CCFinder 工具计算代码相似度。CCFinder 是一款基于 token 比较的开源代码克隆检测工具。检测过程通过基于编程语言的词汇分析器将源代码解析成 token 序列，通过一系列转换规则与变量替代对 token 序列进行处理，从处理后的 token 序列的所有字符串子集中分解出克隆比较对进行比较，最终确定克隆代码所在源文件的位置。CCFinder 提供了一系列度量指标来衡量代码克隆的程度，本文选择使用基于 token 数量的文件间相似度来进行计算。

本文修改了 CCFinder，使其批量分析安卓应用源代码，根据克隆对计算得出安卓应用两两间的整体代码相似度值。在此基础上，本文根据应用的树形代码组织方式对应用进行初步功能模块划分，将叶子文件夹即最底层文件夹中的文件划分为同一模块，在计算应用之间整体代码相似度的同时根据相似代码段计算模块与模块之间的代码相似度，并将其存储于数据库中。

3　经验研究

本文通过分析市场上安卓应用的通用信息，选取功能描述、权限与源代码 3 个维度，利用自然语言处理、相似度比较、代码克隆检测等技术对应用信息进行处理，计算安卓应用间功能描述、权限及代码相似度，通过对 3 个维度相似度进行交叉分析，找出潜在规律，为进一步研究与开发提供基础。

3.1　功能描述相似度与权限相似度交叉分析

本文针对 1173 个不同应用，分别计算了不同应用之间的功能描述相似度与权限相似度，一共得到 683859 条有效记录。图 2 为通过功能描述相似度与权限相似度交叉分析得到的应用分布情况，横坐标为功能描述相似度，纵坐标为权限相似度，每个点代表一个应用对。从图可知不同功能描述相似度与权限相似度的点基本覆盖各种情况。其中功能描述相似度与权限描述相似度均较低的情况占大部分，并且绝大多数点重合于(0,0)这一点上，这是由于安卓应用多种多样，绝大多数应用对的功能差异较大，对于权限声明的需求也各不相同。通过人工观察和鉴定，描述相似度 $desSimilarity(D)$ 选择 0.5 作为区分值，权限相似度 $perSimilarity(P)$ 选择 0.5 作为区分值。即 $D\geqslant 0.5$ 视为一组应用间描述相似度较高，$D<0.5$ 视为一组应用间描述相似度较低；$P\geqslant 0.5$ 视为一组应用间权限相似度较高，$P<0.5$ 视为一组应用间权限相似度较低。因此一组应用对根据功能描述相似度及权限相似度高低可能存在 4 种情况，图 3 示出每种情况对应的应用对数量。除了功能描述相似度与权限相似度均较低的情况占绝大多数以外，功能描述相似度较低、权限相似度较高的应用对最多，有 40681 对，可以得知功能差异较大的应用对可能拥有较高的权限相似度。

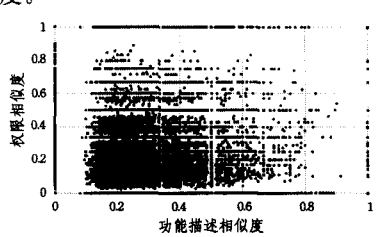

图 2　功能描述相似度与权限相似度交叉分析中的应用分布情况

针对上述情况，本文对应用权限声明情况进行进一步分析，发现当前安卓系统定义的权限主要针对设备的不同资源

进行安全性访问限制，因此权限与功能描述并非完全对应，而系统权限中存在某些比较常用的权限，提供某些基础资源访问，许多应用会对这些权限进行声明，导致功能不同的应用间权限声明相似度较高。图 4 所示为本文所采集的 1173 个应用对于 151 个权限声明的数量，其中共有 11 个权限声明的应用数量高于 100 个，值最大的为 INTERNET 权限，共有 603 个应用声明该权限，而这些应用获取 INTERNET 权限后完成的实际功能各不相同。因此，即使应用功能差异性较大，其权限声明也可能高度相似。如对于画图工具 Acrylic Paint 与文件阅读器 Document Viewer，这两个应用功能描述相似度为 0，而权限相似度为 100%，两者都声明了 INTERNET 权限及 WRITE_EXTERNAL_STORAGE 权限，这两个权限均为常用的基础功能所需的权限。

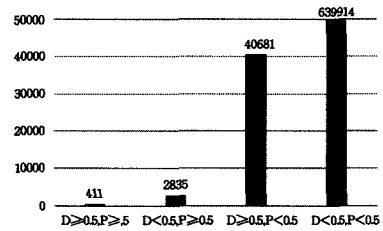

图 3　功能描述相似度与权限相似度交叉分析应用对数量统计

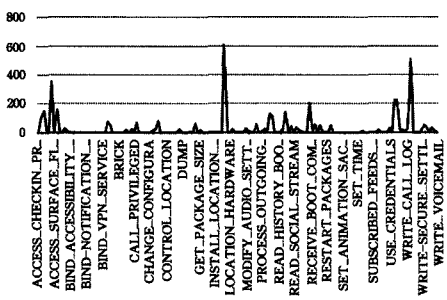

图 4　安卓系统权限在 1173 个应用中的声明情况

为了进一步探究安卓应用功能描述与权限之间的相关规律，本文根据功能描述相似度值，利用基于 weka 的谱聚类算法(Spectral Clustering)[12,13]对 1173 个应用进行聚类。谱聚类算法通过计算特征值的方式对数据进行降维后选择合适的特征向量聚类不同的数据点，其输入为数据集中不同对象之间的相似度矩阵。本文以功能相似度矩阵作为谱聚类的输入集，经计算后共获得 62 个簇，平均每个簇包含 19 个应用。每个簇中包含功能较为相似的一组应用，而不同簇间的应用功能存在较大差异。

针对不同簇中的应用进行权限分析，发现属于同簇的应用的权限声明大多局限于某一范围内，即集中在某些权限上，与某些系统资源相关。通过对不同应用簇权限声明的统计和观察发现，当超过 40%的应用声明某种权限时，可以认为该权限是该类型中的常用权限。图 5 示出簇 23 中的应用(包括 Worldmap、Where am I、Avare 等共 24 个应用)对应权限的声明情况，这类应用主要为地图、导航等与地理位置相关的应用，其中声明的应用数量大于 10 的权限有 ACCESS_COARSE_LOCATION 、ACCESS_FINE_LOCATION、ACCESS_NETWORK_STATE、INTERNET、WAKE_LOCK、WRITE_EXTERNAL_STORAGE 这 6 项；图 6 示出簇 34 中的应用(包括 SMSdroid、QuickMSG、SMS Filter 等共 9 个应用)对应权限的声明情况，这类应用主要为信息收发、读写以及信息过滤相关的应用，其中声明的应用数量大于等于 4 的权限有 ACCESS_NETWORK_STATE、INTERNET、READ_CONTACTS、READ_PHONE_STATE、READ_SMS、RECEIVE_SMS、SEND_SMS、VIBRATE、WAKE_LOCK、WRITE_EXTERNAL_STORAGE、WRITE_SMS 等 11 项。由此可以得出，对于相同功能的一类应用，其敏感操作集中在与该功能相关的设备资源上，因此权限声明也集中在相应权限上。不同类别的应用集中声明的权限不同，与该类别应用相关；而对于同属某一类别的应用，由于某些功能或实现方式的差异性，可能声明不同的权限，导致它们之间权限相似度可能较低，但它们的声明权限主要集中在与该类别应用相关的权限范围内。

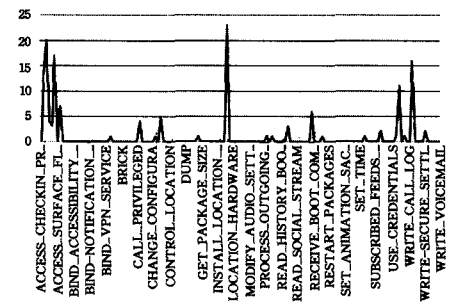

图 5　与地图、GPS 相关的 24 个应用对应权限的声明情况(簇 23)

由于某一类别相关应用的权限声明主要集中在一定范围内，因此通过分析某一应用与其他同类应用的权限差异，可以得出此应用区别于同类应用的权限所在，从而分析出是否存在安全隐患。如果某一应用中包含的某一权限并未或很少在其他同类应用中被声明，则该权限有可能为过度声明或存在安全隐患。

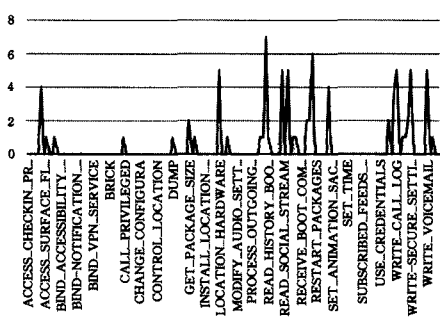

图 6　与信息接收和读写相关的 9 个应用对应权限的声明情况(簇 34)

表 3 展现了 GPSLogger 与同簇中抽取的几个应用之间的权限差异情况，其中打勾项代表该应用声明了对应权限。从表中可以得知，GPSLogger 在 ACCESS_MOCK_LOCATION、ACCESS_NETWORK_STATE、GET_ACCOUNTS、RECEIVE_BOOT_COMPLETED、USE_CREDENTIALS 这 5 个权限上与同类别权限有较大差异，其中 GPSLogger 声明而其他大多数应用未声明的权限包括 ACCESS_MOCK_LOCATION、GET_ACCOUNTS、RECEIVE_BOOT_COMPLETED、USE_CREDENTIALS 这 4 项。ACCESS_MOCK_LOCATION 为获取额外地理位置信息，GET_ACCOUNTS 为访问账户列表，RECEIVE_BOOT_COMPLETED 为允许在系统启动后接收到广播，USE_CREDENTIALS 允许一个应用程序向 AccountManager 申请授权标记。上述几项权限除 ACCESS_MOCK_LOCATION 之外均与 GPS 定位关系较

小，特别是 GET_ACCOUNTS 与 USE_CREDENTIALS 权限，使应用能够获取账户并申请授权标记，属于敏感操作，且与地理位置显示这一功能差异较大，因此分析得出可能存在安全隐患。

表 3　GPSLogger 与同簇中应用权限的差异情况

	Avare	Open Aviation Map	GetBack GPS	RMaps	Helsinki Testbed Viewer 2.0	Where am I?	GPSLogger
ACCESS_COARSE_LOCATION	√	√		√	√		√
ACCESS_MOCK_LOCATION							√
ACCESS_NETWORK_STATE			√				√
ACCESS_WIFI_STATE	√		√	√	√	√	√
GET_ACCOUNTS							√
RECEIVE_BOOT_COMPLETED							√
USE_CREDENTIALS							√
WAKE_LOCK	√		√		√	√	√
WRITE_EXTERNAL_STORAGE	√	√		√		√	√

3.2　功能描述相似度与代码相似度交叉分析

功能描述代表应用对外声明的功能，而代码则能体现应用的实际行为。本小节对安卓应用的功能描述相似度与代码相似度之间的关系进行了分析。图 7 为通过功能描述相似度与代码相似度交叉分析得到的安卓应用的分布情况，横坐标为功能描述相似度，纵坐标为代码相似度，每个点代表一组应用对。从图中可以看出不同功能描述相似度下代码相似度大多趋近或等于 0，其中绝大部分的坐标点重合于横轴，并且图 7 的分布覆盖情况较图 2 的分布明显稀疏。这是由于功能的实现取决于开发人员的理解，不同的开发人员实现相同功能的方式多种多样，如对一组对象进行排序可能使用快速排序、归并排序等多种方式，针对进一步的需求还可能做出不同的改变，不同开发人员的实现方式和代码风格导致即使功能描述相似度较高的应用之间的代码也完全不同，因此功能描述相似并不代表代码也相似。针对功能描述相似度与代码相似度交叉分析情况进行统计，图 8 展示了不同的功能描述相似度与代码相似度下安卓应用的数量统计情况，D 代表功能描述相似度，C 代表代码相似度，根据人工观察和鉴定，功能描述相似度使用 0.5 作为区分值，代码相似度则使用 0.3 作为区分值。由统计结果得出，功能描述相似度与代码相似度均大于区分值的应用对数量最少，仅 26 对；功能描述相似度小于 0.5 而代码相似度大于 0.3 的应用对有 942 对；功能描述相似度大于 0.5 而代码相似度小于 0.3 的应用对有 2378 对；功能描述相似度与代码相似度均小于区分值的有 665750 对。为了进一步研究应用之间的代码相似规律，本文通过对应用进行模块化划分，进一步细化，计算应用之间不同模块的代码相似度，并进行深入分析。

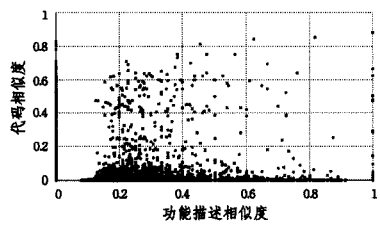

图 7　功能描述相似度与代码相似度交叉分析中应用的分布情况

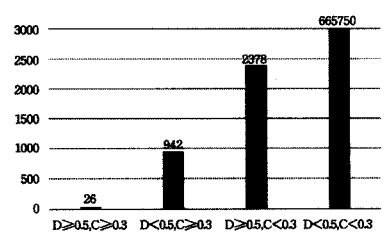

图 8　功能描述相似度与代码相似度交叉分析应用对数量统计

针对功能描述相似度较低而代码相似度较高的情况进行分析。通过研究发现，此类应用对的代码相似之处主要来源于使用了相同的第三方库、模块或框架。现有安卓系统库并不能囊括所有常用开发所需的功能库，而安卓平台支持第三方开发人员扩展常用功能库，因此安卓开发人员经常使用第三方库进行开发，导致功能无关的应用可能具有较高的相似度。如应用 E numbers 与 Open Explorer Beta，前者为食品添加剂参照程序，用来对照不同食品中存在的食品添加剂，而后者为文件管理程序，用来管理安卓系统中的用户文件，两者功能描述相似度为 0%，但代码相似度却高达 69.46%。图 9 所示为 E numbers 与 Open Explorer Beta 树形代码模块组织结构，子节点代表的文件夹包含在父节点中，如某节点内部仅有单个文件夹，使用“/”符号分隔将其合并至该节点中。从图中可以得知，由于上述两个应用均使用了第三方开源框架 Actionbarsherlock[14]，该框架支持 2.X 及以上的安卓系统对应用界面动作条的定制与开发，被广泛使用在不同应用中。E numbers 应用功能较为单一，使得第三方库 Actionbarsherlock 占据了绝大部分代码量，导致两个应用虽然功能毫不相关，但代码相似度却很高。

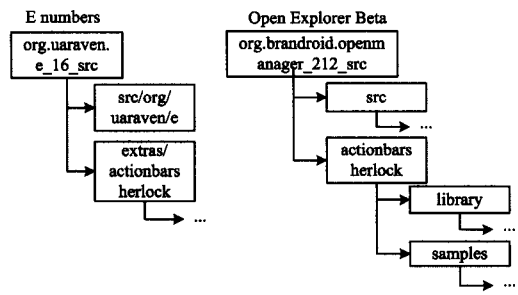

图 9　E numbers 与 Open Explorer Beta 树形代码模块组织结构

对于功能描述相似度与代码相似度均大于界限值的应用而言，这类应用对存在高度相似性，可能存在重打包现象。根据整体代码相似度及模块间代码相似度，进一步分析应用与应用间的相似代码，能有效检测高度相似或重打包现象。

如应用 OpenKeychain 与 APG，两者均用来加密或解密文件。图 10 为两者简化后的树形代码模块组织结构，在 OpenKeychain 中，sufficientlysecure/keychain 与 extern/spongy-castle为主要部分，占整体代码量的 95.7%，APG 中 sufficientlysecure/keychain 与 libs/bouncycastle 为主要部分，占整体代码量的 97%，两个应用整体代码相似度为 87.4%；从其代码树形组织结构可以得知两者代码结构十分相似，spongycastle 为经过 bouncycastle 重打包所得，包含加密解密算法，因此两者代码内容也极为相似，可能存在高度借鉴或重

打包现象。因此，基于功能描述相似度与应用之间代码相似度比较，可找出潜在的重打包、盗版应用组。

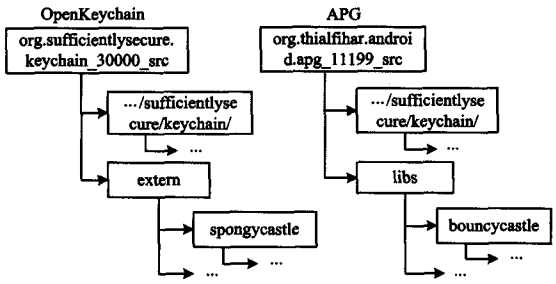

图 10 OpenKeychain 与 APG 简化后的树形代码模块组织结构

由于安卓应用种类丰富，大多数应用之间功能相似度较小，整体代码相似度也较小，本文通过对应用进行模块化划分，比较不同应用模块与模块之间的相似度，发现存在大量应用对的功能描述与整体代码相似度均较低，但是某些模块却高度相似。这些模块一部分为第三方库或开源框架，比如开源框架 Actionbarsherlock，另一部分则是开发人员对某些功能的相似实现或某些模块的复用，这些模块尚未形成规范的第三方库，造成不同的开发人员进行无意义的重复工作且不利于维护。通过比较不同应用模块间代码的相似度，可以找出较为相似的功能模块，对这些模块可进行统一整合，形成第三方库，使开发人员能够快速方便地调用这些功能。如 DroidUPnP、F-Droid、Orweb、VotAR 这 4 个应用都包含涉及构建网络交互服务的模块，并且这些模块之间代码相似度较高，可进一步提取出完善的第三方库以供开发人员使用。通过分析模块间代码相似度可找出一些待规范的通用代码模块，帮助减少开发人员的重复工作，提升开发效率。

4 相关工作

安卓应用的开发、维护和安全问题是软件工程领域值得关注的问题。目前已有许多针对安卓应用的研究工作。与本文相关的研究方向主要有：安卓应用描述、安卓应用安全、安卓应用代码重用等领域。

在安卓应用描述方面，He Jiang[15] 等通过众包和机器学习的方式发现了一些决定应用描述质量的属性，深入分析了影响安卓应用描述质量的各种原因。与该工作相比，本文采用自然语言处理技术对应用描述进行分析，并且利用 wup 算法计算了应用功能描述之间的语义文本相似度。

在安卓应用安全领域中针对安卓应用功能描述与权限之间关系的研究包括：Rahul Pandita[16] 等提出了一个基于自然语言处理技术的 WHYPER 框架，WHYPER 能识别应用描述中反映应用权限声明的语句，其研究结果表明使用自然语言处理技术有望进一步辅助移动应用的风险评估；Zhu Jiao[4] 等通过信息检索和语义分析等技术研究应用类别和权限的关系，以此得到各个类别的安卓应用和系统权限的关联关系模型，从而查找安全隐患；Alessandra Gorla[17] 则利用机器学习挖掘安卓应用描述主题以及和敏感权限对应的 API 的关系，以此识别恶意软件。与这些工作的目标不同，本文通过对描述相似度进行谱聚类，分析各个簇中的应用权限相似度，研究应用权限声明的合理性与安全性，利用应用间的相似关系聚类更为直接有效，能得到更加准确的结果。

针对安卓应用代码重用的研究包括：Steve Hanna[18] 等提出了一个分析安卓应用代码相似性的可扩展框架，以帮助解决安卓安全相关问题，包括确定应用程序是否包含存在 bug 的代码的副本、检测重打包和盗版以及检测已知的恶意软件的实例等；Xin Sun[19] 等构建了一个检测代码重用的系统 DroidSim，DroidSim 基于组件的控制流图（CB-CFG）计算代码相似性，以检测重新打包的应用程序和恶意软件的变体。本文修改了基于 token 比较的开源代码克隆检测工具 CCFinder，利用代码克隆对计算安卓应用间的代码相似度，并进一步检测功能模块相似情况，这有助于理解应用的代码特性，有利于抽取通用的代码模块以便于重用；此外本文还优化了处理流程，以适用于批量检测。

结束语 本文采用多种技术对安卓应用的各类信息进行处理，从功能描述、权限信息及源代码 3 个维度计算不同应用之间的相似度，并对相似度进行交叉分析，初步得到了安卓应用多个维度的相关规律，从而为安卓应用的开发与进一步研究提供基础，帮助规范安卓应用市场。在今后的研究中，将对样本数量及通用性进行进一步拓展，同时探索更多维度的关系，进一步构建安卓应用的关系网络模型。

参 考 文 献

[1] Smartphone OS market share, Q3 2014[OL]. [2014-12-14]. http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[2] Zhou W, Zhou Y, Jiang X, et al. Detecting repackaged smartphone applications in third-party android marketplaces[C]// Proceedings of the 2nd ACM Conference on Data and Application Security and Privacy, 2012. ACM, 2012: 317-326

[3] Zhou Y, Jiang X. Dissecting android malware: Characterization and evolution[C]// IEEE Symposium on Security and Privacy, 2012. IEEE, 2012: 95-109

[4] Zhu Jiao, Li Hong-wei, Peng Xin, et al. On Relationship of Functions and Permissions in Android Applications[J]. Computer Applications and Software, 2014, 31(10): 27-33

[5] F-Droid | Free and Open Source Android App Repository[OL]. [2014-12-14]. https://f-droid.org/

[6] Manifest.permission[OL]. [2014-12-14]. http://developer.android.com/reference/android/Manifest.permission.html

[7] Miller G A, Beckwith R, Fellbaum C, et al. Introduction to wordnet: An on-line lexical database[J]. International Journal of Lexicography, 1990, 3(4): 235-244

[8] Wu Z, Palmer M. Verbs semantics and lexical selection[C]// Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, 1994. 1994: 133-138

[9] Jaccard_index[OL]. [2014-12-14]. http://en.wikipedia.org/wiki/Jaccard_index

[10] Kamiya T, Kusumoto S, Inoue K. CCFinder: a multilinguistic token-based code clone detection system for large scale source code[J]. IEEE Transactions on Software Engineering, 2002, 28(7): 654-670

[11] Google Play [OL]. [2014-12-14]. https://play.google.com/store

[12] Spectral Clusterer for WEKA [OL]. [2014-12-14]. http://www.luigidragone.com/software/spectral-clusterer-for-weka/

[13] Spectral Clustering[OL]. [2014-12-14]. http://en.wikipedia.org/wiki/Spectral_clustering

[14] ActionBarSherlock [OL]. [2014-12-14]. http://actionbarsherlock.com

[15] Jiang H, Ma H, Ren Z, et al. What makes a good app description? [C]// Proceedings of the 6th Asia-Pacific Symposium on Internetware, 2014. ACM, 2014: 45-53

（下转第 219 页）

plications,2011,22(9):36-43

[3] Santoso H A,Haw S C,Abdul-Mehdi Z T. Ontology Extraction from Relational Database: Concept Hierarchy as Background Knowledge[J]. Knowledge-Based Systems,2011,24(3):457-464

[4] Wong W,Liu W,Bennamoun M. Ontology Learning from Text: A Look back and into the Future[J]. ACM Computing Surveys (CSUR),2012,44(4):1-36

[5] Yang Jun-hui,Liu Zong-tian,Liu Wei,et al. Extraction Method of Text Summarization Based on Event Network [J]. Computer Science,2015,42(3):210-213(in Chinese)
杨俊辉,刘宗田,刘炜,等.基于文本事件网络自动摘要的抽取方法[J].计算机科学,2015,42(3):210-213

[6] Petasis G,Karkaletsis V,Paliouras G,et al. Ontology Population and Enrichment:State of the Art[C]//Knowledge-Driven Multimedia Information Extraction and Ontology Evolution,2011. Berlin:Springer-Verlag,2011:134-166

[7] WordNet[EB/OL]. [2014-12]. http://WordNet. princeton. edu

[8] Cunningham H,Maynard D,Bontcheva K,et al. A Framework and Graphical Development Environment for Robust NLP Tools and Applications[C]//ACL,2002. ACM Press,2002:168-175

[9] Beautiful Soup [EB/OL]. [2014-12]. http://www. crummy. com/software/BeautifulSoup

[10] NLTK [EB/OL]. [2014-12]. http://www. nltk. org

[11] Stanford CoreNLP [EB/OL]. [2014-12]. http://nlp. stanford. edu/ software/corenlp. shtml

[12] Stanford Parser [EB/OL]. [Dec. 2014]. http://nlp. stanford. edu /software/ lex-parser. shtml

[13] Zhou De-mao, Li Zhou-jun. Survey of High-Performance Web Crawler[J]. Computer Science,2009,36(8):26-29(in Chinese)
周德懋,李舟军.高性能网络爬虫:研究综述[J].计算机科学,2009,36(8):26-29

[14] Davulcu H,Vadrevu S,Nagarajan S. OntoMiner:Bootstrapping Ontologies from Overlapping Domain Specific Web Sites[C]// Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters,2004. ACM Press,2004:500-501

[15] Wang Chao, Li Shu-qin, Xiao Hong. Research on Literature-based Automatic Ontology Construction Method for Agricultural Domain[J]. Computer Applications and Software,2014,31(8):71-74(in Chinese)
王超,李书琴,肖红.基于文献的农业领域本体自动构建方法研究[J].计算机应用与软件,2014,31(8):71-74

[16] Tang Qing,Lv Xue-qiang,Li Zhuo,et al. Research on Term Extraction for Domain Ontology[J]. New Technology of Library and Information Service,2014,30(1):43-50(in Chinese)
汤青,吕学强,李卓,等.领域本体术语抽取研究[J].现代图书情报技术,2014,30(1):43-50

[17] Maynard D,Li Y,Peters W. NLP Techniques for Term Extraction and Ontology Population[C]//Proceeding of the 2008 Conference on Ontology Learning and Population:Bridging the Gap between Text and Knowledge,2008. IEEE Press,2008:107-127

[18] Maynard D,Funk A,Peters W. SPRAT:A Tool for Automatic Semantic Pattern-Based Ontology Population[C]//International Conference for Digital Libraries and the Semantic Web. 2009

[19] Wu Y, Zhang S, Zhao W. Towards Learning Domain Ontology from Legacy Documents:Digital Society,2010[C]//Fourth International Conference on ICDS'10. IEEE Press,2010:164-171

[20] Sirin E,Parsia B,Grau B C,et al. Pellet:A Practical Owl-Dl Reasoner[J]. Web Semantics:Science,Services and Agents on the World Wide Web,2007,5(2):51-53

[21] Chen Yu,Zhu Jian-feng,Wu Yi-jian,et al. New Term Expansion Method Based on Domain Ontology[J]. Computer Engineering,2011,37(7):24-27(in Chinese)
陈宇,朱建锋,吴毅坚,等.一种基于领域本体的新术语扩充方法[J].计算机工程,2011,37(7):24-27

[22] Zablith F. Evolva:A Comprehensive Approach to Ontology Evolution[C]// The Semantic Web:Research and Applications,2009. Berlin:Springer,2009:944-948

[23] Li Jiang-hua, Shi Peng, Hu Chang-jun. Ontology Concept Learning Method for Compound Terms[J]. Computer Science,2013,40(5):168-172(in Chinese)
李江华,时鹏,胡长军.一种适用于复合术语的本体概念学习方法[J].计算机科学,2013,40(5):168-172

[24] Gu Jun,Xu Xin. Study on Ontology Relation Extraction in Chinese Patent Documents[J]. Computer Engineering,2013(10):73-78(in Chinese)
谷俊,许鑫.中文专利中本体关系获取研究[J].现代图书情报技术,2013(10):73-78

[25] Paiva L,Costa R,Figueiras P,et al. Discovering semantic relations from unstructured data for ontology enrichment:Asssociation rules based approach[C]//2014 9th Iberian Conference on Information Systems and Technologies (CISTI). IEEE,2014:1-6

[26] Faria C,Serra I,Girardi R. A domain-independent process for automatic ontology population from text[J]. Science of Computer Programming,2014,95(1):26-43

(上接第 205 页)

[16] Pandita R,Xiao X,Yang W,et al. WHYPER:Towards Automating Risk Assessment of Mobile Applications[C]//Proceedings of the 22nd Conference on USENIX Security Symposium,2013. 2013:527-542

[17] Gorla A, Tavecchia I, Gross F, et al. Checking app behavior against app descriptions[C]// Proceedings of the 36th International Conference on Software Engineering,2014. ACM,2014:1025-1035

[18] Hanna S,Huang L,Wu E,et al. Juxtapp:A scalable system for detecting code reuse among android applications[M]//Detection of Intrusions and Malware, and Vulnerability Assessment. Springer Berlin Heidelberg,2013:62-81

[19] Sun X,Zhongyang Y,Xin Z,et al. Detecting Code Reuse in Android Applications Using Component-Based Control Flow Graph [M]//ICT Systems Security and Privacy Protection. Springer Berlin Heidelberg,2014:142-155